

PRÁCTICA**Tema:** Configuración de un PLC CompactLogix**1. Objetivos****1.1. Objetivo general**

Reconocer las principales herramientas del software Studio 5000 Logix Designer para la programación en lenguaje escalera usando entradas y salidas tanto digitales como analógicas del PLC CompactLogix L33 ERM.

1.2. Objetivos específicos

1. Identificar las herramientas del software Studio 5000 para la creación de un proyecto.
2. Diferenciar las partes del PLC CompactLogix para la realización de aplicaciones industriales.
3. Crear una red Ethernet a través del software Studio 5000 para la configuración una arquitectura integrada CIP.

2. Equipos y herramientas

- CompactLogix L33 ERM.
- Studio 5000.
- RsLinx Classic.
- Cables Ethernet.
- Switch administrable.
- Módulo de E/S digitales del CompactLogix.
- Computadora.
- Cables bananas.
- Botoneras y luz piloto del tablero de automatización.

3. Marco teórico**Studio 5000**

El software Studio 5000 contiene, Logix Designer y Factory Talk View. Logix Designer permite la programación y configuración de los controladores autómatas programables ControlLogix y CompactLogix de la plataforma Rockwell Automation.

Logix Designer permite programar en diferentes lenguajes como: ladder, texto estructurado, diagrama de bloques funcionales, y Graphcet.





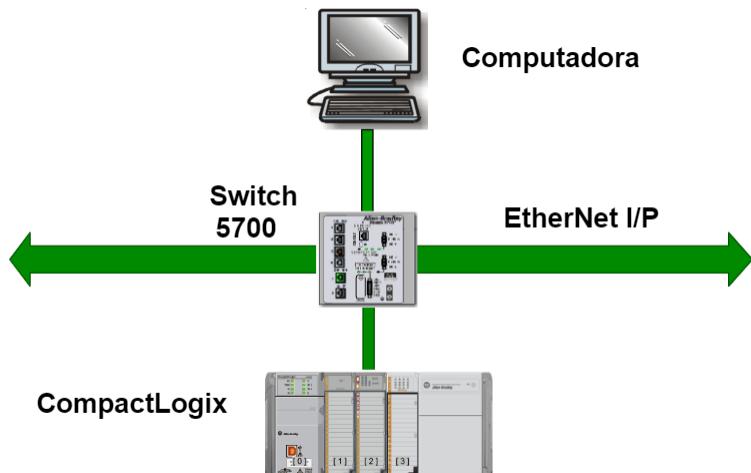
CompactLogix 5370-L3

Tiene puerto dual ethernet con capacidad DLR, memoria de 2 MB, 16 módulos de expansiones de E/S, 32 Nodos Ethernet IP, 8 ejes de movimiento CIP con función cinemática.

- Admite Integrated Motion en EtherNet / IP™ (modelos seleccionados)
- Anillo de nivel de dispositivo (DLR) para mejorar la resistencia de la red
- Capacidad para agregar control localizado módulos de E / S CompactLogix™ 1769
- E/S distribuida a través de EtherNet / IP ofrece flexibilidad adicional en E/S opcionales.
- Capacidad de conexión abierta para dispositivos como impresoras, lectores de códigos de barras y servidores
- El puerto USB incorporado proporciona un fácil acceso para acceder descargar firmware y programación.



4. Conexiones



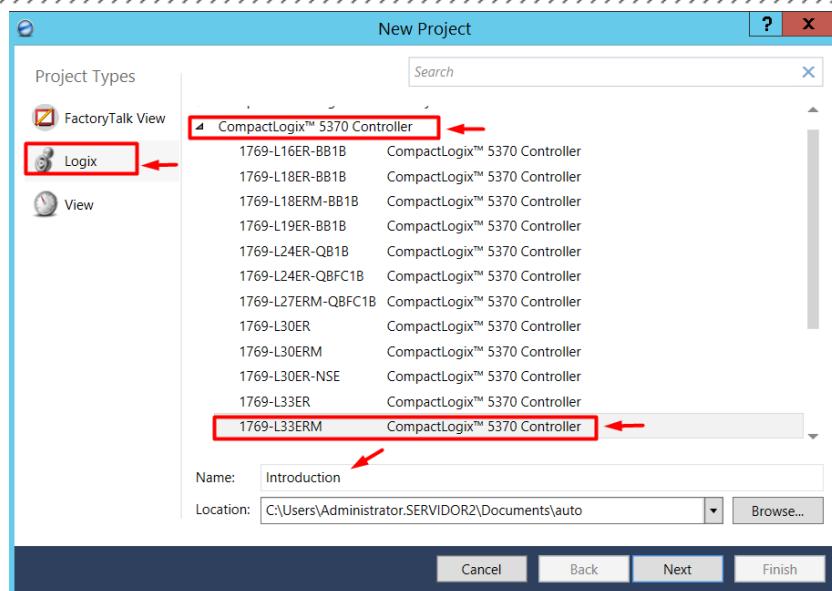
5. Procedimiento

Crear un proyecto en Studio 5000

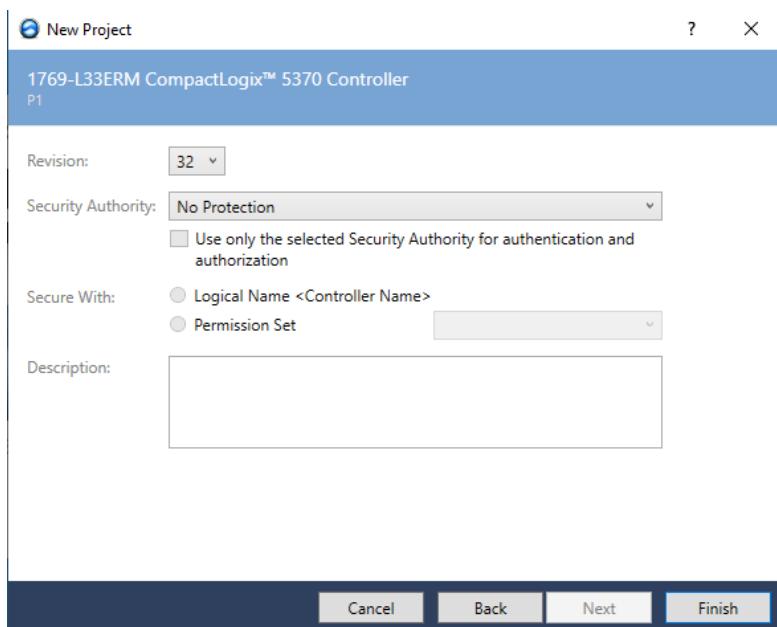
1. Abrir el software **Studio 5000**, seleccionar **New Project**.



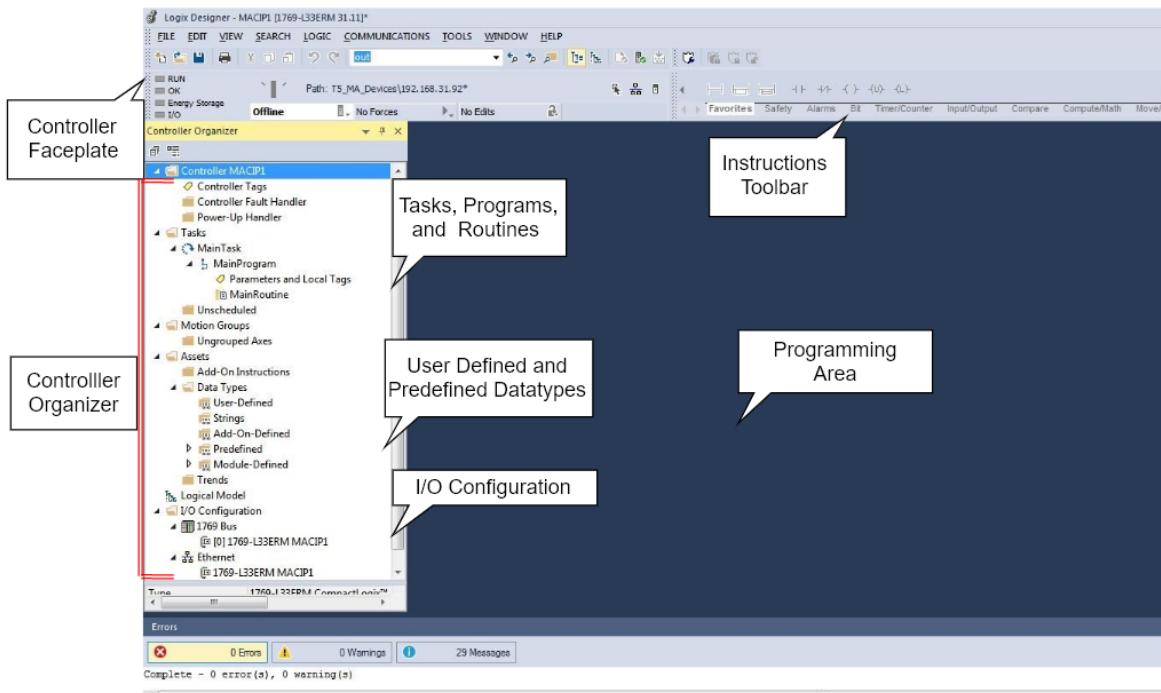
2. En la ventana **New Project**, seleccionar **1769-L33ERM CompactLogix**. Asignar un nombre al proyecto, y la ubicación a guardar del proyecto. La extensión del tipo de archivo del proyecto será. ACD



3. La ventana de Configuraciones del proyecto emergirá, donde se asignará la revisión del controlador, las otras configuraciones se deben dejar las que se encuentran por defecto. La revisión del controlador corresponde al Firmware del dispositivo, la revisión se puede observar en el software RSLinx, al dar clic derecho en el equipo y seleccionando sus propiedades. En algunos controladores suele presentar el tipo de chasis donde se selecciona la cantidad de números de módulos se pueden agregar al chasis, slot corresponde en asignar la posición que se encuentra el CPU del controlador, esta posición comienza desde cero.

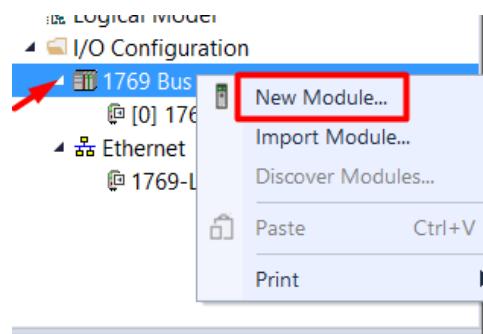


- La pantalla de **Logix Designer** aparecerá. En dicha pantalla se encuentran la ventana **Controller Organizer** que contiene diferentes carpetas y archivos de toda la información de los programas del proyecto.

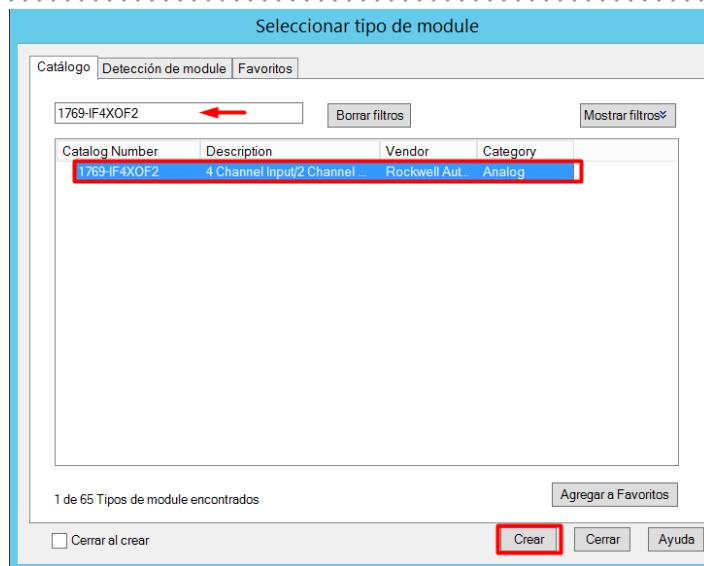


Añadir módulos a un controlador dentro de un proyecto de Studio 5000

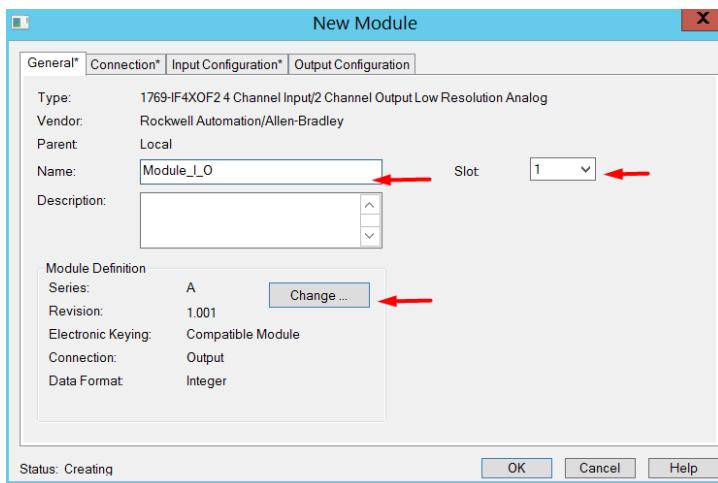
- En la carpeta **I/O Configuration**, dar clic derecho en el bus del controlador y seleccionar **New Module**.



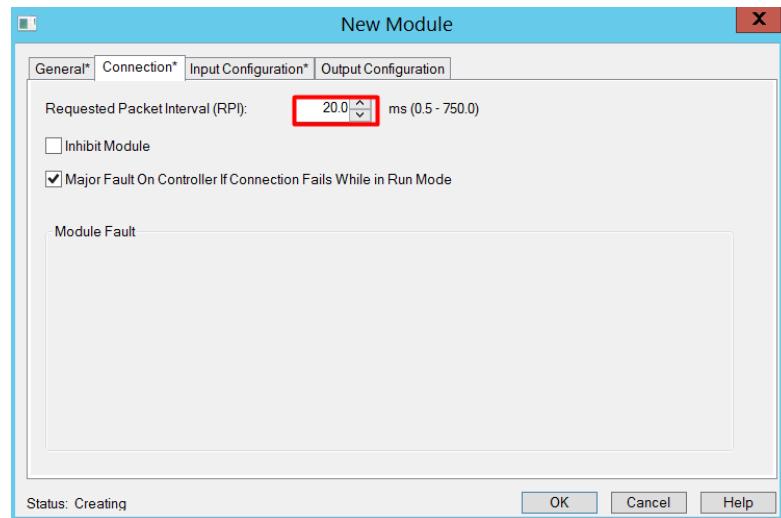
- La ventana **Select Module Type** aparecerá para buscar por el número de catálogo del módulo que se desea agregar. Los módulos que deben crearse son los siguientes, módulo de E/S digitales **1769-IQ6XOW4**, y módulo de E/S analógicas **1769-IF4XOF2**. En este caso seleccionar el número de catálogo del módulo de E/S digitales y dar clic en **Create**.



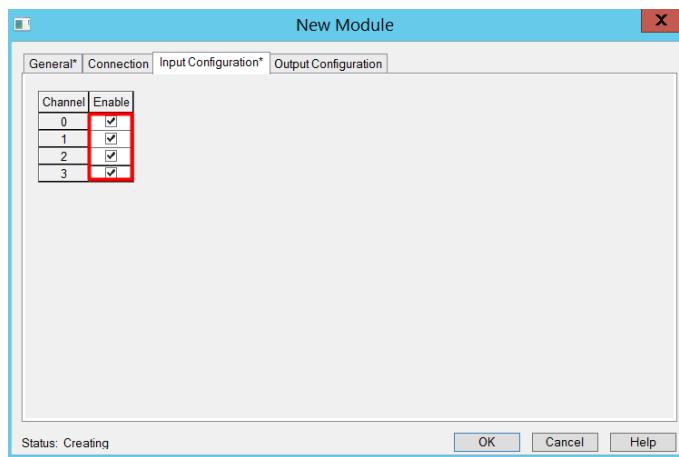
3. A continuación, se muestra la ventana **New Module** se debe escribir un nombre en el parámetro **Name** y asignar la posición que se encuentra el módulo agregado en el chasis en el parámetro **slot**. En la sección **Module Definition**, dar clic en **Change**.



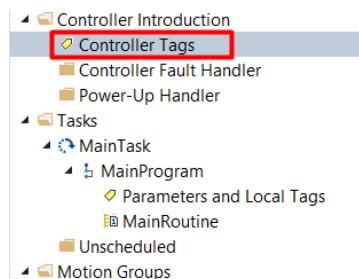
4. En la pestaña **Connection** de la ventana **New Module**, asignar 20ms en RPI: Requested Packet Interval, este parámetro corresponde al período en el cual el dato es actualizado y es configurado en milisegundos. Luego, clic en **OK**.



5. Luego, crear un nuevo módulo de E/S analógicas cuyo número de catálogo es **1769-IF4XOF2**. Además, en las pestañas **Input Configuration** y **Output Configuration** se deben habilitar todos los canales de sus entradas y salidas.



6. En la carpeta **Controller Introduction**, dar doble clic en **Controller Tags**.



- En la ventana **Controller Tags**, las variables de los módulos agregados aparecerán. A continuación, se muestra un ejemplo de estas variables:

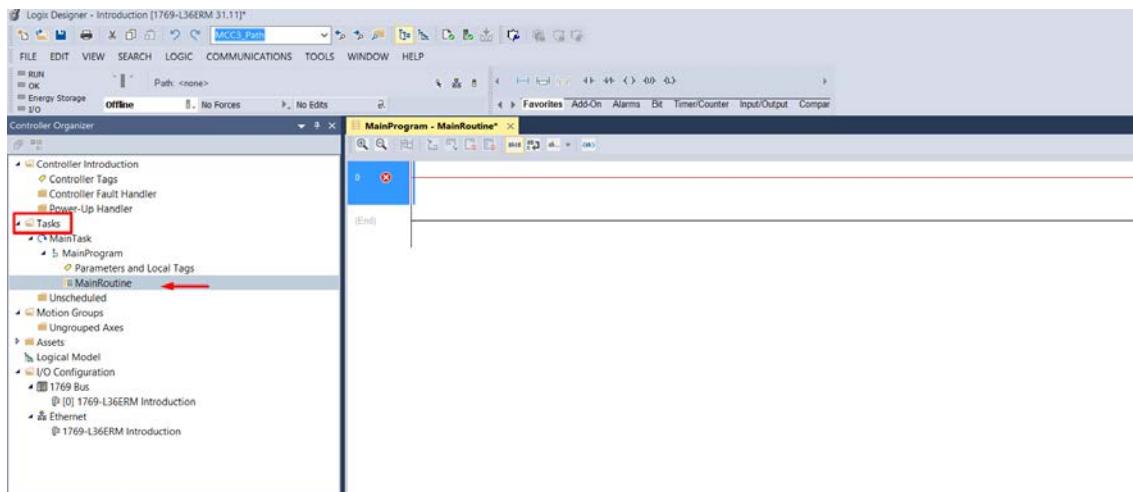
Local:1:I.Data.4

- “Local” significa que el módulo está conectado para un controlador en frente del backplane o con un enlace paralelo.
- “1” significa que el módulo es el numero #1 (localizado en el segundo slot en el rack).
- “I” significa que el bit es una entrada.
- “Data” indica que el tipo de dato.
- “4” significa que es el quinto bit de la salida.

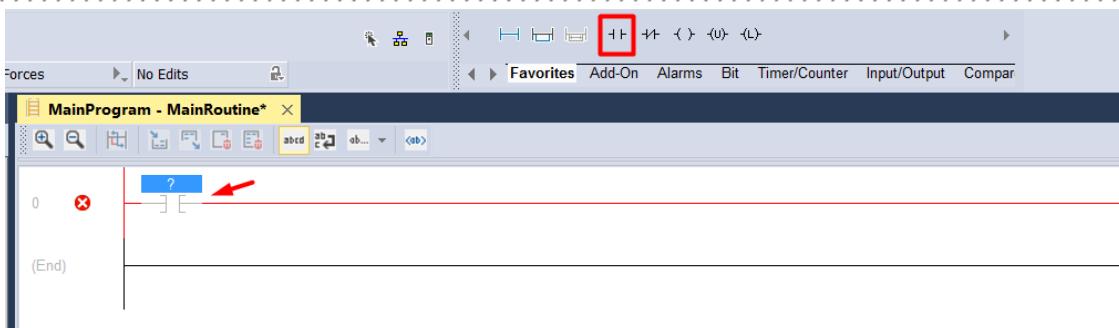
Name	Value	Force Mask	Style	Data Type
Local:1:C	(...)	(...)	(...)	AB:1769_IF4XOF2:...
Local:1:I	(...)	(...)	(...)	AB:1769_IF4XOF2:...
Local:1:O	(...)	(...)	(...)	AB:1769_IF4XOF2:...
Point_I_O:I	(...)	(...)	(...)	AB:1734_5SLOT:I:0
Point_I_O:O	(...)	(...)	(...)	AB:1734_5SLOT:O:0

Añadir elementos de lenguaje de programación ladder

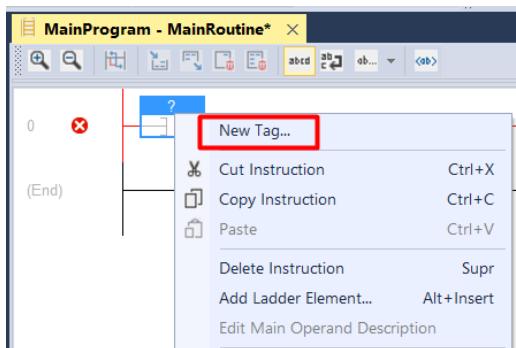
- En el Organizador del controlador en la carpeta “Task”, se encuentra “Main Program”, y dentro de ella “MainRoutine”. Al dar doble clic aparecerá el editor de rutinas.



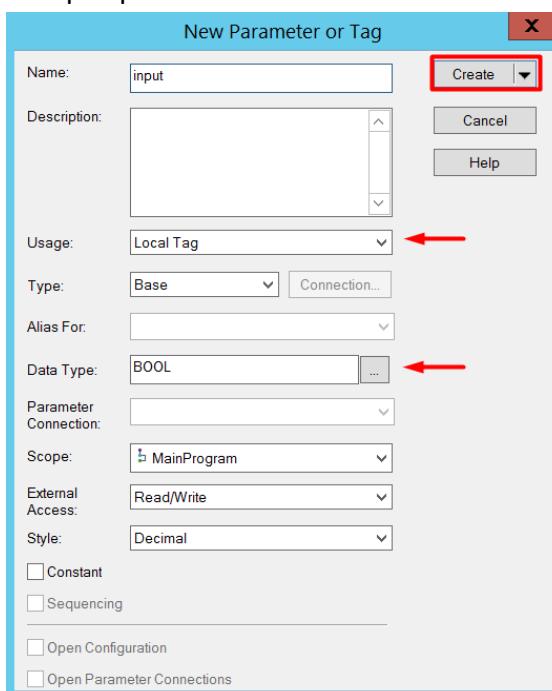
- En la barra de herramientas aparecerán las diferentes librerías del software con sus respectivos elementos, los cuales podrán ser agregados en la rutina.



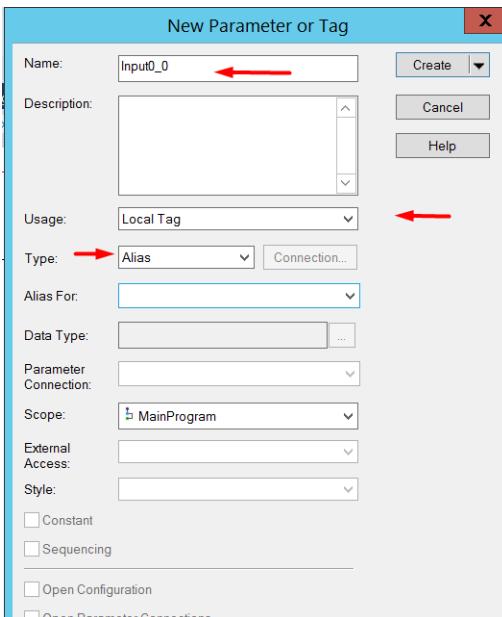
- Para la vinculación de algún elemento de la rutina con una variable, se debe crear una etiqueta, dar clic derecho en elemento, luego seleccionar **New Tag**.



- Ingresar el nombre de la variable, **tipo base**, en este caso se elegirá el tipo base la cual no está vinculado con entradas o salida físicas del controlador (memoria). Luego seleccionar el tipo de datos y llenar los campos pertinentes.



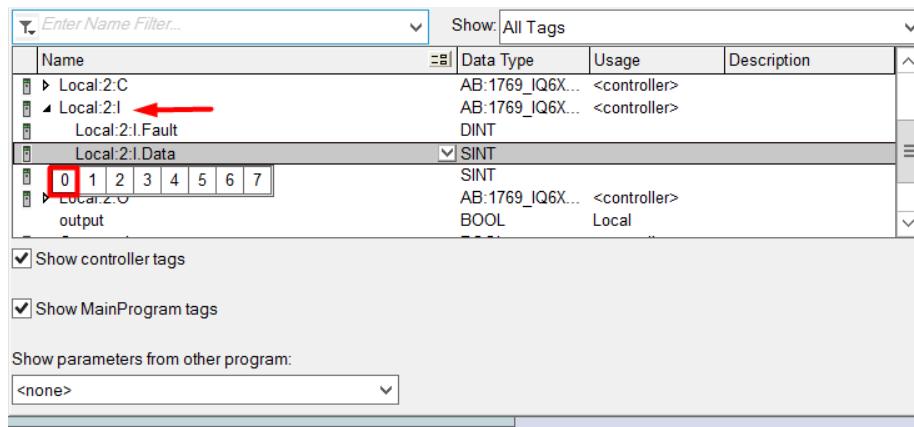
5. La variable **tipo Alias Tag** permite crear una variable que representa otra variable, es decir las variables de los módulos I/O pueden ser representados por otra variable. Dar clic derecho en un elemento de la rutina (contacto, bobina, etc.). En la ventana **New Parameter or tag**, asignar un nombre a la variable, luego en **Type** escoger **Alias** y se habilitará el campo de **Alias For**.



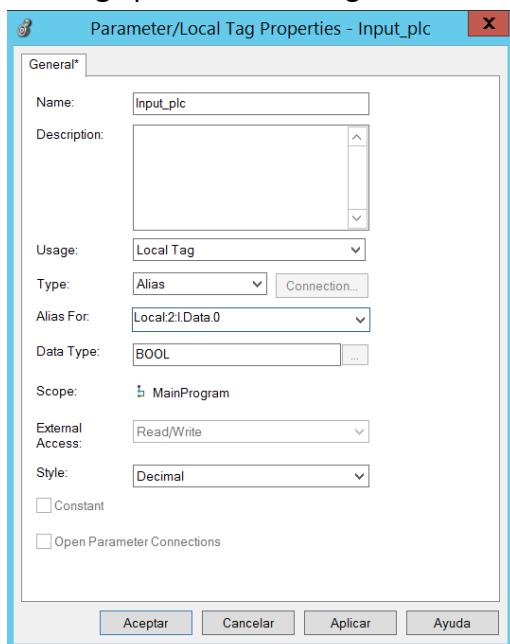
6. En la ventana del buscador se mostrará las variables del controlador y del programa, seleccionar **Local** que representan a las variables al momento de crear los módulos I/O.

Name	Data Type	Usage	Description
input	BOOL	Local	
Input_plc	BOOL	Local	
▶ Local:1:C	AB:1769_IF4X...	<controller>	
▶ Local:1:I	AB:1769_IF4X...	<controller>	
▶ Local:1:O	AB:1769_IF4X...	<controller>	
output	BOOL	Local	
Output_plc	BOOL	<controller>	

7. Seleccionar el bit que queremos vincular con la programación, para esto se debe seleccionar el arreglo de interés, es decir escoger un bit del arreglo 8 bits de **Input** u **Output** respectivamente.



8. La ventana **Parameter/Local Tag** aparecerá de la siguiente forma.

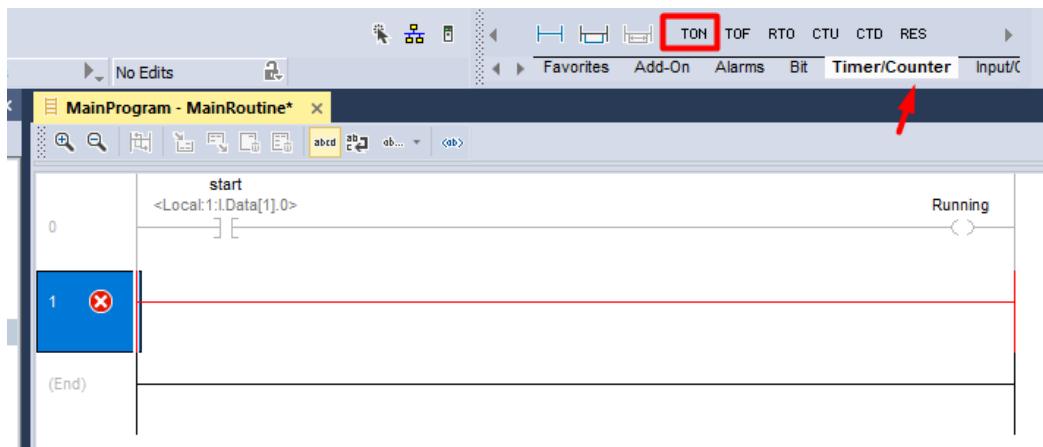


9. Por último, realizar la programación en el peldaño #0 como se muestra en la siguiente imagen.

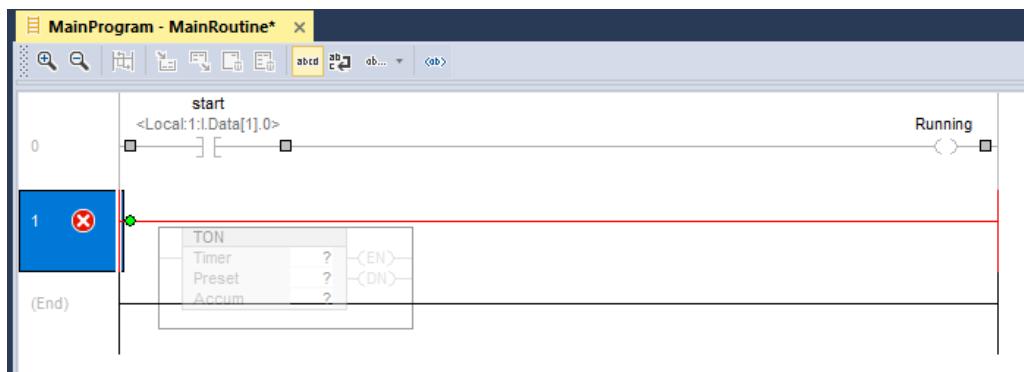


Añadir contadores, temporizadores y comparadores

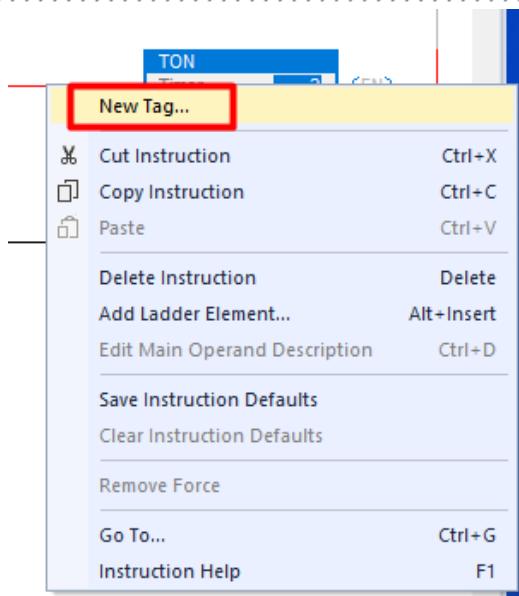
1. En la pestaña **Timer/Counter** de **Instructions Toolbar**, escoger TON que significa temporizador con retardo a la conexión.



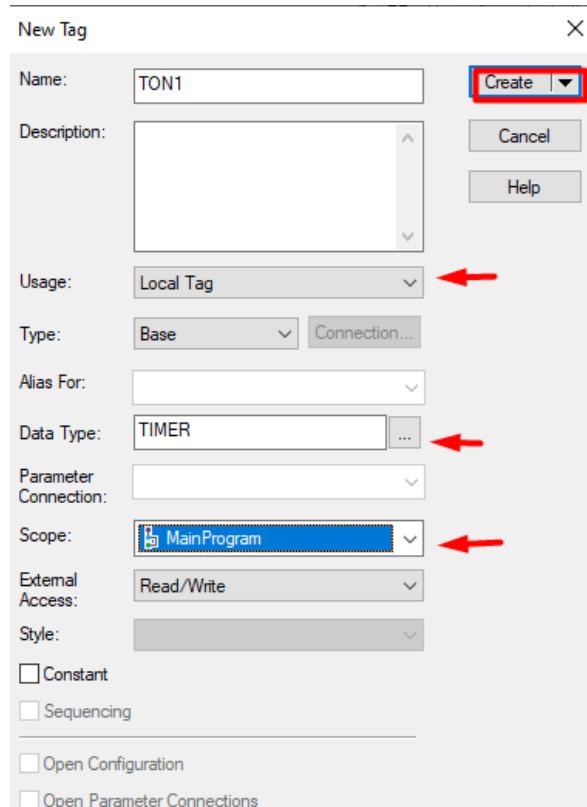
2. A continuación, arrastrar el temporizador seleccionado al peldaño #1 de la programación.



3. Clic derecho a "?" en la sección **Timer** del bloque Ton, luego escoger **New Tag** para la creación de una etiqueta tipo timer del bloque.



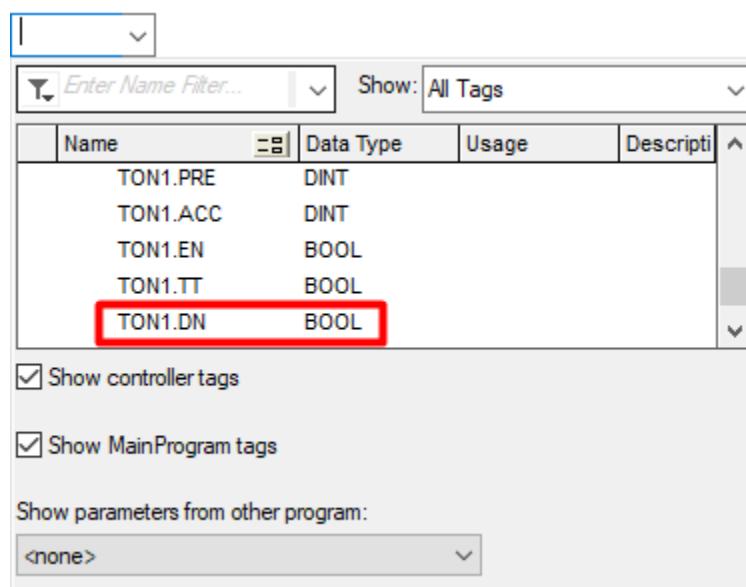
4. La ventana **New Tag** aparecerá. Escribir un nombre a la etiqueta, luego escoger **MainProgram** en **Scope** y que en la sección **Data Type** sea **TIMER**. Dar clic en **Create**.



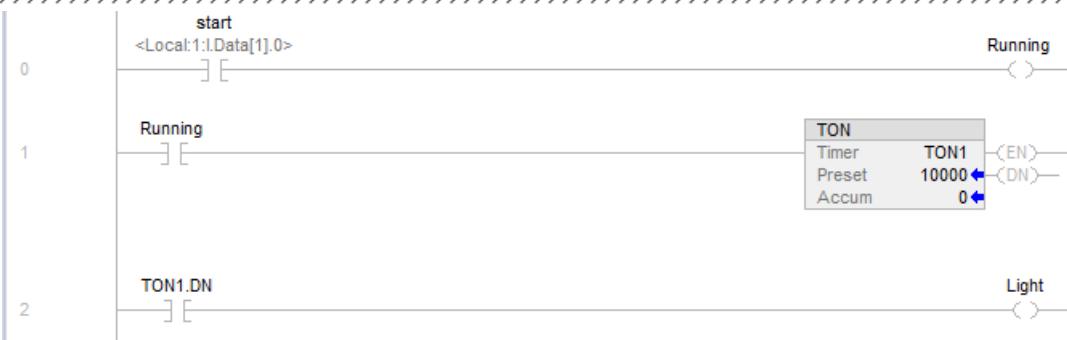
5. El peldaño debe mostrarse como la siguiente figura. En la sección **Preset** del bloque, especificar el valor (unidades en milisegundos) que **Accum** debe alcanzar para que el bit DN se active.



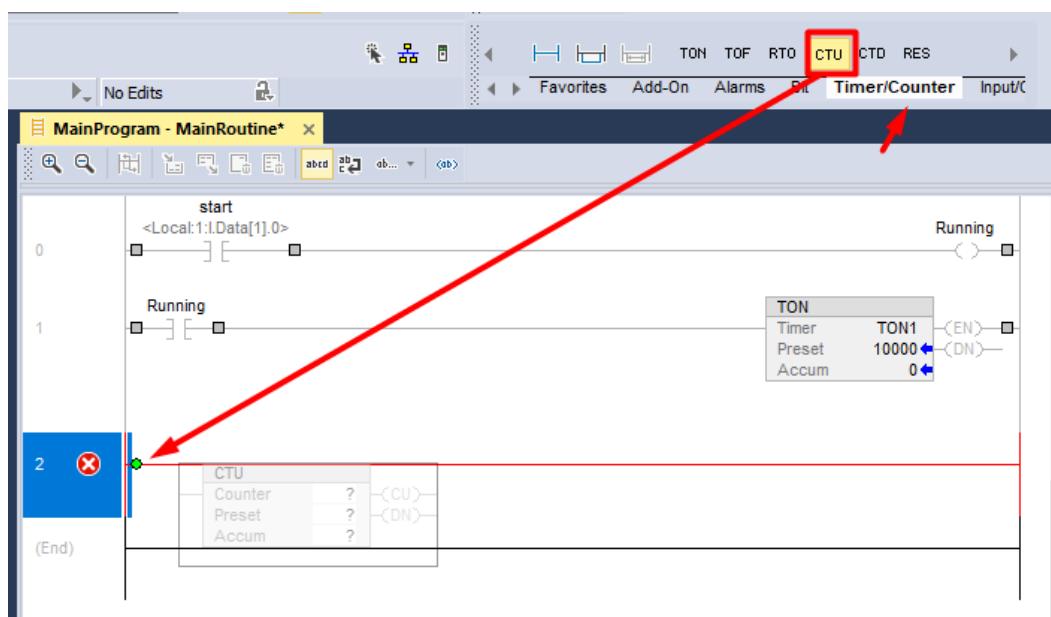
6. Añadir un peldaño y contacto en la programación, luego asignar el **bit DN** de la variable tipo timer al contacto.



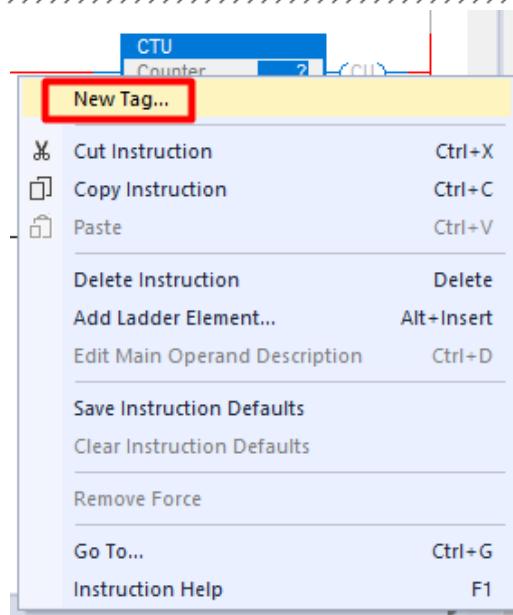
7. La programación debe mostrarse como la siguiente figura.



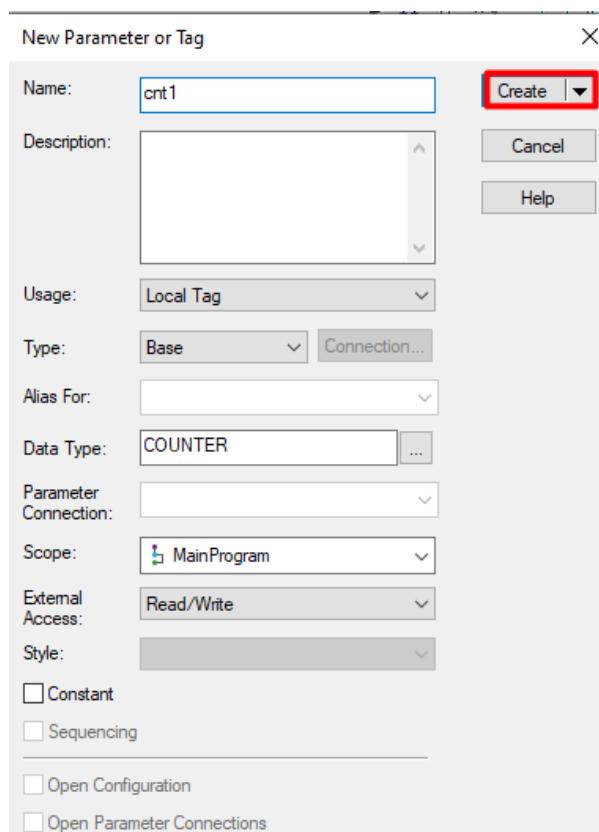
8. En la pestaña **Timer/Counter** de **Instructions Toolbar**, escoger CTU que cuenta hacia arriba cada vez que las transiciones de la condición del peldaño van de un estado de falso a verdadero. Luego, arrastrar el contador seleccionado al peldaño #3 de la programación.



9. Clic derecho a "?" en la sección **Counter** del bloque CTU, luego escoger **New Tag** para la creación de una etiqueta tipo counter del bloque.



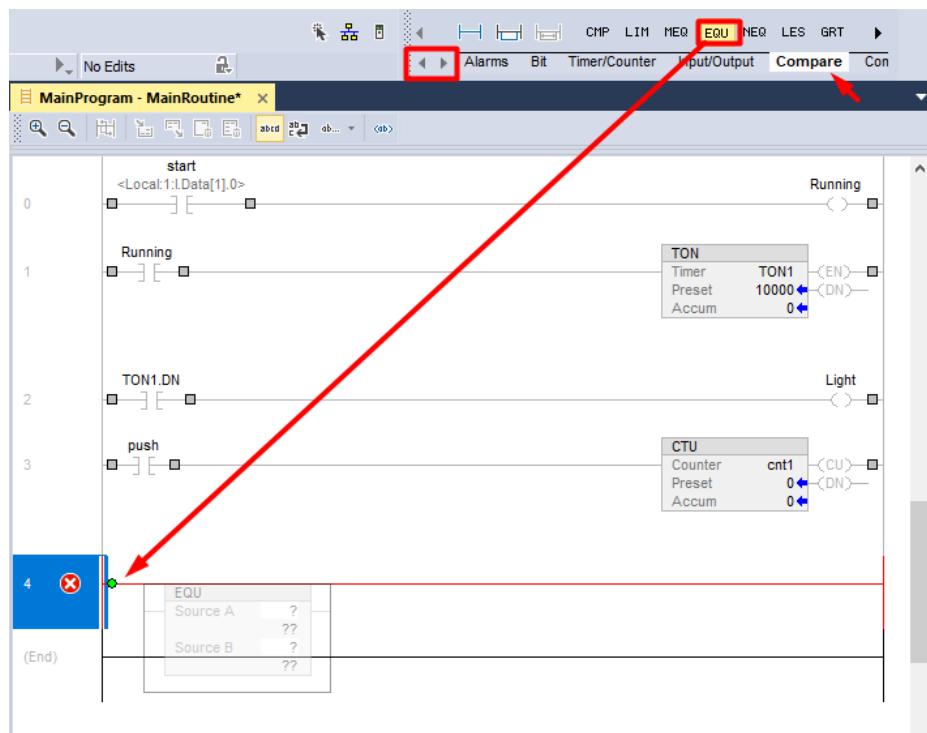
- La ventana **New Tag** aparecerá. Escribir un nombre a la etiqueta, luego escoger **MainProgram** en **Scope** y que en la sección **Data Type** sea **COUNTER**. Dar clic en **Create**.



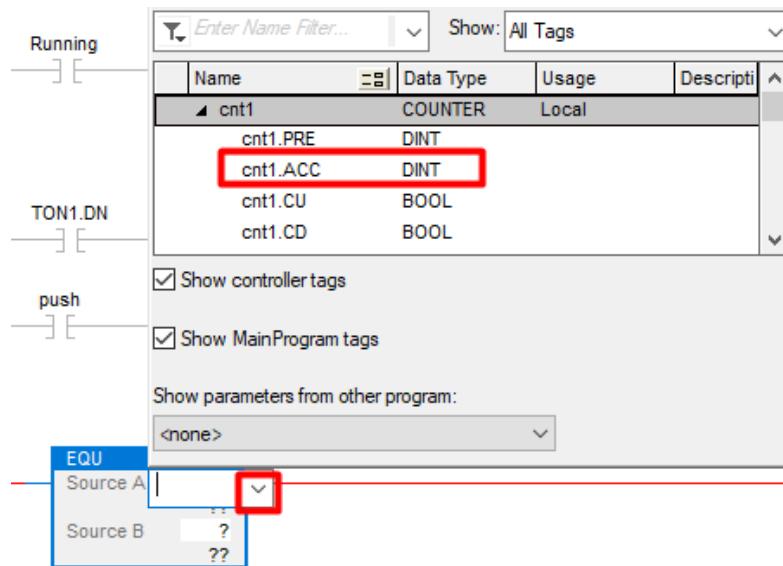
11. El peldaño debe mostrarse como la siguiente figura. En la sección **Preset** del bloque, especificar el valor en que **Accum** debe alcanzar para que el bit DN se active.



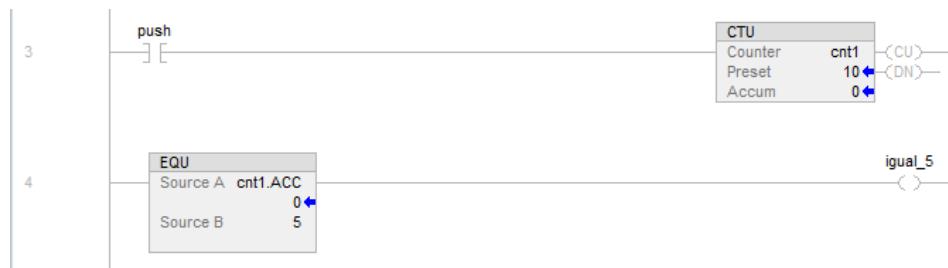
12. En la pestaña **Compare** de **Instructions Toolbar**, escoger **EQU** que compara si dos valores son iguales. Luego, arrastrar el temporizador seleccionado al peldaño #3 de la programación.



13. Desplegar “?” hacia abajo en la sección **Source A** del bloque CTU, luego escoger la etiqueta tipo counter del bloque y seleccionar el parámetro .ACC que especifica el número de transiciones de la instrucción que se ha contado.

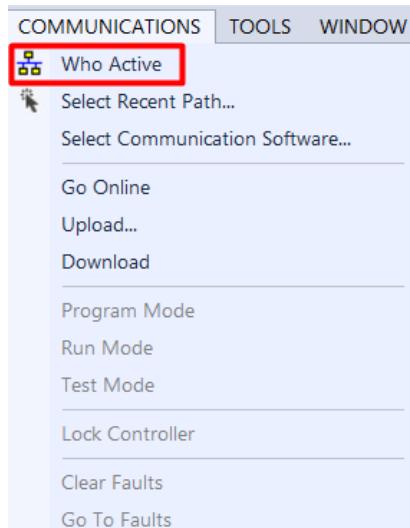


14. La programación debe mostrarse como la siguiente figura.

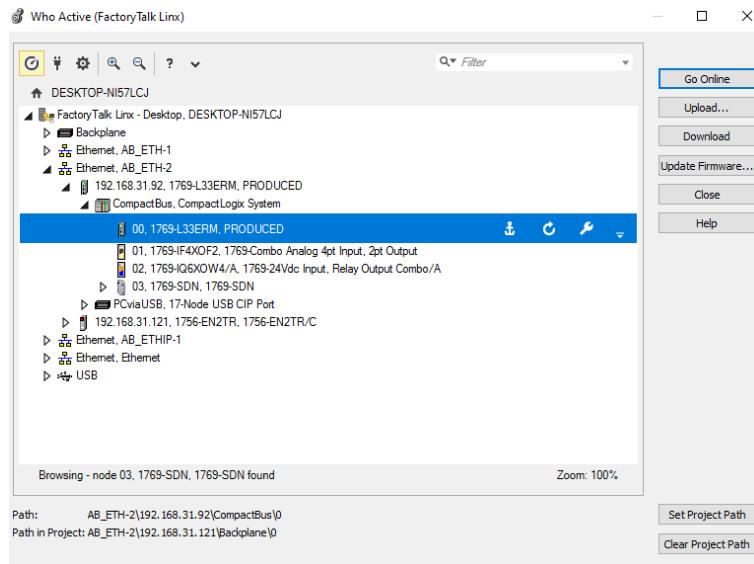


Descargar un proyecto de Logix Designer hacia un controlador

1. Ir a **Communications**, luego seleccionar **Who Active**.



2. En la ventana **Who Active**, abrir la red creada. En esta red se encontrará todos los equipos conectados en la misma. En la red seleccionar el CPU del controlador, dar clic en **Set Project Path**, y finalmente en **Download**.



3. Cambiar al controlador en modo **Run**, dar clic en **Yes**.

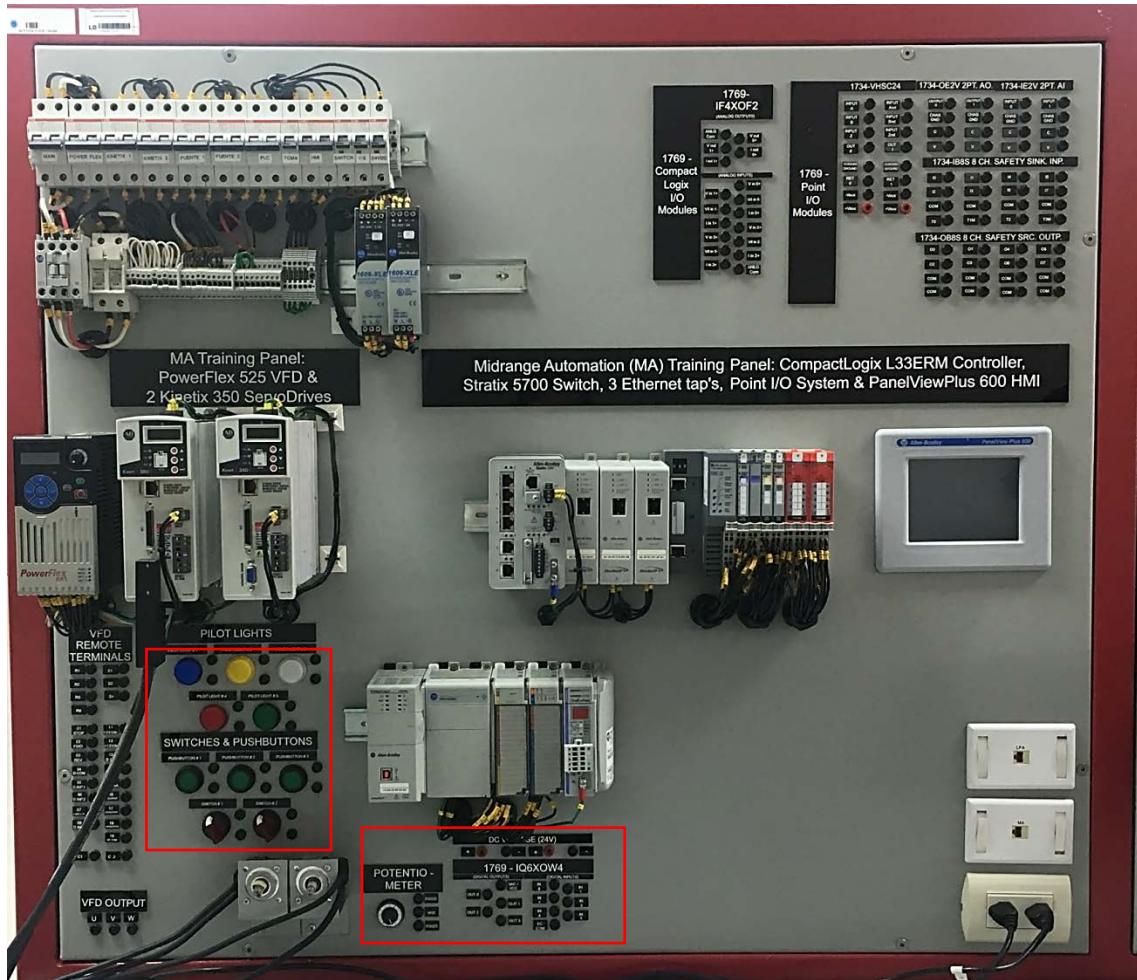


4. Finalmente se puede observar en el estado del controlador que la descarga ha sido realizada con éxito, y el estado del controlador se encuentra en **Run**.



6. Actividades por desarrollar

Conectar entradas y salidas digitales en el tablero Midrange Automation.



Reto:

Realizar la programación de la siguiente aplicación en diagrama de lógica escalera, utilizando el software Studio 5000.

Ejercicio 1

Se pretende automatizar un control de garaje que realice lo siguiente: El garaje dispone de una entrada con barrera, semáforo (rojo/verde) y detector de llegada de coche; y de una salida sin barrera y en la que únicamente hay un detector de salida de coches, como se ilustra en la siguiente figura:

El funcionamiento debe ser el siguiente:

- Inicialmente en el garaje no hay coches, se cuenta el número de coches que entran y salen para controlar el total de coches en el garaje en cada momento.
- Si llega un coche y quedan plazas libres, se debe abrir la barrera por 10 segundos (se ignora el cierre para simplificar).
- Si llega un coche y no quedan plazas libres, no se abre la barrera.
- El semáforo estará verde cuando queden plazas libres y rojo cuando el garaje esté completo. Se supone que el garaje tiene 10 plazas.
- Existe también para reiniciar el controlador a cero manualmente.

