



DEPARTAMENTO DE INGENIERÍA
DE SISTEMAS Y AUTOMÁTICA

Master en Automática e Informática Industrial

Trabajo Final de Master

Desarrollo de un sistema de monitorización de planta piloto mediante servidor OPC-UA empotrado

Alumno: Edison Santiago Maila Andrango

Tutor: Juan Francisco Blanes Noguera

Agradecimientos

A mi familia, amigos y profesores; por su ayuda, apoyo y paciencia, para la realización de este proyecto.

Índice General

1.	Introducción y objetivos	6
1.1	Introducción	6
1.2	Objetivo	7
1.3	Propuesta de desarrollo	7
2.	Hardware utilizado	9
2.1	BeagleBone Black	9
2.2	Control Lógico Programable	10
2.3	Human-Machine Interface	12
3.	Tecnología utilizada	13
3.1	Sysmac Studio	13
3.2	Ignition.....	14
3.3	Sistema de Supervisión, Control y Adquisición de Datos.	15
3.4	OPC	15
3.4.1	OPC DA	16
3.4.2	OPC HDA	17
3.4.3	OPC A&E Server	17
3.4.4	OPC UA	18
3.5	NODE- RED.....	19
4.	Desarrollo del Proyecto	20
4.1	Programación Automata.....	21
4.1.1	Planta piloto de manufactura - Fischertechnik.....	21
4.1.2	Programación y Lógica Automata.....	23
4.2	Desarrollo HMI	25
4.3	Programación BeagleBone Black	29
4.3.1	Planta piloto de carga y descarga – Fischertechnik	29
4.3.2	Programación y Lógica BeagleBone Black	32
4.4	Desarrollo sistema SCADA	45
5.	Conclusiones y trabajos futuros	53
5.1	Conclusiones	53
5.2	Trabajos futuros.....	54
6.	Bibliografía.....	55
7.	Anexos	56
7.1	Manual de usuario nodos GPIO y OPC-UA BeagleBone Black	56
7.2	Programa Ladder PLC Omron NJ101	67

Índice de Ilustraciones

Ilustración 1 Arquitectura de comunicación propuesto.....	8
Ilustración 2 Aspecto BeagleBone Black.....	9
Ilustración 3 Aspecto PLC OMRON NJ101	11
Ilustración 4 Aspecto Modulo E/S CJ1W - MD232.....	11
Ilustración 5 Aspecto HMI OMRON Serie NA	12
Ilustración 6 Aspecto software Sysmac Studio	13
Ilustración 7 Aspecto Gateway Software Ignition	14
Ilustración 8 Pirámide de automatización	15
Ilustración 9 Problemática sin tecnología OPC.....	16
Ilustración 10 Conexión OPC DA.....	16
Ilustración 11 Conexión OPC HDA	17
Ilustración 12 Conexión OPC A&E	18
Ilustración 13 Etapas funcionalidad OPC UA	18
Ilustración 14 Aspecto del editor de NODE RED.....	19
Ilustración 15 Aspecto del nodo GPIO de Node-Red.....	19
Ilustración 16 Arquitectura de comunicación del sistema a desarrollar	20
Ilustración 17 Aspecto Planta Piloto Manufactura	21
Ilustración 18 Distribución de pines conector IDC Planta Manufactura.....	22
Ilustración 19 Flujograma de proceso Planta de Manufactura	23
Ilustración 20 Variables globales del PLC usadas en el HMI.....	25
Ilustración 21 Pagina portada HMI.....	26
Ilustración 22 Acción Show Page para interactuar entre paginas	26
Ilustración 23 Pagina Modo HMI	27
Ilustración 24 Pagina Automático HMI.....	27
Ilustración 25 Asignar una variable del controlador al HMI	28
Ilustración 26 Pagina Manual HMI	29
Ilustración 27 Aspecto Planta Piloto Carga y Descarga	29
Ilustración 28 Señales E/S planta descarga – descarga	30
Ilustración 29 Distribución de pines conector IDC Planta Carga-Descarga	31
Ilustración 30 Diagrama conexión encoder y pulsos de dirección de rotación	31
Ilustración 31 Flujograma de proceso Planta carga – descarga	32
Ilustración 32 Aspecto visual Node-Red BBB.....	33
Ilustración 33 Aspecto nodo GPIO.....	34
Ilustración 34 Agregar nuevo Board node GPIO.....	34
Ilustración 35 Selección de Board node GPIO	34
Ilustración 36 Selección de entrada digital input node GPIO	35
Ilustración 37 Pin utilizado node GPIO	35
Ilustración 38 Nodo Exec y configuración de pin a qep.....	37
Ilustración 39 Configuración qep BeagleBone Black	38
Ilustración 40 Nodos OPC-UA Node-Red.....	38
Ilustración 41 Configuración OPC-UA server	39
Ilustración 42 Información de ayuda nodo OPC-UA.....	39
Ilustración 43 Agregar variable nodo OPC-UA.....	40
Ilustración 44 Envío de datos por variable OPC-UA	40
Ilustración 45 Encendido motor eje rotación a posición de set-point.....	41
Ilustración 46 Secuencia del primer movimiento para descarga	41
Ilustración 47 Secuencia del segundo movimiento hacia banda de carga	42
Ilustración 48 Secuencia de movimientos para proceso de carga y descarga en Node-Red.....	43
Ilustración 49 Variables y envío de datos a cliente OPC-UA.....	44

Ilustración 50 Aspecto Gateway Ignition.....	45
Ilustración 51 Crear un nuevo dispositivo OPC-UA SERVER	45
Ilustración 52 Conexión a Omron NJ Driver en Ignition	45
Ilustración 53 Edición de dispositivo NJ101 OMRON	46
Ilustración 54 Exportar variables globales del PLC	46
Ilustración 55 Variables globales esportadas como CX-Designer.....	47
Ilustración 56 Agregar tags Ignition.....	47
Ilustración 57 Importar variables globales a la lista de tags.....	47
Ilustración 58 Crear una nueva conexión OPC server	48
Ilustración 59 Agregar una nueva conexión OPC server	48
Ilustración 60 End-point del Servidor en BBB.....	48
Ilustración 61 OPC-UA running y conexión Ignition conectado.....	49
Ilustración 62 Abrir o crear nuevo proyecto	49
Ilustración 63 OPC Browser – tags.....	50
Ilustración 64 Image Browser Ignition.....	51
Ilustración 65 2 - State Toogle Ignition.....	51
Ilustración 66 Numeric Text Field Ignition.....	52
Ilustración 67 Pantalla SCADA	52

Índice de tablas

Tabla 1 Canales eQEP BeagleBone Black	10
Tabla 2 Parámetro de los canales eQEP	10
Tabla 3 Comparación productos HMI Omron serie NA	12
Tabla 4 Señales E/S planta manufactura	22
Tabla 5 Pines admitidos del nodo GPIO en BeagleBone Black	37

Resumen

El presente trabajo corresponde al desarrollo de un sistema de monitorización para dos plantas piloto, mediante un servidor OPC-UA empotrado.

Para el control de la primera planta piloto; la encargada del desarrollo de mecanizado, se utiliza un controlador lógico programable (PLC) de la marca OMRON NJ101, de tal manera que, controlada la planta, adquirimos las señales de entrada y salida y las enviamos a un sistema de supervisión, control y adquisición de datos (SCADA) del software IGNITION, mediante el protocolo de comunicación OPC-UA.

La segunda planta piloto; gestiona la carga y descargar de las piezas para la primera planta piloto, para el control de esta planta se utiliza una placa BeagleBone Black y la herramienta de programación Node-Red, permitiendo adquirir las señales de entrada y salida, para luego ser enviadas al sistema SCADA mediante un servidor OPC-UA.

Finalmente, se realizan las pruebas pertinentes, obteniendo como resultado, la monitorización y control de las dos plantas piloto.

Por lo tanto, se ha realizado un sistema de control y monitorización con dos controladores de distintas características, pero que, mediante el protocolo de comunicación OPC-UA permite comunicarse entre ellas.

1. Introducción y objetivos

1.1 Introducción

El presente trabajo corresponde al proyecto final de master en Automática e Informática Industrial de la Universidad Politécnica de Valencia.

La investigación se refiere al desarrollo de un sistema de monitorización de planta piloto mediante servidor OPC-UA.

Principalmente contamos con dos plantas piloto de la marca Fischertechnik, la primera planta se ocupa de la mecanización de piezas, esta es controlada mediante un programador lógico programable (PLC) de la marca Omron NJ101, la misma que se encarga de controlar los transportadores, el área de fresado, el área de perforado, los desplazadores y las señales de entradas proporcionadas por fototransistores.

Una vez obtenido el control, se recogen las señales y se envían a un sistema de supervisión, control y adquisición de datos (SCADA) del software IGNITION mediante el protocolo de comunicación OPC-UA.

La segunda planta se encarga de la carga y descarga de piezas para abastecimiento de la primera planta, esta planta es controlada mediante una placa computadora de hardware libre de bajo coste y consumo producida por Texas Instruments conocida como BeagleBone Black, esta se ocupa de adquirir las señales de los fines de carrera y las señales de encoder proporcionadas por los tres motores de la planta.

Una vez adquirida las señales, usando la herramienta de programación NODE-RED, se realiza el control de posición de los motores, para posteriormente enviar las señales mediante un servidor OPC-UA al sistema SCADA para su monitorización.

Una de las principales ventajas de la placa BeagleBone Black , aparte de ser de bajo coste y consumo, nos permite utilizar sus entradas especiales de lectura de encoder, habilitando los canales de Quadrature Encoder Pulse (eQEP).

La placa contiene 4 entradas eQEP, una de ellas solo puede ser usada si el video esta deshabilitado, y otra puede ser usada siempre y cuando no se usen al mismo tiempo, en total se puede decir que para nuestro caso tenemos a disposición tres entradas de encoder, los cuales no permiten realizar el control de posición de los motores para la segunda planta piloto.

A parte de las señales de lectura de encoder, otra de las ventajas es que podemos utilizar la herramienta de programación Node-Red, siendo un motor de flujos con enfoques de Internet Of Things (IOT), que permite definir flujos de servicio de forma gráfica a través de varios protocolos de comunicación como Mqtt, WebSocket y para nuestro caso mediante un servidor OPC-UA.

De tal manera que mediante las librerías existentes en Node-Red, obtendremos las señales de los GPIO, la señal de los eQEP y las enviaremos a un sistema de monitorización.

1.2 Objetivo

El objetivo principal del trabajo es el desarrollo de un sistema de monitorización de planta piloto mediante servidor OPC-UA empotrado. De tal manera que se pueda integrar distintas tecnologías para el control de dos plantas piloto y sean monitoreadas y controladas por en un sistema SCADA, mediante el protocolo de comunicación OPC-UA.

Para conseguirlo se definen los siguientes objetivos específicos:

- Implementar el programa de control que actúe sobre un controlador lógico programable para gobernar las entradas y salidas de la planta piloto de mecanizado.
- Enviar las señales adquiridas por el controlador lógico programable a un sistema de supervisión, control y adquisición de datos SCADA mediante el protocolo de comunicación OPC-UA.
- Desarrollar el programa de control que permita adquirir señales de entrada y salida de la planta piloto de carga y descarga mediante el uso de las GPIO de la BeagleBone Black.
- Enviar las señales adquiridas por la BeagleBone Black a un sistema SCADA mediante un servidor OPC-UA empotrado.
- Desarrollar la interfaz gráfica para el sistema SCADA y el HMI, de manera que, permita observar el comportamiento de las dos plantas piloto.

1.3 Propuesta de desarrollo

El continuo desarrollo y evolución de la automatización, genera que muchos procesos distribuidos sean capaces de comunicarse entre sí. Este problema lo podemos ver en el desarrollo de este proyecto utilizando dos plantas piloto y distintas tecnologías hardware como software que permitan comunicarse a un sistema de supervisión y control.

Para abordar el desarrollo del proyecto, se trabaja en varios niveles de la pirámide de automatización, buscando soluciones en cada una de ellas. Primero se trabajaba sobre la red de proceso para adquirir las señales de las plantas piloto, luego sobre la red de control con un PLC y una BeagleBone Black, finalizando sobre el nivel de supervisión donde se realiza un SCADA y se envía señales a través del protocolo OPC-UA.

En primer lugar se trabaja sobre la planta de manufactura, el planteamiento del proyecto se detalla en los siguientes pasos:

1. Para la red de proceso, se obtiene las señales físicas de la planta piloto de manufactura enviándola a un PLC por medio de un cable 26-28 AWG con conector IDC de 26 pines (planta piloto) a uno de 20 pines (E/S PLC).
2. En la red de control, se realiza un programa mediante el software Sysmac Studio de Omron y el lenguaje de programación Ladder para el control y la supervisión de la planta piloto.

3. Para la red de supervisión, se desarrolla el sistema de supervisión y control HMI (Human-Machine Interface) usando Sysmac Studio y se envía las señales del PLC usando variables globales del control por medio de ethernet.
4. Para la misma red de supervisión, mediante el protocolo CIP (Common Industrial Protocol), se exporta las variables del PLC a un servidor OPC-UA de Ignition y creamos un sistema SCADA mediante el software Ignition.

Para la segunda planta el planteamiento es el siguiente:

1. En la red de procesos, se adquiere las señales físicas de la planta piloto de carga descarga y se utiliza la placa de potencia para adaptar la señal de E/S a la BeagleBone Black.
2. En la red de control, se utiliza la herramienta de programación node-red, y se realiza un programa que controla el proceso de la planta de manera que lea las señales de encoder de los motores, las señales digitales y las salidas a los actuadores.
3. En la etapa de supervisión, mediante los nodos de node-red, se crea un servidor OPC-UA empotrado y se envía las señales a un cliente OPC-UA de Ignition, agregando variables de control al sistema SCADA.

El enfoque global a la propuesta de desarrollo se lo muestra en la ilustración 1.

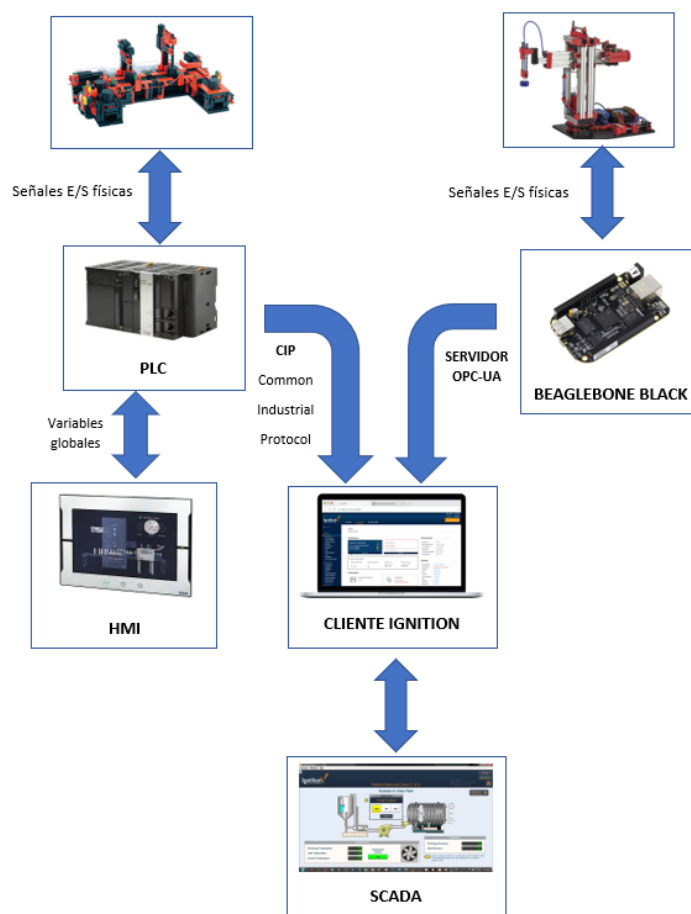


Ilustración 1 Arquitectura de comunicación propuesto

2. Hardware utilizado

Para la realización del proyecto, se ha utilizado diversos elementos hardware que nos permiten el control y la supervisión de las dos plantas piloto. Entre ellos se ha utilizado una computadora de bajo coste como es la placa BeagleBone Black para adquirir señales mediante las GPIO, un controlador lógico programable NJ 101 y una pantalla HMI de la serie NA de la marca OMRON.

2.1 BeagleBone Black

La placa BeagleBone Black es una pequeña computadora de base Linux de bajo coste y consumo producida por Texas Instrument, siendo desarrollada como una placa educativa para enseñar las capacidades del software y hardware libre.

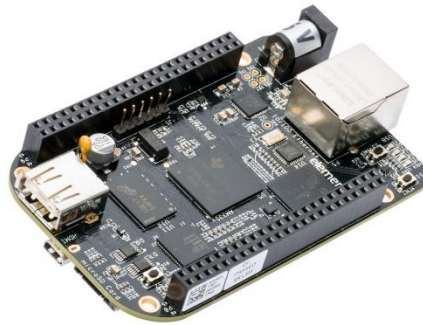


Ilustración 2 Aspecto BeagleBone Black

Debido a su gran versatilidad y precio, facilita el desarrollo de aplicación permitiendo el control de actuadores y sensores mediante sus GPIO.

Adicional a sus capacidades, permite utilizar la herramienta de programación Node-Red, de tal manera que se puede utilizar varios protocolos de comunicación, entre ellas comunicaciones por OPC-UA.

Sus especificaciones técnicas son:

- Procesador AM335x 1GHz ARM@ Cortex-A8
- 512 MB de RAM DDR3
- Almacenamiento flash integrado eMMC de 4GB y 8bits.
- Acelerador de gráficos 3D.
- NEON acelerador de punto flotante.
- 2x microcontroladores PRU de 32 bits.

Su compatibilidad de software con:

- Debian
- Androide
- Ubuntu

- Clud9 IDE en Node.js con biblioteca BoneScript
- Node-Red

Conectividad mediante:

- Cliente USB para alimentación y comunicaciones.
- Puerto USB
- Ethernet
- HDMI
- 2 cabezales de 46 pines.

Dentro de las ventajas ya mencionadas, podemos resaltar que BeagleBone Black permite la lectura de señales encoder, accediendo a los canales de acceso a Quadrature Encoder Pulse (eQEP).

Los canales disponibles se pueden observar en la tabla 1.

CANAL	PIN A	PIN B	NOTA
eQEPO	P9.27	P9.42	Disponible
eQEP1	P8.33	P8.35	Disponible con video deshabilitado
eQEP2	P8.11	P8.12	Disponible con eQEP2b sin usar
eQEP2b	P8.41	P8.42	Disponible con video deshabilitado y eQEP2 sin usar

Tabla 1 Canales eQEP BeagleBone Black

Los parámetros de los canales se observa en la tabla 2.

Enabled	Devuelve 1 si el canal eQEP está habilitado, 0 si esta deshabilitado.
Mode	Devuelve 0 si el canal eQEP está configurado en modo absoluto, 1 si está configurado en modo relativo
Period	Periodo de la señal
Position	Devuelve la posición actual del codificador.

Tabla 2 Parámetro de los canales eQEP

2.2 Control Lógico Programable

Un control lógico programable conocido por sus siglas en ingles como PLC (Programmable Logic Controller) o autómatas programables, es un dispositivo electrónico utilizado para de control de procesos en distintas etapas de la automatización industria.

Los PLC son muy utilizados en la industria ya que están diseñados para soportar rangos de temperatura amplios, inmunidad al ruido eléctrico, resistencia a vibraciones e impacto, entre otros.

Sus estructura interna consta de una CPU (Unidad Central de Procesos) e interfaces de entrada y salida. La CPU se encarga de ejecutar el programa escrito por el usuario que se encuentra almacenado en la memoria, además permite comunicarse con el exterior mediante sus puertos de comunicación y realizar funciones de autodiagnóstico. La interfaz de entrada y salida permite captar las señales que llegan de sensores, botoneras, fines de carrera, entre otros, para luego mediante la secuencia realizada por el usuario, estas realicen una acción sobre sus señales de salidas como motores, pistones, electroválvulas, entre otros.

Para el desarrollo de este proyecto, disponemos en laboratorio un PLC de la marca OMRON de la serie NJ101. La serie NJ1 es un nuevo controlador Sysmac para el control de secuencias de lógica y motion, diseñado para maquinas compactas.



Ilustración 3 Aspecto PLC OMRON NJ101

Dentro de sus características relevantes tenemos:

- Tiempo de ciclo más rápido: 1ms.
- Numero de ejes: 2.
- Núcleo de motion sincronizado.
- Funciones: lógica, motion y conexión con base de datos.
- Multitarea.
- Puertos EtherCAT y EtherNet/IP integrados.

De igual manera disponemos en el laboratorio la unidad de entrada y salida CJ1W-MD232, la misma que actúa de interfaz del PLC entre en el control del programa y los distintos actuadores.



Ilustración 4 Aspecto Modulo E/S CJ1W - MD232

Las unidades CJ esta disponibles con distinta densidades de E/S y tecnologías de conexión. Para el desarrollo de nuestro proyecto disponemos de una unidad de E/S de 32 puntos de alta densidad equipadas con conectores de cable plano de 40 pines.

2.3 Human-Machine Interface

La interfaz hombre maquina o mejor conocida como HMI por sus siglas en inglés (Human-Machine Interface) es una panel de control diseñado para conseguir una comunicación interactiva entre el operador y la máquina, se trata básicamente de un panel de instrumentos del operario.

La función consiste en mostrar información operativa en tiempo real, proporcionando gráficos de procesos visuales, acciones de estados a los actuadores, control de operaciones, entre otros, que permitan el control de la maquina y sus procesos.

Para el desarrollo de nuestro proyecto disponemos en laboratorio el HMI de la marca OMRON de la serie NA.



Ilustración 5 Aspecto HMI OMRON Serie NA

Siendo una terminal HMI dinámica, intuitiva y predictiva que hace que las maquinas industriales sean más competitivas. La terminal HMI Sysmac permite que los procesos de monitorización y control sean más rápidos y eficientes, proporcionando una relación proactiva entre el operador y máquina.

Las especificaciones más relevantes que podemos encontrar entre las distintas versiones se muestran en la siguiente tabla.

HMI	NA5-15	NA5-12	NA5-9	NA5-7
DISPLAY	15" widescreen TFT colour	12" widescreen TFT colour	9" widescreen TFT colour	7" widescreen TFT colour
RESOLUTION	1280x800	1280x800	800x480	800x480
COLOURS	24bit	24bit	24bit	24bit
CONNECTIVITY	3x USB, 2x Ethernet, 1x RS232, SD Card, 24VDC	3x USB, 2x Ethernet, 1x RS232, SD Card, 24VDC	3x USB, 2x Ethernet, 1x RS232, SD Card, 24VDC	3x USB, 2x Ethernet, 1x RS232, SD Card, 24VDC
DIMENSIONS	420x291, 391x267 (cut-out)	340x244, 309x220 (cut-out)	290x190, 260x165 (cut-out)	236x165, 196x140 (cut-out)

Tabla 3 Comparación productos HMI Omron serie NA

3. Tecnología utilizada

Es este capítulo se describe las diferentes tecnologías de software y protocolos de comunicación que se han utilizado para el desarrollo del proyecto.

3.1 Sysmac Studio

Sysmac Studio es un entorno de desarrollo integrado, creado para ofrecer a los desarrolladores de maquinaria un control total desde un entorno único. Integra funciones de configuración, programación, simulación y monitorización, siendo la única herramienta de software que se necesita para el desarrollo de controladores de la serie NJ e interfaces HMI.

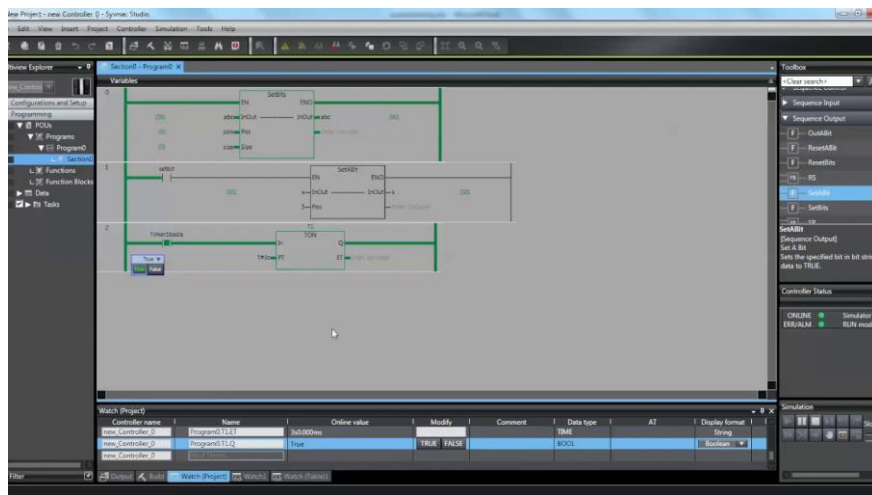


Ilustración 6 Aspecto software Sysmac Studio

Los principales productos hardware que se pueden integrar este entorno son:

- Controladores de la serie NJ/NX.
- HMI de la serie NA.
- E/S y de seguridad de la serie NX.
- Servo de la serie G5.
- Variadores de frecuencia de las series MX2 y RX.
- Sistema de visualización y sensor FH y FQM.
- Componentes de red GX.
- Sensores E3 N-Smart.
- Sensores de medición de la serie ZW.

Dentro de sus ventajas esta la programación con variables, donde Sysmac asigna automáticamente memoria a la CPU para variables que se crean necesarias, facilitando el desarrollo y reduciendo los errores.

Estas variables nos permitirán transformarlas a variables globales, para luego ser controladas por el MHI y también ser exportadas como CX-Designer para el control del SCADA mediante el protocolo OPC-UA.

3.2 Ignition

Ignition es una plataforma de software integrada para sistemas SCADA lanzada por Inductive Automation en enero del 2010.

Es la primera plataforma de aplicaciones industriales verdaderamente universal para conectar datos y diseñar e implementar aplicaciones industriales en toda la empresa. La plataforma y los módulos, permiten la creación de aplicaciones como SCADA, MES, IIOT, entre más.

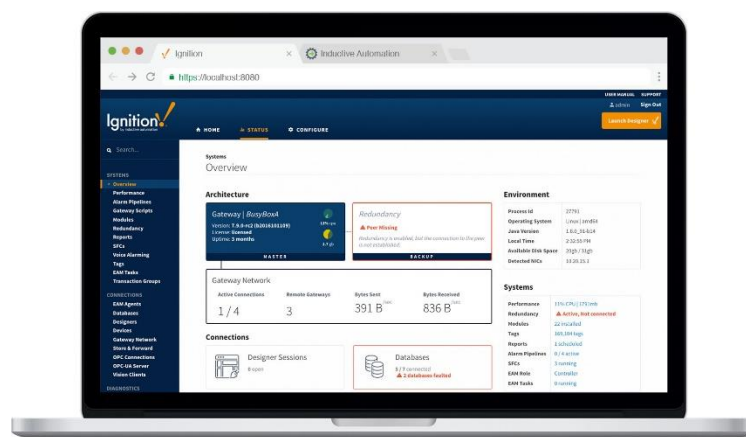


Ilustración 7 Aspecto Gateway Software Ignition

Los módulos de encendido SCADA proporcionan características tales como: control de estado en tiempo real, alarmas, informes, adquisición de datos, secuencia de comandos, programación y soporte móvil.

El módulo de encendido OPC-UA es un servidor OPC que admite controladores modulares para PLC y otros dispositivos y conexiones de red, permitiendo a los usuarios leer y escribir valores de registro del PLC a través de una página web alojada en la puerta de enlace de encendido.

3.3 Sistema de Supervisión, Control y Adquisición de Datos.

Un sistema de supervisión, control y adquisición de datos o mejor conocido por su siglas en inglés SCADA (Supervisory Control and Data Acquisition), es un término que se emplea para realizar una aplicación que permita el control y supervisión de procesos a distancia.

El control en tiempo real con sensores y actuadores, permite una realimentación o feedback que controla el proceso automáticamente. el objetivo es compartir observaciones, preocupaciones, y sugerencias para recabar información a nivel individual o grupal y de esa manera mejorar o modificar distintos procesos por el bien de la organización.

Para entender mejor la funcionalidad de los sistemas SCADA, la siguiente figura muestra la distribución jerárquica de un sistema de automatización,



Ilustración 8 Pirámide de automatización

Esto permite una mejor toma de decisiones en las empresas, ya que se obtiene información en todos los niveles de producción y sus departamentos interconectados, que van desde el proceso de la planta de producción, almacenaje, distribución, compra logística, etc.

La diferencia entre un HMI y un SCADA es que el HMI se utiliza para monitorizar los sistemas SCADA, es decir que si unimos ambos conceptos, tendremos ventajas como: potenciar el control y monitorización del proceso, informar de alarmas o tiempos muertos, efectividad del usuario pudiendo controlar más datos, reducción de costes operacionales y de mantenimiento, entre otros.

3.4 OPC

OPC conocida si por sus siglas en inglés (OLE for Process Control) , es una tecnología diseñada para comunicar datos de forma segura y operativa entre aplicaciones tanto en la industria como en otros sectores, que inicialmente nació para funcionar únicamente en sistema operativo Windows.

Se originó debido a la problemática de comunicar varias tecnologías existentes en el proceso de automatización, ya que esto conlleva la adquisición de varias licencias, drivers y pasarelas, que en muchas ocasiones es difícil disponer de los mismos.

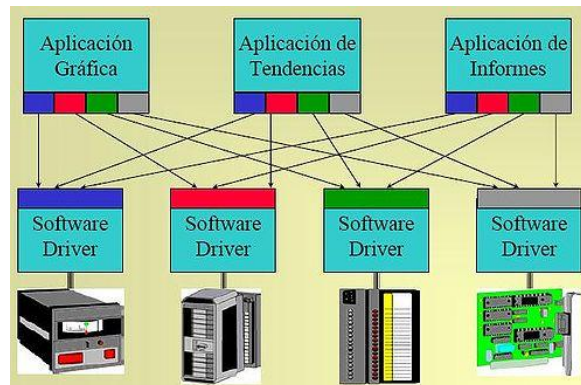


Ilustración 9 Problemática sin tecnología OPC

OPC es una plataforma independiente de los fabricantes y por lo tanto abierta ya que asegura el flujo de información entre componentes de distintos proveedores.

Permite a un servidor manejar varios dispositivos, operar en paralelo varios servidores, atender varios clientes, todo ellos con una interfaz uniforme, de manera que se puede leer/escribir variables, manejar alarmas, almacenar datos históricos, comunicación entre aplicaciones e incluso con base de datos.

Existen cuatro tipos de servidores OPC definidos por la fundación OPC, siendo:

- Servidor OPC DA
- Servidor OPC HDA
- Servidor OPC A&E Server
- Servidor OPC UA

3.4.1 OPC DA

Proviene de OPC Data Access, es una especificación que permite la comunicación y transferencia de datos entre una fuente y una aplicación cliente, por ejemplo entre un PLC y un SCADA sin necesidad de que cada uno conozca el protocolo nativo del otro.



Ilustración 10 Conexión OPC DA

Los datos obtenidos de sensores o posiciones de memoria de los PLC's, pueden ser leídas por los procesos clientes cuando cambian, bajo demanda o cuando pasa un determinado tiempo, gestionando la actualización de sus valores de acuerdo a la política que se defina.

Fue la primera definida por la fundación OPC, antes de ella los fabricantes de dispositivos obligaban a cualquier aplicación que desee conectarse a ellas, tener un driver propietario capaz de traducir los datos entre ellos

3.4.2 OPC HDA

Proviene de OPC Historical Data Access, se utiliza para recuperar y analizar datos de proceso históricos con el propósito de análisis, optimización, control de inventario, cumplimiento normativo, etc.

Esto permite que las aplicaciones escriban datos históricos en una base de datos, ya que muchas veces los usuarios necesitan ingresar lecturas, o hacer correcciones de información previa.



Ilustración 11 Conexión OPC HDA

OPC HDA también se puede utilizar para la transferencia automática de datos, es decir se puede compensar las interrupciones en la transferencia e tiempo real de los datos del proceso, almacenando los datos en el buffer durante la interrupción para cuando se restablece, recuperar la información almacenada.

3.4.3 OPC A&E Server

Proviene de alarmas y eventos, se utiliza para intercambiar alarmas y eventos de procesos. Como es conocido dentro de la industria todo cambio que se produce en la planta o estados anormales requieren de atención. Incluso el personal de mantenimiento puede utilizarlo para implementar una gestión de mantenimiento basado en tiempos, condiciones o incluso informar de fallas del equipo.

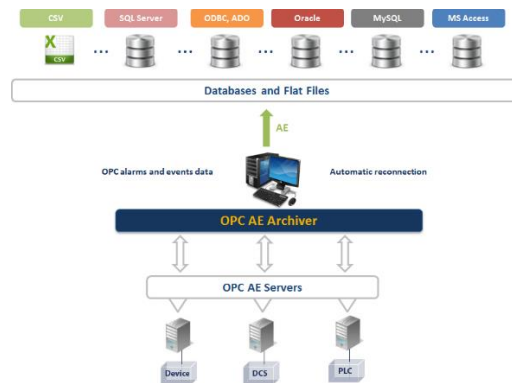


Ilustración 12 Conexión OPC A&E

3.4.4 OPC UA

Proviene de Open Productivity Collaboration Unified Architecture, es un protocolo de comunicación flexible e independiente. La necesidad de encontrar simplicidad y seguridad ha llevado a la Fundación OPC crear un método de comunicación unificado para las actuales especificaciones OPC DA, HDA y A&E.

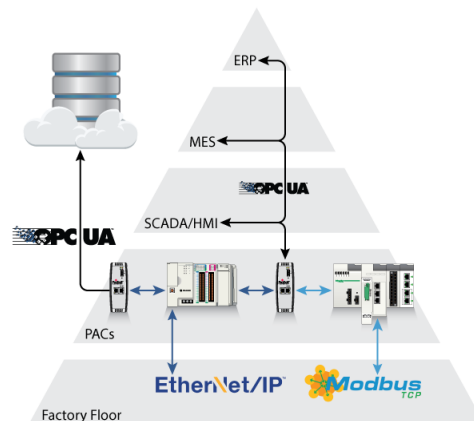


Ilustración 13 Etapas funcionalidad OPC UA

Las principales características y beneficios de los OPC UA son:

- Plataforma neutral que funciona en cualquier sistema operativo.
- Transmisión segura y fiable entre niveles de producción.
- Metodología Cliente/Servidor.
- Prepara y permite la comunicación con sistemas antiguos.
- Tecnología orientada a servicios.
- Define una arquitectura robusta, con mecanismos fiables.
- Fácil configuración y mantenimiento.
- Alto rendimiento y escalable.

3.5 NODE- RED

Es una herramienta de programación visual, que permite conectar dispositivos de hardware, API y servicios en línea de manera distinta.

Proporciona un editor de flujo basado en un navegador web, facilitando la conexión de flujos, mediante la utilización de los distintos nodos en la paleta.

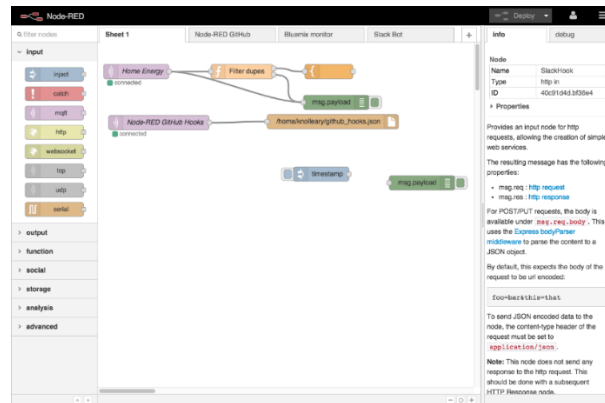


Ilustración 14 Aspecto del editor de NODE RED

Existen dos tipos de nodos, un nodo de inyección y uno de función. Los nodos de inyección producen un mensaje sin necesidad de conectar una entrada, solo lanzan el mensaje al siguiente nodo que este conectado a éste. Los nodos de función, tienen una entrada y una salida, y realizan algún trabajo dentro de él. Debido a su modelo sin bloqueo controlado por eventos, permite ejecutar el mismo en hardware de bajo costo, como Raspberry Pi, BeagleBone Black, Arduino , así como la nube.

Para el desarrollo del proyecto se utiliza Node-Red sobre un dispositivo BeagleBone Black. Antes de cualquier cosa es recomendable utilizar las últimas imágenes basadas en Debian 9. Las imágenes de 4GB para placas BeagleBone Black ya tienen un Node-Red preinstalado y configurado para inicio automático, por lo que puede iniciar y apuntar su navegador al puerto 1880.

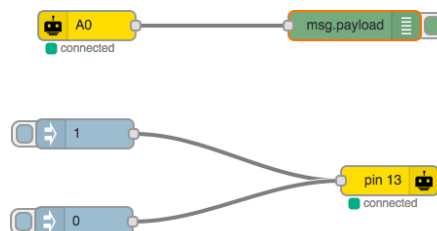


Ilustración 15 Aspecto del nodo GPIO de Node-Red

El conjunto de nodos que nos permitirá adquirir las señales de entrada de la BeagleBone Black, es node-red-contrib-gpio, este nodo también viene preinstalado en las últimas versiones y es aquella que nos permitirá adquirir las señales de los sensores y las salidas para los actuadores.

4. Desarrollo del Proyecto

En este capítulo se describe de forma detallada todo el trabajo realizado durante la elaboración del proyecto. Para la realización del proyecto se toma en cuenta todos los objetivos anteriormente mencionados. En primer lugar se trabaja sobre la adquisición de señales de la planta piloto de manufactura, el autómatas programable y el envío de señales al sistema SCADA. En segundo lugar abordaremos la adquisición de señales de la planta piloto de carga y descarga, el uso de los GPIO en la placa BeagleBone Black mediante NODE-RED, como levantar un servidor OPC-UA y el envío de datos al sistema SCADA.

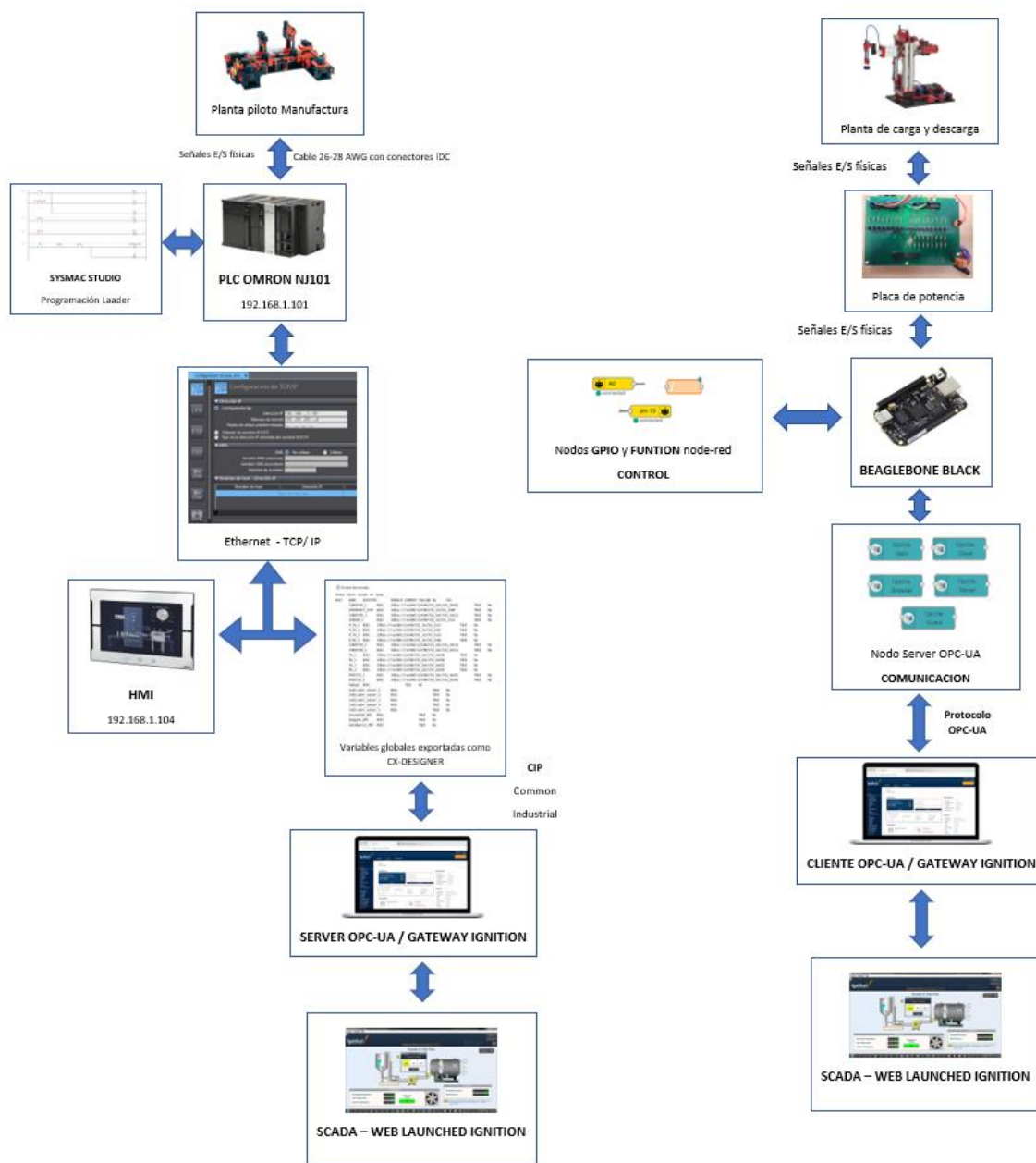


Ilustración 16 Arquitectura de comunicación del sistema a desarrollar

4.1 Programación Automata

Como se ha mencionado anteriormente, la primera etapa del desarrollo del proyecto es la adquisición, control y envío de las señales de la planta piloto de manufactura al sistema SCADA.

4.1.1 Planta piloto de manufactura - Fischertechnik

Para comenzar a programar primero tenemos que conocer todas las señales que vienen de la planta piloto de manufactura de la marca Fischertechnik, la podemos encontrar con el nombre de Indexed Line with two Machining Station 24V

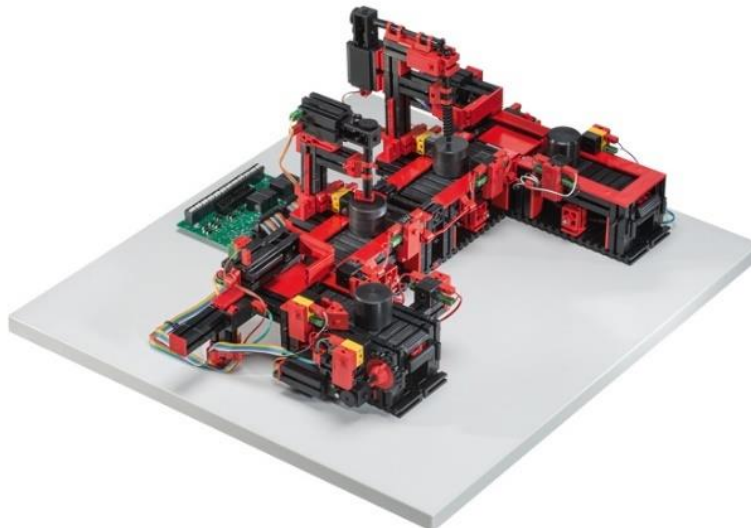


Ilustración 17 Aspecto Planta Piloto Manufactura

Esta planta piloto es una línea transportadora con una estación de fresado y una de taladrado con cuatro cintas transportadoras que forman una U.

Dispone de una placa electrónica que funciona a 24 V DC, contiene un circuito con relés para la adquisición de entrada y salidas y para invertir el sentido de rotación de algunos motores.

Se puede conectar a un conector IDC de 26 pines con paso de 2.54 mm, o a terminales en serie de inserción.

Las señales de entrada, salida y alimentación, se pueden ver en la siguiente tabla:

Numero de terminal	Descripción	Entrada / Salida
1	Fuente de alimentación actuadores (+)	24V DC
2	Fuente de alimentación sensores (+)	24V DC
3	Fuente de alimentación (-)	0V DC
4	Fuente de alimentación (-)	0V DC
5	Final de carrera deslizador 1 adelante	Entrada 1
6	Final de carrera deslizador 1 atrás	Entrada 2
7	Final de carrera deslizador 2 adelante	Entrada 3
8	Final de carrera deslizador 2 atrás	Entrada 4
9	Fototransistor acción deslizador 1	Entrada 5
10	Fototransistor estación fresadora	Entrada 6
11	Fototransistor estación de carga	Entrada 7
12	Fototransistor estación taladro	Entrada 8
13	Fototransistor estación de descarga	Entrada 9
14		
15	Motor deslizador 1 atrás	Salida 1
16	Motor deslizador 1 adelante	Salida 2
17	Motor deslizador 2 atrás	Salida 3
18	Motor deslizador 2 adelante	Salida 4
19	Motor cinta transportadora estación de carga	Salida 5
20	Motor cinta transportadora estación de fresado	Salida 6
21	Motor máquina de fresado	Salida 7
22	Motor cinta transportadora estación de taladro	Salida 8
23	Motor maquina taladro	Salida 9
24	Motor cinta transportadora estación de descarga	Salida 10

Tabla 4 Señales E/S planta manufactura

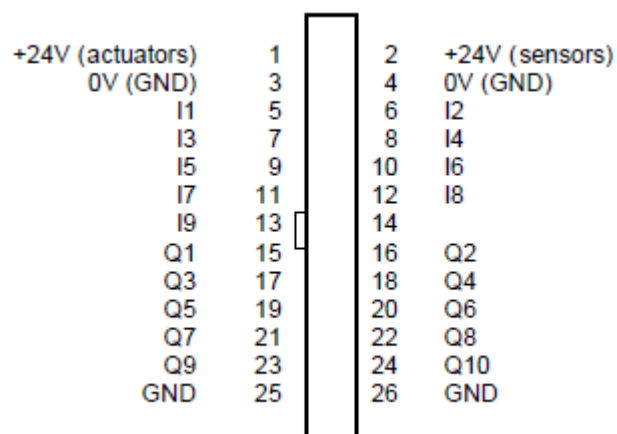


Ilustración 18 Distribución de pines conector IDC Planta Manufactura

4.1.2 Programación y Lógica Autómata

Para entender la secuencia de programación, se ha realizado un flujograma que expresa la idea general de todos los sensores, actuadores y temporizadores que van a interactuar.

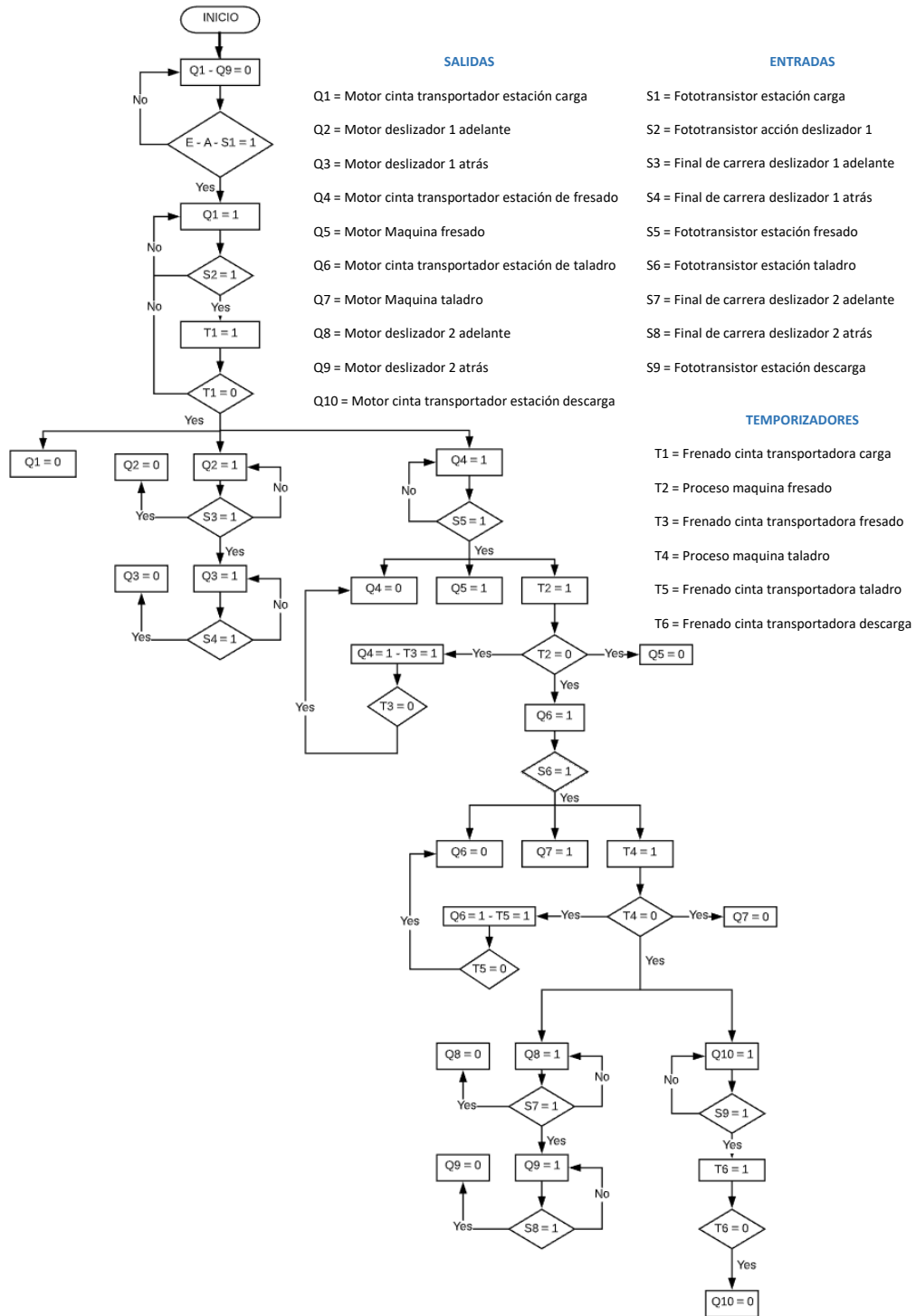


Ilustración 19 Flujograma de proceso Planta de Manufactura

La programación del autómatas se ha realizado con el software Sysmac Studio, utilizando el lenguaje ladder. Se toma en cuenta la secuencia de etapas mencionadas en el flujograma anterior pero adicionalmente se agregan otras, como un paro de emergencia que apaga todos los actuadores, variables para el manejo de HMI y seguridades para los actuadores.

La programación en ladder se encuentra en el apartado de anexos, sin embargo se detalla la programación realizada.

En las 4 primeras líneas de código, se realiza la secuencia para que el arranque el proceso pueda ser desde el HMI o desde una entrada externa como una botonera al igual que él paro.

También ya en pruebas muchas veces los deslizadores de la planta piloto no estaban en una posición inicial, por lo que se ha considerado que al arrancar el programa lo primero que hace la planta es ubicarse en una posición inicial o de origen.

Para el proceso del primer conveyor, el autómatas toma en cuenta que para empezar la secuencia el proceso debe estar seleccionado en automático, luego verifica que hay pieza disponible mediante el fototransistor de carga, si existe pieza arranca el transportador de carga y desplaza la pieza hasta llegar al segundo fototransistor el que acciona el deslizador, desplazando la pieza al segundo transportador de fresado.

Es la siguiente etapa del segundo conveyor, el deslizador desplaza la pieza al segundo transportador, una vez ahí al llegar al fototransistor del área de fresado, el transportador para y se enciende la máquina de fresado, al acabar su tiempo de ejecución, enciende nuevamente el transportador y también enciende el transportador de taladro, así desplaza la pieza hasta la siguiente etapa.

En la etapa del tercer conveyor, la pieza se desplaza hasta llegar al fototransistor de taladro, el mismo que le indica detenerse al transportador y accionar la máquina de taladro, una vez que acaba el proceso del taladro, se enciende nuevamente el transportador y lo lleva hasta el segundo desplazador para que lo pueda mover al cuarto transportador el de descarga.

En la última etapa la del cuarto conveyor, el desplazador mueve la pieza hacia el transportador de descarga, este se enciende y mueve la pieza hasta que es detectada por el fototransistor de descarga, el mismo que le da un tiempo de espera para frenar el transportador ya que existe un espacio entre la detección y la parte final del transportador.

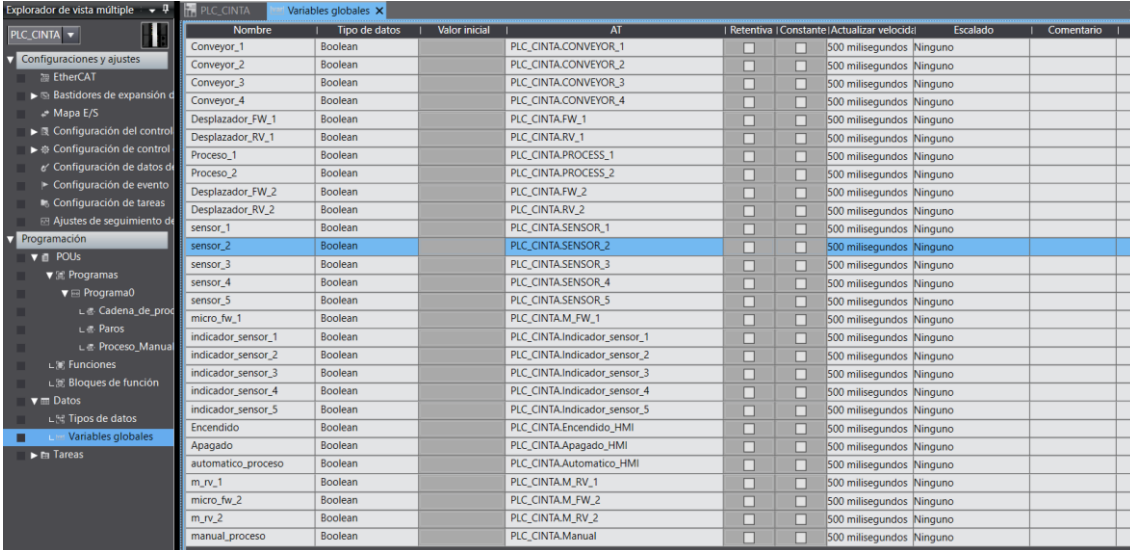
Como también necesitamos un proceso manual, se crea una segunda sección que permita manejar la planta piloto de forma manual.

Este proceso funciona siempre y cuando este accionado, en el HMI o en un botonera externa la opción manual y además solo funciona si tenemos pulsado el botón de accionamiento de cada actuador, es decir, que si lo dejamos de aplastar este para.

4.2 Desarrollo HMI

El diseño del HMI se ha realizado de igual manera con el software Sysmac Studio. Para empezar debemos ir a configuraciones, referencia de dispositivos y dispositivos internos y agregamos el PLC con el que estamos trabajando, y sobre todo la dirección IP, ya que de esta manera se comunicara el dispositivo con el HMI.

De esta manera podremos extraer todas las variables globales que existen en el PLC y así poder trabajar con las mismas en el HMI.



Nombre	Tipo de datos	Valor inicial	AT	Retentiva	Constante	Actualizar velocidad	Escalado	Comentario
Conveyor_1	Boolean		PLC_CINTA.CONVEYOR_1	<input type="checkbox"/>	<input type="checkbox"/>	500 milisegundos	Ninguno	
Conveyor_2	Boolean		PLC_CINTA.CONVEYOR_2	<input type="checkbox"/>	<input type="checkbox"/>	500 milisegundos	Ninguno	
Conveyor_3	Boolean		PLC_CINTA.CONVEYOR_3	<input type="checkbox"/>	<input type="checkbox"/>	500 milisegundos	Ninguno	
Conveyor_4	Boolean		PLC_CINTA.CONVEYOR_4	<input type="checkbox"/>	<input type="checkbox"/>	500 milisegundos	Ninguno	
Desplazador_FW_1	Boolean		PLC_CINTA.FW_1	<input type="checkbox"/>	<input type="checkbox"/>	500 milisegundos	Ninguno	
Desplazador_RV_1	Boolean		PLC_CINTA.RV_1	<input type="checkbox"/>	<input type="checkbox"/>	500 milisegundos	Ninguno	
Proceso_1	Boolean		PLC_CINTA.PROCESS_1	<input type="checkbox"/>	<input type="checkbox"/>	500 milisegundos	Ninguno	
Proceso_2	Boolean		PLC_CINTA.PROCESS_2	<input type="checkbox"/>	<input type="checkbox"/>	500 milisegundos	Ninguno	
Desplazador_FW_2	Boolean		PLC_CINTA.FW_2	<input type="checkbox"/>	<input type="checkbox"/>	500 milisegundos	Ninguno	
Desplazador_RV_2	Boolean		PLC_CINTA.RV_2	<input type="checkbox"/>	<input type="checkbox"/>	500 milisegundos	Ninguno	
sensor_1	Boolean		PLC_CINTA.SENSOR_1	<input type="checkbox"/>	<input type="checkbox"/>	500 milisegundos	Ninguno	
sensor_2	Boolean		PLC_CINTA.SENSOR_2	<input type="checkbox"/>	<input type="checkbox"/>	500 milisegundos	Ninguno	
sensor_3	Boolean		PLC_CINTA.SENSOR_3	<input type="checkbox"/>	<input type="checkbox"/>	500 milisegundos	Ninguno	
sensor_4	Boolean		PLC_CINTA.SENSOR_4	<input type="checkbox"/>	<input type="checkbox"/>	500 milisegundos	Ninguno	
sensor_5	Boolean		PLC_CINTA.SENSOR_5	<input type="checkbox"/>	<input type="checkbox"/>	500 milisegundos	Ninguno	
micro_fw_1	Boolean		PLC_CINTA.M_FW_1	<input type="checkbox"/>	<input type="checkbox"/>	500 milisegundos	Ninguno	
indicador_sensor_1	Boolean		PLC_CINTA.Indicador_sensor_1	<input type="checkbox"/>	<input type="checkbox"/>	500 milisegundos	Ninguno	
indicador_sensor_2	Boolean		PLC_CINTA.Indicador_sensor_2	<input type="checkbox"/>	<input type="checkbox"/>	500 milisegundos	Ninguno	
indicador_sensor_3	Boolean		PLC_CINTA.Indicador_sensor_3	<input type="checkbox"/>	<input type="checkbox"/>	500 milisegundos	Ninguno	
indicador_sensor_4	Boolean		PLC_CINTA.Indicador_sensor_4	<input type="checkbox"/>	<input type="checkbox"/>	500 milisegundos	Ninguno	
indicador_sensor_5	Boolean		PLC_CINTA.Indicador_sensor_5	<input type="checkbox"/>	<input type="checkbox"/>	500 milisegundos	Ninguno	
Encendido	Boolean		PLC_CINTA.Encendido_HMI	<input type="checkbox"/>	<input type="checkbox"/>	500 milisegundos	Ninguno	
Apagado	Boolean		PLC_CINTA.Apagado_HMI	<input type="checkbox"/>	<input type="checkbox"/>	500 milisegundos	Ninguno	
automatico_proceso	Boolean		PLC_CINTA.Automatico_HMI	<input type="checkbox"/>	<input type="checkbox"/>	500 milisegundos	Ninguno	
m_rv_1	Boolean		PLC_CINTA.M_RV_1	<input type="checkbox"/>	<input type="checkbox"/>	500 milisegundos	Ninguno	
micro_fw_2	Boolean		PLC_CINTA.M_FW_2	<input type="checkbox"/>	<input type="checkbox"/>	500 milisegundos	Ninguno	
m_rv_2	Boolean		PLC_CINTA.M_RV_2	<input type="checkbox"/>	<input type="checkbox"/>	500 milisegundos	Ninguno	
manual_proceso	Boolean		PLC_CINTA.Manual	<input type="checkbox"/>	<input type="checkbox"/>	500 milisegundos	Ninguno	

Ilustración 20 Variables globales del PLC usadas en el HMI

Para la monitorización y control de esta primera planta se ha creado 4 páginas, las mismas que tendrán una portada, un modo de selección ya sea automático o manual y otra donde se ejecuta el proceso automático y otra donde se ejecuta el proceso manual.

Al arrancar el sistema del HMI se puede observar que su primera página es la portada, donde se muestra el logo de la empresa en este caso la Universidad Politécnica de Valencia y las plantas piloto.

El botón INGRESAR nos permite desplazar a la siguiente página, de este modo generamos una interacción entre páginas.



Ilustración 21 Pagina portada HMI

Para interactuar entre páginas, nos dirigimos a la pestaña ver, y seleccionamos eventos y acciones. Esto nos permite decirle al botón que acción debe realizar, aquí seleccionaremos la opción Show Page.

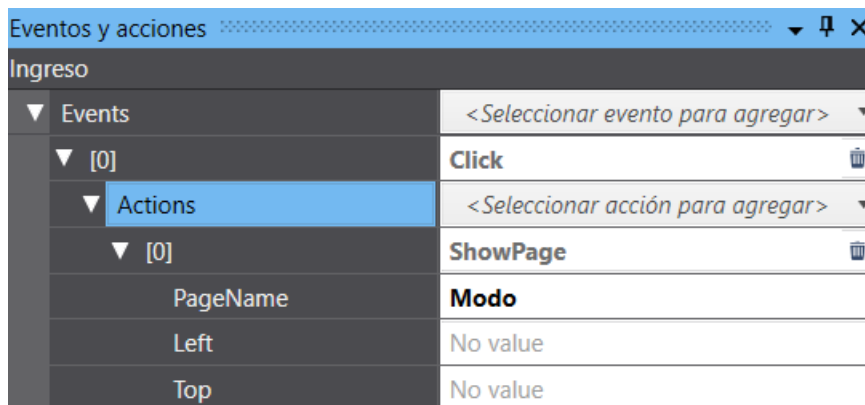


Ilustración 22 Acción Show Page para interactuar entre paginas

Y le indicamos a que pagina debe dirigirse en PageName, para este caso, al momento de pulsar INGRESAR el botón se dirigirá a la siguiente página que es MODO.

En la página MODO, podremos seleccionar dos opciones entre automático o manual y al igual que en la portada este nos dirige a la página correspondiente, adicional a estos botones disponeos en la parte inferior izquierda un botón con la forma de una casa, esto nos permitirá regresar a la página de portada en cualquier comento.



Ilustración 23 Pagina Modo HMI

En la página AUTOMATICO, tendremos dos simples opciones, encendido y apagado del proceso, además monitorizaremos que actuadores se está ejecutando mediante las flechas indicadoras y si ha sido detectado por los fototransistores.

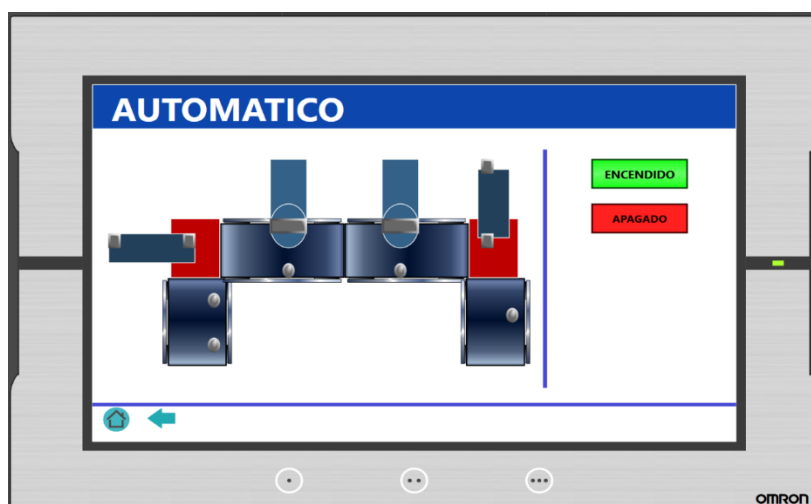


Ilustración 24 Pagina Automático HMI

Para poder asociar los botones a las variables globales del PLC lo realizamos de la siguiente manera:

- En las propiedades del botón en TYPE escogemos la opción que mejor nos convenga de acuerdo a la acción que queremos que ejecute, en este caso se requiere de un pulso momentáneo, por lo que escogemos la opción Momentary Button.
- Luego en COMPORTAMIENTO en la opción VARIABLE agregamos una nueva variable y seleccionamos la opción ASIGNAR A LA VARIABLE DEL CONTROLADOR.

- En DISPOSITIVO seleccionamos la variable a la que queremos asociar y creamos una nueva variable. De esta manera ya tendremos asociada a la variable del controlador.

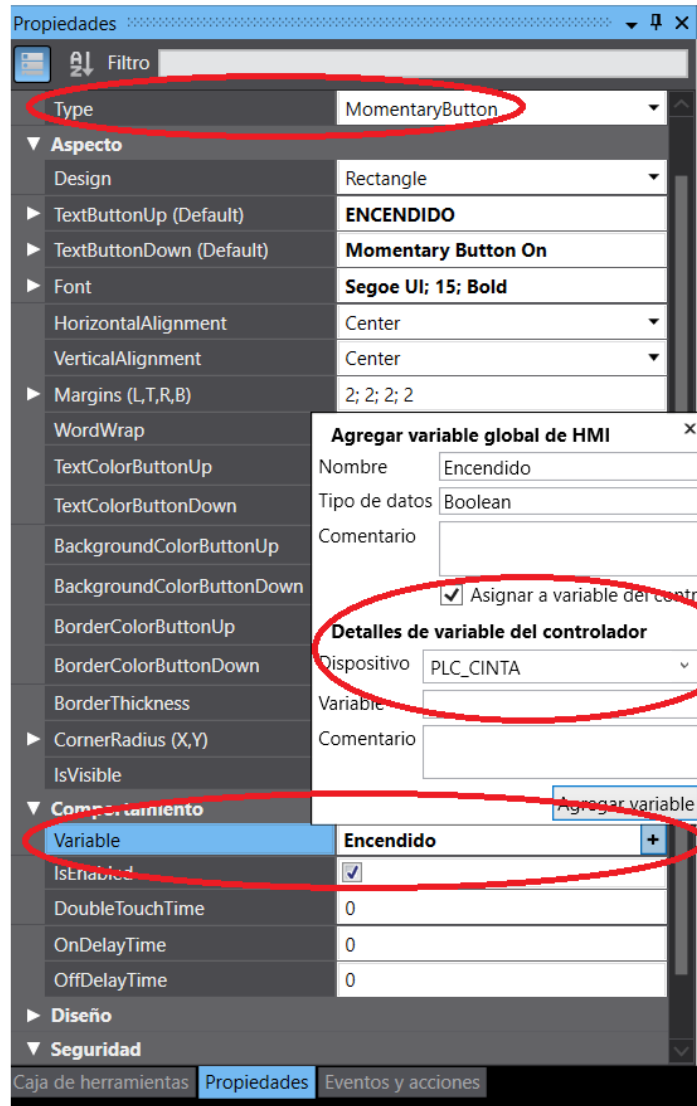


Ilustración 25 Asignar una variable del controlador al HMI

De esta forma se asocia todas las variables del controlador que deseemos usar para monitorizar o controlar tanto para el proceso manual como el automático.

En la página MANUAL, tenemos la opción de manejar de manera independiente cada uno de los actuadores y además nos permite inspeccionar si los indicadores están funcionando correctamente.

Los botones de accionamiento son de tipo momentáneo eso quiere decir, que si dejamos de presionar el botón, lo actuadores dejaran de funcionar.

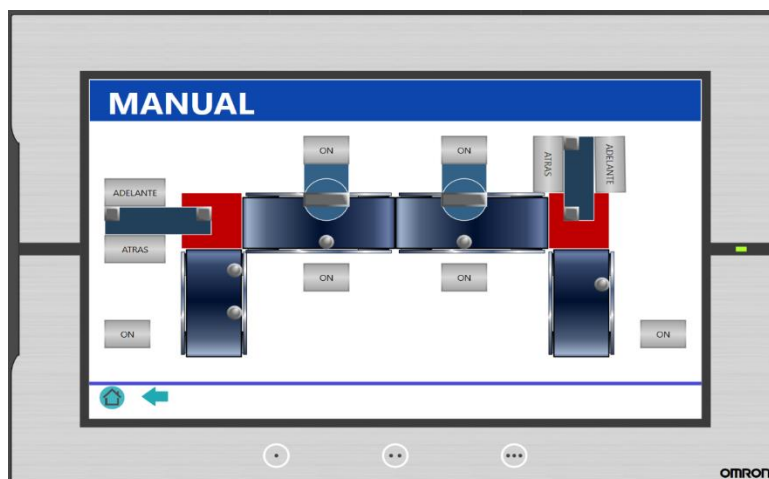


Ilustración 26 Pagina Manual HMI

4.3 Programación BeagleBone Black

En la segunda etapa del proyecto, se maneja el control de la planta piloto de carga y descarga, complemento para la ejecución de la planta de manufactura.

Para este procedimiento se adquiere las señales de la planta mediante una placa BeagleBone Black y posteriormente se envía las señales a un cliente OPC-UA mediante el levantamiento de un servidor OPC-UA.

4.3.1 Planta piloto de carga y descarga – Fischertechnik

Al igual que como se menciona en la planta piloto de manufactura, en este apartado vamos a identificar las señales que provienen de la planta de carga y descarga de la marca Fischertechnik y entender su funcionamiento. Se la puede encontrar con el nombre de Vacuum Gripper Robot 24V – Simulation.

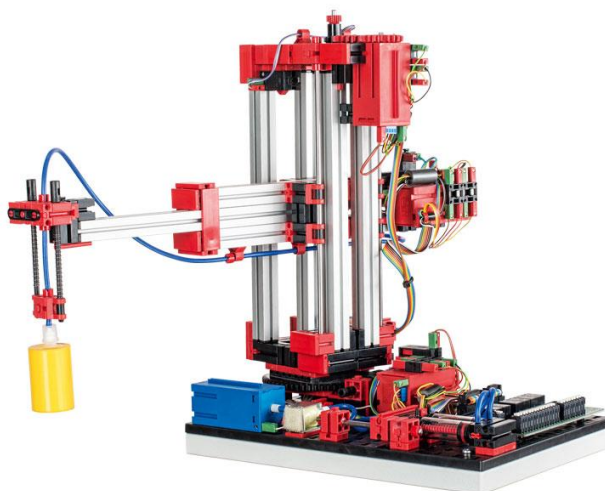


Ilustración 27 Aspecto Planta Piloto Carga y Descarga

Esta planta es un robot de 3 ejes con pinza de vacío que funciona de forma rápida, posicionando las piezas en un espacio tridimensional. Es un modelo ideal para entrenamiento, simulación y demostración de automatización industrial.

Posee un rango de trabajo: eje x (270°), eje y (adelante/atrás) 140 mm, eje z (arriba/abajo) 120 mm, 3 entradas digitales, 8 salidas digitales, 3 encoder 24V DC a una frecuencia de 1KHz.

Al igual que en la planta de manufactura, se puede conectar a un conector IDC de 34 pines con paso de 2.54 mm, o a terminales en serie de inserción.

Las señales de entrada, salida, alimentación y encoder se muestran en la siguiente tabla.

Numero de terminal	Descripción	Entrada / Salida
1	Fuente de alimentación actuadores (+)	24V DC
2	Fuente de alimentación sensores (+)	24V DC
3	Fuente de alimentación (-)	0V DC
4	Fuente de alimentación (-)	0V DC
5	Final de carrera eje vertical	Entrada 1
6	Final de carrera eje horizontal	Entrada 2
7	Final de carrera eje rotación	Entrada 3
8		
9	Impulso 1 encoder eje vertical	Encoder 1 A
10	Impulso 2 encoder eje vertical	Encoder 2 A
11	Impulso 1 encoder eje horizontal	Encoder 1 B
12	Impulso 2 encoder eje horizontal	Encoder 2 B
13	Impulso 1 encoder eje rotación	Encoder 1 C
14	Impulso 2 encoder eje rotación	Encoder 2 C
15		
16		
17	Motor eje vertical Arriba	Salida 1
18	Motor eje vertical Abajo	Salida 2
19	Motor eje horizontal Atrás	Salida 3
20	Motor eje horizontal Adelante	Salida 4
21	Motor eje rotación Horario	Salida 5
22	Motor eje rotación Antihorario	Salida 6
23	Compresor	Salida 7
24	Válvula	Salida 8

Ilustración 28 Señales E/S planta descarga – descarga

+24V (Aktoren / actuators)	1	2	+24V (Sensoren / sensors)
0V (GND)	3	4	0V (GND)
I1	5	6	I2
I3	7	8	
B1	9	10	B2
B3	11	12	B4
B5	13	14	B6
	15	16	
Q1	17	18	Q2
Q3	19	20	Q4
Q5	21	22	Q6
Q7	23	24	Q8
	25	26	
	27	28	
	29	30	
	31	32	
GND	33	34	GND

Ilustración 29 Distribución de pines conector IDC Planta Carga-Descarga

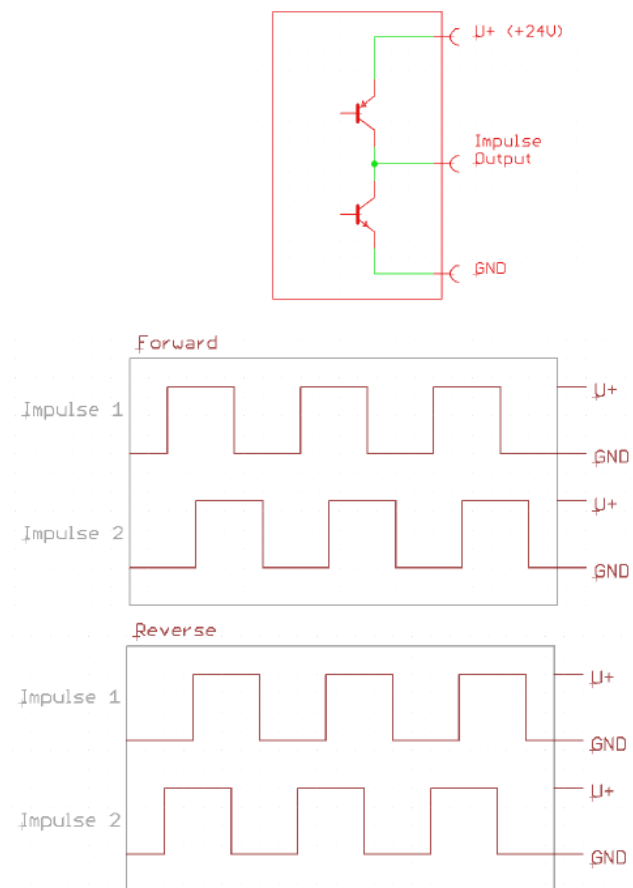


Ilustración 30 Diagrama conexión encoder y pulsos de dirección de rotación

4.3.2 Programación y Lógica BeagleBone Black

Al igual que en la planta anterior, en esta sección se muestra la lógica que lleva el proceso mediante un flujograma, en esta se identifica las entradas, salidas y señales de encoder que se usan para la ejecución del proceso.

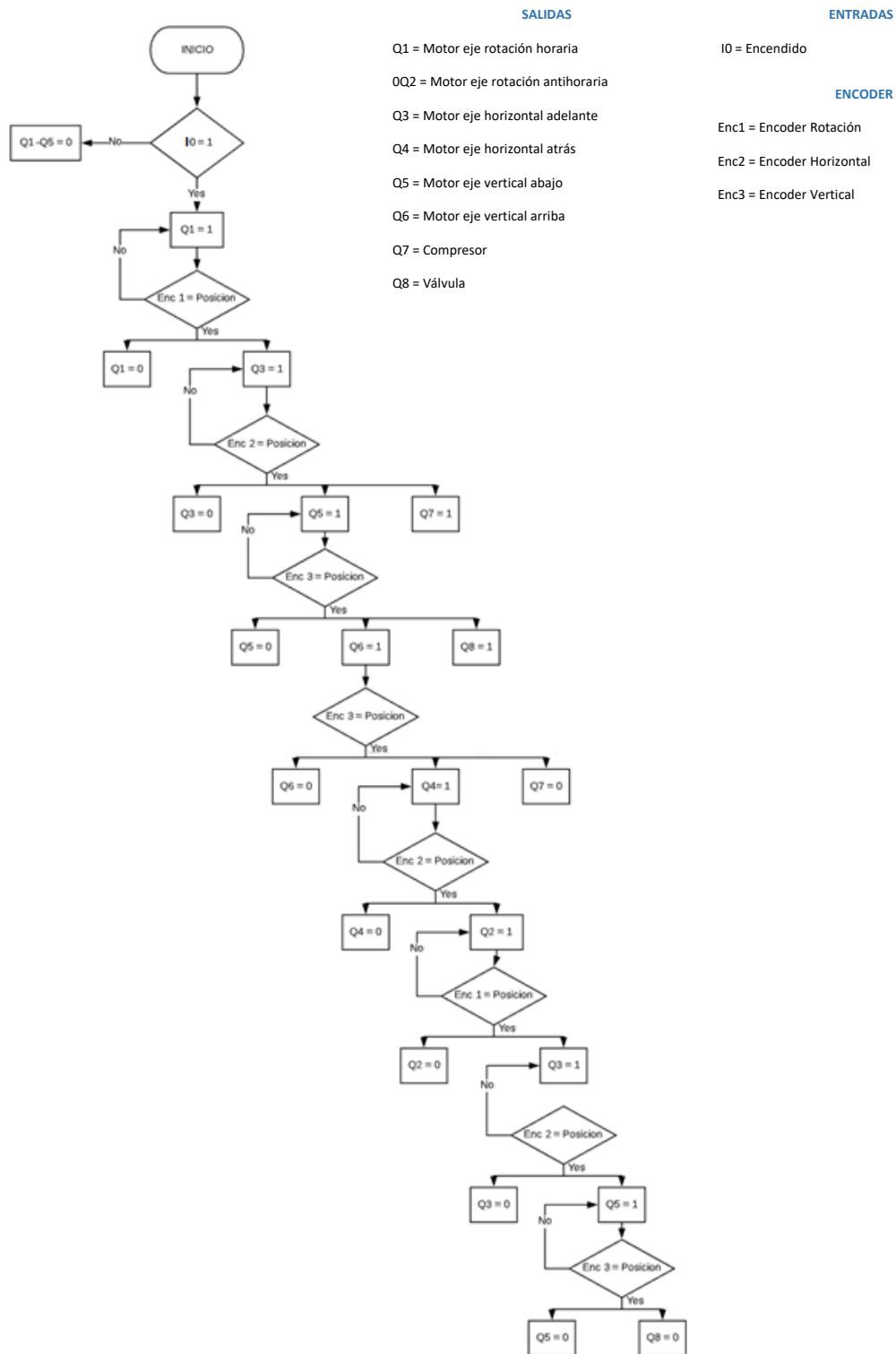


Ilustración 31 Flujograma de proceso Planta carga – descarga

La programación del proceso se ha realizado sobre la herramienta de programación node-red. Como se mencionó anteriormente, las últimas imágenes basadas en Debian 9 ya vienen preinstaladas y configuradas para un inicio automático.

Para la realización de este proyecto se ha utilizado la imagen bone-debian-9.5-iot-armhf-2018-10-07-4gb que se encuentra en la página <https://beagleboard.org/latest-images>.

Es decir se trabaja sobre las siguientes especificaciones:

- Operating System: Debian GNU/Linux 9 (stretch)
- Kernel: Linux 4.14.71-ti-r80
- Architecture: arm
- Node-RED version: v0.19.4
- Node.js version: v6.14.4

Para comenzar a trabajar ingresaremos a la parte visual de node-red usando el navegador, donde pondremos la dirección IP y el puerto 1880.

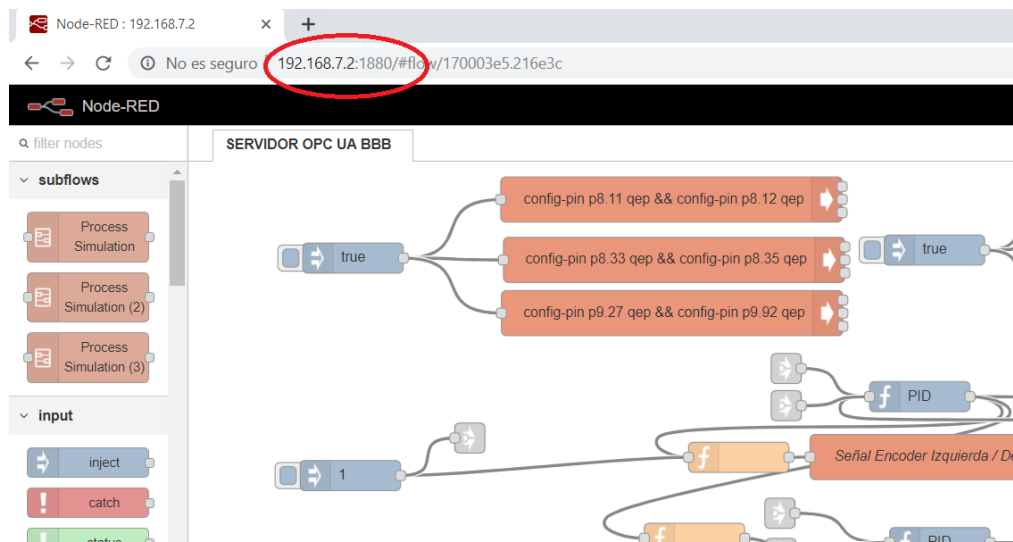


Ilustración 32 Aspecto visual Node-Red BBB

La adquisición de señales se realiza mediante los nodos de la paleta “node-red-contrib-gpio”, este nodo ya viene preinstalado en las últimas imágenes de la BeagleBone Black, sin embargo, si no viene instalada solo hay que poner “npm install node-red-contrib-gpio” en la terminal de la BeagleBone e instalarla.



Ilustración 33 Aspecto nodo GPIO

Una vez identificada los nodos GPIO de entrada y salida, la configuración es la siguiente:

- Damos doble click sobre la GPIO de entrada (GPIO IN) y agregamos un nuevo Board.

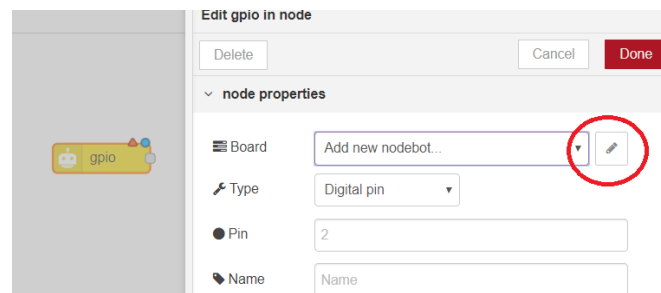


Ilustración 34 Agregar nuevo Board node GPIO

- Este nodo puede soportar varios dispositivos, en este caso escogemos nuestro dispositivo que es una BeagleBone Black y agregamos.

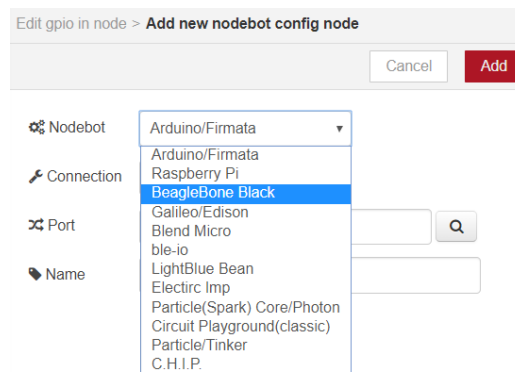


Ilustración 35 Selección de Board node GPIO

- Seleccionamos el tipo de entrada ya sea digital o análoga, para nuestro caso se esta trabajando con entradas digitales, y escogemos Digital pin

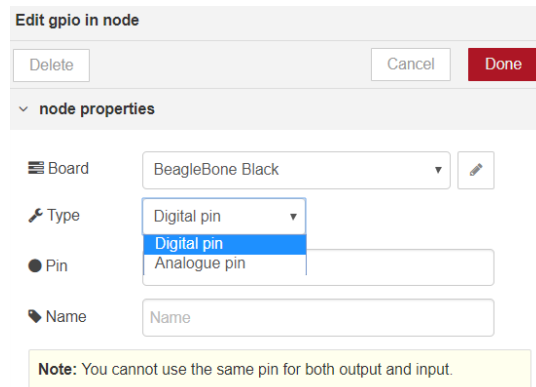


Ilustración 36 Selección de entrada digital input node GPIO

- Agregamos que GPIO vamos a utilizar como entrada en Pin y le ponemos un nombre para identificarlo, en este caso se ha identificado el pin P9_30 como entrada, el mismo que recibirá la señal del fin de carrera del eje vertical.

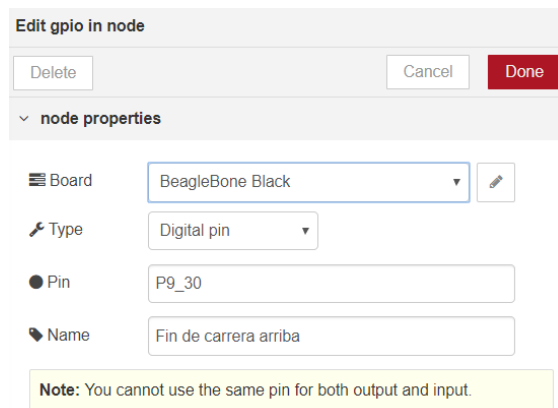


Ilustración 37 Pin utilizado node GPIO

Es necesario tomar en cuenta la nomenclatura para identificar las los pines que vamos a utilizar, ya sea para entrada o salida, ya que si se escribe mal por ejemplo “P9 30” o “P9-30” el nodo no identificara cual es el pin que estamos utilizando y saltará un error.

La nomenclatura utilizada se muestra en la siguiente tabla:

Pin ID	Soporte	Comentario
P8_7 o GPIO66	INPUT, OUTPUT	
P8_8 o GPIO67	INPUT, OUTPUT	
P8_9 o GPIO69	INPUT, OUTPUT	
P8_10 o GPIO68	INPUT, OUTPUT	
P8_11 o GPIO45	INPUT, OUTPUT	
P8_12 o GPIO44	INPUT, OUTPUT	

P8_13 o GPIO23	INPUT, OUTPUT, SERVO, PWM	
P8_14 o GPIO26	INPUT, OUTPUT	
P8_15 o GPIO47	INPUT, OUTPUT	
P8_16 o GPIO46	INPUT, OUTPUT	
P8_17 o GPIO27	INPUT, OUTPUT	
P8_18 o GPIO65	INPUT, OUTPUT	
P8_19 o GPIO22	INPUT, OUTPUT, SERVO, PWM	
P8_26 o GPIO61	INPUT, OUTPUT	
P9_11 o GPIO30	INPUT, OUTPUT	
P9_12 o GPIO60	INPUT, OUTPUT	
P9_13 o GPIO31	INPUT, OUTPUT	
P9_14 o GPIO50	INPUT, OUTPUT, SERVO, PWM	
P9_15 o GPIO48	INPUT, OUTPUT	
P9_16 o GPIO51	INPUT, OUTPUT, SERVO, PWM	
P9_17 o GPIO5	INPUT, OUTPUT	
P9_18 o GPIO4	INPUT, OUTPUT	
P9_19	I2C2 SCL	Reservado para I2C
P9_20	I2C2 SDA	Reservado para I2C
P9_21 or GPIO3	INPUT, OUTPUT, SERVO, PWM	
P9_22 o GPIO2	INPUT, OUTPUT, SERVO, PWM	
P9_23 o GPIO49	INPUT, OUTPUT	
P9_24 o GPIO15	INPUT, OUTPUT	
P9_26 o GPIO14	INPUT, OUTPUT	
P9_27 o GPIO115	INPUT, OUTPUT	
P9_30 o GPIO112	INPUT, OUTPUT	
P9_33 o A4	ANALOG	No ingrese más de 1.8V
P9_35 o A6	ANALOG	No ingrese más de 1.8V
P9_36 o A5	ANALOG	No ingrese más de 1.8V
P9_37 o A2	ANALOG	No ingrese más de 1.8V
P9_38 o A3	ANALOG	No ingrese más de 1.8V
P9_39 o A0	ANALOG	No ingrese más de 1.8V
P9_40 o A1	ANALOG	No ingrese más de 1.8V

P9_41 o GPIO20	INPUT, OUTPUT	
P9_42 o GPIO7	INPUT, OUTPUT, SERVO, PWM	
USR0	OUTPUT	Usuario incorporado LED 0
USR1	OUTPUT	Usuario incorporado LED 1
USR2	OUTPUT	Usuario incorporado LED 2
USR3	OUTPUT	LED de usuario incorporado 3 / LED predeterminado

Tabla 5 Pines admitidos del nodo GPIO en BeagleBone Black

En la tabla anterior se detalla como escribir los pines si son de entrada, salida, servo, pwm o análogo y que pines son admitidos para cada ejecución. El mismo procedimiento se realiza para las señales de salida.

Como se mencionó anteriormente, BeagleBone Black dispone de tres canales de lectura para encoder, por lo que se utiliza el nodo "EXEC" para realizar una llamada al sistema y leer el valor de posición de encoder y configurar los pines como qep.

Para configurarlos como qep, ponemos la siguientes línea en Command en cada nodo Exec de manera independiente:

- config-pin p8.11 qep && config-pin p8.12 qep
- config-pin p8.33 qep && config-pin p8.35 qep
- config-pin p9.27 qep && config-pin p9.92 qep

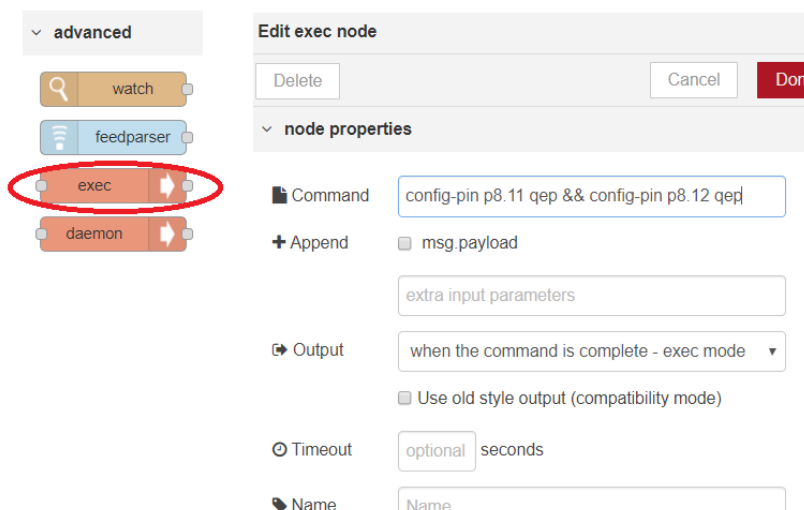


Ilustración 38 Nodo Exec y configuración de pin a qep

Mediante un nodo “INJECT” de entrada como TRUE, se ejecuta la configuración de los qep.

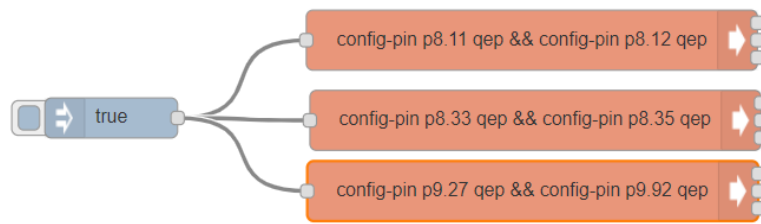


Ilustración 39 Configuración qep BeagleBone Black

Para leer los valores de posición del encoder utilizaremos las siguientes líneas en Command:

- `cat /sys/devices/platform/ocp/48300000.epwmss/48300180.eqep/position`
- `cat /sys/devices/platform/ocp/48304000.epwmss/48304180.eqep/position`
- `cat /sys/devices/platform/ocp/48302000.epwmss/48302180.eqep/position`

De esta manera obtendremos los valores de posición de los encoder cada vez que hagamos un llamado al sistema.

Una vez obtenido el control de los procesos, se levanta un servidor OPC-UA y se envía las señales a un cliente.

Para instalar el nodo OPC-UA primero nos dirigimos a la carpeta donde están ubicado los nodos “usr/local/lib/node_modules” y ejecutamos el comando de instalación: “npm install -g node-red-contrib-opcua --unsafe-perm --build-from-source”, de manera que sea una instalación global y sin permisos, ya que en nuestro caso sino quitábamos los permisos no permitía la instalación y saltaba errores.

Una vez realizado el anterior proceso, ya obtendremos en nuestra paleta los nodos OPC-UA, de manera que podremos levantar un servidor OPC-UA o un cliente OPC-UA, para poder leer, escribir y suscribirse por el servidor OPC-UA.

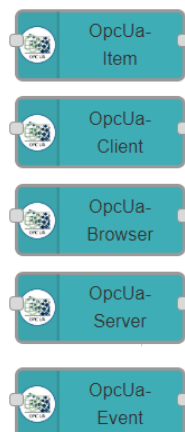


Ilustración 40 Nodos OPC-UA Node-Red

Para el desarrollo de este proyecto, se ha de utilizar un servidor OPC-UA por lo tanto utilizamos el nodo “OPC-UA server” y configuramos el servidor ingresando el “port” y el “Endpoint”, como se muestra en la siguiente imagen:

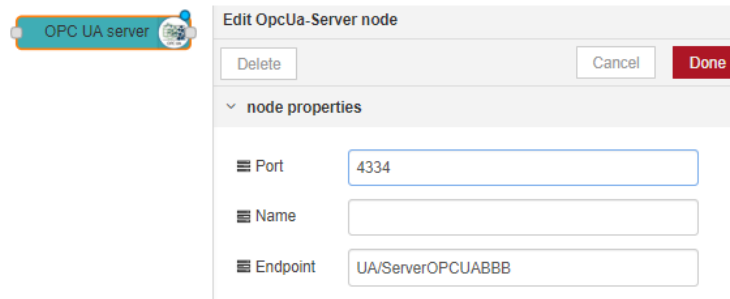


Ilustración 41 Configuración OPC-UA server

Los comandos que podemos utilizar para agregar equipos o borrar nodos mediante un “INJECT”, se los puede ver en la siguiente imagen, y son aquellas que nos ofrecen en las ayudas del nodo.



Ilustración 42 Información de ayuda nodo OPC-UA

Para el desarrollo del proyecto crearemos varias variables que permitirán el envío de datos desde el servidor OPC-UA al cliente y posteriormente mostrarlo en el sistema SCADA.

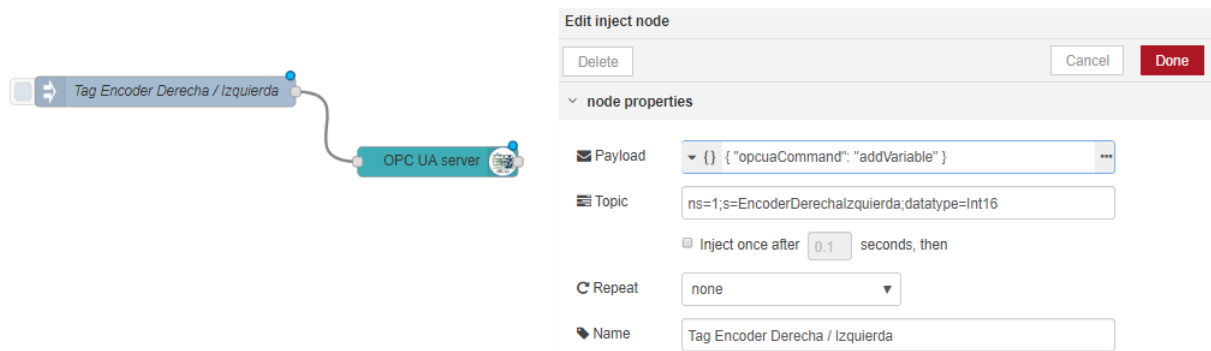


Ilustración 43 Agregar variable nodo OPC-UA

Para crear la variable se utiliza un nodo INJECT y enviamos un PAYLOAD con {"opcuaCommand":"addVariable"} y en TOPIC identificamos la variable agregándole un name space index (ns) = 1, para que este en la parte principal de la tabla de espacio de nombres, un identificador (s) y el tipo de dato (datatype).

Para enviar los datos al cliente en la variable creada, utilizamos un nodo de función (function node), en donde le indicaremos el tipo de mensaje, el nombre de la variable y el valor. En la siguiente imagen se muestra la variable, el nodo de función y su configuración.

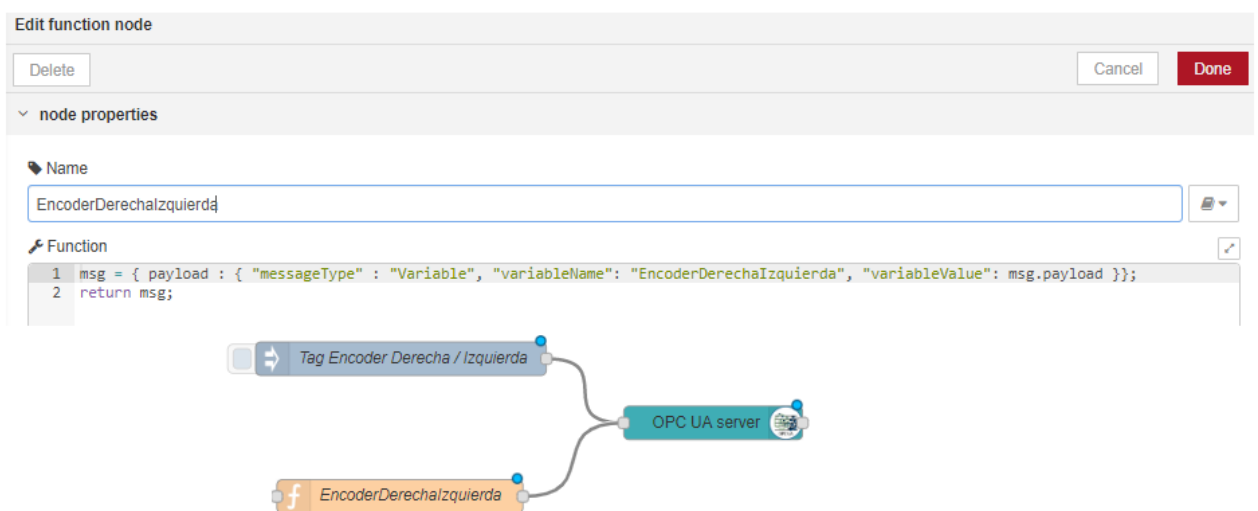


Ilustración 44 Envió de datos por variable OPC-UA

De esta manera retornamos los datos que llegan al nodo function y lo enviamos al cliente OPC-UA por la variable creada anteriormente con nombre de "EncoderDerechalzquierda".

Para el desarrollo de este proyecto vamos a supervisar varias variables que vienen de la planta de carga y descarga y todas esas se configuran de la misma manera lo único que cambiamos es el nombre de la variable y de donde obtendrán sus valores.

Una vez mencionado la configuración tanto para la adquisición de señales por el nodo GPIO y el envío por un servidor OPC-UA, se describe a continuación el proceso realizado para la ejecución del programa.

- Configuramos los pines como qep como muestra la ilustración 44.
- Mediante bloques de función indicaremos la lógica que identifique si la señal de inicio ha sido activada, de manera que encienda el motor correspondiente y mediante un nodo PID, se detiene el motor al setpoint indicado.

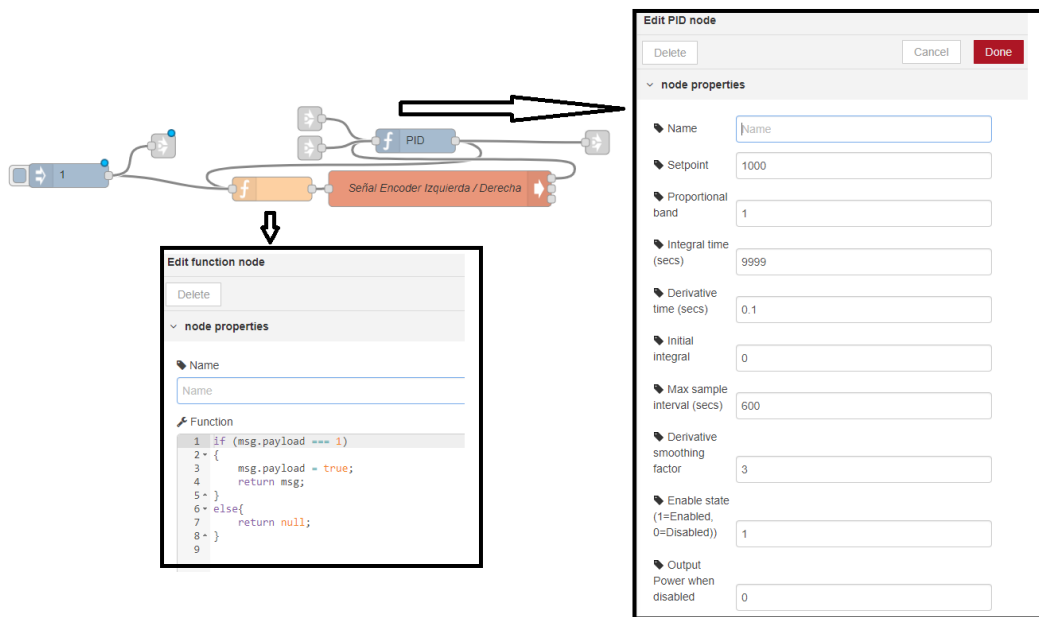


Ilustración 45 Encendido motor eje rotación a posición de set-point

- Una vez que se ha movido el motor del eje de rotación a la posición indicada, continua en secuencia el eje horizontal y luego el vertical a la posición de setpoint.

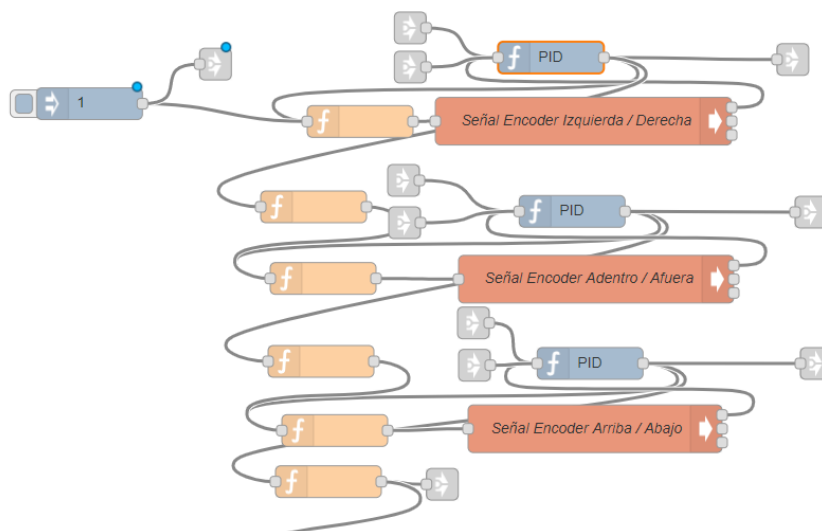


Ilustración 46 Secuencia del primer movimiento para descarga

- Una vez que ha recogido la pieza se realiza su segundo movimiento, donde se acciona el motor del eje vertical para separar la pieza de la banda de descargar y moverlo a la banda de carga, en este caso no se ha utilizado un nodo PID ya que no lo estamos moviendo a un lugar específico solo lo estamos llevando a una posición donde no se golpee con algún otro eslabón.

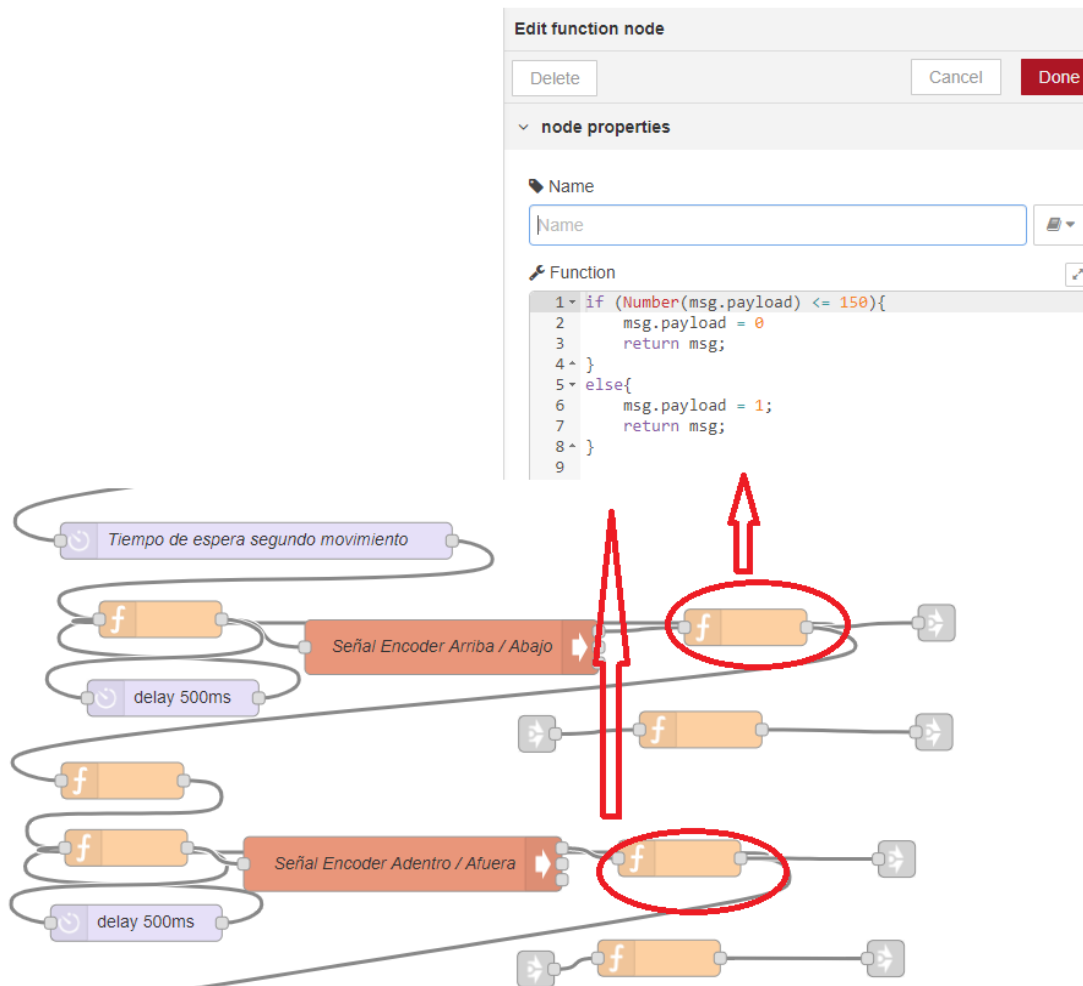


Ilustración 47 Secuencia del segundo movimiento hacia banda de carga

- Una vez que se encuentra lejos de la banda de descarga, realiza su tercer movimiento y lo llevamos hacia la banda de carga. Este paso es igual que el primer movimiento, es decir utilizamos un nodo PID para moverlo hacia la posición indicada y su proceso es igual que la indicada en la ilustración 51.
- Finalmente ya dejada la pieza en la banda de carga, realizamos un cuarto movimiento que es idéntico al segundo y lo que hace es alejarse de la banda de carga y quedarse en una posición inicial hasta que la señal indique que existe una pieza que debe ser descargada.

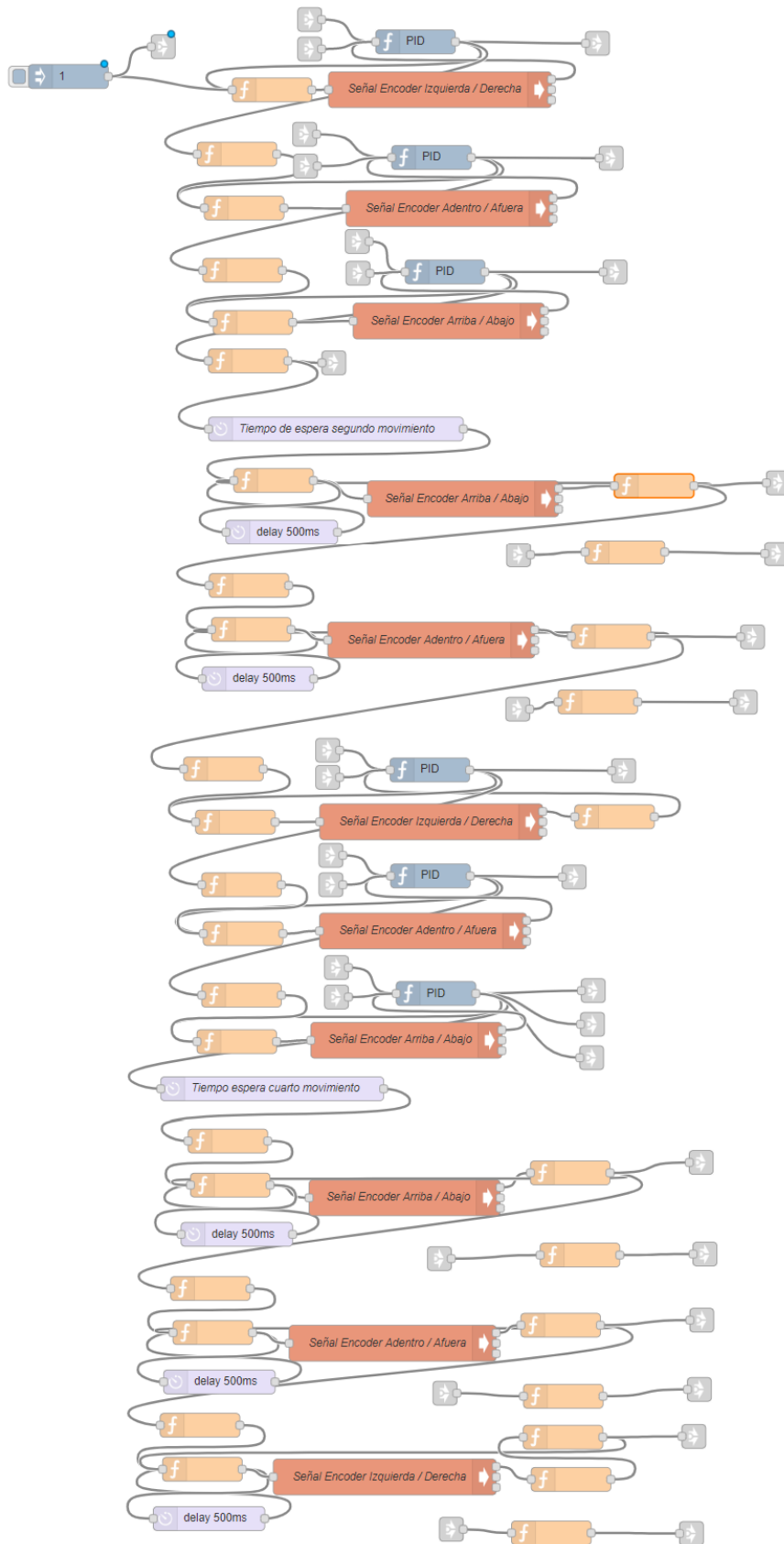


Ilustración 48 Secuencia de movimientos para proceso de carga y descarga en Node-Red

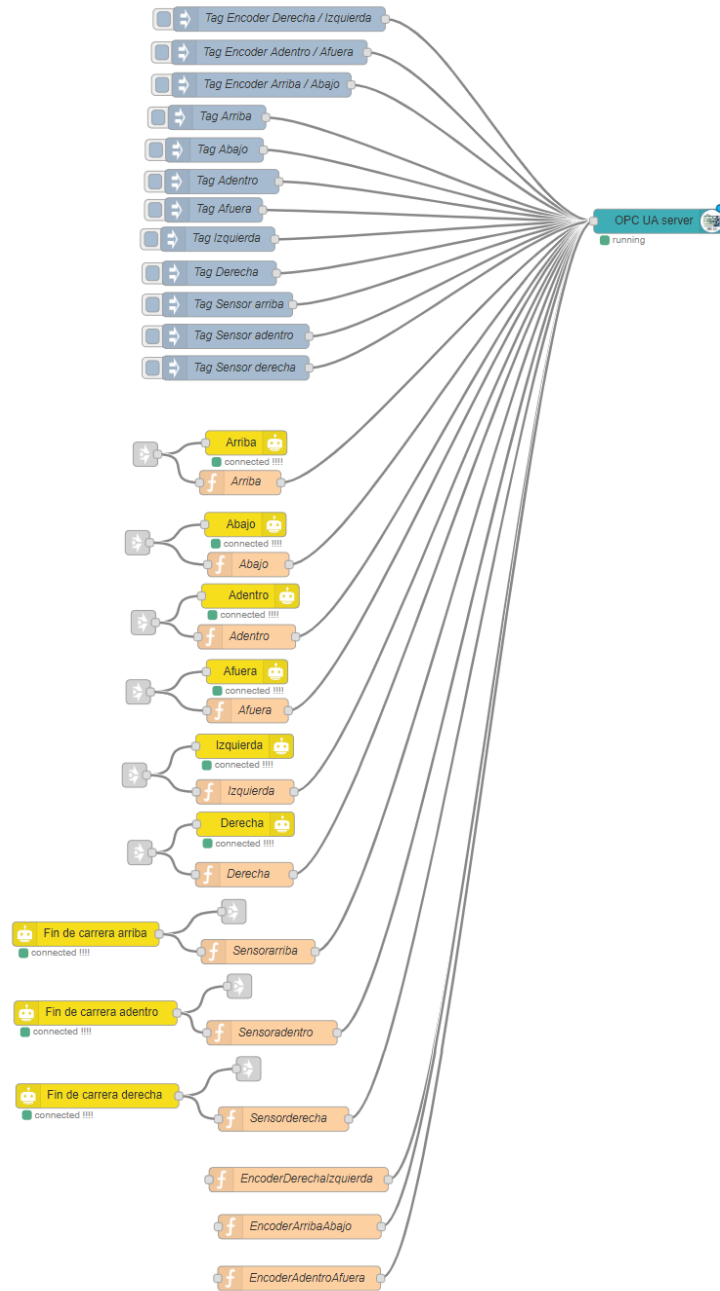


Ilustración 49 Variables y envío de datos a cliente OPC-UA

El proceso realizado cumple con el propósito de adquirir señales mediante las GPIO y enviarlas por medio de un servidor OPC-UA a un sistema SCADA, sin embargo, se puede notar que el proceso de control funciona bien cuando no existe demasiada carga de variables para el OPC-UA, esto se lo ha probado enviando solamente las tres señales de encoder al SCADA y el sistema de control va bien, pero cuando agregamos mas variables el sistema se satura y tienen a dejar de funcionar correctamente, de manera que los motores tienen a no parar a la medida correcta fallando en la medida del encoder o las variables indicadoras como los fines de carrera se demoran en responder.

4.4 Desarrollo sistema SCADA

En este capítulo se muestra el desarrollo y ejecución del sistema SCADA usando el software Ignition. Como se ha mencionado anteriormente, Ignition es una potente plataforma de desarrollo de sistemas SCADA y MES.

Ingresamos al Gateway del programa, introduciendo en la barra de direcciones de un navegador <http://localhost:8088/main> de manera que nos mostrara la pantalla de bienvenida del Gateway e iniciamos sesión ingresando el admin y password correspondientes.

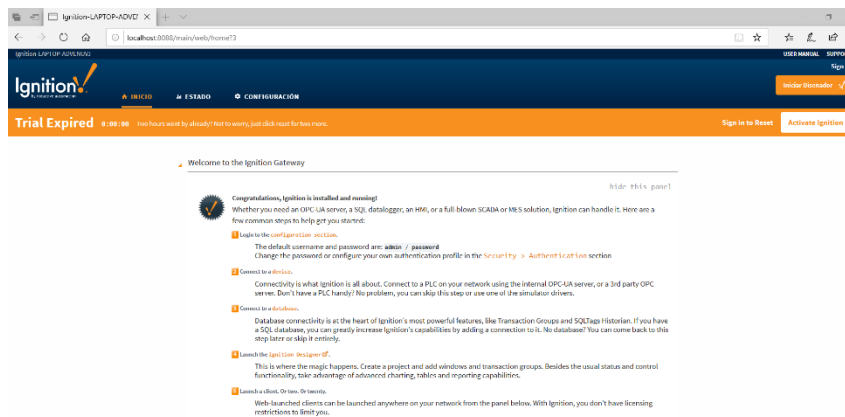


Ilustración 50 Aspecto Gateway Ignition

En primer lugar vamos a configurar el Gateway para recibir las señales que provienen del PLC de la siguiente manera:

- Nos dirigimos a las opciones de OPC-UA SERVER, entramos en DEVICES y creamos un nuevo device.



Ilustración 51 Crear un nuevo dispositivo OPC-UA SERVER

- Ignition nos permite conectarnos con los PLC Omron de la serie NJ así que escogemos esa opción y damos a next.

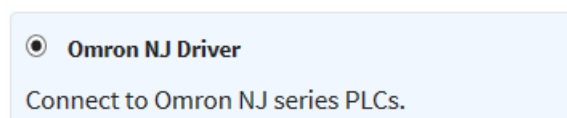


Ilustración 52 Conexión a Omron NJ Driver en Ignition

- Editamos el dispositivo, dándole un nombre, una descripción y colocamos el hostname correspondiente al dispositivo, en nuestro caso la IP del PLC NJ101 es 192.168.1.101 y le guardamos los cambios.

General	
Name	Omron
Description	PLC_Omron
Enabled	<input checked="" type="checkbox"/> (default: true)

Main	
Hostname	192.168.1.101 The hostname or IP address of the device.
Timeout	2000 The request timeout, specified in milliseconds. (default: 2.000)
Concurrency	2 The number of concurrently issued requests allowed. (default: 2)

Ilustración 53 Edición de dispositivo NJ101 OMRON

- De esta manera ya tenemos configurado el equipo, lo siguiente es agregar los tags de las variables que vienen del PLC en Ignition, la manera más sencilla es exportar las variables globales del PLC como XC-Designer y la copiamos en un editor de texto como un bloc de notas

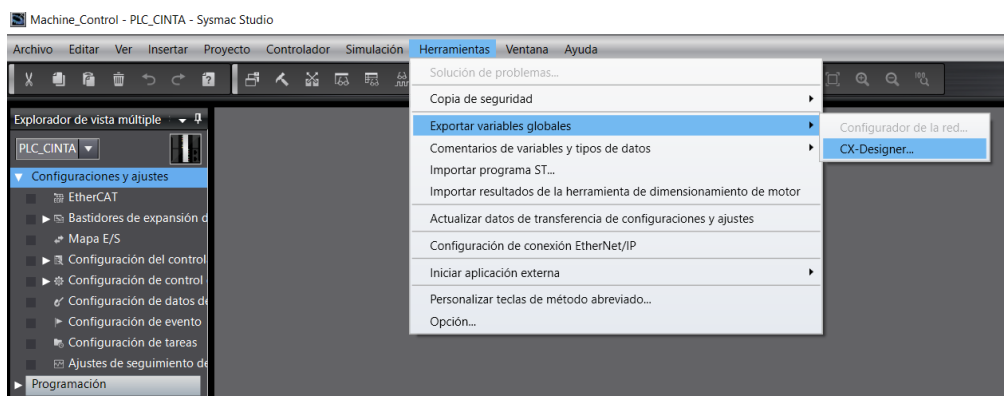


Ilustración 54 Exportar variables globales del PLC

*Sin título: Bloc de notas

HOST	NAME	DATATYPE	ADDRESS	COMMENT	TAGLINK	RW	POU
	CONVEYOR_1	BOOL	IOBus://rack#0/slot#0/Ch1_Out/Ch1_Out02			TRUE	RW
	EMERGENCY_STOP	BOOL	IOBus://rack#0/slot#0/Ch1_In/Ch1_In00			TRUE	RW
	CONVEYOR_3	BOOL	IOBus://rack#0/slot#0/Ch1_Out/Ch1_Out11			TRUE	RW
	SENSOR_5	BOOL	IOBus://rack#0/slot#0/Ch1_In/Ch1_In11			TRUE	RW
	M_FW_1	BOOL	IOBus://rack#0/slot#0/Ch1_In/Ch1_In15			TRUE	RW
	M_RV_1	BOOL	IOBus://rack#0/slot#0/Ch1_In/Ch1_In07			TRUE	RW
	M_FW_2	BOOL	IOBus://rack#0/slot#0/Ch1_In/Ch1_In14			TRUE	RW
	M_RV_2	BOOL	IOBus://rack#0/slot#0/Ch1_In/Ch1_In06			TRUE	RW
	CONVEYOR_2	BOOL	IOBus://rack#0/slot#0/Ch1_Out/Ch1_Out10			TRUE	RW
	CONVEYOR_4	BOOL	IOBus://rack#0/slot#0/Ch1_Out/Ch1_Out12			TRUE	RW
	FW_1	BOOL	IOBus://rack#0/slot#0/Ch1_Out/Ch1_Out00			TRUE	RW
	RV_1	BOOL	IOBus://rack#0/slot#0/Ch1_Out/Ch1_Out08			TRUE	RW
	FW_2	BOOL	IOBus://rack#0/slot#0/Ch1_Out/Ch1_Out01			TRUE	RW
	RV_2	BOOL	IOBus://rack#0/slot#0/Ch1_Out/Ch1_Out09			TRUE	RW
	PROCESS_1	BOOL	IOBus://rack#0/slot#0/Ch1_Out/Ch1_Out03			TRUE	RW
	PROCESS_2	BOOL	IOBus://rack#0/slot#0/Ch1_Out/Ch1_Out04			TRUE	RW
	Manual	BOOL		TRUE	RW		
	Indicador_sensor_1	BOOL			TRUE	RW	
	Indicador_sensor_2	BOOL			TRUE	RW	
	Indicador_sensor_3	BOOL			TRUE	RW	
	Indicador_sensor_4	BOOL			TRUE	RW	
	Indicador_sensor_5	BOOL			TRUE	RW	
	Encendido_HMI	BOOL		TRUE	RW		
	Apagado_HMI	BOOL		TRUE	RW		
	Automatico_HMI	BOOL		TRUE	RW		

Ilustración 55 Variables globales esportadas como CX-Designer

- Volvemos al Gateway de Ignition y en la opción more, damos a la opción tags.

Devices

Name	Type	Description	Enabled	Status	
Omron	Omron NJ Driver	PLC_Omron	true	Connecting	More edit tags delete

Ilustración 56 Agregar tags Ignition

- En el Manage tags, examinamos el archivo donde guardamos las variables y la importamos, de esta manera ya tenemos agregado las variables globales que vienen del PLC OMRON y que las usaremos para el control y supervisión en el sistema SCADA.

Manage Tags

Import a List of Tags. Warning: This will replace the contents of the tag table

C:\Users\santi\Downloads\VariablesGlobalesPLC.txt Examinar...

overwrite append Import

Tags Export Table to TSV

Name	DataType	Chars	Elements	R/W
<input type="checkbox"/> Apagado_HMI	BOOL			RW
<input type="checkbox"/> Automatico_HMI	BOOL			RW
<input type="checkbox"/> CONVEYOR_1	BOOL			RW
<input type="checkbox"/> CONVEYOR_2	BOOL			RW
<input type="checkbox"/> CONVEYOR_3	BOOL			RW

Ilustración 57 Importar variables globales a la lista de tags

Lo siguiente es configurar el Gateway para que se conecte con el servidor OPC-UA levantado en la placa BeagleBone Black. Al igual que en el proceso anterior seguiremos los siguientes pasos:

- os dirigimos a la opción OPC CONNECTIONS y es cogemos la opción SERVERS y le damos Create new OPC Server Connection.



Ilustración 58 Crear una nueva conexión OPC server

- Escogemos la opción OPC-UA y le damos a next.

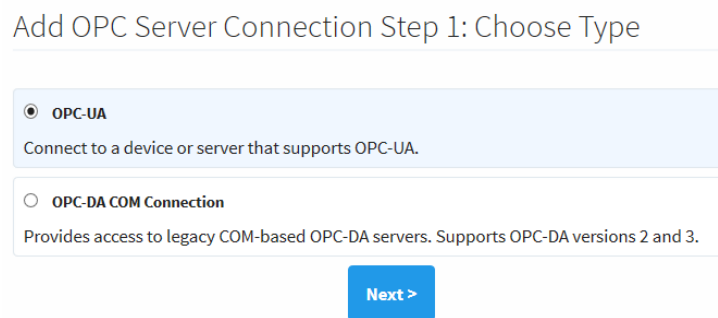


Ilustración 59 Agregar una nueva conexión OPC server

- Agregamos el endpoint del OPC-UA del servidor y el puerto establecido en la ilustración 46. En este caso es: `opc.tcp://192.168.7.2:4334` y le damos a discover.

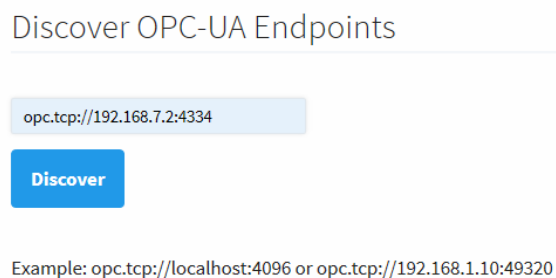


Ilustración 60 End-point del Servidor en BBB

- De esta manera ya tenemos enlazado el servidor OPC-UA del sistema empujado con el sistema SCADA de Ignition, mostrando el estado como conectado.

IGNITION



BEAGLEBONE BLACK

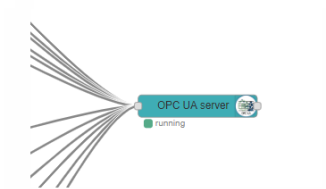


Ilustración 61 OPC-UA running y conexión Ignition conectado

Una vez configurado el Gateway, abrimos el editor de diseño de Ignition para realizar la interfaz del SCADA. El Username es: admin y el password: password.

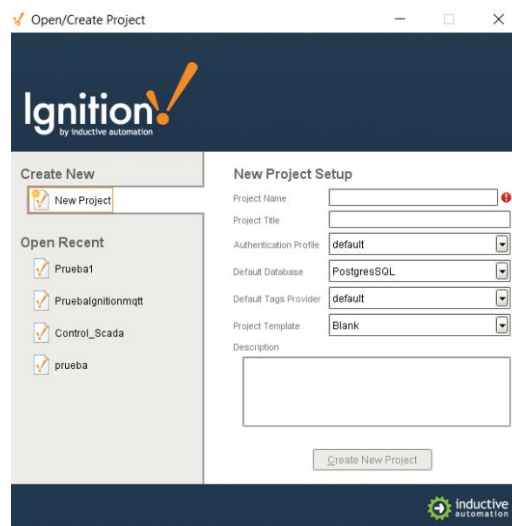


Ilustración 62 Abrir o crear nuevo proyecto

Si las conexiones OPC-UA están conectadas, se puede encontrar en el apartado de OPC BROWSER el contenido de las conexiones definidas en la placa BeagleBone Black y las importadas del PLC OMRON NJ101.

Esto significa que estas variables están disponibles para el diseño del SCADA, y para usarlas simplemente hay que arrastrarlas al componente de trabajo, ya sea un indicador, un pulsador, etc.

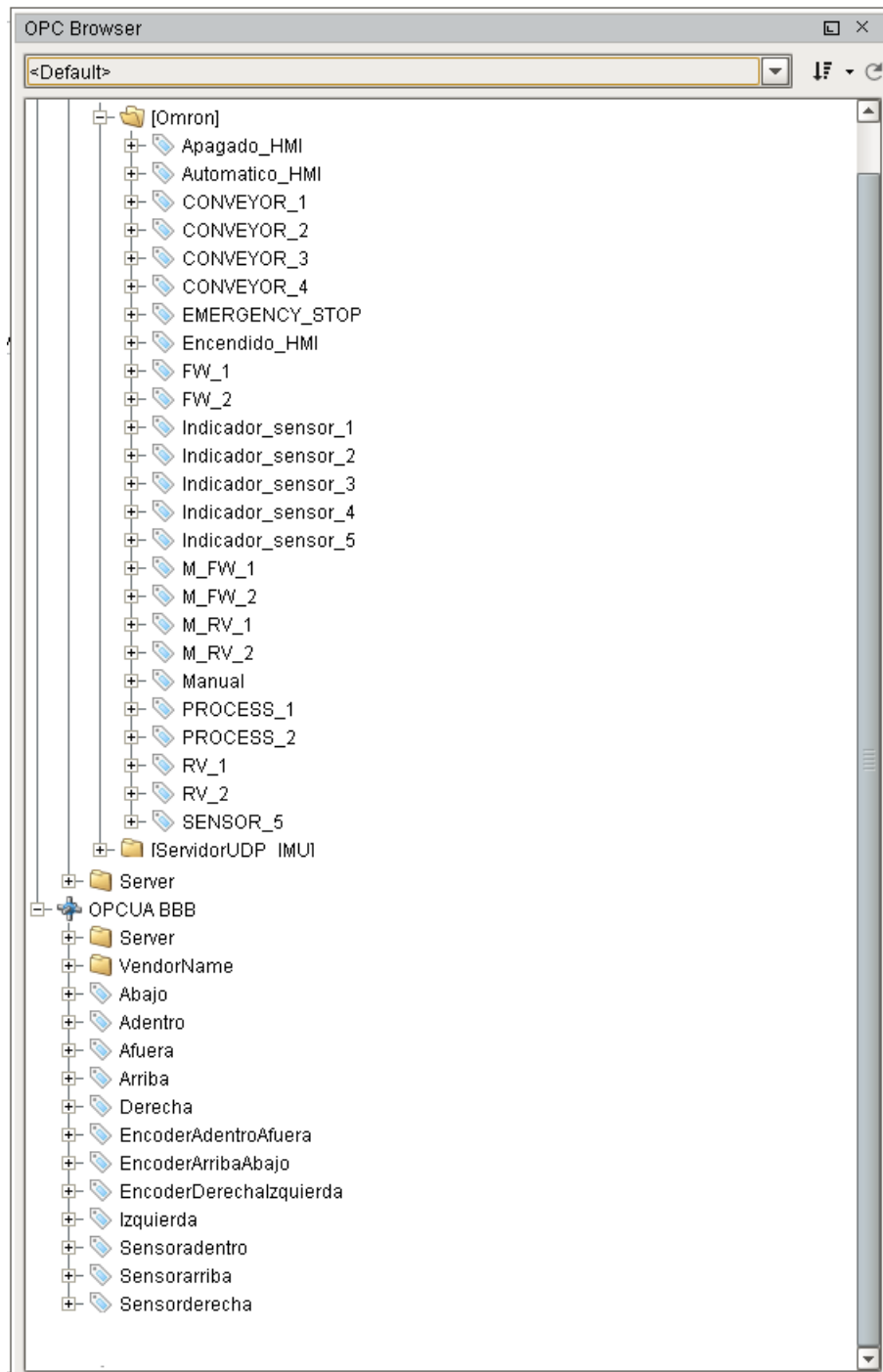


Ilustración 63 OPC Browser – tags

Para el desarrollo de este proyecto se ha utilizado varios componentes de trabajo que son:

- Display – image: Agrega imágenes externas.
- 2-state toggle button: Cambia entre dos estados.
- Numeric text field: Asocia el valor que se le asigne a la variable.
- Label: Texto fijo.

Como se ha mencionado si se quiere agregar imágenes externas como el logo de la universidad, se debe escoger en display la opción image, y se escoge la imagen que requerida.

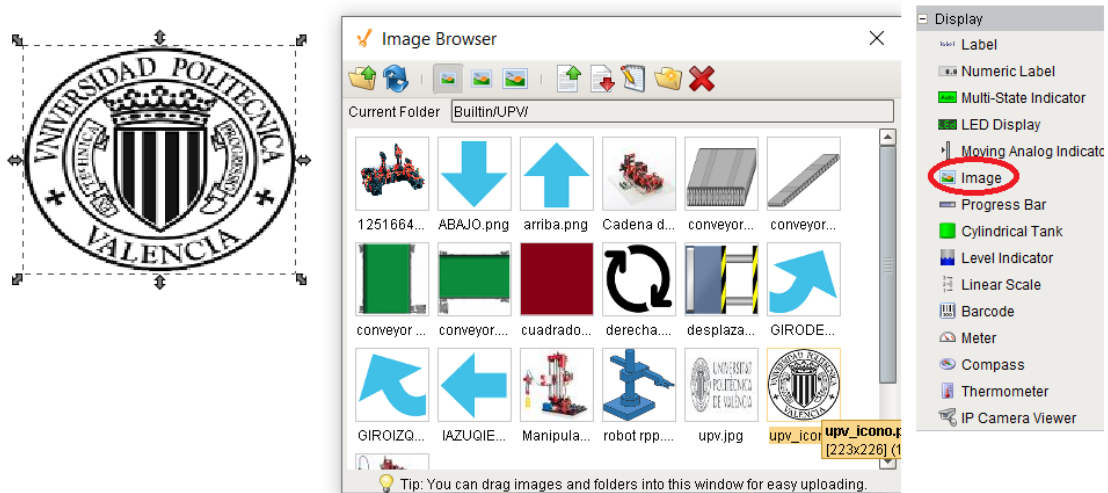


Ilustración 64 Image Browser Ignition

Los botones de dos estados, permiten realizar el control de algún proceso es decir pueden ser de lectura y escritura, de tal manera que podemos accionar algún motor y con otro podemos inspeccionar algún indicador..

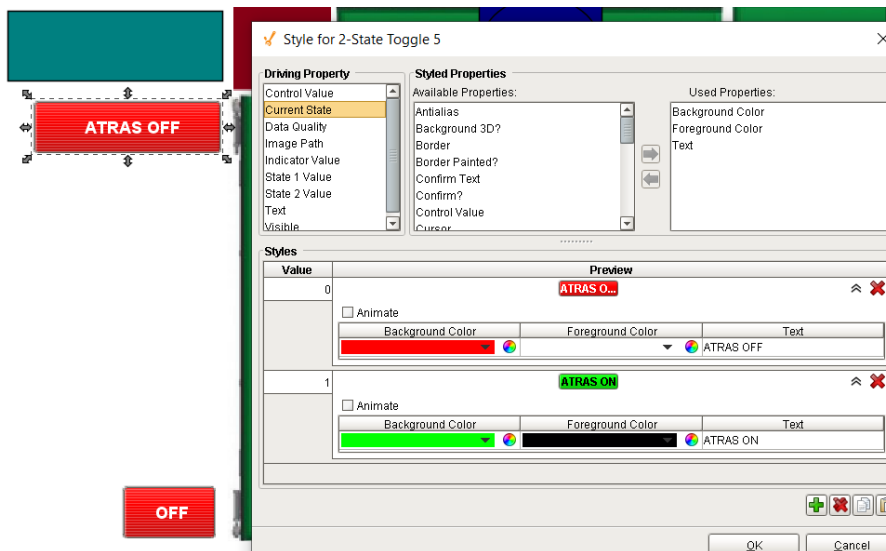


Ilustración 65 2 - State Toogle Ignition

Para visualizar el encoder, el componente numeric text field, permite visualizar su estado.

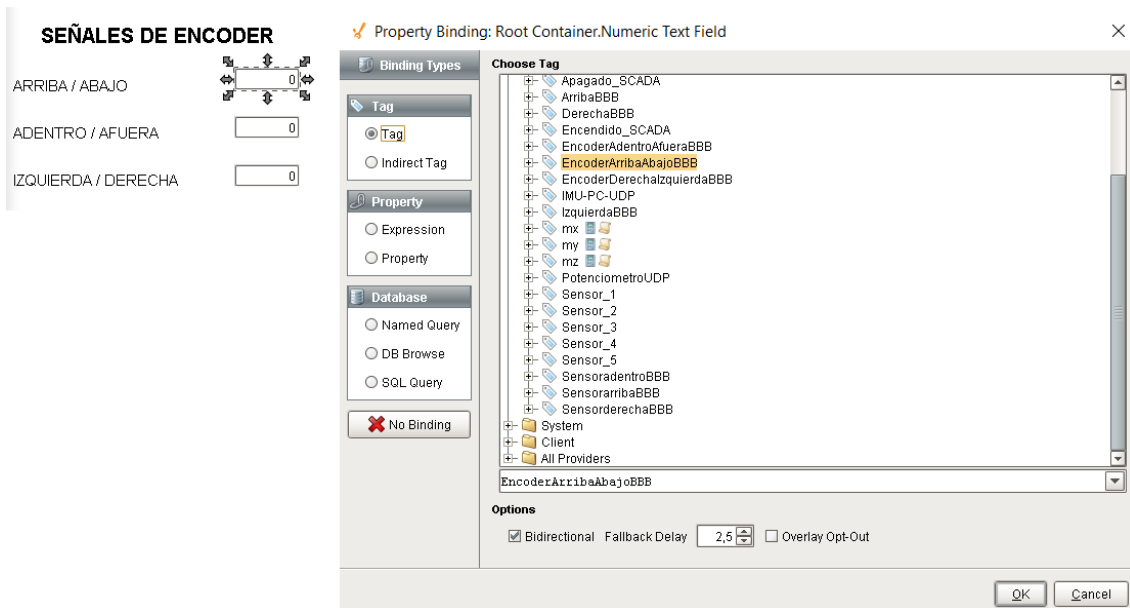


Ilustración 66 Numeric Text Field Ignition

De esta manera, usando los diferentes componentes de edición, se ha logrado diseñar la pantalla principal de control y supervisión SCADA de las plantas piloto, como se muestra en la siguiente imagen.

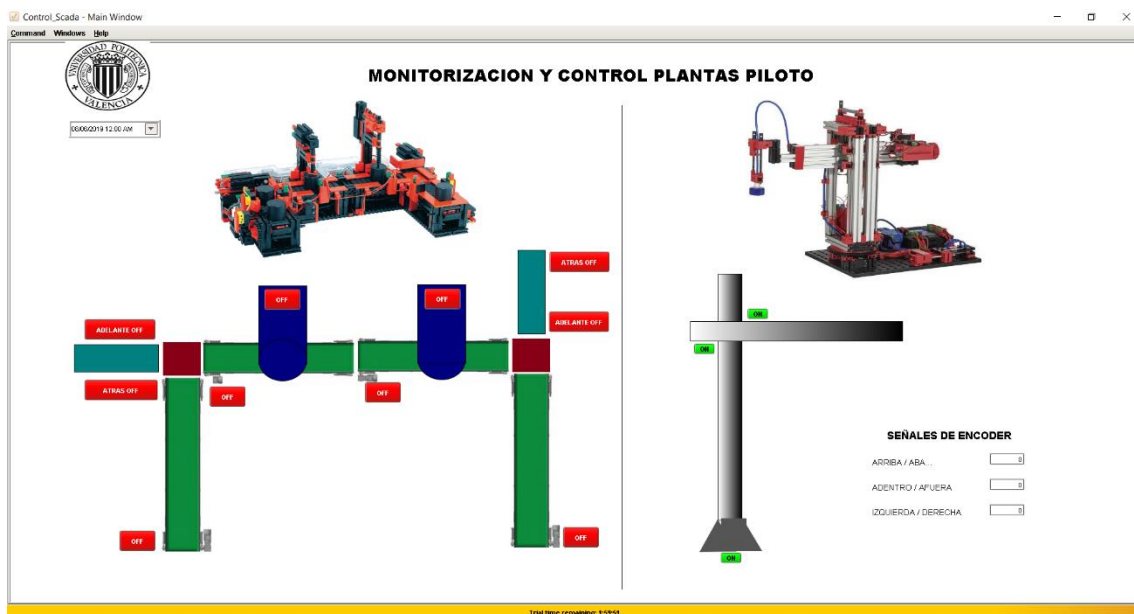


Ilustración 67 Pantalla SCADA

5. Conclusiones y trabajos futuros

En este capítulo se exponen las conclusiones obtenidas del trabajo realizado de acuerdo a los objetivos propuestos, además se exponen los trabajos futuros que pueden ser relevantes para la mejora del desarrollo del proyecto.

5.1 Conclusiones

De acuerdo a lo expuesto a lo largo de este trabajo podemos afirmar que se ha conseguido desarrollar un sistema de monitorización de planta piloto mediante servidor OPC-UA empotrado, enlazando un autómatas programable, un dispositivo HMI y un sistema empotrado como la placa BeagleBone Black.

Se ha logrado implementar un programa que actúe sobre un controlador lógico programable y enviar sus señales usando sus variables globales para el control mediante un HMI, cumpliendo con el primer objetivo específico propuesto, el mismo que se expone en los capítulos 4.1 y 4.2.

Por medio del protocolo CIP se ha logrado enviar las variables globales del PLC Omron a un servidor OPC-UA en Ignition, de manera que se puede controlar la planta piloto desde un sistema SCADA, cumpliendo con el segundo objetivo específico y expuesto en el capítulo 4.4.

La utilización de la herramienta de programación Node-Red como sus nodos y paletas, permitió que el desarrollo de un sistema OPC-UA empotrado sea más sencillo, ya que simplemente hay que extraer el nodo y configurarlo. Cabe recalcar, que node-red está desarrollado para conectar dispositivos, por lo que el desarrollar un programa de control dentro de su interfaz resulta un poco complicado, ya que muchas condiciones e instrucciones resultan más difíciles de hacerlas, sin embargo, se ha conseguido realizar un sistema de control en la herramienta de programación node-red y enviar las señales a un cliente OPC-UA mediante el levantamiento de un servidor OPC-UA empotrado, de manera que se ha logrado cumplir con el tercer y cuarto objetivo específico los mismos que se exponen en los capítulos 4.3 y 4.4, respectivamente.

Mediante Ignition que es un software que permite el desarrollo de sistemas SCADA de manera más sencilla y escalable, se ha logrado realizar la interfaz gráfica expuesta en el quinto objetivo específico, de manera que se puede controlar y supervisar el proceso de las plantas piloto, la misma que se encuentra en el capítulo 4.4.

El desarrollo de la últimas imágenes basadas en debian de la placa BeagleBone Black, permitió que el desarrollo de un servidor OPC-UA empotrado resulte más intuitiva, ya que solo debemos ingresar a la interfaz gráfica de node-red e instalar el nodo de OPC-UA, de manera que al ser una placa muy pequeña, potente y tener entradas especiales como la lectura de encoder, permite el desarrollo de varios proyectos que presentan una complejidad alta.

5.2 Trabajos futuros

De acuerdo al desarrollo del proyecto y las mejoras que se ha podido observar para el rendimiento del proceso, se exponen a continuación sugerencias a trabajos futuros que sean relevantes a la mejora del proyecto:

- Desarrollar un sistema MES (Manufacturing Execution System, de manera que se controle datos como tiempos de producción, tiempo de ejecución, controles de procesamiento y controles de rendimiento para la toma de decisiones. Al igual que para el desarrollo del sistema SCADA, el software Ignition también dispone de herramientas para sistemas MES.
- A nivel industrial es muy importante controlar alarmas y eventos que suscitan en un proceso, ya que tanto para el área de producción, el área de mantenimiento, y otras áreas, estos sucesos repentinos pueden ocasionar perdidas, por lo que se presenta desarrollar un sistema que genere históricos de alarmas y eventos para el sistema de control, de manera que pueda ser útil para las áreas en sus toma de decisiones. Al igual que en el sistema anterior, Ignition posee herramientas que nos ayudan a la gestión de alarmas y eventos.
- Node-red permite crear nuevos nodos e instalarlos en la paleta de diseño, por lo que se sugiere desarrollar un nodo que permita extraer los valores de posición de encoder de la placa BeagleBone Black usando código nativo y que tenga variables configurables como tiempo de muestreo, configuración de pines como entrada de lectura de encoder y sistema de conversión para medidas, de manera que se mejore el rendimiento de adquisición de datos y el sistema de control sin saturar los demás procesos.

6. Bibliografía

- beagleboard.org. (s.f.). *BeagleBone Black*. Obtenido de <https://beagleboard.org/black>
- Inductiveautomation. (s.f.). *Ignition*. Obtenido de <https://inductiveautomation.com>
- InductiveAutomation. (s.f.). *InductiveAutomation*. Obtenido de <https://docs.inductiveautomation.com/display/DOC/Omron+Driver>
- Landsdorf, K. (2016). *node-red-contrib-opcua* . Obtenido de <https://flows.nodered.org/node/node-red-contrib-opcua>
- Manzano Alcide, J. A. (28 de 09 de 2018). Implementación de un sistema de supervisión y control distribuido en myRIO con servidor OPC-UA. Valencia, España.
- Matrikon. (s.f.). *OPC UA*. Obtenido de <https://www.matrikonopc.es/opc-ua/index.aspx>
- NodeRed. (s.f.). *node-red-contrib-pid*. Obtenido de <https://flows.nodered.org/node/node-red-contrib-pid>
- NodeRed. (s.f.). *Running on BeagleBone Boards*. Obtenido de <https://nodered.org/docs/getting-started/beaglebone>
- NPM. (s.f.). *node-red-contrib-gpio*. Obtenido de <https://www.npmjs.com/package/node-red-contrib-gpio>
- Rodriguez, M. (s.f.). *OPC*. Obtenido de <https://revistadigital.inesem.es/gestion-integrada/protocolo-opc-ua-caracteristicas-y-aplicaciones/>
- Wonderware. (s.f.). *¿Que es SCADA?* Obtenido de <https://www.wonderware.es/hmi-scada/que-es-scada/>

7. Anexos

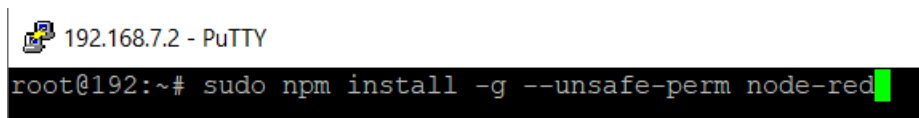
7.1 Manual de usuario nodos GPIO y OPC-UA BeagleBone Black.

En este capítulo se da las directrices para levantar los nodos para uso de los GPIO, el servidor OPC-UA en la placa BeagleBone Black y su configuración mediante la herramienta de programación Node-Red. Cabe recalcar que la imagen utilizada en la placa y la versión de node-red para el desarrollo de este proyecto son las siguientes:

- Operating System: Debian GNU/Linux 9 (stretch)
- Kernel: Linux 4.14.71-ti-r80
- Architecture: arm
- Node-RED version: v0.19.4
- Node.js version: v6.14.4

La imagen utilizada es bone-debian-9.5-iot-armhf-2018-10-07-4gb y se la puede descargar en la página <https://beagleboard.org/latest-images>. En esta imagen ya viene preinstalada la última versión de node-red por lo que no es necesario descargar a una versión actual.

Sin embargo si es necesario descargar la versión actual de node-red, usamos la terminal de la BeagleBone, la descargamos y la instalamos escribiendo:

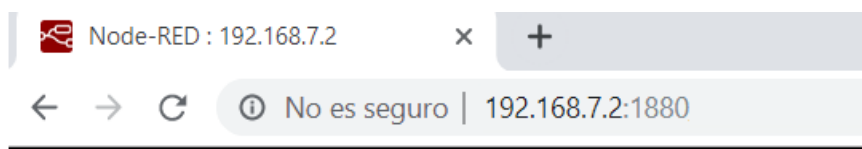


```
192.168.7.2 - PuTTY  
root@192:~# sudo npm install -g --unsafe-perm node-red
```

De igual manera ya vienen instalada en node-red los nodos GPIO para usar los pines como entradas o salidas.

Esto lo podemos verificar ingresando a la interfaz gráfica de node-red.

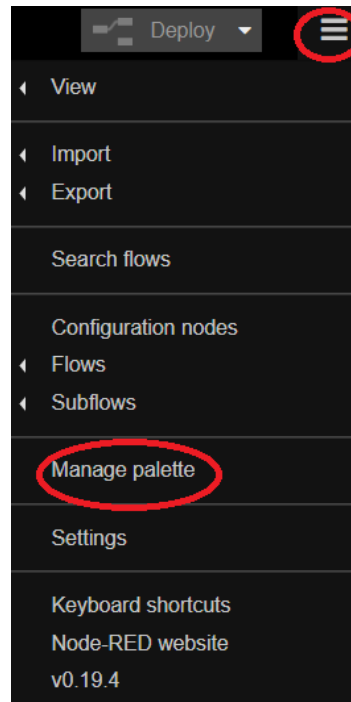
Escribimos en el navegador 192.168.7.2:1880, especificando la IP y el puerto de enlace que es el 1880.



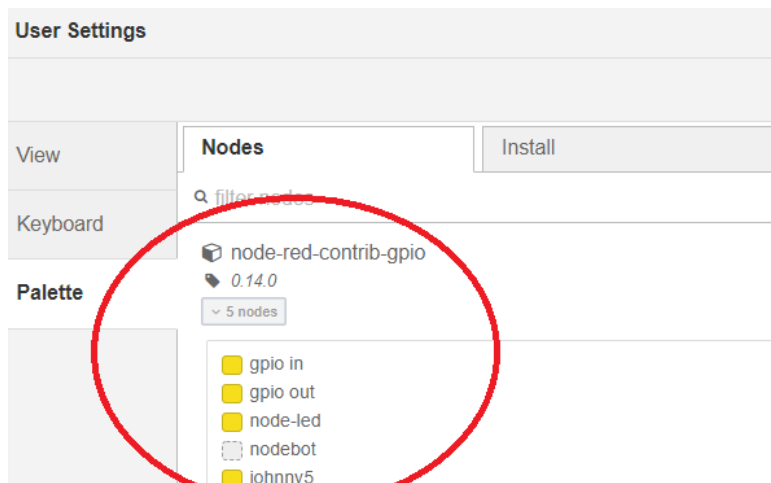
De esta manera se nos abre la interfaz gráfica de node-red

INSTALAR NODO GPIO

Verificamos que los nodos GPIO estén instalados, nos dirigimos a las opciones y escogemos la opción Manage Palette.



Y verificamos que el nodo node-red-contrib-gpio este instalada en la paleta.



Si es el caso y no se encuentra instalada, mediante la terminal nos dirigimos la ruta:

- /usr/local/lib/node_modules/ aquí es donde se encuentran todos los nodos e
- Instalamos el nodo GPIO usando npm install node-red-contrib-gpio

```
root@192:~# cd /usr/local/lib/node_modules/  
root@192:/usr/local/lib/node_modules# npm install node-red-contrib-gpio
```

De esta manera ya tendremos los nodos de GPIO. Cada vez que instalamos algún nuevo nodo es necesario reiniciar el servicio de node red para ver su cambio.

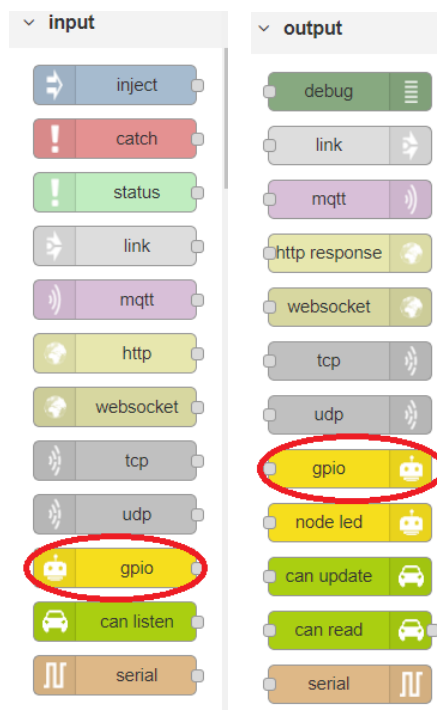
Esto lo podemos hacer parando el servicio de node red mediante:

- `sudo service node-red stop`

y luego lo iniciamos nuevamente con:

- `sudo service node-red start`

Los nodos de GPIO se ubican en los paneles de Inputs y Outputs, dependiendo de la utilización se escoge la configuración del pin ya sea como entrada o salida, hay que decir, que no se puede utilizar el mismo pin como salida y entrada ya que daría conflictos.



GPIO – INPUT


Escogemos el nodo de gpio del panel de inputs y damos doble click para empezar a configurarlo.

- Agregamos un nuevo **Board**

Edit gpio in node

Delete Cancel Done

node properties

Board Add new nodebot... 

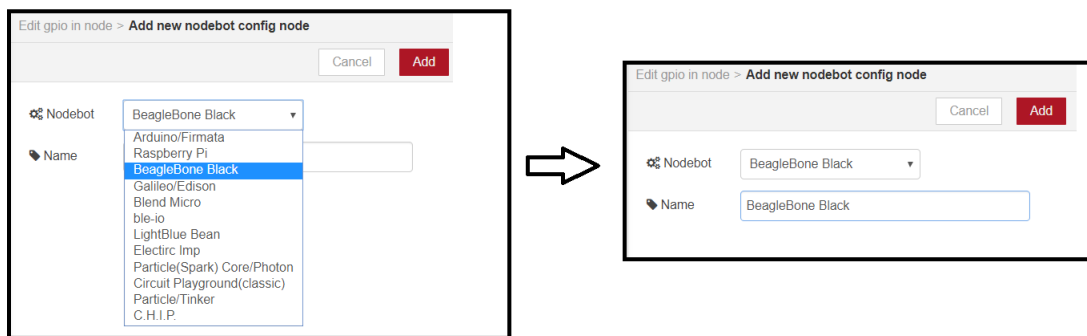
Type Digital pin

Pin 2

Name Name

Note: You cannot use the same pin for both output and input.

- Escogemos nuestra placa BeagleBone Black, le damos un nombre y lo agregamos.



Edit gpio in node - Add new nodebot config node

Cancel Add

Nodebot BeagleBone Black

Name

BeagleBone Black


BeagleBone Black

- Seleccionamos en **Type** la entrada como digital o análoga.

Edit gpio in node

Delete Cancel Done

node properties

Board BeagleBone Black 

Type Digital pin

Pin

Name Name

Note: You cannot use the same pin for both output and input.

- En **Pin** ubicamos el pin con el que vamos a trabajar. La nomenclatura utilizada y los pines disponibles se los ve en la siguiente tabla, de esta manera ya tenemos configurado el GPIO como entrada

Pin ID	Soporte	Comentario
P8_7 o GPIO66	INPUT, OUTPUT	
P8_8 o GPIO67	INPUT, OUTPUT	
P8_9 o GPIO69	INPUT, OUTPUT	
P8_10 o GPIO68	INPUT, OUTPUT	
P8_11 o GPIO45	INPUT, OUTPUT	
P8_12 o GPIO44	INPUT, OUTPUT	
P8_13 o GPIO23	INPUT, OUTPUT, SERVO, PWM	
P8_14 o GPIO26	INPUT, OUTPUT	
P8_15 o GPIO47	INPUT, OUTPUT	
P8_16 o GPIO46	INPUT, OUTPUT	
P8_17 o GPIO27	INPUT, OUTPUT	
P8_18 o GPIO65	INPUT, OUTPUT	
P8_19 o GPIO22	INPUT, OUTPUT, SERVO, PWM	
P8_26 o GPIO61	INPUT, OUTPUT	
P9_11 o GPIO30	INPUT, OUTPUT	
P9_12 o GPIO60	INPUT, OUTPUT	
P9_13 o GPIO31	INPUT, OUTPUT	
P9_14 o GPIO50	INPUT, OUTPUT, SERVO, PWM	
P9_15 o GPIO48	INPUT, OUTPUT	
P9_16 o GPIO51	INPUT, OUTPUT, SERVO, PWM	
P9_17 o GPIO5	INPUT, OUTPUT	
P9_18 o GPIO4	INPUT, OUTPUT	
P9_19	I2C2 SCL	Reservado para I2C
P9_20	I2C2 SDA	Reservado para I2C
P9_21 or GPIO3	INPUT, OUTPUT, SERVO, PWM	
P9_22 o GPIO2	INPUT, OUTPUT, SERVO, PWM	
P9_23 o GPIO49	INPUT, OUTPUT	
P9_24 o GPIO15	INPUT, OUTPUT	
P9_26 o GPIO14	INPUT, OUTPUT	
P9_27 o GPIO115	INPUT, OUTPUT	

P9_30 o GPIO112	INPUT, OUTPUT	
P9_33 o A4	ANALOG	No ingrese más de 1.8V
P9_35 o A6	ANALOG	No ingrese más de 1.8V
P9_36 o A5	ANALOG	No ingrese más de 1.8V
P9_37 o A2	ANALOG	No ingrese más de 1.8V
P9_38 o A3	ANALOG	No ingrese más de 1.8V
P9_39 o A0	ANALOG	No ingrese más de 1.8V
P9_40 o A1	ANALOG	No ingrese más de 1.8V
P9_41 o GPIO20	INPUT, OUTPUT	
P9_42 o GPIO7	INPUT, OUTPUT, SERVO, PWM	
USR0	OUTPUT	Usuario incorporado LED 0
USR1	OUTPUT	Usuario incorporado LED 1
USR2	OUTPUT	Usuario incorporado LED 2
USR3	OUTPUT	LED de usuario incorporado 3 / LED predeterminado

Edit gpio in node

node properties

Board

Type

Pin

Name

Note: You cannot use the same pin for both output and input.

GPIO – OUTPUT

La configuración es igual que en el caso de los INPUTS.

- Agregamos el **Board**.
- En **Type** escogemos el tipo de salida en este caso tenemos disponibles varias salidas como, digital, analógica, servo e I2C.

Edit gpio out node

Delete Cancel Done

node properties

Board BeagleBone Black

Type Digital (0/1)

Pin

Name I2C Delay

Note: You cannot use the same pin for both output and input.

- Escogemos el Pin que vamos a utilizar y le agregamos un nombre.

Edit gpio out node

Delete Cancel Done

node properties

Board BeagleBone Black

Type Digital (0/1)

Pin P8_8

Name pin P8_8

Note: You cannot use the same pin for both output and input.

INSTALAR NODO OPC-UA

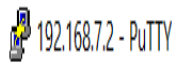
El protocolo OPC-UA no viene instalada previamente en la imagen, por lo que es necesario descargarla en la ruta donde se encuentran los nodos. Al igual que en el caso anterior nos dirigimos a la ruta:

- `usr/local/lib/node_modules/`

Descargamos e instalamos el node OPC-U mediante:

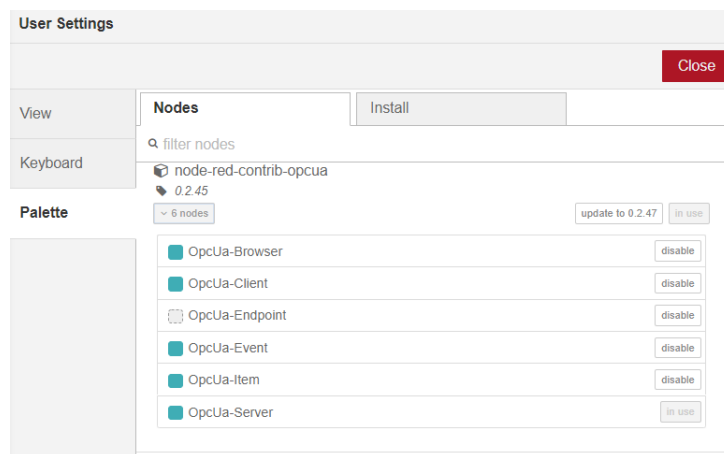
- `npm install -g node-red-contrib-opcua --unsafe-perm --build-from-source`

Para este caso es necesario deshabilitar los permisos ya que caso contrario nos dará un error y nos dejara instalarlo.



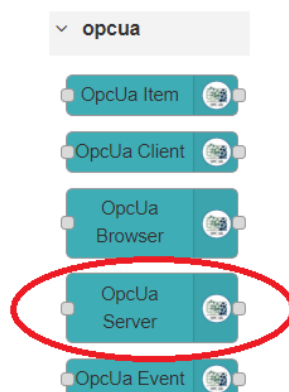
```
root@192:/usr/local/lib/node_modules# npm install -g node-red-contrib-opcua --unsafe-perm --build-from-source
```

Al igual que en el acaso anterior, verificamos que se ha instalado correctamente y se encuentra en la paleta de nodos de node-red.



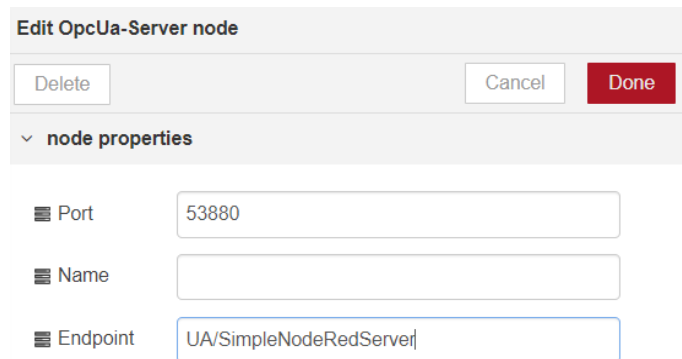
Servidor OPC-UA

Nos dirigimos al panel de nodos de OPCUA y escogemos el nodo OPC-UA Server



Y lo configuramos:

- Simplemente agregamos el **PORT** y el **ENDPOINT** y el servidor OPC-UA estaría listo para usar.



Edit OpcUa-Server node

Delete Cancel Done

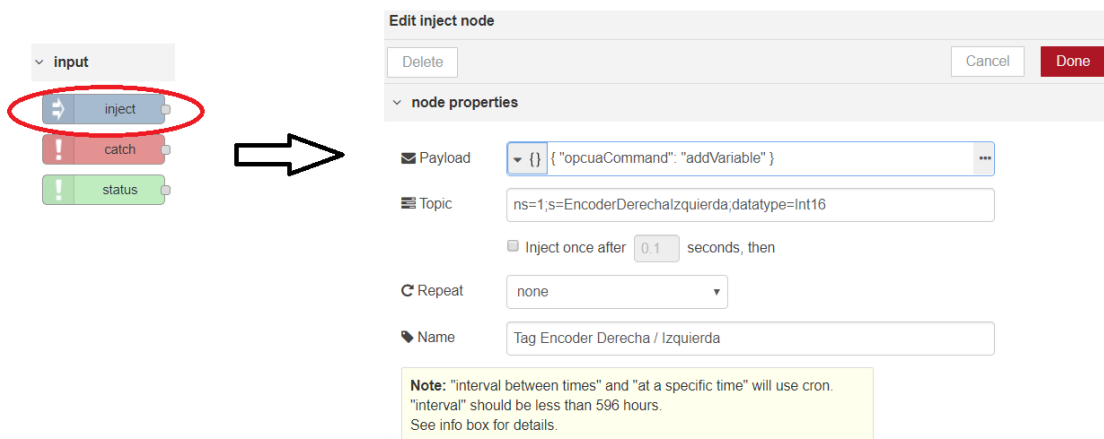
node properties

Port 53880

Name

Endpoint UA/SimpleNodeRedServer

Si queremos agregar una variable al servidor utilizaremos un nodo **Inject** de la sección de Inputs y la configuramos escribiendo en **Payload** el comando para agregar variables y en **Topic** agregamos el name space index (ns) = 1, el identificador (s) y el tipo de dato (datatype).



input

inject

catch

status

Edit inject node

Delete Cancel Done

node properties

Payload {{ "opcuaCommand": "addVariable" }}

Topic ns=1,s=EncoderDerechalzquierda,datatype=Int16

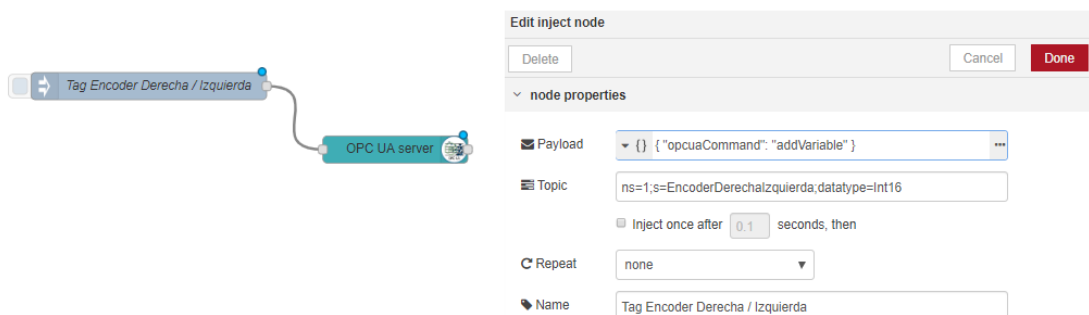
Inject once after 0.1 seconds, then

Repeat none

Name Tag Encoder Derecha / Izquierda

Note: "interval between times" and "at a specific time" will use cron. "interval" should be less than 596 hours. See info box for details.

Finalmente unimos los nodos y le damos a INJECT para que este ejecute el comando y agregue una variable al servidor.



Tag Encoder Derecha / Izquierda

OPC UA server

Edit inject node

Delete Cancel Done

node properties

Payload {{ "opcuaCommand": "addVariable" }}

Topic ns=1,s=EncoderDerechalzquierda,datatype=Int16

Inject once after 0.1 seconds, then

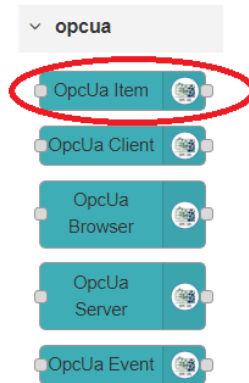
Repeat none

Name Tag Encoder Derecha / Izquierda

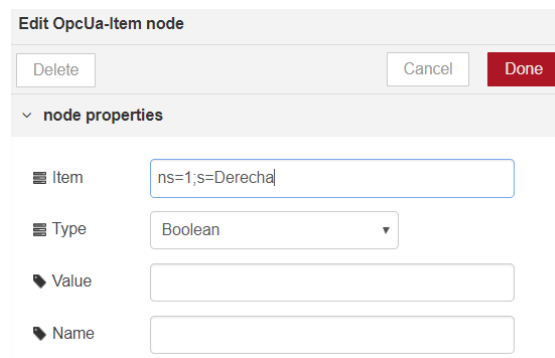
Ciente OPC-UA

Para trabajar con una variable del Servidor OPC-UA, vamos a necesitar de dos nodos de los OPCUA, el nodo OpcUaItem y OpcUaClient.

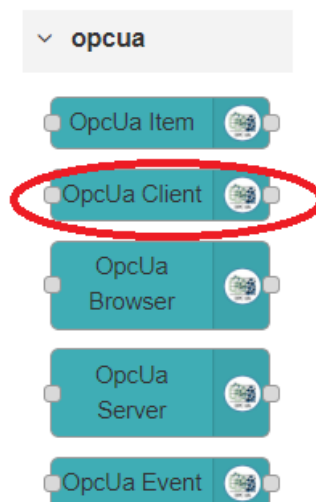
- Seleccionamos OpcUaItem.



- Lo configuramos agregando el **Item** del cual queremos recibir la información y el **Type** de la variable.

A screenshot of the 'Edit OpcUa-Item node' dialog box. It has a title bar, a 'Delete' button, a 'Cancel' button, and a 'Done' button. Under the 'node properties' section, there are four fields: 'Item' with the value 'ns=1;s=Derecha', 'Type' with a dropdown menu set to 'Boolean', 'Value' (empty), and 'Name' (empty).

- Seleccionamos OpcUaClient.



- Y configuramos, agregamos el **ENDPOINT** del servidor y la acción que quiere que esa variable haga en **ACTION**.

Endpoint: opc.tcp://localhost:4334/UA/ServerOPCUABBB

Action: READ

Certificate: None, no certificate used

Local file: ./pki/server_certificate.pem

Name:

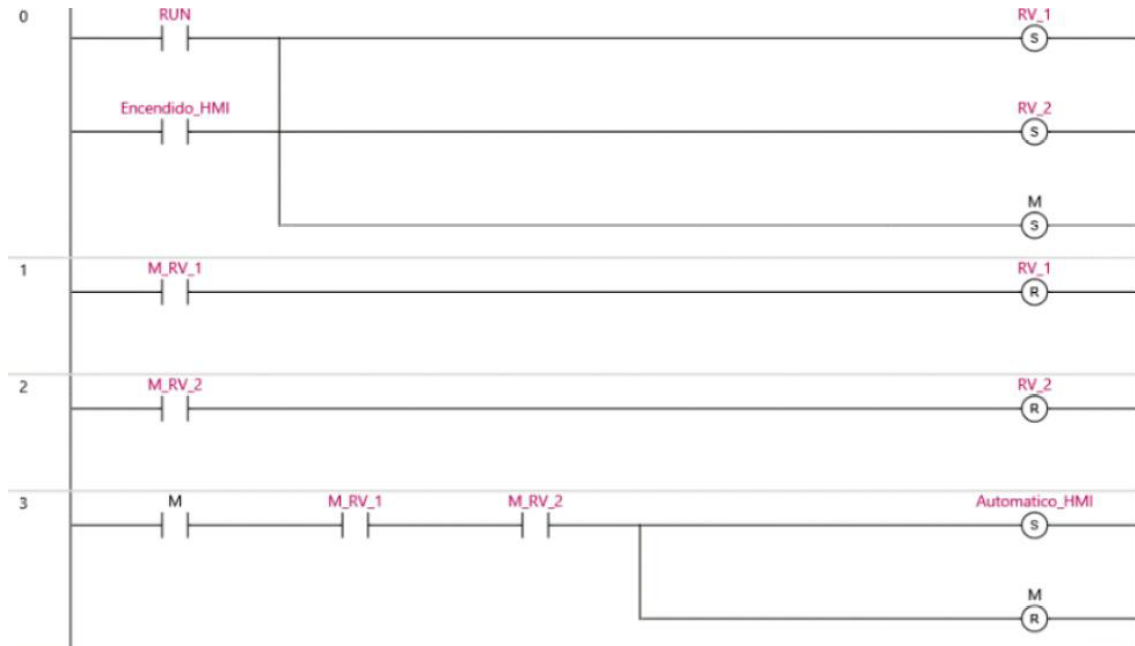
- Unimos los dos nodos, de esta manera ya estará trabajando el cliente OPC-UA



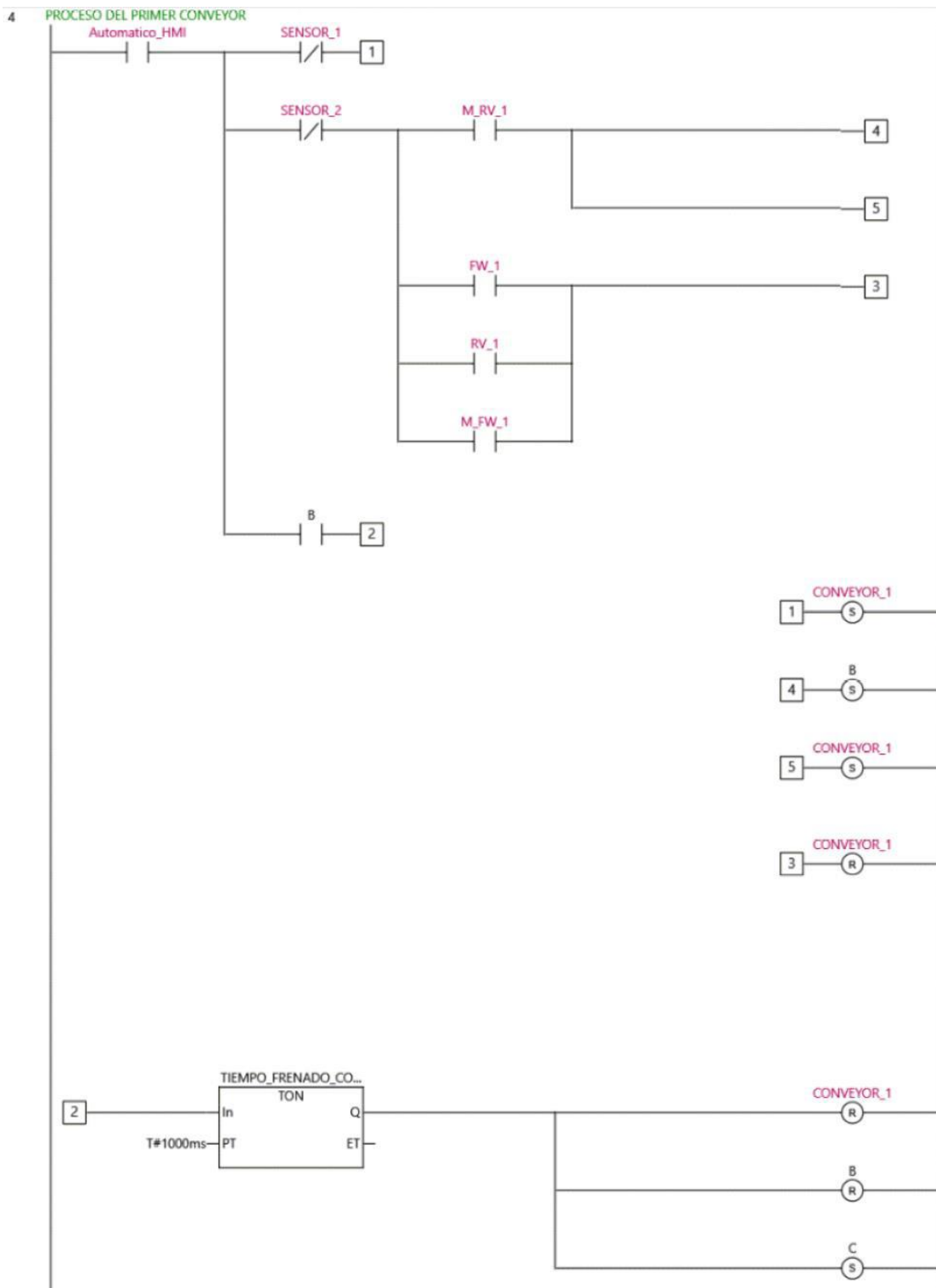
7.2 Programa Ladder PLC Omron NJ101

En este capítulo se encuentra la programación en ladder para el control de la planta piloto de manufactura.

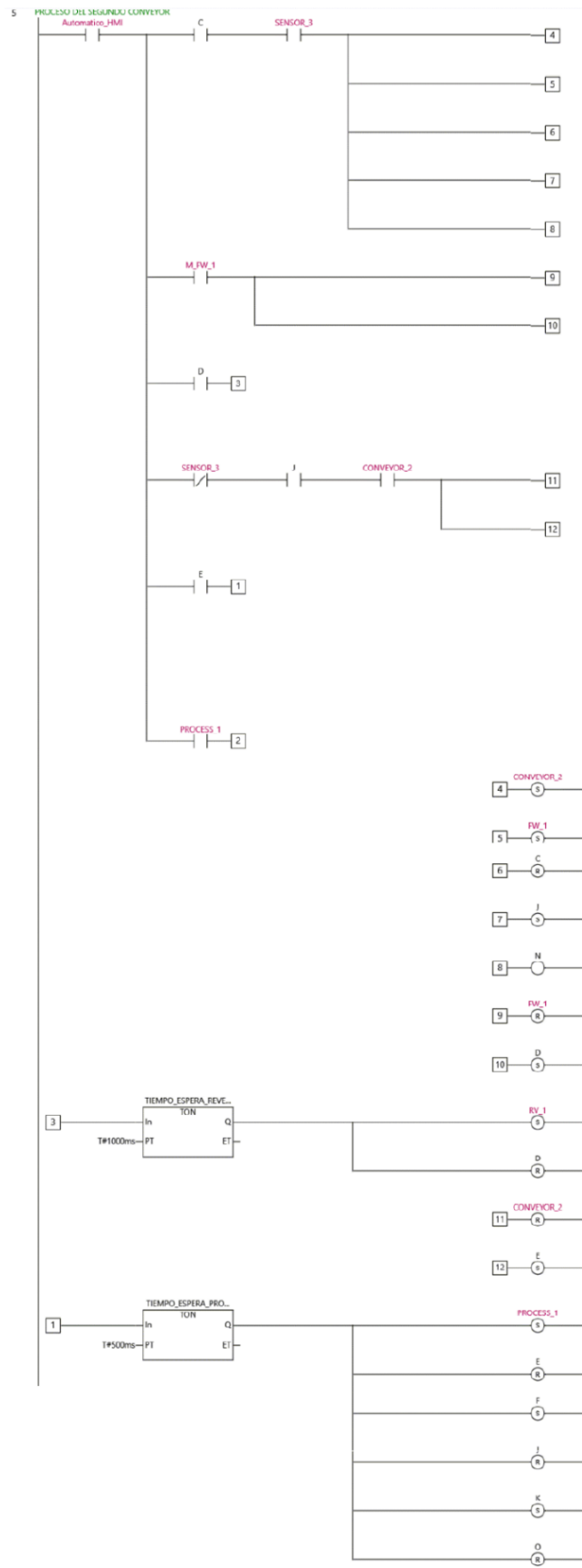
- El proceso arranca cuando la planta piloto está en automático y los deslizadores se encuentran en una posición inicial.



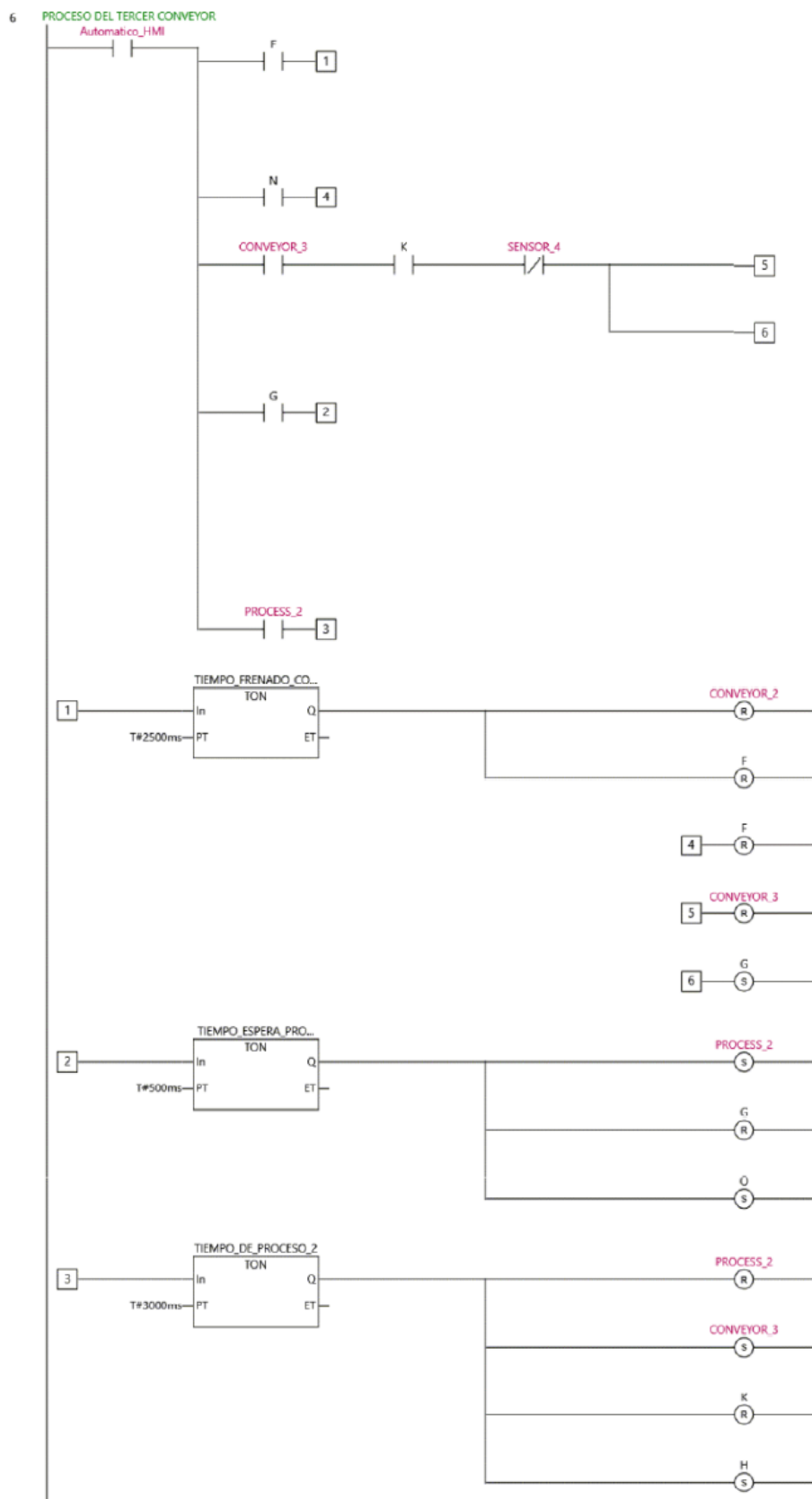
- Proceso de la primera etapa



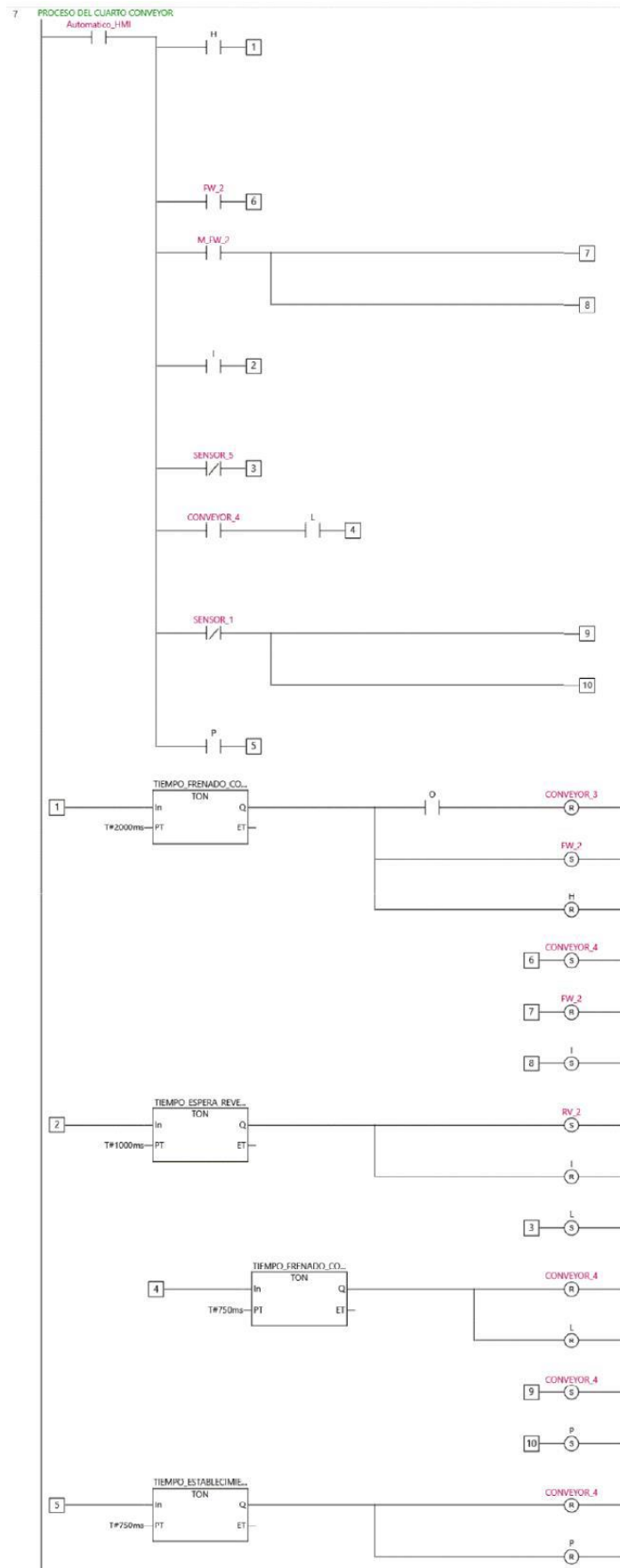
- Segunda etapa



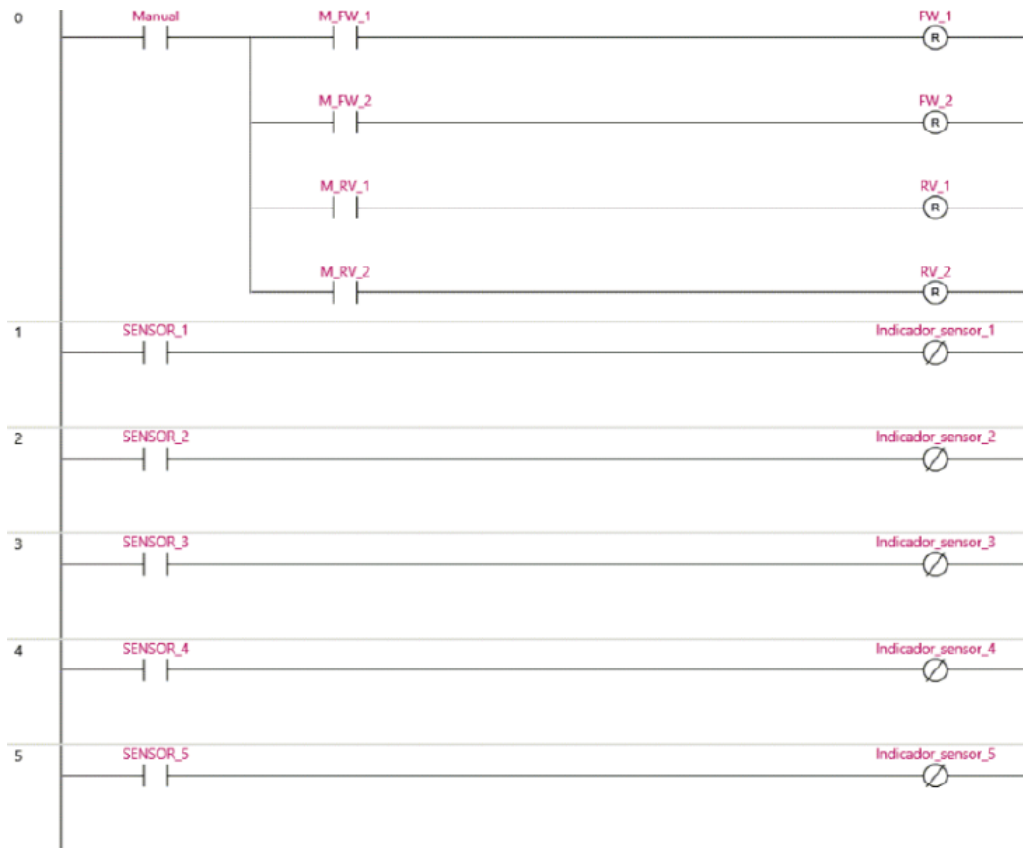
- Tercera etapa



- Cuarta etapa



- Etapa de control Manual.



El mapa de entradas y salidas utilizadas se muestra en la siguiente imagen al igual que el de las variables globales utilizadas para el control del HMI.

Posición	Puerto	Descripción	R/W	Tipo de dato	Variable	Comentario de Variable	Tipo de Variable
	▼ Ch1_Out	Output CH1	RW	WORD			
	Ch1_Out00	Output CH1 bit 00	RW	BOOL	FW_1		Variables globales
	Ch1_Out01	Output CH1 bit 01	RW	BOOL	FW_2		Variables globales
	Ch1_Out02	Output CH1 bit 02	RW	BOOL	CONVEYOR_1		Variables globales
	Ch1_Out03	Output CH1 bit 03	RW	BOOL	PROCESS_1		Variables globales
	Ch1_Out04	Output CH1 bit 04	RW	BOOL	PROCESS_2		Variables globales
	Ch1_Out05	Output CH1 bit 05	RW	BOOL			
	Ch1_Out06	Output CH1 bit 06	RW	BOOL			
	Ch1_Out07	Output CH1 bit 07	RW	BOOL			
	Ch1_Out08	Output CH1 bit 08	RW	BOOL	RV_1		Variables globales
	Ch1_Out09	Output CH1 bit 09	RW	BOOL	RV_2		Variables globales
	Ch1_Out10	Output CH1 bit 10	RW	BOOL	CONVEYOR_2		Variables globales
	Ch1_Out11	Output CH1 bit 11	RW	BOOL	CONVEYOR_3		Variables globales
	Ch1_Out12	Output CH1 bit 12	RW	BOOL	CONVEYOR_4		Variables globales
	Ch1_Out13	Output CH1 bit 13	RW	BOOL			
	Ch1_Out14	Output CH1 bit 14	RW	BOOL			
	Ch1_Out15	Output CH1 bit 15	RW	BOOL			
	▼ Ch1_In	Input CH1	R	WORD			
	Ch1_In00	Input CH1 bit 00	R	BOOL	EMERGENCY_STOP		Variables globales
	Ch1_In01	Input CH1 bit 01	R	BOOL	RUN		Variables globales
	Ch1_In02	Input CH1 bit 02	R	BOOL	STOP		Variables globales
	Ch1_In03	Input CH1 bit 03	R	BOOL			
	Ch1_In04	Input CH1 bit 04	R	BOOL	SENSOR_4		Variables globales
	Ch1_In05	Input CH1 bit 05	R	BOOL	SENSOR_3		Variables globales
	Ch1_In06	Input CH1 bit 06	R	BOOL	M_RV_2		Variables globales
	Ch1_In07	Input CH1 bit 07	R	BOOL	M_RV_1		Variables globales
	Ch1_In08	Input CH1 bit 08	R	BOOL			
	Ch1_In09	Input CH1 bit 09	R	BOOL			
	Ch1_In10	Input CH1 bit 10	R	BOOL			
	Ch1_In11	Input CH1 bit 11	R	BOOL	SENSOR_5		Variables globales
	Ch1_In12	Input CH1 bit 12	R	BOOL	SENSOR_1		Variables globales
	Ch1_In13	Input CH1 bit 13	R	BOOL	SENSOR_2		Variables globales
	Ch1_In14	Input CH1 bit 14	R	BOOL	M_FW_2		Variables globales
	Ch1_In15	Input CH1 bit 15	R	BOOL	M_FW_1		Variables globales

Mapa E/S		Variables globales x						
Nombre	Tipo de datos	Valor inicial	AT	Retentiva	Constante	Publicación en red	Comentario	
CONVEYOR_1	BOOL		IOBus://ra...	<input type="checkbox"/>	<input type="checkbox"/>	Solo publicar		
EMERGENCY_STOP	BOOL		IOBus://ra...	<input type="checkbox"/>	<input type="checkbox"/>	Solo publicar		
CONVEYOR_3	BOOL		IOBus://ra...	<input type="checkbox"/>	<input type="checkbox"/>	Solo publicar		
RUN	BOOL		IOBus://ra...	<input type="checkbox"/>	<input type="checkbox"/>	No publica		
STOP	BOOL		IOBus://ra...	<input type="checkbox"/>	<input type="checkbox"/>	No publica		
SENSOR_1	BOOL		IOBus://ra...	<input type="checkbox"/>	<input type="checkbox"/>	No publica		
SENSOR_2	BOOL		IOBus://ra...	<input type="checkbox"/>	<input type="checkbox"/>	No publica		
SENSOR_3	BOOL		IOBus://ra...	<input type="checkbox"/>	<input type="checkbox"/>	No publica		
SENSOR_4	BOOL		IOBus://ra...	<input type="checkbox"/>	<input type="checkbox"/>	No publica		
SENSOR_5	BOOL		IOBus://ra...	<input type="checkbox"/>	<input type="checkbox"/>	Solo publicar		
M_FW_1	BOOL		IOBus://ra...	<input type="checkbox"/>	<input type="checkbox"/>	Solo publicar		
M_RV_1	BOOL		IOBus://ra...	<input type="checkbox"/>	<input type="checkbox"/>	Solo publicar		
M_FW_2	BOOL		IOBus://ra...	<input type="checkbox"/>	<input type="checkbox"/>	Solo publicar		
M_RV_2	BOOL		IOBus://ra...	<input type="checkbox"/>	<input type="checkbox"/>	Solo publicar		
CONVEYOR_2	BOOL		IOBus://ra...	<input type="checkbox"/>	<input type="checkbox"/>	Solo publicar		
CONVEYOR_4	BOOL		IOBus://ra...	<input type="checkbox"/>	<input type="checkbox"/>	Solo publicar		
FW_1	BOOL		IOBus://ra...	<input type="checkbox"/>	<input type="checkbox"/>	Solo publicar		
RV_1	BOOL		IOBus://ra...	<input type="checkbox"/>	<input type="checkbox"/>	Solo publicar		
FW_2	BOOL		IOBus://ra...	<input type="checkbox"/>	<input type="checkbox"/>	Solo publicar		
RV_2	BOOL		IOBus://ra...	<input type="checkbox"/>	<input type="checkbox"/>	Solo publicar		
PROCESS_1	BOOL		IOBus://ra...	<input type="checkbox"/>	<input type="checkbox"/>	Solo publicar		
PROCESS_2	BOOL		IOBus://ra...	<input type="checkbox"/>	<input type="checkbox"/>	Solo publicar		
Manual	BOOL			<input type="checkbox"/>	<input type="checkbox"/>	Solo publicar		
Indicador_sensor_1	BOOL			<input type="checkbox"/>	<input type="checkbox"/>	Solo publicar		
Indicador_sensor_2	BOOL			<input type="checkbox"/>	<input type="checkbox"/>	Solo publicar		
Indicador_sensor_3	BOOL			<input type="checkbox"/>	<input type="checkbox"/>	Solo publicar		
Indicador_sensor_4	BOOL			<input type="checkbox"/>	<input type="checkbox"/>	Solo publicar		
Indicador_sensor_5	BOOL			<input type="checkbox"/>	<input type="checkbox"/>	Solo publicar		
Encendido_HMI	BOOL			<input type="checkbox"/>	<input type="checkbox"/>	Solo publicar		
Apagado_HMI	BOOL			<input type="checkbox"/>	<input type="checkbox"/>	Solo publicar		
Automatico_HMI	BOOL			<input type="checkbox"/>	<input type="checkbox"/>	Solo publicar		