

Guía para conversión de aplicaciones Simatic S7 a Logix5000



Solución de aplicación

Información importante para el usuario

Las características de funcionamiento de los equipos de estado sólido son distintas a las de los equipos electromecánicos. En la publicación Safety Guidelines for the Application, Installation and Maintenance of Solid State Controls (publicación SGI-1.1 que podrá obtener a través de la oficina local de ventas de Rockwell Automation o en línea en la dirección <http://literature.rockwellautomation.com>) se describen algunas diferencias importantes entre los equipos de estado sólido y los dispositivos electromecánicos de lógica cableada. Debido a estas diferencias y a la gran diversidad de usos que se puede dar a los equipos de estado sólido, las personas responsables de la utilización de este equipo deberán asegurarse de la idoneidad de cada una de las aplicaciones concebidas con estos equipos.

Rockwell Automation, Inc. no será responsable en ningún caso de daños directos o indirectos resultantes del uso o la aplicación de este equipo.

Los ejemplos y diagramas incluidos en este manual tienen exclusivamente un fin ilustrativo. Debido al gran número de variables y requisitos asociados con cualquier instalación en particular, Rockwell Automation, Inc. no puede asumir ninguna responsabilidad u obligación por el uso que se haga a partir de los ejemplos y diagramas.

Rockwell Automation, Inc. no asume ninguna obligación de patente relativa al uso de la información, circuitos, equipo o software descritos en este manual.

Se prohíbe la reproducción total o parcial del contenido de este manual sin la autorización por escrito de Rockwell Automation, Inc.

En este manual, cuando es necesario, se utilizan notas para alertarle respecto a consideraciones de seguridad.

| | |
|--|--|
| ADVERTENCIA  | Identifica información sobre prácticas o circunstancias que pueden ocasionar una explosión en un ambiente peligroso y que pueden provocar lesiones personales, la muerte, daños materiales o pérdidas económicas. |
| IMPORTANTE | Identifica información crítica para una correcta aplicación y comprensión del funcionamiento del producto. Sírvase tomar nota de que en esta publicación se usa el punto decimal para separar la parte entera de la decimal de todos los números. |
| ATENCIÓN  | Identifica información sobre prácticas o circunstancias que pueden provocar lesiones personales, la muerte, daños materiales o pérdidas económicas. Las notas de atención le ayudan a identificar un peligro, evitar un peligro y reconocer las consecuencias. |
| PELIGRO DE CHOQUE  | Puede haber etiquetas colocadas sobre el equipo o en el interior del mismo (por ejemplo, en el variador o en el motor) que indican que puede haber un voltaje peligroso. |
| PELIGRO DE QUEMADURA  | Puede haber etiquetas colocadas sobre el equipo o en el interior del mismo (por ejemplo, en el variador o en el motor) para alertar al personal que las superficies pueden alcanzar temperaturas peligrosas. |

Allen-Bradley, Rockwell Automation y TechConnect son marcas comerciales de Rockwell Automation, Inc.

Las marcas comerciales no pertenecientes a Rockwell Automation son propiedad de sus respectivas empresas.

| | | |
|---|---|----|
| Prefacio | Propósito | 7 |
| | Conversión frente a traducción | 7 |
| | Terminología | 8 |
| | Recursos adicionales | 8 |
| | Servicios de conversión de lógica PLC proporcionados por Rockwell Automation | 9 |
| | Características de servicio | 9 |
| | Servicios integrados de conversión de programa PLC | 9 |
| | Ventajas del servicio. | 10 |
| | Servicios ofrecidos | 10 |
| | Paquete de conversión básica | 10 |
| | Paquete de conversión más limpieza inicial. | 10 |
| | Opciones adicionales. | 11 |
| | Conversiones de programas adicionales disponibles. | 11 |
| | | |
| | Capítulo 1 | |
| Conversión de hardware | Introducción. | 13 |
| | Controladores S7. | 13 |
| | Sistemas de E/S | 14 |
| | E/S locales S7 | 14 |
| | Selección y configuración de componentes de E/S de S7. | 14 |
| | E/S locales de Logix. | 16 |
| | Selección y configuración de componentes de E/S de Logix. | 18 |
| | E/S remotas de S7 | 20 |
| | Configuración de E/S remotas Profibus DP de S7 | 21 |
| | E/S distribuidas de Logix | 22 |
| | Configuración de E/S distribuidas de Logix. | 22 |
| | Redes. | 25 |
| | Redes en S7 | 25 |
| | Redes en Logix | 27 |
| | Conversión de HMI. | 31 |
| | Conversión de sistemas que contienen controladores distribuidos | 32 |
| | Implementación de hardware y software | 32 |
| | Conexión de dispositivos de Siemens y Rockwell Automation | 34 |
| | Controladores. | 34 |
| | Dispositivos distribuidos | 34 |
| | | |
| | Capítulo 2 | |
| Características de Logix con las que los usuarios de S7 pueden no estar familiarizados | Introducción. | 35 |
| | Bloques de organización de S7 comparados con las tareas de Logix | 36 |
| | Bloques de organización en S7 | 36 |
| | Tareas en Logix. | 41 |
| | Monitor de tareas. | 46 |
| | Tags en vez de direcciones. | 47 |
| | Áreas de datos en S7 | 47 |
| | Datos en Logix | 50 |
| | E/S y tags de alias | 51 |

| | |
|--|----|
| Lenguajes de programación..... | 53 |
| Diagrama de lógica de escalera Logix..... | 54 |
| Texto estructurado Logix..... | 54 |
| Diagrama de bloques de funciones Logix..... | 55 |
| Diagrama de función secuencial Logix..... | 55 |
| Conversión del código STEP 7 a Logix..... | 55 |
| Matrices en vez de punteros..... | 56 |
| Instrucciones Add-On..... | 57 |
| Resumen de la instrucción Add-On..... | 57 |
| Tags de respaldo..... | 58 |
| El protocolo industrial común (CIP)..... | 58 |
| Visualización de la red..... | 59 |
| Intercambio de datos entre controladores..... | 60 |
| Enviar / recibir en STEP 7..... | 60 |
| Tags producidos / consumidos en Logix..... | 60 |
| Tipos de datos definidos por el usuario..... | 61 |
| Actualización de E/S asíncronas..... | 62 |
| El tipo de datos DINT..... | 62 |
| Gestor de fases..... | 63 |
| Gestión de fases en STEP 7..... | 63 |
| PhaseManager en Logix..... | 63 |
| Hora coordinada del sistema (CST)..... | 65 |
| Entradas de sello de hora..... | 65 |
| Salidas secuenciadas..... | 65 |
| Sin variables temporales..... | 66 |
| No se necesitan acumuladores o registros especiales..... | 66 |

Capítulo 3

Conversión del software del sistema y funciones estándar

| | |
|--|----|
| Introducción..... | 67 |
| Funciones del sistema Logix..... | 68 |
| Copiar..... | 68 |
| Establecimiento y lectura de fecha y hora..... | 69 |
| Leer hora del sistema..... | 69 |
| Manejo de interrupciones..... | 70 |
| Errores..... | 70 |
| Estado – Controlador..... | 71 |
| Estado – Módulo..... | 71 |
| Estado – para OB y tareas..... | 72 |
| Temporizadores..... | 72 |
| Rutinas de conversión..... | 73 |
| Rutinas de manejo de cadenas..... | 73 |
| Ejemplos de llamadas de función del sistema..... | 74 |
| Puesta en hora del reloj..... | 74 |
| Inhabilitación de interrupciones..... | 76 |
| Leer hora del sistema..... | 78 |
| Obtener fallos..... | 79 |
| Información del módulo..... | 80 |
| Obtener tiempo de escán..... | 81 |

Conversión de estructuras de programa típicas

Capítulo 4

| | |
|--|-----|
| Introducción..... | 83 |
| Ejemplos de código de conversión | 83 |
| Traducción de lógica de escalera..... | 83 |
| Saltos y toma de decisiones | 90 |
| Matrices | 94 |
| Tipos de datos de usuario..... | 99 |
| Punteros y matrices | 102 |
| Máquina de estados | 103 |
| Máquina de estados STEP 7 | 104 |
| Cadenas | 108 |
| Variables temporales STEP 7 | 110 |
| Funciones..... | 110 |
| Block Copy, COP y CPS | 114 |
| Expresiones matemáticas | 116 |
| Otros temas relacionados con la programación | 120 |
| Alcance de variables..... | 120 |
| OB, tareas y secuenciación..... | 120 |
| Un ejemplo mayor – módulo de control | 121 |
| Componentes del CM | 121 |
| Tipo datos de usuario Valve | 122 |
| La instrucción Add-On | 123 |
| Datos locales de la instrucción Add-On..... | 124 |
| Llamada | 127 |

Errores comunes que se cometen al convertir a Logix

Capítulo 5

| | |
|--|-----|
| Introducción..... | 129 |
| No seleccionar el hardware apropiado | 129 |
| Subestimar el impacto de la secuenciación de tareas..... | 130 |
| Realizar traducción en lugar de conversión..... | 130 |
| No usar los lenguajes Logix más apropiados..... | 130 |
| Implementación de tipos de datos incorrectos – DINT frente a INT... .. | 131 |
| Añadir DINT | 131 |
| Añadir INT | 131 |
| Resultados de temporización..... | 131 |
| Código de usuario que emula instrucciones existentes..... | 132 |
| Código de usuario | 132 |
| Instrucción COP | 132 |
| Uso incorrecto de COP, MOV y CPS..... | 133 |
| Uso incorrecto de CPT | 133 |
| No manejar las cadenas de una manera óptima | 133 |
| Uso frecuente de saltos..... | 133 |
| No usar tag de alias | 133 |

| | |
|--|--|
| Glosario de S7 a Logix | <p>Capítulo 6</p> <p>Introducción..... 135</p> <p>Terminología de hardware 135</p> <p>Terminología de software 136</p> |
| Componentes de S7 300 y S7 400 y equivalentes de RA | <p>Apéndice A</p> <p>Introducción..... 139</p> <p>CPU S7 300 compactas 140</p> <p>CPU S7 300 estándar..... 140</p> <p>CPU S7 300 de alta tecnología S7 300..... 141</p> <p>CPU S7 300 a prueba de fallos..... 142</p> <p>Módulos de entradas digitales S7 300..... 142</p> <p>Módulos de salidas digitales S7 300 143</p> <p>Módulos de salidas de relés S7 300 144</p> <p>Módulos digitales combinados S7 300..... 144</p> <p>Módulos de entradas analógicas S7 300..... 144</p> <p>Módulos de salidas analógicas S7 300..... 145</p> <p>Módulos analógicos combinados S7 300..... 146</p> <p>Controladores estándar S7 400 146</p> <p>Controladores redundantes y a prueba de fallo 147</p> <p>Módulos de entradas digitales 147</p> <p>Módulos de salidas digitales 147</p> <p>Módulos de entradas analógicas..... 148</p> <p>Módulos de salidas analógicas..... 148</p> |
| Tabla de referencias cruzadas de HMI Siemens | <p>Apéndice B</p> <p>Micropaneles SIMATIC y equivalentes de Rockwell Automation ... 149</p> <p>Paneles SIMATIC serie 7x y equivalentes de Rockwell Automation.. 151</p> <p>Paneles SIMATIC serie 17x y equivalentes de Rockwell Automation .. 152</p> <p>Paneles SIMATIC serie 27x y equivalentes de Rockwell Automation .. 155</p> <p>Multipaneles SIMATIC serie 27x y equivalentes de Rockwell Automation..... 157</p> <p>Multipaneles SIMATIC serie 37x y equivalentes de Rockwell Automation..... 159</p> |

Propósito

Este manual del usuario proporciona orientación a usuarios e ingenieros que han usado sistemas de control basados en una de estas dos plataformas:

- Controladores S7 de Siemens
- Controladores de automatización programable (PAC) Logix de Rockwell Automation

Y, además:

- desean o necesitan aprovechar las funciones de los PAC o están en las primeras etapas de migración de un sistema S7 a Logix.
- tienen código de programa STEP 7 específico que desean convertir a código eficiente y eficaz de RSLogix 5000.

Use este manual como ayuda para adoptar buenas prácticas y evitar errores comunes al convertir un proyecto a Logix.

Conversión frente a traducción

El tema de conversión frente a traducción se presenta repetidamente en esta guía de conversión de aplicaciones. Una traducción simple se concentra sólo en la línea de código y en encontrar un equivalente en los lenguajes Logix. Para convertir una aplicación de manera óptima, usted necesita algo más que simplemente traducir. Por ejemplo, puede beneficiarse de elegir un lenguaje de programación diferente, utilizar técnicas de programación y designar un esquema de secuenciación diferente para resolver la misma tarea. Por lo tanto, la conversión se realiza en un contexto de diseño de más alto nivel y conocimiento de las cualidades del sistema Logix.

Si tiene que convertir código de aplicación, necesitará entender su programa STEP 7 antes de comenzar la conversión. Para entenderlo necesitará haber participado personalmente en su desarrollo o haber leído la documentación del programa y del proceso que controla. Si el programa o el proceso no se conocen bien o su documentación es deficiente, será difícil realizar una correcta conversión y se obtendrá una simple traducción con menos probabilidad de éxito. Por ejemplo, en Logix hay un espacio de nombre global, mientras que en el entorno Siemens hay bloques de datos que pueden cargarse/descargarse por código de aplicación. El poder apreciar esto le ayudará a diseñar una estrategia de conversión.

En algunos casos, si la documentación del proceso y del programa es deficiente, puede ser más eficiente en lo que respecta a duración/costo de todo el proyecto generar una nueva especificación y comenzar su programa Logix dedicando un tiempo mínimo a traducir el programa antiguo.

Terminología

STEP 7 es el entorno de software de programación para los controladores SIMATIC S7 de Siemens. El software RSLogix 5000 se usa con los controladores de automatización programables Logix de Rockwell Automation. Nos referimos al sistema Logix como un controlador de automatización programable porque va más allá que un PLC para uso general tradicional. Proporciona una plataforma de control excelente para control multidisciplinario, un espacio de nombre común, una hora coordinada del sistema para arquitecturas de múltiples CPU realmente escalables, tipos de datos definidos por el usuario y una total conectividad NetLinx.

El término “Logix” se usa para referirse a cualquiera de los controladores ControlLogix, CompactLogix, GuardLogix, FlexLogix, DriveLogix o SoftLogix, o al entorno de programación RSLogix 5000 donde se hace evidente del contexto en el que se hace mención.

Recursos adicionales

Cada sección de esta guía de conversión de aplicaciones se refiere a otros manuales de usuario, guías de selección y documentos de Rockwell Automation en donde puede encontrarse más información.

| Número de publicación | Título de la publicación |
|-----------------------|--|
| 1756-SG001 | ControlLogix Controllers Selection Guide |
| 1769-SG001 | 1769 CompactLogix Controllers Selection Guide |
| 1768-UM001 | 1768 CompactLogix Controllers User Manual |
| 1769-SG002 | Compact I/O Selection Guide |
| 1756-RM094 | Logix5000 Controllers Design Considerations Programming Manual |
| 1756-PM001 | Logix5000 Controllers Common Procedures Programming Manual |
| 1756-RM003 | Logix5000 Controllers General Instructions Reference Manual |
| 1734-SG001 | POINT I/O Selection Guide |
| 1738-SG001 | ArmorPoint I/O Selection Guide |
| 1792-SG001 | ArmorBlock MaXum I/O and ArmorBlock I/O Selection Guide |
| 1794-SG002 | FLEX I/O and FLEX Ex Selection Guide |
| NETS-SG001 | NetLinx Selection Guide |
| VIEW-SG001 | Visualization Platforms Selection Guide |
| IA-RM001 | Integrated Architecture: Foundations of Modular Programming |
| 6873-SG004 | Encompass Program Product Directory |
| 1756-PM010 | Logix5000 Controllers Add-On Instructions Programming Manual |
| 1756-RM087 | Logix5000 Controllers Execution Time and Memory Use Reference Manual |
| IASIMP-RM001 | IA Recommended Literature Reference Manual |

Servicios de conversión de lógica PLC proporcionados por Rockwell Automation

Rockwell Automation proporciona servicios adicionales para la conversión de lógica de PLC.

- Características del servicio
- Servicios integrados de conversión de programas PLC
- Ventajas del servicio
- Servicios ofrecidos
- Paquete de conversión básica
- Paquete de conversión más limpieza inicial
- Conversiones de programas adicionales disponibles

Características del servicio

Los servicios de conversión de programas convertirán su programa de controlador programable de otros fabricantes o de PLC de Allen-Bradley de versiones anteriores a fin de que puedan ejecutarse en un sistema de control de automatización programable Logix, o en los controladores programables SLC 500/MicroLogix o PLC-5.

Generalmente es costoso ofrecer servicio de asistencia técnica a los productos de versiones anteriores y es difícil repararlos, lo cual puede aumentar el tiempo improductivo y reducir la producción. Por esta razón el servicio de asistencia técnica al cliente de Rockwell Automation ofrece ahora servicios de conversión de programas. Estos servicios están diseñados para reducir el costo y el tiempo requerido para migrar de un PLC de una versión anterior a una de nuestras familias actuales de plataformas de control PLC o PAC.

Servicios integrados de conversión de programas PLC

La migración a una plataforma de control Allen-Bradley actual desde un producto de una versión anterior mejorará su proceso de fabricación, la confiabilidad y flexibilidad del sistema, le dará más acceso a la potencia de procesamiento de la aplicación, y reducirá los costos de reparación del servicio y el inventario de repuestos. Con los servicios de conversión de programas de la unidad del grupo de asistencia técnica al cliente de Rockwell Automation, su programa de controlador programable actual se convertirá rápida y eficientemente a la nueva familia de controladores. Los ingenieros de asistencia técnica al cliente de Rockwell Automation pueden ayudar en la tarea de migración de equipos Allen-Bradley de versiones anteriores o convertir sus sistemas de PLC a productos de Rockwell Automation, minimizando a la vez el tiempo improductivo y maximizando el éxito de la operación.

Ventajas del servicio

Durante el proceso de conversión del programa participarán especialistas en cada plataforma de productos. No hay anomalía que sea difícil de encontrar en la lógica debido a errores de escritura. En la mayoría de los casos se reproduce toda la tabla de datos y no se pierde ningún dato, y además se conserva la documentación original y no se necesita volver a escribir comentarios ni símbolos. Los programas originales de la marca Allen-Bradley pueden estar en formato de la serie AI, APS o 6200. Los nuevos programas estarán en el formato RSLogix apropiado.

Servicios ofrecidos

Hay dos paquetes de conversión de programas disponibles, además de paquetes personalizados para proyectos específicos que se ofrecen en aplicaciones particulares.

Paquete de conversión básica

- El programa de controlador programable original se convertirá al formato apropiado de ControlLogix, CompactLogix, PLC-5 o SLC 500/MicroLogix.
- El paquete proporciona una lista de errores generada durante la conversión que incluye instrucciones que no se convierten directamente y cualquier dirección que pueda no haberse convertido, lo cual podría incluir punteros y direccionamiento indirecto.
- El programa y la lista de errores se devolverían al cliente para su corrección y depuración manual.

Paquete de conversión más limpieza inicial

- El programa de controlador programable original se convertirá al formato apropiado de ControlLogix, PLC-5 o SLC 500/MicroLogix.
- Corregiremos y convertiremos todas las instrucciones y/o los errores de direccionamiento a la nueva familia de procesadores.
- Una vez completado, el programa se devolverá al cliente para la puesta en marcha y depuración final.

Opciones adicionales

Las opciones adicionales a cualquiera de los paquetes incluyen lo siguiente:

- Asistencia técnica por teléfono a nivel de la aplicación durante la puesta en marcha y la fase de depuración del proyecto.
- Consultoría sobre reingeniería del sistema, interface de operador, arquitectura y estrategias de comunicación, para aprovechar al máximo las capacidades de control de la nueva plataforma que no forman parte del esfuerzo de traducción de código, capacitación y puesta en marcha en las instalaciones, se ofrecen como valor agregado a través de la oficina local de GSS (Global Sales and Solutions).
- La oficina local de GSS/sistemas con ingeniería incorporada ofrecen servicios de migración completa llave en mano o actualizaciones.

Conversiones de programas adicionales disponibles

- Formato PLC-2 a formato ControlLogix, CompactLogix, PLC-5, SLC500/MicroLogix
- Formato PLC-3 a formato ControlLogix, CompactLogix o PLC-5.
- Formato PLC-5/250 a formato ControlLogix o CompactLogix.
- Modicon – Quantum, 984, 584, 380, 381, 480, 485, 780, 785 a formato ControlLogix o CompactLogix.
- Siemens – S-5, S-7 a formato ControlLogix o CompactLogix.
- TI – 520, 520C, 525, 530, 530C, 535, 560, 560/565, 565, 560/560T, 560T, 545, 555, 575 a formato ControlLogix o CompactLogix.
- GE serie 6 a formato ControlLogix o CompactLogix.

Conversiones de programas a controladores programables de otros fabricantes o programas de controladores Allen-Bradley también disponibles. Comuníquese con el grupo de asistencia técnica para mayores detalles.

Para programar un proyecto de conversión o para obtener más información acerca de los servicios de conversión de programas, comuníquese con la oficina local de ventas o con el distribuidor autorizado de Rockwell Automation: envíenos un correo electrónico a raprogramconversion@ra.rockwell.com, o visite <http://support.rockwellautomation.com/> y vea el documento G19154 de la base de conocimientos.

IMPORTANTE

Use los servicios de consultoría para reingeniería, típicamente con el fin de ampliar la funcionalidad del sistema y no para cambiar hardware debido a obsolescencia o razones relacionadas. Las conversiones de formato SLC a Logix y de PLC-5 a Logix, así como la generación de comentarios PCE están incorporadas en el software RSLogix 5000.

Notas:

Conversión de hardware

Introducción

El objetivo de este capítulo es orientar al usuario o ingeniero que necesita determinar el hardware Logix correcto que servirá de reemplazo para el equipo S7 existente.

El capítulo describe cómo seleccionar controladores, E/S locales, E/S remotas, redes y HMI, además de incluir una sección sobre la arquitectura de controladores distribuidos y proporcionar ejemplos de conversión de hardware de los módulos S7 usados más frecuentemente.

| Tema | Página |
|---|--------|
| Controladores S7 | 13 |
| Sistemas de E/S | 14 |
| Redes | 25 |
| Conversión de HMI | 31 |
| Conversión de sistemas que contienen controladores distribuidos | 32 |
| Conexión de dispositivos Siemens y Rockwell Automation | 34 |

Controladores S7

Esta tabla contiene una selección de ejemplos relevantes de los controladores Siemens S7 actuales, los cuales se usan en una amplia gama de aplicaciones.

Ejemplo de selección de controladores Siemens S7 actuales

| Controlador | Número de parte | Equivalente Logix |
|-------------|---------------------|----------------------|
| 313C | 6ES7 313-5BF03-0AB0 | L23 en serie |
| 314C-DP | 6ES7 314-6CG03-0AB0 | L23 EtherNet/IP, L31 |
| 315-2 DP | 6ES7 315-2AG10-0AB0 | L32E, L32C |
| 317-2 DP | 6ES7 317-6TJ10-0AB0 | L35CR, L35E |
| 317T-2 DP | 6ES7 317-6TJ10-0AB0 | L43, L45 |
| 319-3 PN/DP | 6ES7 318-3EL00-0AB0 | L45, L61 |
| 414-2 | 6ES7 414-2XK05-0AB0 | L61, L62 |
| 414-3 | 6ES7 414-3XM05-0AB0 | L62, L63, L64, L65 |
| 414-3 PN/DP | 6ES7 414-3EM05-0AB0 | |

Ejemplo de selección de controladores Siemens S7 actuales

| | | |
|---------------------------------|--|--|
| 315F-2 PN/DP (seguridad) | 6ES7 315-2FH13-0AB0 6ES7 317-2FK13-0AB0 | GuardLogix L61S, L62S, L63S |
| 414-H (redundante) 417-H | 6ES7 414-4HM14-0AB0 6ES7 417-4HT14-0AB0 | L61-L65 con SRM |
| PCS7 – Usa el controlador 417-4 | | L3x, L4x, L6x + software FactoryTalk Batch, FactoryTalk View |

A continuación se proporciona una guía donde se indica la idoneidad de algunos de los controladores S7 usados más frecuentemente:

- S7 315-2DP – Máquinas de tamaño pequeño a mediano.
- S7 317-2DP – Máquinas de tamaño mediano a mediano-grande y aplicaciones de control de proceso pequeñas a medianas.
- S7 414-2 – Aplicaciones exigentes de control de máquinas y control de procesos.
- S7 414-3 – Aplicaciones exigentes de control de máquinas y control de procesos grandes.

La gama completa de controladores S7 se presenta en el [Apéndice A](#).

Sistemas de E/S

Estas secciones describen los sistemas de E//S Logix usados para reemplazar el equipo S7 existente.

E/S locales S7

Existe una amplia gama de módulos de E/S S7-300 y S7-400. Los módulos S7-300 se montan en un riel DIN estándar y se conectan a las tarjetas adyacentes mediante los conectores en U suministrados con los módulos. Los módulos S7-400 se montan en el rack S7-400.

Selección y configuración de los componentes de E/S de S7

Las capturas de pantalla siguientes provienen del programa de configuración de hardware STEP 7, un programa separado en el conjunto de aplicaciones STEP 7. En el software RSLogix 5000, esta funcionalidad está totalmente integrada, como verá posteriormente en este manual del usuario.

Programa de configuración de hardware STEP 7

| Slot | Module | Order number | Firmware | MPI address | I address | Q address | Comment |
|------|--------------|---------------------|----------|-------------|-----------|-----------|---------|
| 1 | PS 407 10A | 6ES7 407-0KA02-0AA0 | | | | | |
| 3 | CPU 414-3 DP | 6ES7 414-3XJ04-0AB0 | V4.1 | 2 | | | |
| X2 | DP | | | | 8151* | | |
| X1 | MPI/DP | | | 2 | 8150* | | |
| IF1 | | | | | | 8147 | |
| 5 | CP 443-1(1) | 6GK7 443-1EX11-0-E0 | V2.5 | | | | |
| 6 | | | | | | | |
| 7 | | | | | | | |
| 8 | | | | | | | |
| 9 | | | | | | | |
| 10 | | | | | | | |
| 11 | | | | | | | |
| 12 | | | | | | | |
| 13 | | | | | | | |
| 14 | | | | | | | |
| 15 | | | | | | | |

6ES7 421-1BL00-0AA0
Digital input module DI32, 24 VDC, grouping 32

Arrastre el módulo seleccionado a la pantalla de configuración de rack.

| Slot | Module | Order number | Firmwar |
|------|---------------|---------------------|---------|
| 1 | PS 407 4A | 6ES7 407-0DA01-0AA0 | |
| 2 | | | |
| 3 | CPU414-2DP(1) | 6ES7 414-2XG03-0AB0 | V3.1 |
| X2 | DP | | |
| X1 | MPI/DP | | |
| 4 | DI32xDC 24V | 6ES7 421-1BL00-0AA0 | |
| 5 | | | |

E/S locales Logix

Hay una amplia gama de módulos de E/S ControlLogix y CompactLogix disponibles. El módulo de E/S 1769 está optimizado en cuanto a costos, justo con funcionalidad suficiente que solicitan a menudo los fabricantes de equipos originales, mientras que la familia de E/S 1756 proporciona un mayor nivel de características y funcionalidad para satisfacer las aplicaciones más exigentes que solicitan a menudo los usuarios finales y que algunas veces se requieren para satisfacer niveles de rendimiento específicos.

Los módulos CompactLogix se montan en un riel DIN estándar y un sistema de acoplamiento especial asegura la conexión eléctrica y mecánica a los módulos adyacentes. Los ingenieros le darán una gran acogida al sistema de acoplamiento mecánico –con el S7-300– mediante el cual los módulos se fijan a un riel especial y no uno a otro (excepto por el conector en U eléctrico).

Los módulos ControlLogix se montan en los racks 1756.

- Para los controladores 1769-L31, 1769-L32C, 1769-L32E y 1768-L43, el número máximo de módulos de E/S conectados al rack del controlador es 16, en hasta 3 bancos.
- Para los controladores 1769-L35CR, 1769-L35E y 1768-L45, el número máximo de módulos de E/S conectados al rack del controlador es 30, también en 3 bancos.
- Para los controladores 1756, el número de ranuras en el rack define el número máximo de módulos de E/S. Pueden ser 4, 7, 10, 13 ó 17.

En ambas plataformas es posible conectar en red E/S adicionales mediante redes CIP, donde las redes EtherNet/IP y ControlNet proporcionan la integración de E/S más sólida y transparente.

Esta tabla presenta los equivalentes Logix para algunos módulos de E/S S7 populares.

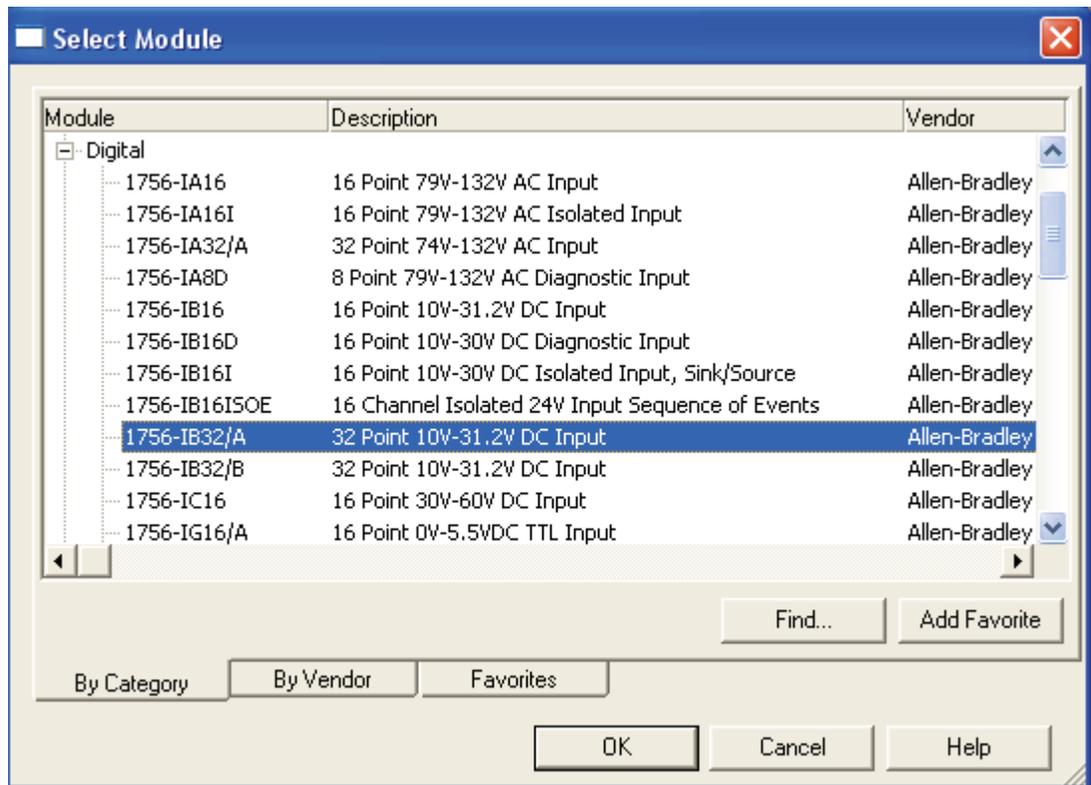
Equivalentes Logix para módulos de E/S S7

| Módulo de E/S S7 | Descripción | Equivalente Logix | Descripción |
|-------------------------|---------------------------------------|--------------------------|---|
| 6ES7 321-1BL00-0AA0 | S7-300, entrada digital de 32 canales | 1769-IQ32 | CompactLogix, entrada digital de 32 canales |
| 6ES7 322 - 1BH01-0AA0 | S7-300, salida digital de 16 canales | 1769-OB16 | CompactLogix, salida digital de 16 canales |
| 6ES7 421-1BL01-0AA0 | S7-400, entrada digital de 32 canales | 1756-IB32 | ControlLogix, entrada digital de 32 canales |
| 6ES7 422-1BH01-0AA0 | S7-400, salida digital de 16 canales | 1756-OB16E | ControlLogix, salida digital de 16 canales |

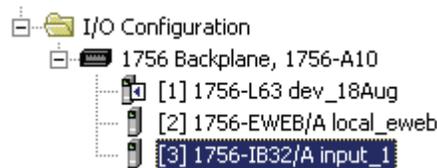
Consulte en el [Apéndice A](#) las tablas de conversión de módulos de E/S más detalladas.

Selección y configuración de componentes de E/S de Logix

Puede tener acceso a la biblioteca Logix de perfiles de dispositivos desde la bifurcación I/O Configuration de su árbol de proyectos. Estos perfiles proporcionan una configuración completa mediante asistentes de software para una integración completa y fácil de usar en la tabla de datos y control programable intuitivo sobre cada una de las funcionalidades del módulo, tales como escalado, alarmas y diagnósticos.



Seleccione un ítem y éste aparecerá en el rack en su configuración de E/S.



Los tags de perfil de dispositivo para el nuevo módulo de E/S se han añadido automáticamente a la base de datos de tags bajo el control del controlador.

| | | | |
|---|-----------|--|----------------|
| + | Local:3:C | | AB:1756_DI:C:0 |
| + | Local:3:I | | AB:1756_DI:I:0 |

La vista siguiente muestra los tags parcialmente expandidos.

| | | | |
|---------------------------------|--|--|----------------|
| [-] Local:3:C | | | AB:1756_DI:C:0 |
| [+] Local:3:C.FilterOffOn_0_7 | | | SINT |
| [+] Local:3:C.FilterOnOff_0_7 | | | SINT |
| [+] Local:3:C.FilterOffOn_8_15 | | | SINT |
| [+] Local:3:C.FilterOnOff_8_15 | | | SINT |
| [+] Local:3:C.FilterOffOn_16_23 | | | SINT |
| [+] Local:3:C.FilterOnOff_16_23 | | | SINT |
| [+] Local:3:C.FilterOffOn_24_31 | | | SINT |
| [+] Local:3:C.FilterOnOff_24_31 | | | SINT |
| [+] Local:3:C.COSOnOffEn | | | DINT |
| [+] Local:3:C.COSOffOnEn | | | DINT |
| [+] Local:3:I | | | AB:1756_DI:I:0 |

El perfil contiene datos de configuración y estado así como datos de E/S.

| | | | |
|---------------------------------|--|--|----------------|
| [-] Local:0:C | | | AB:1756_DI:C:0 |
| [+] Local:0:C.FilterOffOn_0_7 | | | SINT |
| [+] Local:0:C.FilterOnOff_0_7 | | | SINT |
| [+] Local:0:C.FilterOffOn_8_15 | | | SINT |
| [+] Local:0:C.FilterOnOff_8_15 | | | SINT |
| [+] Local:0:C.FilterOffOn_16_23 | | | SINT |
| [+] Local:0:C.FilterOnOff_16_23 | | | SINT |
| [+] Local:0:C.FilterOffOn_24_31 | | | SINT |
| [+] Local:0:C.FilterOnOff_24_31 | | | SINT |
| [+] Local:0:C.COSOnOffEn | | | DINT |
| [+] Local:0:C.COSOffOnEn | | | DINT |
| [-] Local:0:I | | | AB:1756_DI:I:0 |
| [+] Local:0:I.Fault | | | DINT |
| [+] Local:0:I.Data | | | DINT |
| [+] Local:0:O | | | DINT |

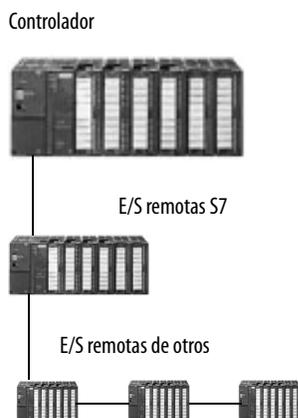
Consulte el [Capítulo 4](#) para obtener más información.

E/S remotas de S7

Es una práctica común dividir las E/S entre el rack local del controlador y las estaciones de E/S remotas, con las comunicaciones por la red Profibus DP. Estos son los tipos de nodos Profibus DP:

- E/S remotas S7, en cuyo caso los módulos de E/S S7-300 estándar se montan en un panel de E/S remoto y se interconectan con el bus Profibus DP mediante un módulo especial. El controlador ve estas E/S como E/S locales y les asigna direcciones de E/S estándar. A esto se le llama ET200M.
- Otras E/S remotas Siemens son el ET200S (similar al sistema POINT I/O) y el ET200L (similar al sistema FLEX I/O).
- E/S remotas de otros fabricantes. Hay una serie de fabricantes de sistemas de E/S y válvulas que producen una interfaz para vincular sus sistemas con el bus Profibus DP de la misma manera que el sistema de E/S remotas S7. Para estos sistemas, quizás se necesite importar un archivo de integración especial (archivo GSD) para la instalación de STEP 7.
- Algunos fabricantes de dispositivos más complejos, tales como básculas y variadores de velocidad variable (VSD), producen interfaces Profibus DP para sus productos. Para estos sistemas, se necesitará importar un archivo de integración especial (archivo GSD) para la instalación de STEP 7. Con frecuencia se necesita consultar la documentación del fabricante para conocer el significado de las áreas de datos.

Configuración típica de E/S de S7



Configuración de E/S remotas Profibus DP de S7

Para instalar un módulo de interface Profibus DP en la configuración de hardware, se arrastra desde el catálogo de hardware hasta el gráfico del bus Profibus DP. Una vez instalado el módulo de interface, pueden agregarse módulos S7-300 estándares como si fuera un sistema de E/S locales.

| Slot | Module | Order Number | I Address | Q Address |
|------|------------------|---------------------|-----------|-----------|
| 1 | | | | |
| 2 | IM 153-1 | 6ES7 153-1AA03-0AB0 | 8187 | |
| 3 | | | | |
| 4 | AI8x12Bit | 6ES7 331-7KF02-0AB0 | 512...527 | |
| 5 | AO4x12Bit | 6ES7 332-5HD01-0AB0 | | 512...519 |
| 6 | D116xDC24V | 6ES7 321-1BH02-0AA0 | 0...1 | |
| 7 | D18/DO8x24V/0.5A | 6ES7 323-1BH00-0AA0 | 24 | 0 |
| 8 | | | | |

La tabla de datos define direcciones de E/S asociadas con el variador. Los símbolos para estas direcciones se añadirían manualmente en la tabla de símbolos. La configuración de hardware ahora está completa.

Es posible usar dispositivos remotos en la red Profibus DP junto con Logix, pero con las mismas restricciones/limitaciones de uso que usted tiene en el entorno S7.

E/S distribuidas de Logix

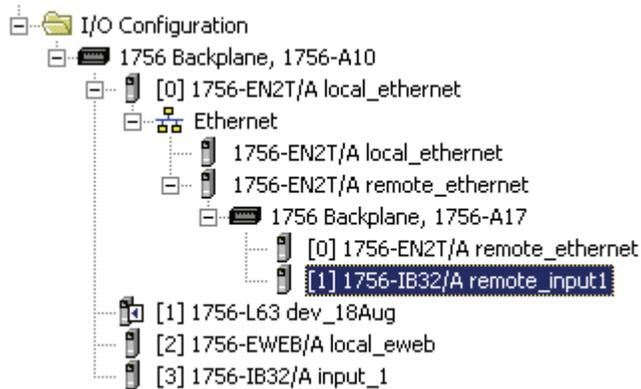
Los sistemas de E/S distribuidas de Rockwell Automation incluyen E/S remotas mediante los módulos de E/S 1756 ó 1769, así como varias plataformas de E/S distribuidas tales como los sistemas POINT I/O, FLEX I/O, ArmorPoint y ArmorBlock.

Los módulos de E/S se conectan a la red mediante un módulo de comunicación o un adaptador de comunicación, o bien directamente usando una interface de comunicación incorporada.

Configuración de E/S distribuidas de Logix

Toda la configuración de E/S se hace en el árbol de proyectos del software RSLogix 5000. Desde la bifurcación I/O Configuration, inserte un módulo de comunicación para el tipo de red que usted haya elegido.

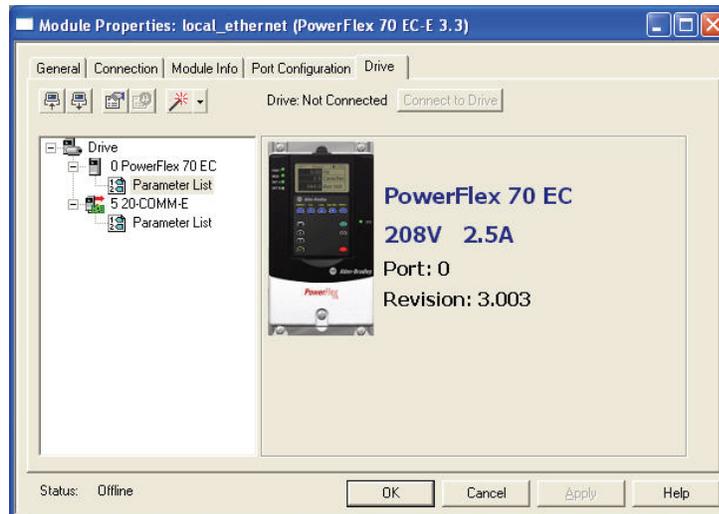
La captura de pantalla muestra una adición de un módulo de E/S 1756-IB32 remoto conectado mediante una red EtherNet/IP.



Observe que los tags correspondientes al módulo de E/S remotas se han añadido automáticamente a la base de datos de tags bajo el control del controlador.

| | | | |
|--------------------------------------|--|--|---------------------------|
| [-] remote_ethernet:I | | | AB:1756_ENET_17SLOT:I:0 |
| [+] remote_ethernet:I.SlotStatusBits | | | DINT |
| [+] remote_ethernet:I.Slot | | | AB:1756_ENET_SLOT:I:0[17] |

De la misma manera puede añadirse un variador de velocidad variable conectado en red, tal como el variador PowerFlex.



Nuevamente, el software RSLogix 5000 generará los nuevos tags automáticamente para cualquier dispositivo con un perfil en el software RSLogix 5000 y conectado a una red EtherNet/IP o ControlNet. En el caso de la red DeviceNet, GuardLogix Safety I/O se integra de la misma manera. Otros dispositivos DeviceNet necesitan configurarse mediante el software de configuración RSNNetWorx y los archivos EDS que operan esencialmente de forma equivalente al software administrador STEP 7 Profibus y los archivos GSD.

A continuación se muestran tags de perfil de dispositivo en el software RSLogix 5000, disponibles para cientos de dispositivos de Rockwell Automation.

| | |
|--|--|
| - PowerFlex_Drive:I | |
| + PowerFlex_Drive:I.DriveStatus | |
| - PowerFlex_Drive:I.DriveStatus_Ready | |
| - PowerFlex_Drive:I.DriveStatus_Active | |
| - PowerFlex_Drive:I.DriveStatus_CommandDir | |
| - PowerFlex_Drive:I.DriveStatus_ActualDir | |
| - PowerFlex_Drive:I.DriveStatus_Accelerating | |
| - PowerFlex_Drive:I.DriveStatus_Decelerating | |
| - PowerFlex_Drive:I.DriveStatus_Alarm | |
| - PowerFlex_Drive:I.DriveStatus_Faulted | |
| - PowerFlex_Drive:I.DriveStatus_AtSpeed | |
| - PowerFlex_Drive:I.DriveStatus_LocalID0 | |
| - PowerFlex_Drive:I.DriveStatus_LocalID1 | |
| - PowerFlex_Drive:I.DriveStatus_LocalID2 | |
| - PowerFlex_Drive:I.DriveStatus_SpdRefID0 | |
| - PowerFlex_Drive:I.DriveStatus_SpdRefID1 | |
| - PowerFlex_Drive:I.DriveStatus_SpdRefID2 | |
| - PowerFlex_Drive:I.DriveStatus_SpdRefID3 | |
| + PowerFlex_Drive:I.OutputFreq | |
| - PowerFlex_Drive:O | |
| + PowerFlex_Drive:O.DriveLogicRslt | |
| - PowerFlex_Drive:O.DriveLogicRslt_Stop | |
| - PowerFlex_Drive:O.DriveLogicRslt_Start | |

Redes

Consulte estas secciones para obtener información acerca de las redes.

Redes en S7

Red Profibus DP, DPV1, DPV3

En el mundo del S7, el principal tipo de red para comunicación con dispositivos es la red Profibus DP en una variedad de implementaciones. Algunos controladores de la gama superior S7-300 y todos los controladores S7-400 tienen puertos maestros Profibus incorporados.

Red Profibus - Otros

Los sistemas Profibus FMS y FDL son para comunicación de datos entre controladores. Estos cumplen una función similar a la red Ethernet industrial, y la configuración es casi idéntica. Las diferencias son que se requieren procesadores de comunicación Profibus en lugar de la red Ethernet, y que se usará el cableado Profibus.

Puede usarse Profibus DPv2 para conectar servovariadores en los controladores S7-315T y S7-317T para control de movimiento de la gama inferior.

Red Ethernet industrial

La red Ethernet industrial Siemens es la variedad Siemens de la red Ethernet en un entorno industrial. Se usa principalmente para comunicación entre controladores, y para comunicación entre el controlador y la computadora de programación.

Aparte de algunos recientes controladores equipados para Profinet, los controladores S7 no tienen puertos Ethernet incorporados. Un sistema S7 que usa Ethernet industrial tendrá procesadores de comunicación montados en los racks.

Dependiendo del procesador de comunicación, pueden usarse los siguientes protocolos:

- S7 (protocolo de propiedad exclusiva para comunicación entre controladores S7)
- Sockets generales TCP (Transmission Control Protocol)
- ISO por TCP (TCP extendido con verificación adicional)
- Sockets generales UDP (User Datagram Protocol)

Se requiere código de aplicación para administrar la mayoría de los aspectos de comunicación en estas redes.

En el entorno de Rockwell Automation, esta funcionalidad puede implementarse mediante puertos EtherNet/IP integrados, módulos de puente EtherNet/IP y/o módulos EWEB.

Profinet

Profinet proporciona una funcionalidad similar Profibus DP en una red Ethernet industrial con los mismos requisitos de tiempo de procesamiento interno de programación. Una red que usa Profinet es similar a una Profibus, excepto que usa cables y conectores diferentes, y módulos de interface de campo Ethernet en lugar de Profibus. Para conexión a la red se usan controladores con interface Profinet incorporada o un procesador de comunicación equipado para Profinet.

Como alternativa, puede conectarse en puente una red Profibus DP existente a Profinet, ya sea con un proxy o usando el puerto Profibus DP de un controlador equipado con Profinet.

Algunos módulos de interface de campo Profinet tienen múltiples puertos RJ45 con un conmutador integrado para permitir una topología de bus de línea tipo Profibus, si fuera necesario.

Profinet proporciona estas tres posibilidades de comunicación:

- Profinet CBA (Component Based Automation), que se usa principalmente para comunicación de controlador a controlador y utiliza hardware Ethernet estándar y la pila de software TCP/IP.
- Profinet IO para transferencias secuenciadas, tales como variadores o módulos de E/S, y utiliza hardware Ethernet estándar, pero se salta la pila de software TCP/IP.
- Profinet IRT (Isochronous Real Time) para aplicaciones de control de movimiento, que utiliza hardware Profinet específico y también se salta la pila de software TCP/IP y debe existir en un segmento de red protegido.

Si se usa el entorno Profinet CBA, las redes Profibus, Profinet y Ethernet industrial pueden integrarse mediante configuración gráfica, con menor necesidad de programación adicional. Las redes EtherNet/IP de Rockwell Automation proporcionan esta funcionalidad mediante hardware estándar y la pila de software TCP/IP estándar, utilizando funciones incorporadas como la instrucción Message (MSG) y tags producidos/consumidos.

Redes en Logix

NetLinx es el término que identifica la solución de Rockwell Automation en el área de tecnologías de conexión en red. Las siguientes son las principales redes usadas en los sistemas Logix:

- EtherNet/IP
- ControlNet
- DeviceNet

Estas redes tienen varias funciones dignas de mención. Todas están diseñadas bajo el protocolo industrial común (CIP) que le permite controlar, configurar y recolectar datos mediante cualquiera de las redes NetLinx. Como resultado, los datos pueden fluir entre diferentes redes sin necesidad de software de traducción de protocolo o proxies.

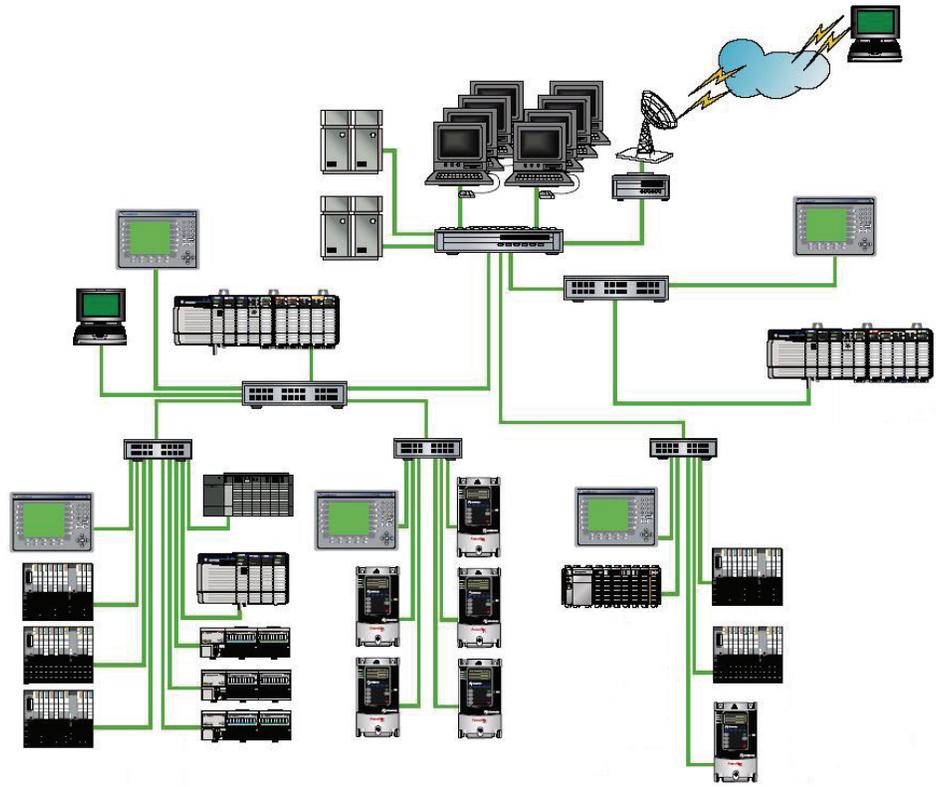
Un ingeniero familiarizando con los sistemas Logix puede quedar impresionado por la naturaleza integrada y la elegancia en la configuración de las redes Logix.

Red EtherNet/IP

La red EtherNet/IP ofrece un conjunto completo de servicios de control, configuración y recolección de datos. Utiliza TCP/IP para transmisión general de mensajes e intercambio de información, y UDP/IP para transmisión de mensajes de E/S. A menudo se usa en estos tipos de configuraciones:

- Control general de E/S
- Intercambio de datos entre controladores
- Conexión de muchas computadoras
- Conexión de muchos dispositivos
- Conectividad con sistemas empresariales
- Integración de dispositivos de seguridad
- Control de movimiento (futuro)

Ejemplo típico de Ethernet/IP

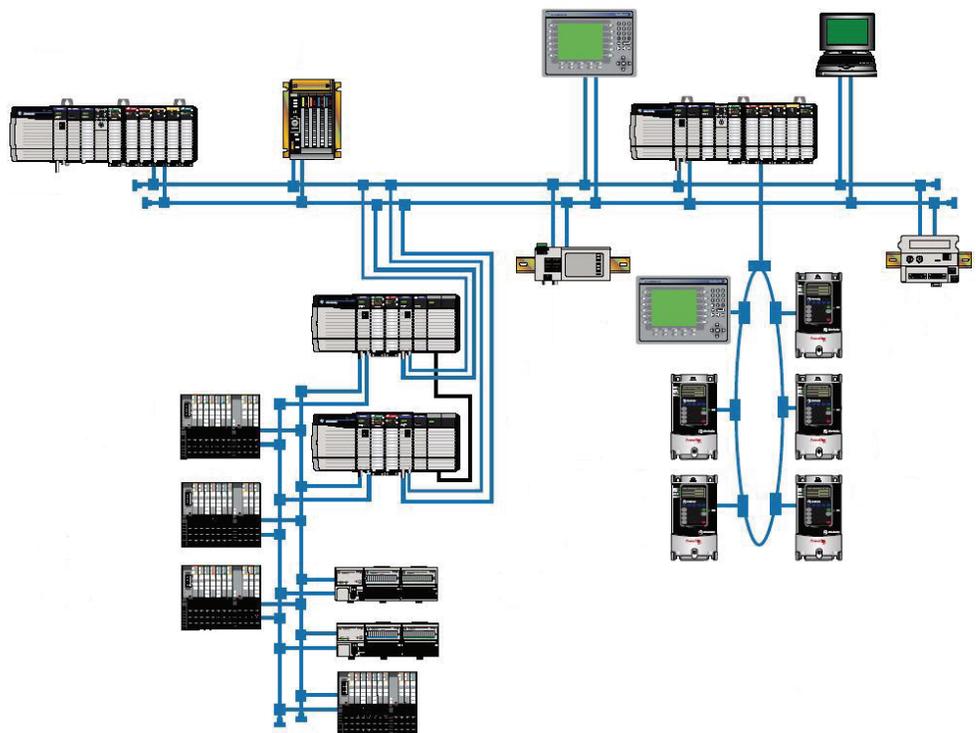


Red ControlNet

ControlNet es una red de control en tiempo real que proporciona transporte de datos de enclavamiento y E/S críticas en cuanto a tiempo, así como datos de transmisión de mensajes, incluida la carga y descarga de datos de programación y configuración en un solo vínculo de medio físico. A menudo se usa en estos tipos de configuraciones:

- Control general de E/S
- Intercambio de datos entre controladores
- Elemento principal a múltiples redes DeviceNet distribuidas

Ejemplo típico de ControlNet

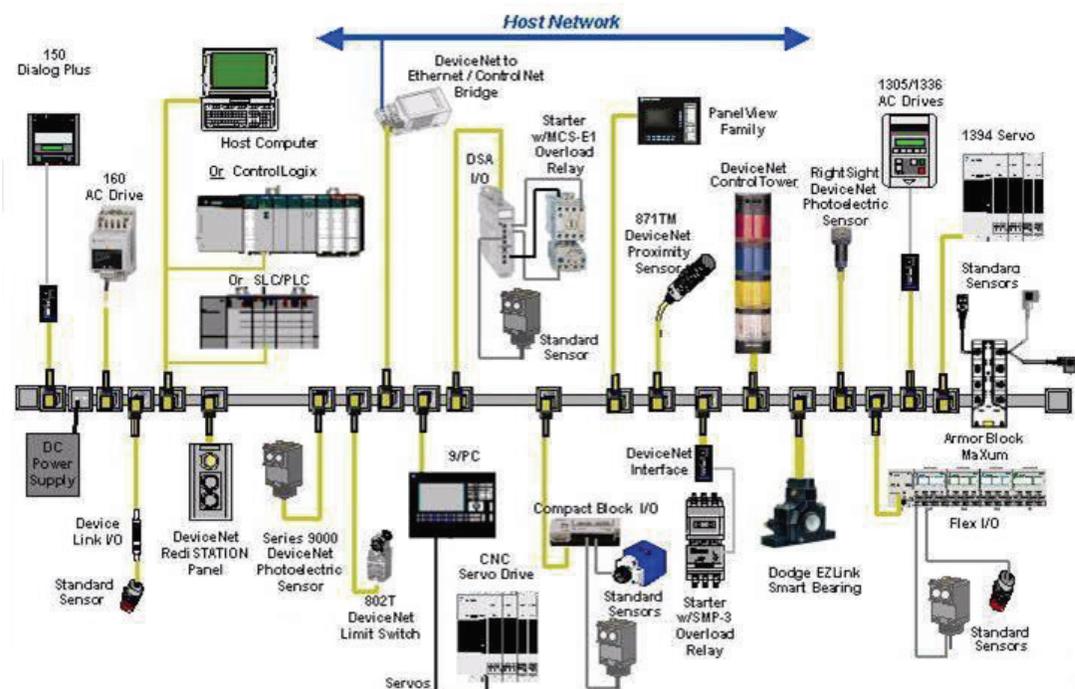


Red DeviceNet:

La red DeviceNet es una solución para conexión de dispositivos industriales de la gama inferior. Fue diseñada para dispositivos con un bajo volumen de datos por dispositivo para operación en tiempo real. A menudo se usa en estos tipos de configuraciones:

- Aplicaciones que contienen dispositivos distribuidos con pocos puntos
- Red de variadores de otros fabricantes y otros dispositivos “simples” de otros fabricantes
- Sistemas en los cuales los dispositivos necesitan conectarse directamente a la red con datos y alimentación eléctrica en la misma conexión
- Cuando se requiere información de diagnósticos avanzados

Ejemplo típico de DeviceNet



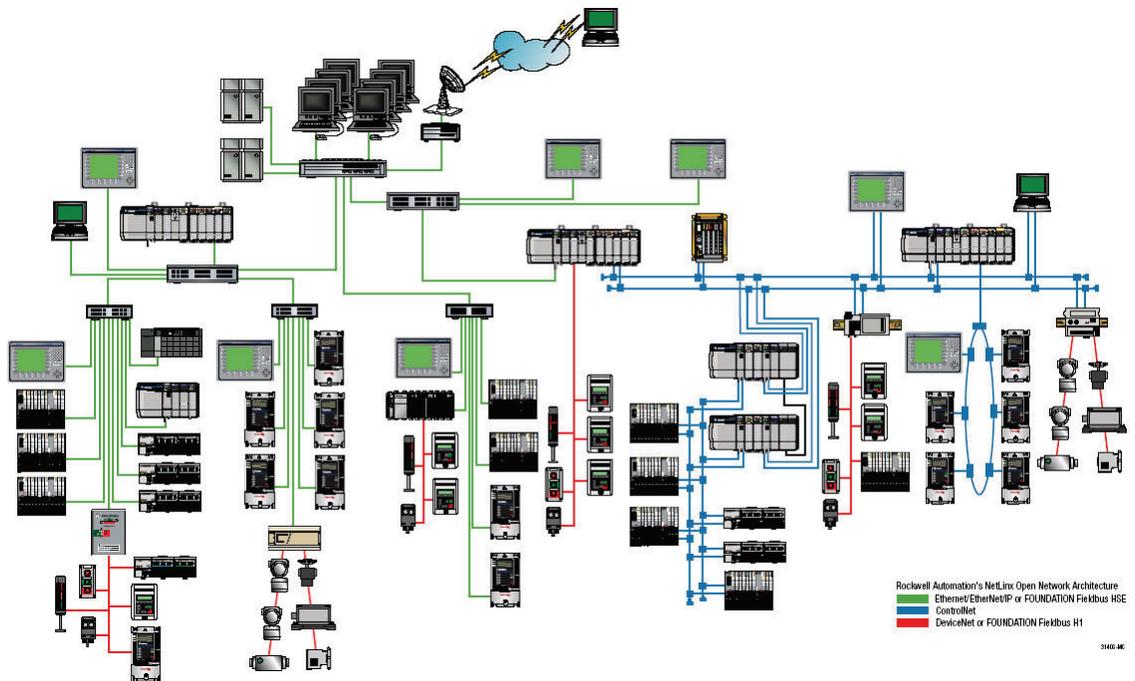
Interconexión de redes NetLinx

Hay dos maneras comunes de interconectar las redes NetLinx.

- Backplane de comunicación, que permite múltiples vínculos de red simultáneamente.
- Dispositivos de vínculo de comunicación, que vincula dos redes de manera transparente.

No se requiere ningún controlador ni programación con ninguno de estos enfoques.

Ejemplo de un sistema de control basado en las redes NetLinx



Conversión de HMI

Consulte el [Apéndice B](#).

Conversión de sistemas que contienen controladores distribuidos

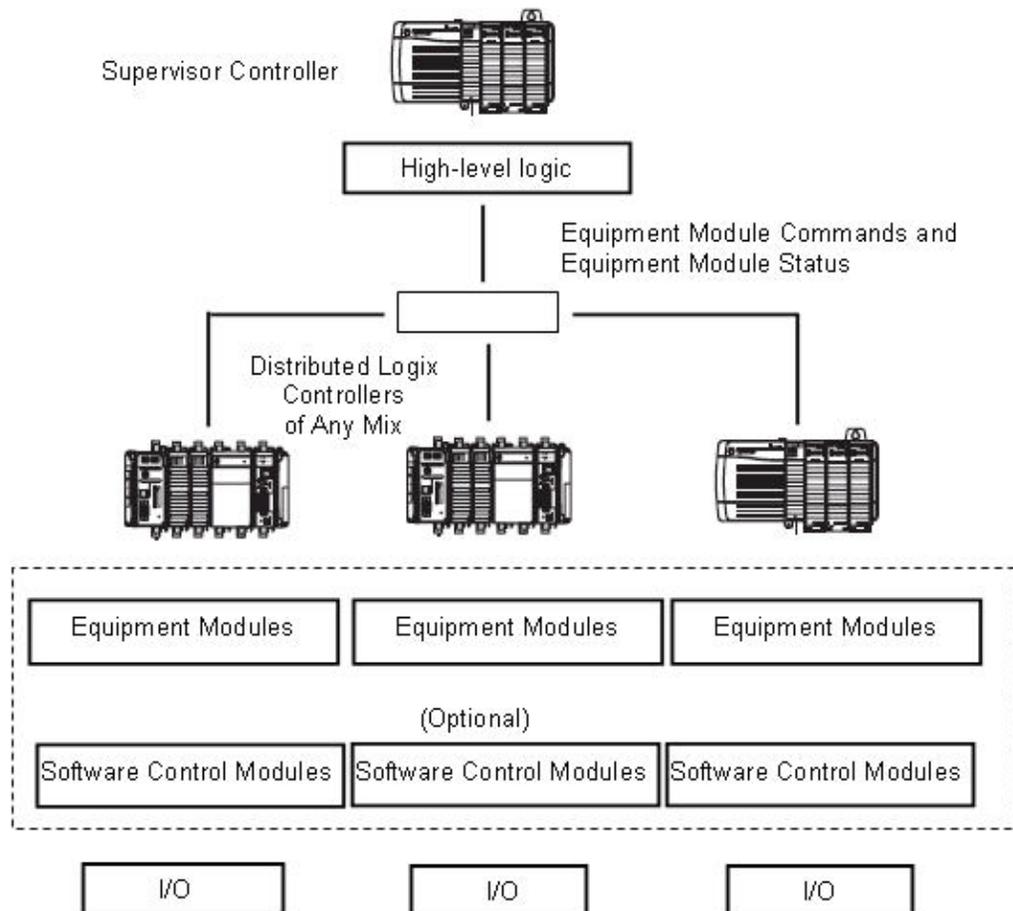
Esta sección abarca lo siguiente:

- cómo diseñar una aplicación de control discreto general que contiene un grupo de unidades funcionales usando varios controladores.
- cómo aplicar un método similar a una aplicación de control diseñada según el estándar S88.

Implementación de hardware y software

Control discreto general

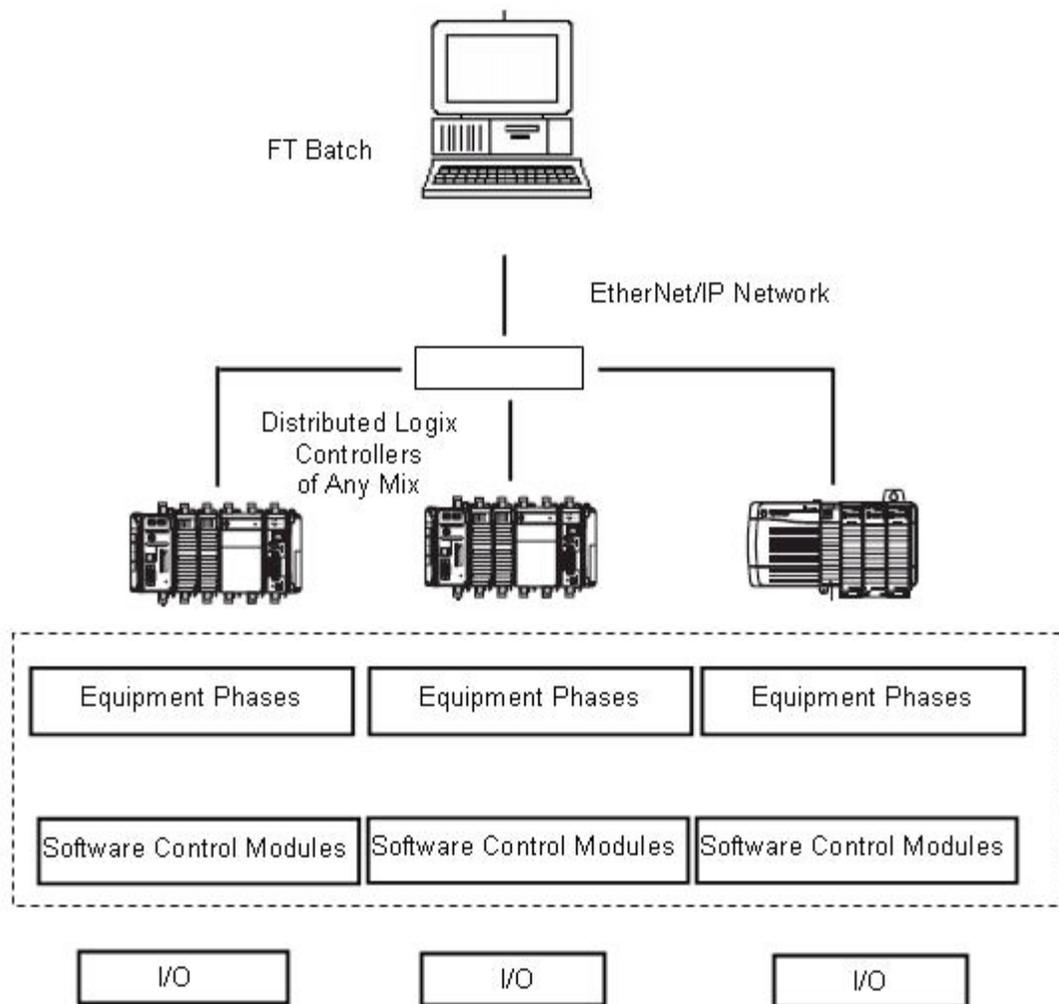
A continuación se presenta el modelo de hardware y software para lógica distribuida para control discreto general. En este caso un controlador tendrá la función supervisoria. Puede usarse la red EtherNet/IP o ControlNet para interconectar los controladores. Puede usarse transmisión de mensajes producidos/consumidos o explícitos para intercambiar datos dentro del sistema.



Control de procesos

El diagrama siguiente ilustra la estructura de hardware y software para una aplicación de control de procesos S88. La PC ejecutará el software FactoryTalk Batch, un paquete de software que permite la producción de lotes de producción mediante recetas. El software FactoryTalk Batch reside en una PC y se comunica con cada controlador mediante la red EtherNet/IP.

Las fases del equipo se configuran bajo PhaseManager, como se describe posteriormente en el [Capítulo 2](#). Éstas ejecutan la lógica de fases y se comunican con las E/S del sistema de control mediante módulos de control.



Conexión de dispositivos Siemens y Rockwell Automation

Hay situaciones en las que usted necesita interconectar equipos Siemens y Rockwell Automation. Recomendamos el uso de productos de las empresas asociadas que agrupa el programa Encompass.

Controladores

Los controladores Logix pueden conectarse a las redes S7 mediante:

- módulos en el rack
- gateways de comunicación autónomos.

Dispositivos distribuidos

Parte de la gama de sistemas de E/S de Rockwell Automation, variadores PowerFlex y terminales HMI se conectan a Profibus mediante adaptadores de comunicación, interfaces incorporadas o módulos de interface.

Características de Logix con las que los usuarios del S7 pueden no estar familiarizados

Introducción

Este capítulo describe las funciones de Logix con las que los usuarios del S7 pueden no estar familiarizados.

| Tema | Página |
|--|--------|
| Bloques de organización de S7 comparados con las tareas de Logix | 36 |
| Tags en vez de direcciones | 47 |
| E/S y tags de alias | 51 |
| Lenguajes de programación | 53 |
| Instrucciones Add-On | 57 |
| El protocolo industrial común (CIP) | 58 |
| Intercambio de datos entre controladores | 60 |
| Tipos de datos definidos por el usuario | 61 |
| Actualización de E/S asíncronas | 62 |
| El tipo de datos DINT | 62 |
| PhaseManager | 63 |
| Hora coordinada del sistema (CST) | 65 |
| Entradas de sello de hora | 65 |
| Salidas secuenciadas | 65 |
| Sin variables temporales | 66 |
| No se necesitan acumuladores ni registros especiales | 66 |

Algunas funciones del sistema Logix son más fáciles de usar y mantener que el sistema S7. Por ejemplo, los datos están organizados en bases de datos de tags sin direcciones absolutas, mientras que los ítems de datos del S7 tienen direcciones absolutas que el programador selecciona en áreas definidas de la memoria.

En otros aspectos, la estructura de Logix es muy similar a la del S7, pero se presenta de manera diferente. Por ejemplo, debajo de la superficie, la estructura de tareas es similar a los bloques de organización del S7.

En este capítulo se contrastan las características diferentes (como los tags) y se comparan las funciones que tienen similitudes subyacentes (como las tareas).

El objetivo es:

- proporcionar al usuario del S7 que está convirtiendo a Logix la información necesaria para facilitar y acelerar el proceso de diseño;
- mostrar lo que Logix puede hacer de modo que los ingenieros no traten de recrear lo que ya existe dentro del firmware del controlador.

Bloques de organización de S7 comparados con las tareas de Logix

Esta comparación de bloques de organización y tareas introducirá la estructura de un programa Logix al usuario del S7.

Los bloques de organización y las tareas son similares en el sentido de que ambos son llamados por el sistema operativo del controlador y no por el programa del usuario. En STEP 7 (y en Logix), hay tres tipos de bloques de organización (tareas en Logix).

- OB de ciclo del programa (tarea continua de organización en Logix) donde el OB (bloque de organización) recomienza desde el principio una vez que ha terminado.
- OB de interrupción cíclica (tarea periódica en Logix) donde el OB se ejecuta periódicamente a intervalos de tiempo preconfigurados.
- OB de interrupción de hardware (tarea de evento en Logix), se ejecutan en respuesta a algún estímulo de hardware.

Muchos programadores de STEP 7 no usan los OB de interrupción cíclica.

Logix proporciona un sistema operativo multitareas configurable que permite asignar la capacidad de la CPU según lo requerido por la aplicación.

Bloques de organización en S7

El tipo de OB se define por su número: unos se ejecutan continuamente (OB1 solamente), mientras que otros se ejecutan periódicamente (OB30 – OB38), o bien pueden ejecutarse ante un evento (OB40 – OB47) o pueden ejecutarse cuando se producen ciertos fallos. Con Logix, las tareas no están numeradas, pero se identifican mediante un nombre definido por el usuario.

Si se requiere, puede asignarse un nombre significativo a un OB de STEP 7.

Ciclo de programa de OB1

OB1 se ejecuta en ciclos continuamente. Cuando ha terminado la ejecución, los valores de la tabla de imagen de salida se envían a las salidas, la tabla de imagen de entrada se actualiza a partir de las salidas y OB1 se inicia nuevamente.

Un programa STEP 7 no tiene que incluir OB1 pero, si está incluido, será continuo.

Fragmento de OB1 típico:

Network 3 : Title:

```
callup valve and motor control module
```

```
CALL "ValveMotor_Calls"
```

Network 4 : Title:

```
callup switch control module
```

```
CALL "Switch_Calls"
```

Network 5 : Title:

```
callup flow totalisers control module
```

```
CALL "Totaliser_calls"
```

Network 6 : Title:

```
callup analogue input control module
```

```
CALL "AnalogueIn_calls"
```

OB1 es la raíz de la jerarquía de llamadas para todo el código ejecutado continuamente.

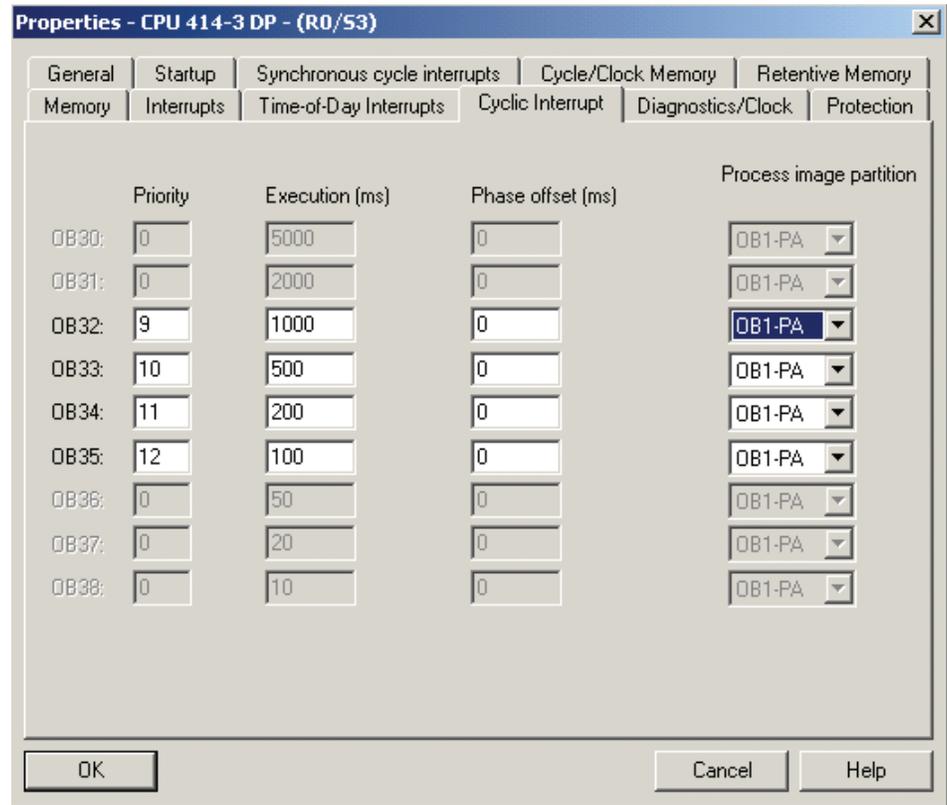
OB1 se parece a la (sólo una es posible, por supuesto) **tarea continua** en Logix. En la terminología S7, OB1 se describe como "ciclo del programa".

Para los lectores que están más familiarizados con Logix que con STEP 7, puede ser útil saber que en la lógica de escalera de STEP 7, una red es igual que un renglón Logix. En la lista de instrucciones STEP 7, las redes están allí, pero sólo sirven para mejorar la apariencia del código. Fragmentan el código en secciones y permiten la adición de comentarios. Todo el código puede colocarse en una red si se desea; se compilará y se ejecutará perfectamente bien.

Interrupciones cíclicas de OB30 – OB38

Estos OB se ejecutan a intervalos fijos y configurables. Su prioridad también puede configurarse. Los OB de mayor prioridad interrumpirán cualquiera de menor prioridad que se estén ejecutando.

Cómo se configuran los OB llamados periódicamente



El número de OB periódicos disponibles depende del tipo de controlador. Un número de baja prioridad representa una prioridad de interrupción superior (la selección de prioridad sólo está disponible con los controladores S7 400). La ejecución (ms) es el período de ejecución del OB. El offset de fase permite escalonar el disparo de interrupciones periódicas relacionadas entre sí. La selección de partición de imagen del proceso permite dividir la tabla de imagen de E/S y que dicha partición se actualice sólo cuando ocurre la interrupción (esta función está disponible con los controladores S7 400 solamente). La opción predeterminada es la tabla completa. En Logix, vea la selección de actualización de E/S de tarea y los comandos IOT.

El contenido de un OB de interrupción periódico se parece al contenido de OB1. Consistirá en llamadas a funciones y bloques de funciones que se ejecutarán según la periodicidad del OB.

Estos OB se parecen a las **tareas periódica** en Logix. En la terminología del S7, OB30 – OB38 se llaman OB de interrupción cíclica.

OB de interrupción de hardware OB40 – OB47

Estos OB pueden configurarse para dispararse ante un evento de entrada. Su prioridad también puede configurarse.

Son **tareas de evento** en Logix. En la terminología del S7, OB40 – OB47 se llaman interrupciones de hardware.

Por ejemplo, el evento de hardware más simple que podría ser manejado por un OB de interrupción de hardware (o tarea de evento) es el cambio de estado de una entrada digital. Una interrupción de hardware (o tarea de evento) garantizaría una respuesta muy rápida al cambio.

Las tareas de evento son más flexibles que los OB de interrupción de hardware, con disparos que provienen no sólo de las E/S, sino también de eventos de red, instrucciones de programa y eventos de movimiento.

Estructura de programa en STEP 7

Un programa típico incluye bloques de organización (OB), bloques de funciones (FB), funciones (FC) y bloques de datos (DB). Generalmente estarán presentes bloques de funciones del sistema (SFB) y funciones del sistema (SFC).

- A partir de los bloques de organización (ciclo del programa o interrupción cíclica, o ambos), se hacen llamadas a bloques de funciones y a funciones.
- Un bloque de funciones contiene código y está asociado con un bloque de datos que contiene los datos estáticos que el FB requiere. Además de los datos estáticos, el FB tiene datos temporales. Los FB se usan cuando la lógica debe conservar los valores entre ejecuciones.
- Una función contiene código pero no tiene datos estáticos. Tiene datos temporales. Las FC se usan cuando la lógica se completa en una sola ejecución; no hay necesidad de conservar los valores.
- Los bloques de datos son áreas de almacenamiento de datos estáticos. Éstos se describirán en la siguiente sección.
- Los SFB y lo SFC son bloques de funciones del sistema y funciones del sistema. Pueden copiarse de las bibliotecas incluidas con la instalación STEP 7 y colocarse en un proyecto.
- Una vez que esto se ha realizado, pueden llamarse desde cualquier lugar del programa.

En STEP 7 no existe una estructura equivalente al programa/rutina de Logix. El OB será la raíz de la cadena de llamadas a los FB y FC, pero la manera en que esto se realiza depende del programador.

Tareas en Logix

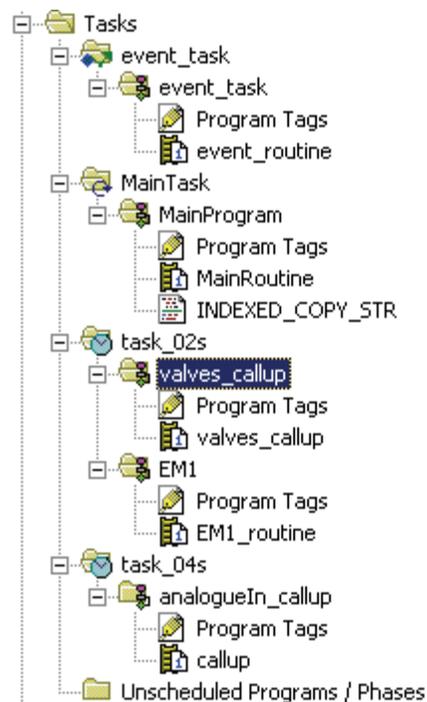
El que llama las tareas es el sistema operativo. Una tarea proporciona la secuenciación así como la prioridad de uno o más programas. Cada programa contiene una sección de datos y una o más rutinas de código.

Las tareas pueden ser periódicas, de eventos o continuas. A cada tarea se le puede asignar una prioridad. La tarea continua, si está presente, siempre tiene la prioridad más baja.

Un proyecto Logix tendrá una tarea, cuyo nombre predeterminado es MainTask. Esta tarea puede ser continua, periódica o de evento. Usted puede cambiar el nombre si lo desea.

Estructura de programas y tareas en Logix

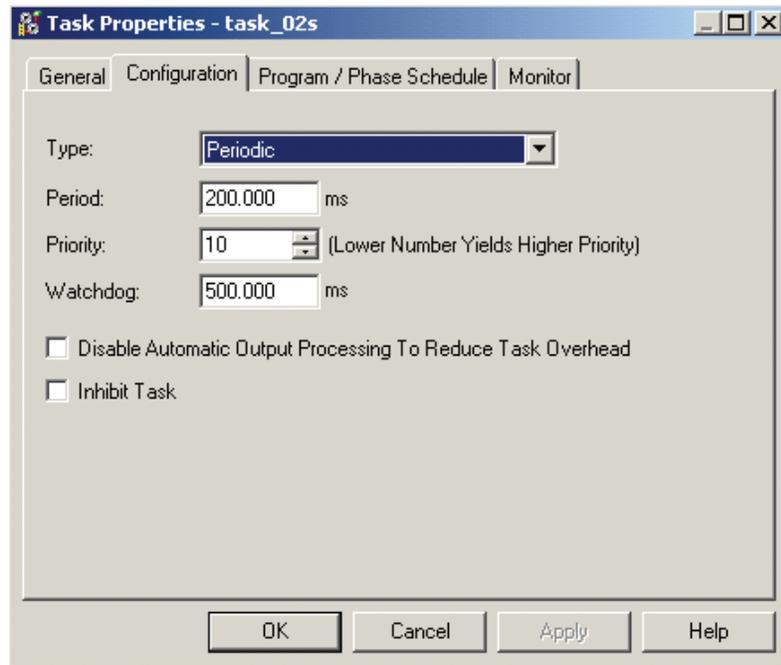
Esta captura de pantalla de un ejemplo de árbol de proyecto RSLogix 5000 ayuda a ilustrar cómo están estructurados los programas y las tareas.



En la captura de pantalla anterior, el icono a la izquierda de “event_task” representa una tarea de evento. El icono a la izquierda de “MainTask” representa una tarea continua y el icono a la izquierda de “task_02s” representa una tarea periódica.

Tareas periódicas

Las tareas periódicas se activarán a un intervalo configurado constante. La configuración del período y la prioridad se muestra a continuación.



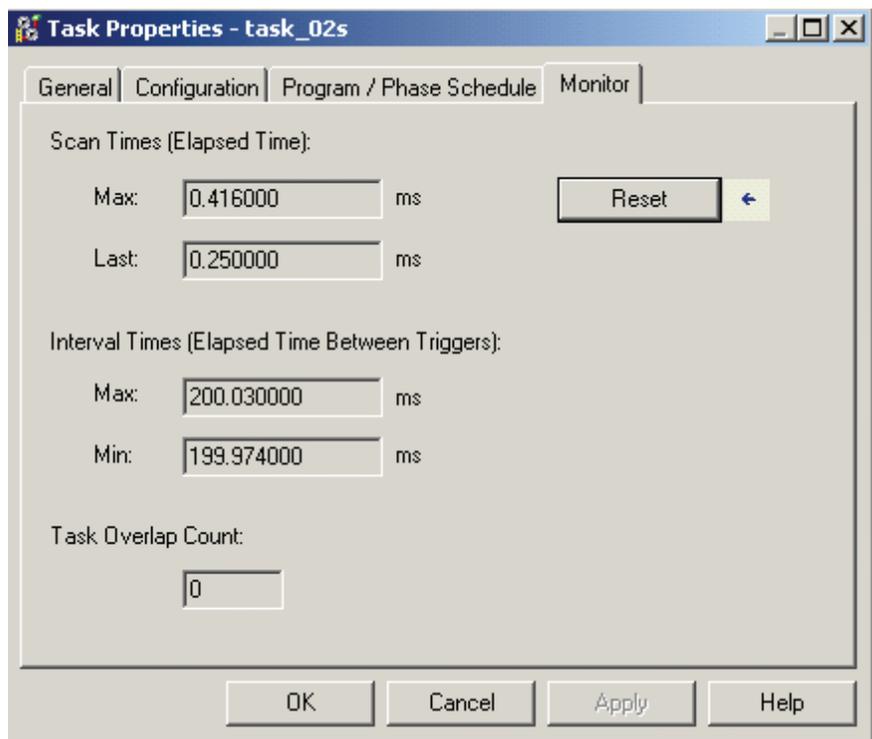
La configuración es similar a la de la página de configuración de OB30 – OB38 descrita en la sección “Interrupciones cíclicas de OB30 – OB38”.

Secuenciación de tareas periódicas

La finalidad del sistema de tareas es:

- Permitir al programador seleccionar las frecuencias apropiadas para la ejecución de los programas. El ejecutar el código con una frecuencia no mayor que la necesaria, la capacidad de las CPU de los controladores se usa con mayor eficiencia en función de las prioridades de la aplicación.
- Usar el sistema de prioridades para permitir que las tareas críticas interrumpen a las de menor prioridad, dándoles así una mayor oportunidad de ejecución a la frecuencia deseada.

Es fácil verificar estos tiempos mediante Task Properties / Monitor.



¿Qué sucederá si ocurre un disparo mientras se está ejecutando una tarea?

- Si el nuevo disparo es para una tarea con una mayor prioridad que la que se está ejecutando, la tarea en ejecución será interrumpida por la nueva y continuará cuando la tarea de mayor prioridad haya concluido.
- Si el nuevo disparo es para una tarea con una menor prioridad que la que se está ejecutando, la tarea en ejecución continuará y la nueva tarea esperará hasta que no se esté ejecutando ninguna tarea de mayor prioridad.
- Si el nuevo disparo es para una tarea con una prioridad igual a la que se está ejecutando, el controlador ejecutará ambas tareas conmutando entre ellas a intervalos de 1 ms.
- Si el nuevo disparo es para la misma tarea que la que se está ejecutando, el nuevo disparo se desechará. Ésta es una condición de **superposición**.

El número de superposiciones que ocurrieron desde que el contador se restableció por última vez se muestra en la ventana Task Properties. Un número diferente de cero indica que el período de interrupción necesita aumentarse.

SUGERENCIA

Debido a la capacidad de procesamiento que se desperdicia durante conmutaciones innecesarias, evite conmutar tareas innecesariamente.

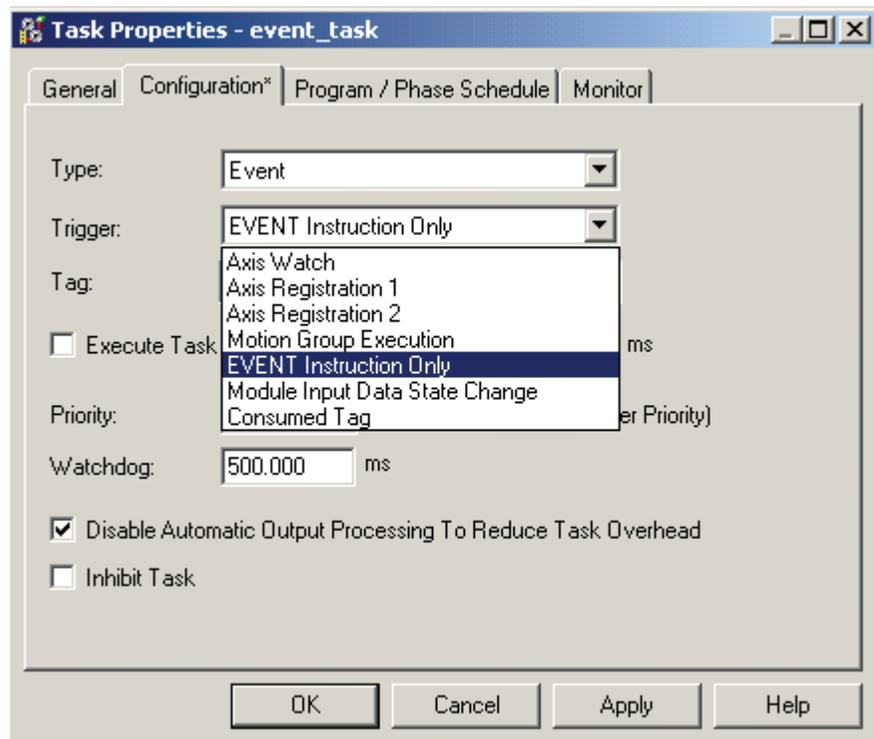
Cuando programe interrupciones periódicas en Logix, observe estas similitudes y diferencias con respecto a STEP 7:

- En STEP 7, las llamadas se realizarán a partir del OB que está configurado para ejecutarse a la frecuencia seleccionada con las funciones y bloques de funciones que desea ejecutar a esta frecuencia. En Logix, usted insertará programas y rutinas en el árbol de proyectos debajo de la tarea.
- Tanto en STEP 7 como en Logix, el código de aplicación en cuestión no será muy diferente del código en una tarea de ejecución continua. Tenga presente que la frecuencia constante y conocida de una tarea periódica da a los programadores la oportunidad de convertir un incremento variable simple en un temporizador.
- En ambos sistemas, usted necesita verificar superposiciones a medida que desarrolla y prueba su código. El tiempo de ejecución del OB o de la tarea debe ser mucho menor que su período de ejecución.
- Es fácil verificar el tiempo de ejecución de las tareas Logix. Utilice la pantalla Task Properties antes mostrada. En STEP 7 necesitará muestrear el reloj del sistema al comienzo y al final del OB, restar los valores y almacenar el resultado en una variable para monitoreo.

- En un controlador S7, la superposición hará que el controlador se detenga, a menos que se añada un OB de fallo que atrape el fallo. Logix es menos estricto y simplemente cuenta el número de superposiciones.
- En STEP 7, es posible escalonar la ejecución de los OB periódicos relacionados entre sí. Esto no está disponible con las tareas Logix.

Tareas de evento

Las tareas de evento se ejecutarán cuando ocurra un evento de disparo configurado. Normalmente se les da mayor prioridad que a las tareas periódicas.



Para configurar una tarea de evento se abre la página Task Properties de la tarea y se selecciona Type Event. Pueden usarse diferentes tipos de tareas de evento en diferentes controladores Logix.

Tarea continua

Un controlador Logix acepta una tarea continua, pero un proyecto no tiene que incluir la tarea continua. Si desea, puede ejecutar todo el programa con tareas periódicas y tareas de evento.

Usted puede especificar en la configuración si la tarea continua actualiza las salidas al final de su ejecución.

Si lo desea puede ajustar el porcentaje del tiempo de la CPU que se dedica a comunicación no programada como porcentaje del tiempo dedicado a la tarea continua.

Monitor de tareas

El software RSLogix 5000 incluye una herramienta llamada Task Monitor que puede ayudar a analizar las tareas secuenciadas y mucho más.

La siguiente captura de pantalla muestra cómo puede verse en una tabla la información sobre las tareas del controlador.

The screenshot shows the 'Logix5000 Task Monitor' window with the 'User Tasks' tab selected. The table below displays the following data:

| Name | Rate | CPU | Priority | Last Scan | Max Scan | Watchdog | Overlap | Stat |
|----------|-------------|-------|----------|-----------|----------|-----------|---------|-------|
| MainTask | * 500,000us | 0.70% | Lowest | 3,524us | 5,852us | 500,000us | 0 | Runr |
| task_02s | 200,000us | 0.16% | 9 | 548us | 894us | 500,000us | 0 | Runr |
| task_04s | 400,000us | 0.04% | 10 | 764us | 1,182us | 500,000us | 0 | Runr |
| task_01s | 100,000us | 0.17% | 5 | 134us | 426us | 500,000us | 0 | Runr |
| event | 10,000us | 0.00% | 10 | 0 | 0 | 500,000us | 0 | Stopp |

Las otras fichas proporcionan un cúmulo de información a nivel del sistema sobre el desempeño del controlador. La herramienta se incluye como estándar en el disco de instalación de RSLogix 5000.

Tags en vez de direcciones

Una de las principales diferencias que un usuario del S7 observará al comenzar a trabajar con Logix es que los datos no tienen direcciones. Los ítems de datos se crean en una base de datos de tags y el software RSLogix 5000 asigna direcciones “en la trastienda”. Así los usuarios no tendrán que entender ni administrar las direcciones de memoria. Esta sección describe la asignación de datos en los dos sistemas.

Áreas de datos en los S7

Áreas de datos en controladores S7

| Área de dirección | Notación en el S7 | Tamaño de unidad |
|--|-------------------|--------------------------|
| Tabla de entrada de imagen del proceso | I | Bit de entrada |
| | IB | Byte de entrada |
| | IW | Palabra de entrada |
| | ID | Doble palabra de entrada |
| Tabla de salida de imagen del proceso | Q | Bit de salida |
| | QB | Byte de salida |
| | QW | Palabra de salida |
| | QD | Doble palabra de salida |
| Memoria de bits | M | Bit de memoria |
| | MB | Byte de memoria |
| | MW | Palabra de memoria |
| | MD | Doble palabra de memoria |
| Temporizadores | T | |
| Contadores | C | |
| Bloque de datos | DBX | Bit de datos |
| | DBB | Byte de datos |
| | DBW | Palabra de datos |

Las secciones siguientes proporcionan más información acerca de las dos áreas más comúnmente usadas en la programación: memoria de bits y bloques de datos.

Memoria de bits

Las ubicaciones de la “memoria de bits” se indican mediante Mx donde, por ejemplo:

- **M5.3** es un bit.
- **MB6** es un byte (BYTE).
- **MW8** es una palabra de 16 bits (WORD).
- **MD10** es una palabra de 32 bits (DWORD).

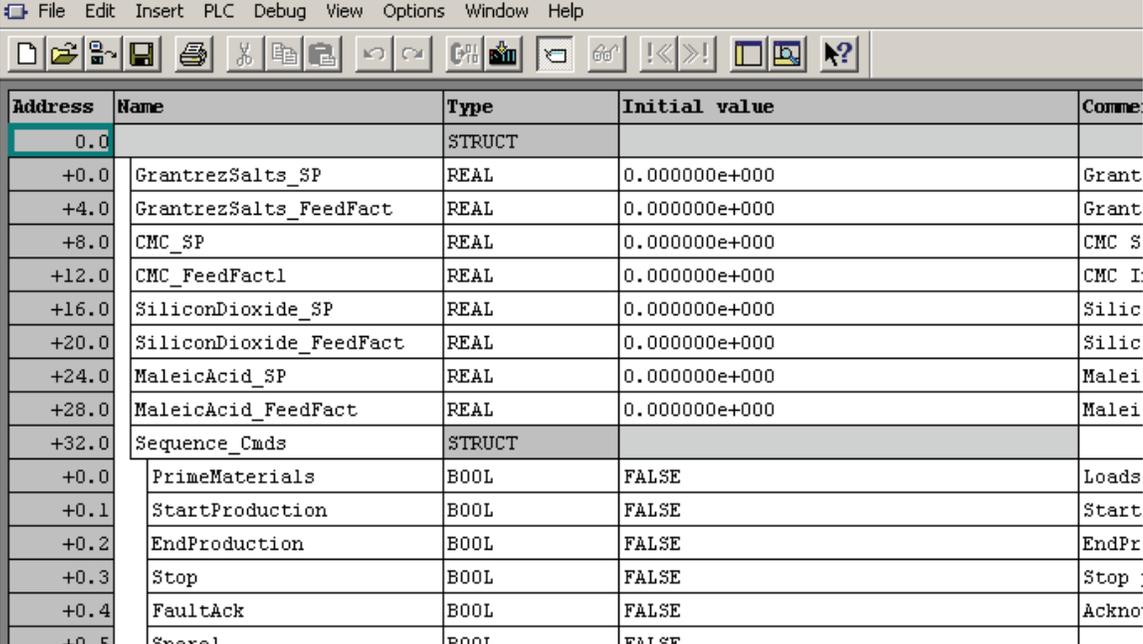
Las ubicaciones de memoria de bits pueden identificarse en la tabla de símbolos (similar a una tabla de símbolos de PLC-5 o SLC), como se muestra en la siguiente captura de pantalla.

| Status | Symbol ▲ | Address | Data type | Comment |
|--------|----------------------|---------|-----------|---|
| | EXT_ZONE2_ON | Q 28.2 | BOOL | EXTRUDER ZONE2 ON |
| | EXT_ZONE3_ON | Q 28.3 | BOOL | EXTRUDER ZONE3 ON |
| | EXT_ZONE4_ON | Q 28.4 | BOOL | EXTRUDER ZONE4 ON |
| | EXT_ZONE5_ON | Q 28.5 | BOOL | EXTRUDER ZONE5 ON |
| | FALSE | M 0.0 | BOOL | |
| | FlowTotaliser | UDT 10 | UDT 10 | |
| | FSL24001 | I 10.0 | BOOL | |
| | FSL24002 | I 41.0 | BOOL | |
| | FSL24003 | I 10.6 | BOOL | |
| | FT24001 | PIW 530 | INT | |
| | FT24002 | PIW 526 | INT | |
| | FT24006 | PIW 570 | INT | |
| | GET_INDEXED_REFE... | FC 111 | FC 111 | |
| | Global_Data | DB 99 | DB 99 | |
| | IFIX_alarms | DB 71 | DB 71 | |
| | INDEXED_COMPARE | FC 102 | FC 102 | |
| | INDEXED_COPY | FC 101 | FC 101 | |
| | Interlocks_Handler | FB 70 | FB 70 | |
| | Interrupt_Execution | OB 35 | OB 35 | |
| | JUNK_BIT | M 0.2 | BOOL | |
| | KTRON_CmdsFromM... | DB 33 | DB 33 | Commands & Setpoints From Manufacturing PLC |
| | KTRON_StatusToMan... | DB 32 | DB 32 | Feeder Status to ManufacturingPLC |
| | LF24001A | Q 17.0 | BOOL | PIG LAUNCH VALVE |
| | LF24002A | Q 18.5 | BOOL | PIG LAUNCHER |
| | LF24003A | Q 10.4 | BOOL | PIG LAUNCHER |

Bloques de datos

Los bloques de datos tienen un estado similar al de los otros bloques –bloques de organización, bloques de funciones y funciones– excepto por el hecho de que contienen datos en lugar de código de programa. La memoria en los bloques de datos es estática; los datos retienen su valor hasta que se cambian.

Ejemplo de un bloque de datos



| Address | Name | Type | Initial value | Comments |
|---------|-------------------------|--------|---------------|----------|
| 0.0 | | STRUCT | | |
| +0.0 | GrantrezSalts_SP | REAL | 0.000000e+000 | Grant |
| +4.0 | GrantrezSalts_FeedFact | REAL | 0.000000e+000 | Grant |
| +8.0 | CMC_SP | REAL | 0.000000e+000 | CMC S |
| +12.0 | CMC_FeedFact1 | REAL | 0.000000e+000 | CMC I |
| +16.0 | SiliconDioxide_SP | REAL | 0.000000e+000 | Silic |
| +20.0 | SiliconDioxide_FeedFact | REAL | 0.000000e+000 | Silic |
| +24.0 | MaleicAcid_SP | REAL | 0.000000e+000 | Malei |
| +28.0 | MaleicAcid_FeedFact | REAL | 0.000000e+000 | Malei |
| +32.0 | Sequence_Cnds | STRUCT | | |
| +0.0 | PrimeMaterials | BOOL | FALSE | Loads |
| +0.1 | StartProduction | BOOL | FALSE | Start |
| +0.2 | EndProduction | BOOL | FALSE | EndPr |
| +0.3 | Stop | BOOL | FALSE | Stop |
| +0.4 | FaultAck | BOOL | FALSE | Ackno |
| +0.5 | Spare1 | BOOL | FALSE | |

Los símbolos de los bloques de datos no aparecen en la tabla de símbolos, pero sí los nombres de los bloques de datos.

Los bloques de datos pueden asignarse para mantener los datos usados por los bloques de funciones. Éstos se llaman bloques de datos de instancia.

Datos en Logix

En el entorno de programación de RSLogix 5000, los datos se configuran en una base de datos de tags. Las direcciones de la memoria permanecen ocultas para el programador, lo cual le facilita las cosas a éste.

Base de datos de tags

| Controller Tags - dev_18Aug(controller) | | | | | | |
|---|------------------|------------------|--------------------------|----------|---------------------|--|
| Scope: dev_18Aug | | Show... | | Show All | | |
| Name | Alias For | Base Tag | Data Type | Style | Description | |
| + analogIn_1 | | | DINT | Decimal | | |
| Blue_Button | Local:3:I.Data.0 | Local:3:I.Data.0 | BOOL | Decimal | | |
| + CompactLogix_1_consume | | | UDT_STEP_SEQUENCE | | Data - step sequ... | |
| + ControlLogix_1_produce | | | UDT_STEP_SEQUENCE | | Data - step sequ... | |
| + Drive:I | | | AB:PowerFlex70EC_Driv... | | | |

Seleccione un tag de un menú desplegable mientras realiza la programación

The screenshot shows a software interface for programming a controller. A menu is open, displaying a list of tags with columns for Name, Data Type, and Description. The tags listed are: -em1Data.step[0], -em1Data.step[1], -em1Data.step[2], -em1Data.step[3], and -em1Data.step[4], all with a Data Type of BOOL and a Description of 'Data - step sequ...'. Below the menu, a ladder logic rung is visible, labeled 'STEP 1 /Jump to step 3'. The rung contains a coil labeled 'end_seq' with a 'JMP' symbol. A dropdown menu is open over the rung, showing the selected tag 'em1Data.step[0]'. The rung is annotated with 'Data - step sequence step number' and 'Data - step sequence EM sequencer step'.

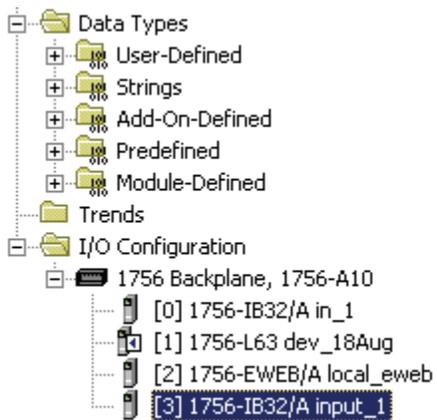
En Logix, hay una **base de datos bajo el control del controlador** y **bases de datos bajo el control del programa** asociadas con cada programa.

- Los tags en la base de datos bajo el control del controlador son globales y puede accederse a ellos mediante rutinas en cualquier parte del programa.
- Sólo puede accederse a los tags bajo el control del programa mediante rutinas en dicho programa.

E/S y tags de alias

Un tag de alias le permite representar a otro tag, mientras ambos tags comparten el mismo valor. Una de las finalidades de los alias es la de referenciar los tags de E/S como se describe a continuación.

Es posible añadir módulos de E/S a un proyecto añadiendo el módulo al backplane del controlador en la carpeta del proyecto.



En este caso, se ha añadido una tarjeta de entrada de 32 puntos en la ranura 3. El número de ranura está entre corchetes al comienzo de la línea. “1756-IB32/A” es el número de parte de la tarjeta. “input_1” es el nombre de la tarjeta, el cual se configura cuando la tarjeta se añade inicialmente al rack.

Al añadir la tarjeta, el software RSLogix 5000 generará automáticamente los tags de perfil de dispositivo relevantes y lo incorporará a la base de datos de tags bajo el control del controlador. Ellos son los tags Local:3:I de entrada y Local:3:C de configuración siguientes.

| Controller Tags - dev_18Aug(controller) | | | | | |
|---|-----------|--|----------------|---|--|
| Scope: <input type="text" value="dev_18Aug"/> | | <input type="button" value="Show..."/> | | <input type="button" value="Show All"/> | |
| Name | Alias For | Base Tag | Data Type | Style | |
| [-] Local:3:I | | | AB:1756_DI:I:0 | | |
| [+] Local:3:I.Fault | | | DINT | Binary | |
| [-] Local:3:I.Data | | | DINT | Binary | |
| [-] Local:3:I.Data.0 | | | BOOL | Decimal | |
| [-] Local:3:I.Data.1 | | | BOOL | Decimal | |
| [-] Local:3:I.Data.2 | | | BOOL | Decimal | |

Usted puede crear un nuevo tag de alias con un nombre más descriptivo. Por ejemplo, puede crearse un alias para la primera entrada llamado Limit_Switch_1, el cual describe físicamente esta entrada.



| Name | Alias For | Base Tag | Data Type | Style |
|----------------|------------------|------------------|-----------|---------|
| Limit_Switch_1 | Local:3:I.Data.0 | Local:3:I.Data.0 | BOOL | Decimal |

En STEP 7, la herramienta de configuración de hardware asignará direcciones a una tarjeta de E/S cuando ésta se añada al sistema. Por ejemplo, pueden asignarse los bytes I16 y I17 a una tarjeta de entradas digitales. De esta forma, el programador identificará la dirección de bits de cada entrada e introducirá un nombre para cada una de ellas en la tabla de símbolos. Una vez hecho esto, el programa hará automáticamente la asociación I16.5 = "ZSC2036".

Lenguajes de programación

Esta sección describe los lenguajes de programación disponibles con los softwares STEP 7 y RSLogix 5000. Todos los lenguajes no son estándar; esto depende de la versión de software comprada. La selección del lenguaje Logix más adecuada para la tarea resultara en un diseño más fácil del programa, una codificación más rápida y un programa más fácil de entender.

Existe una diferencia importante entre los lenguajes S7 y Logix. En S7, la lista de instrucciones es el lenguaje “nativo” del controlador. Otros lenguajes se traducen a STL. En Logix, todos los lenguajes son lenguajes “nativos” en el controlador y cada uno se compila sin referencia a ninguno de los otros. La ventaja de ello es que cuando usted carga un programa desde el controlador, lo ve en el lenguaje en el cual se escribió.

STEP 7 tiene tres lenguajes estándar:

- Lista de instrucciones (STL) – podría describirse como un ensamblador de alto nivel.
- Lógica de escalera (LAD)
- Diagrama de bloques de funciones (FBD)

Y algunos lenguajes opcionales:

- Texto estructurado (ST)
- CFC – diagrama de flujo continuo para aplicaciones de tipo proceso
- HiGraph – control secuencial mediante el software Grahing
- ML – lenguaje de control de movimiento – similar a GML en el antiguo controlador de movimiento dedicado 1394 de Rockwell Automation

Un programa puede consistir en bloques de funciones y funciones escritas en diferentes lenguajes.

El software RSLogix 5000 tiene cuatro lenguajes de programación:

- Diagrama de lógica de escalera (LD) – comparable con el LD de Siemens, con un conjunto de instrucciones expandido.
- Texto estructurado (ST) – Equivalente al ST de Siemens
- Diagrama de bloques de funciones (FBD) – Equivalente al CFC de Siemens
- Diagrama de función secuencial (SFC) – Comparable al hiGraph de Siemens.

Una rutina –la sección básica de código en Logix– puede estar en cualquiera de éstos, y un programa puede componerse de rutinas escritas en diferentes lenguajes. La siguiente captura de pantalla proporciona un ejemplo.

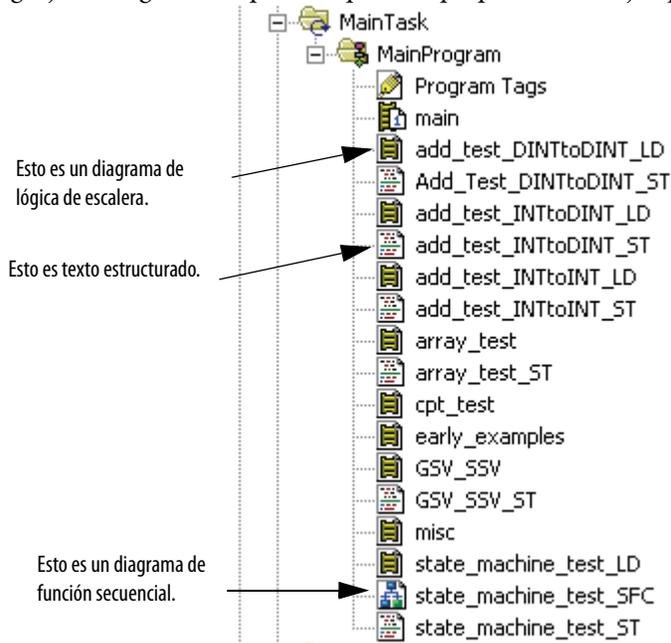


Diagrama de lógica de escalera Logix

Generalmente, el diagrama de lógica de escalera se usa para implementar lógica combinatoria booleana. En Logix, también puede usarse para lógica secuencial, control de movimiento, manejo de datos y cálculos matemáticos, si bien otros lenguajes pueden ser más adecuados para estas tareas.

Texto estructurado Logix

El texto estructurado es un lenguaje procedimental de alto nivel, fácil de aprender por cualquier persona con experiencia en Basic, Pascal o uno de los lenguajes de la familia 'C'. Se usa principalmente para manipulación de datos y cálculos matemáticos, aunque la lógica de control de movimiento, combinatoria y secuencial también puede programarse fácilmente en ST.

Diagrama de bloques de funciones Logix

El diagrama de bloque de funciones describe gráficamente una función (booleana o matemática) que relaciona variables de entrada y variables de salida. Las variables de entrada y salida se conectan a los bloques mediante líneas de conexión. Una salida de un bloque también puede conectarse a una entrada de otro bloque.

Es una buena práctica programar lazos PID en FBD. Éste es el lenguaje más apropiado para control de procesos.

Diagrama de función secuencial Logix

SFC es una herramienta gráfica para describir la lógica secuencial como un conjunto de estados y transiciones. Las salidas pueden asignarse a un estado, y las condiciones booleanas para transiciones a otros estados definidos.

Conversión del código STEP 7 a Logix

- Si tiene lógica de escalera STEP 7 que desea convertir a Logix, LD debe ser su primera opción. El significado de LD es similar en ambos sistemas.
- Si tiene código de diagrama de bloques de funciones STEP 7 que desea convertir a Logix, FBD debe ser su primera opción.
- Tenga presente que el FBD Logix estándar es más avanzado que el FBD STEP 7 y es equivalente al lenguaje CFC STEP 7 opcional.
- Si usted tiene código de lista de instrucciones STEP 7 que desea convertir a Logix, el lenguaje más apropiado dependerá de la naturaleza del bloque STL. Si el bloque STL contiene principalmente evaluaciones booleanas, LD probablemente sería el mejor lenguaje Logix al cual convertir. Si el bloque STL contiene punteros para acceso y manipulación de datos, o ejecuta cálculos matemáticos, probablemente ST sería el mejor lenguaje Logix al cual convertir. Si el bloque STL contiene lógica secuencial, debería considerarse SFC, aunque la lógica secuencial también puede implementarse fácilmente en ST y LD.

Matrices en vez de punteros

En STEP 7, las matrices pueden definirse exactamente como en Pascal o en C, pero los lenguajes básicos (STL, LD y FBD) no tienen soporte de alto nivel para acceder a ellos. En lugar de ello, deben construirse rutinas de punteros.

Las funciones de biblioteca STEP 7 no tienen soporte para acceso a matrices. Los programadores que se sienten cómodos con los punteros pueden escribir sus propias funciones, tales como FC101 "INDEXED_COPY" (ver abajo) pero ello requiere habilidad y tiempo.

"INDEXED_COPY" en STEP 7 hace lo mismo que la instrucción COP de Logix para copia indexada.

```
CALL "INDEXED_COPY"          FC101
  indexSrc:=#index_in
  source  := "Instance_FB2".table  P#DB4.DBX0.0
  indexDst:=1
  dest    := "Instance_FB2".target  P#DB4.DBX96.0
  len     :=8
```

FC111, presentada a continuación, accederá a una matriz.

```
CALL "GET_INDEXED_REFERENCE"  FC111
  refArray := "Instance_FB2".table  P#DB4.DBX0.0
  index    := #index_in
  byteIncr :=32
  startIndex:=TRUE
  retVal   := #ptr
```

El puntero al objeto se retorna en el parámetro #ptr, el cual a continuación se puede desreferenciar para obtener los datos.

En Logix, se puede definir y acceder a las matrices de la manera usual que se hace con un lenguaje de computadora de alto nivel, como lo ilustra el fragmento a continuación.

```
// copy a string from a table of strings #table
// to a target string #target. The index is #index_in

COP(table[index_in], target, target.LEN);
```

Instrucciones Add-On

Resumen de la instrucción Add-On

Las instrucciones Add-on son el equivalente de los bloques de funciones de STEP 7, con opciones de datos privados y parámetros avanzados. En particular, el tipo de parámetro INOUT o “paso por referencia” hace posible pasar de manera eficiente las estructuras de datos al código.

Como la instrucción Add-On es tan similar al bloque de funciones de STEP 7, es probable que el programador de S7 que está realizando la conversión a Logix la use con frecuencia.

Comparación entre las instrucciones Add-On y los FB:

- Ambas se pueden llamar como funciones con nombre desde cualquier lugar del programa.
- Ambas contienen un área de datos privada de datos estáticos, aunque no son realmente privados en el caso de STEP 7.
- Un bloque de funciones STEP 7 también tiene un área de datos temporales.
- En la instrucción Add-On, los datos estáticos locales harán lo mismo.

Ambas tienen tres tipos de parámetros: entrada (paso por valor) salida (paso por valor) y entrada-salida (paso por referencia). El parámetro de paso por referencia es una gran ventaja puesto que permite que se pasen estructuras de datos grandes de manera eficiente.

La instrucción Add-On mantendrá automáticamente un historial de cambios al registrar el sello de hora y el nombre de usuario de Windows al momento del cambio. Esta función no está disponible con los bloques de funciones STEP 7.

Con la instrucción Add-On puede configurarse una rutina previa al escán para que se ejecute cuando el controlador cambia del modo de programación al modo de marcha, o se activa en el modo de marcha. Bajo estas condiciones, la rutina previa al escán se ejecutará una vez y normalmente puede usarse para inicializar los datos. En STEP 7, el bloque de organización OB100 hace lo mismo, pero el código de la rutina previa al escán no puede conectarse específicamente a un FB.

Si se llama a la instrucción Add-On desde un paso SFC y el SFC se configura para restablecimiento automático, una rutina posterior al escán definida en la instrucción Add-On se ejecutará una vez cuando el SFC salga de dicho paso. Podría usarse para el restablecimiento de datos. Un FB STEP 7 no tiene equivalente incorporado (si bien es fácil de programar).

Una instrucción Add-On puede tener una rutina EnableInFalse, la cual se llamará (si está presente) cuando la condición de renglón en la llamada de instrucción Add-On sea falsa. En este caso, los parámetros de entrada y salida pasarán valores. Un FB STEP 7 no tiene equivalente.

Las instrucciones Add-On se describen con mayores detalles en el [Capítulo 4](#).

Tags de respaldo

Muchas instrucciones y tipos de datos utilizan tags de respaldo, tags que se crean específicamente para la instancia de la instrucción o tipos de datos que usted está usando como instancia. Las instrucciones Add-On, y los temporizadores, los contadores, los mensajes y el control PID, utilizan tags de respaldo. El software RSLogix 5000 generará la correspondiente estructura de elementos cada vez que usted cree un tag de dicho tipo; así usted no tendrá que crear los elementos.

| Name | Alias For | Base Tag | Data Type | Style |
|--------------|-----------|----------|-----------|---------|
| -Timer1 | | | TIMER | |
| + Timer1.PRE | | | DINT | Decimal |
| + Timer1.ACC | | | DINT | Decimal |
| - Timer1.EN | | | BOOL | Decimal |
| - Timer1.TT | | | BOOL | Decimal |
| - Timer1.DN | | | BOOL | Decimal |

El protocolo industrial común (CIP)

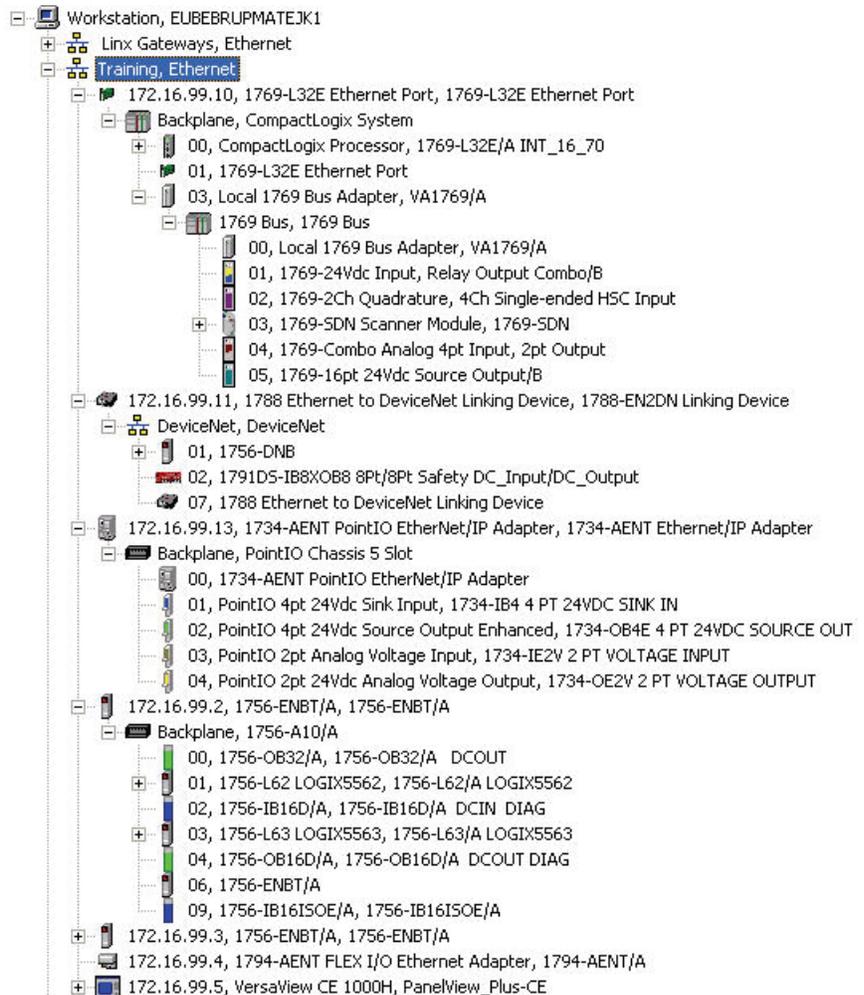
Logix utiliza tres redes principales: Ethernet/IP, ControlNet y DeviceNet. Cada una tiene características apropiadas para áreas diferentes de la aplicación. Los tres tipos de red comparten un protocolo: el ‘protocolo industrial común’.

El CIP hace posible transferir datos mediante cualquiera de tres tipos de redes compatibles con Logix mediante una interface de programación y configuración casi idénticas para las tres. Además, los datos pueden transferirse a través de una red creada a partir de más de uno de los tres tipos de red, sin que el programador necesite traducir protocolos.

En el S7 “tradicional”, los dos protocolos principales son Ethernet industrial para conexión en red a IT y a otros controladores, y Profibus DP para conexión en red a los sistemas de campo. Estos dos protocolos están separados a nivel de hardware y a nivel de datos. Con el más reciente hardware y software S7, “Profinet CBA” se integra a Ethernet industrial, Profinet y Profibus.

Visualización de la red

Los usuarios de S7 pueden sentirse impresionados con la configuración y administración de red de Logix. Como ejemplo, el árbol siguiente muestra los dispositivos conectados al sistema. Este árbol se produjo al entrar en línea; no estaba configurado.



Las redes se describen en mayor detalle en el [Capítulo 1](#).

Intercambio de datos entre controladores

Enviar / recibir en STEP 7

Para preparar la comunicación de controlador a controlador en STEP 7, se realizan los pasos siguientes.

1. Las estaciones remotas se configuran gráficamente en un componente de STEP 7 llamado NetPro.
2. Hay una tabla de conexión incorporada en NetPro que especifica los protocolos y parámetros para cada una de las conexiones.
3. Se copian al proyecto las funciones de biblioteca FC5 AG_SEND y FC6 AG_RECV.
4. Se realizan llamadas desde el programa de usuario a AG_SEND y AG_RECV que especifican parámetros de conexión y las áreas de datos que se están usando para suministrar y recibir datos.

Tags producidos / consumidos en Logix

Los tags producidos y consumidos son la manera en que los datos críticos se transfieren entre controladores Logix conectados en red cada período de tiempo definido. Los datos producidos y consumidos se pueden transmitir por Ethernet/IP o ControlNet y a través del backplane de los controladores ControlLogix.

Los tags producidos y consumidos son tags configurados como producidos o consumidos al momento de crearse. Si un tag se marca como producido, su valor se difundirá a una red EtherNet/IP o ControlNet a la cual está conectado el controlador. Si se marca como consumido, el controlador del cual el tag requiere datos se identificará como parte de la configuración, y el tag consumido recibirá su valor desde el tag producido equivalente en dicho controlador.

Hay canales separados para envío y recepción. El cambiar el valor de un tag consumido no tendrá efecto alguno en el tag producido. Esto se parece a la comunicación de controlador a controlador en S7 y se diferencia de la comunicación de controlador a SCADA, donde cualquier cambio se reflejará en el otro extremo.

No se requiere ninguna programación para configurar conexiones producidas/consumidas. Ello difiere del sistema S7 ya que éste necesita algo de código para comunicación de controlador a controlador (ENVÍO/RECEPCIÓN).

Tipos de datos definidos por el usuario

En Logix, los tipos de datos definidos por el usuario pueden configurarse. Esto permite declarar como tipo la estructura de un tipo de datos complejo. Posteriormente las instancias de dicho tipo pueden definirse en el programa.

Los tipos de datos definidos por el usuario Logix tienen una configuración y un modo de empleo muy similar a los tipos de datos definidos por el usuario STEP 7.

UDT Logix

Name:

Description:

Members: Data Type Size: 28 byte(s)

| | Name | Data Type | Style | Description |
|--------------------------|----------------|-----------|---------|-------------------------------|
| <input type="checkbox"/> | initial_output | REAL | Float | saved initial output |
| <input type="checkbox"/> | increment | REAL | Float | calculated increment |
| <input type="checkbox"/> | RAMP_RATE_ABS | REAL | Float | per second - (set always +ve) |
| <input type="checkbox"/> | RAMP_TARGET | REAL | Float | final value - (set) |
| <input type="checkbox"/> | change | REAL | Float | calculated change over ramp |
| <input type="checkbox"/> | counter | DINT | Decimal | internal counter |
| <input type="checkbox"/> | complete | BOOL | Decimal | ramping is complete |
| <input type="checkbox"/> | _enable | BOOL | Decimal | for enable one shot |
| <input type="checkbox"/> | enabled | BOOL | Decimal | ramper enabled |
| <input type="checkbox"/> | | | | |

Actualización de E/S asíncronas

En los sistemas Logix, las E/S se actualizan asíncronamente con respecto a los períodos de ejecución del programa, a diferencia de la estrategia PLC tradicional como se usa en S7 donde una tabla de imagen de E/S se actualiza al comienzo del ciclo y los valores de entrada no cambian durante la ejecución del programa.

El programador de Logix necesitará considerar si existe la necesidad de almacenar en el búfer los datos de entrada de modo que su valor permanezca constante durante la ejecución del programa.

Es bastante común “consumir” entradas una vez solamente pasándolas como parámetros a un módulo de código. Las entradas no se usarán en ningún otro lugar en el programa. Esto elimina la necesidad de almacenamiento en el búfer. Vea el ejemplo del módulo de control en el [Capítulo 4](#).

El tipo de datos DINT

Los controladores Logix operan en tags DINT (entero de 32 bits) más eficientemente que en INT (entero de 16 bits) o en SINT (entero de 8 bits). Use DINT siempre que sea posible, incluso si el rango de valores con el que está trabajando cabe en un INT o SINT. Estos tipos de datos se proporcionan por razones de compatibilidad con IEC61131-3, pero son convertidos internamente a DINT antes de ser usados por el programa, de modo que el código se ejecute más eficientemente en la mayoría de los casos.

PhaseManager

PhaseManager en STEP 7

STEP 7 no posee herramientas incorporadas para realizar la gestión de fases. Deben programarse las estructuras necesarias en un conjunto de rutinas, llamado normalmente PLI o interface de lógica de fases. Los componentes para un programa PLI basado en S88 son:

- Un secuenciador de pasos cuyo comportamiento cumple con el modelo de estados S88. Ciertos pasos o rangos de pasos definen el estado S88. Los comandos del secuenciador son especificados por S88, y el secuenciador responderá sólo cuando el modelo de estado lo permita. Un secuenciador con estas propiedades se llama una fase.
- Un conjunto de datos para cada fase que se usa para registrar el estado de la fase y para recibir comandos de entrada desde el administrador de recetas. El administrador de recetas se comunica con estos datos. El formato de los datos dependerá del administrador de recetas.
- Un módulo lógico que traduce el estado de la fase al formato requerido por el administrador de recetas, y traduce los comandos del administrador de recetas a comandos de fase.

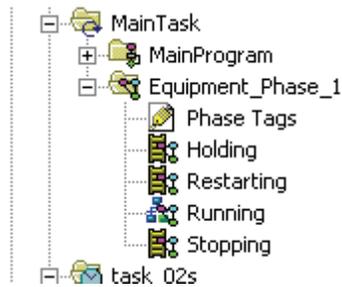
PhaseManager en Logix

En una fase de equipo S88, hay estados especificados de la fase así como las transiciones entre estos estados. PhaseManager es una funcionalidad del software RSLogix 5000 que le permite hacer tres cosas:

- Asignar el código para cada estado de fase a una rutina diferente.
- Ejecutar una máquina de estados “en la trastienda” que maneje las transacciones entre estados de la fase.
- Administrar la ejecución de la fase usando un conjunto de comandos Logix.

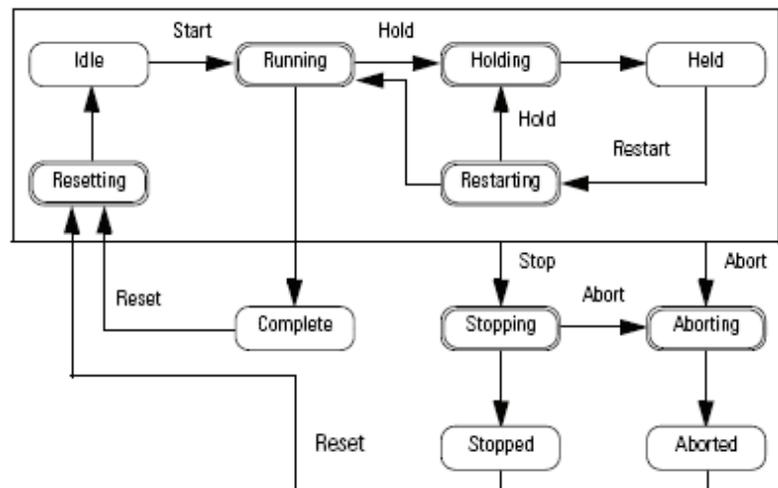
Se usa en una variedad de espacios de aplicación, entre ellos en control de procesos y envasado, porque permite una separación limpia del control de dispositivo/equipo y del control procedimental, lo cual hace que el código sea mucho más modularizado y eficiente, especialmente para sistemas de mayor tamaño con estandarización.

Fase de equipo en el árbol de proyecto



El código para cada estado de la fase puede escribirse en cualquiera de los lenguajes de Logix.

Ésta es la máquina de estados de fase. Es casi idéntica al modelo de estados S88.



Si usted ha programado una rutina PLI / gestor de fases STEP 7 que cumple con S88 y desea convertirla a Logix, puede ser posible evitar la traducción si usa Logix PhaseManager.

Hora coordinada del sistema (CST)

S7 tiene un reloj de sistema cuya hora se representa usando 32 bits y conteos en milisegundos. Sus valores pueden obtenerse (y almacenarse) mediante una llamada al sistema operativo, lo cual permite una medición precisa de los intervalos de tiempo.

Logix usa la hora coordinada del sistema, un número de 64 bits que mide el número de microsegundos desde que el controlador se inició por última vez. Al igual que con S7, los intervalos pueden medirse haciendo llamadas al sistema operativo para obtener el valor CST. Éste proporciona la base para la sincronización de reloj en sistemas de múltiples CPU, la funcionalidad de control de movimiento preciso, la conmutación de salida secuenciada con una precisión de 100 μ s, el sello de hora de eventos de entrada, el muestreo analógico secuenciado, la comunicación y monitoreo de E/S de seguridad, los cálculos de posición de leva de movimiento y la hora del reloj.

Entradas de sello de hora

El sello de hora es una funcionalidad que registra un cambio en los datos de entrada y un tiempo relativo de cuándo ocurrió dicho cambio. Con los módulos de entradas digitales usted puede configurar un sello de hora para cambios de datos. Usted puede usar el sello de hora CST para comparar el tiempo relativo entre muestreos de datos.

Esto brinda al programador una precisión sin paralelo en la vinculación de señales de entrada con referencias de hora para aplicaciones tales como las comúnmente usadas en control de movimiento, sin imponer una carga excesiva en los sistemas de comunicación y procesamiento lógico, así como en el código de aplicación relacionado.

Salidas secuenciadas

Usted puede configurar los módulos de salidas digitales para establecer las salidas a una hora determinada.

Esto brinda al programador una precisión sin paralelo en la vinculación de señales de salida con referencias de hora para aplicaciones tales como posiciones de ejes en control de movimiento o funciones de control de procesos, sin imponer una carga excesiva en los sistemas de comunicación y procesamiento lógico, así como en el código de aplicación relacionado.

Sin variables temporales

S7 tiene una categoría de variables llamada variables temporales. Su alcance es el bloque de programa en el cual están definidas y su vida útil es la ejecución del bloque de programa en el cual están definidas.

Logix no dispone de un equivalente de las variables temporales. Todas las variables son estáticas; es decir, retienen sus valores hasta que se cambian.

Para lograr la funcionalidad que generalmente se busca en las aplicaciones S7, use por ejemplo uno de los siguientes enfoques:

- Use tags bajo el control del programa.
- Si está programando una instrucción Add-On, use tags locales (parte de los datos de la instrucción Add-On).

No se necesitan acumuladores ni registros especiales

Si usted programa en la lista de instrucciones de STEP 7, estará familiarizado con los acumuladores y con los registros de puntero AR1 y AR2. Logix no dispone de equivalentes a estos elementos. Todos los operandos son tags.

Para lograr la funcionalidad que generalmente se busca en las aplicaciones S7, use por ejemplo uno de los siguientes enfoques:

- Use tags bajo el control del programa.
- Si está programando una instrucción Add-On, use tags locales (parte de los datos de la instrucción Add-On).
- Considere si necesita equivalentes Logix de los acumuladores y registros especiales S7. Estos elementos están allí debido a la naturaleza de bajo nivel de la lista de instrucciones S7, pero en un lenguaje como el texto estructurado probablemente no se necesiten.

Conversión del software del sistema y funciones estándar

Introducción

Este capítulo enumera las funciones del sistema S7 más comúnmente usadas, explica cómo se consigue el equivalente en Logix y proporciona varios ejemplos específicos.

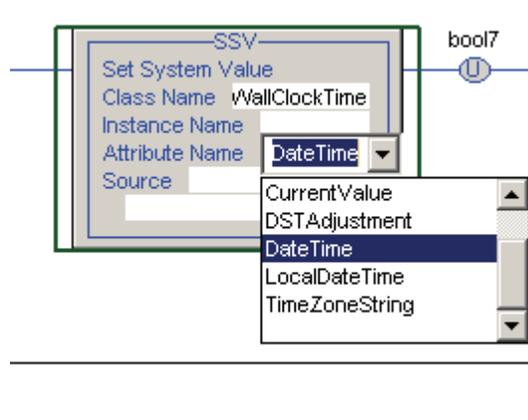
| Tema | Página |
|---|--------|
| Funciones del sistema Logix | 68 |
| Copiar | 68 |
| Establecimiento y lectura de fecha y hora | 69 |
| Leer hora del sistema | 69 |
| Manejo de interrupciones | 70 |
| Errores | 70 |
| Estado – Controlador | 71 |
| Estado – Módulo | 71 |
| Estado – para OB y tareas | 72 |
| Temporizadores | 72 |
| Rutinas de conversión | 73 |
| Rutinas de manejo de cadenas | 73 |
| Ejemplos de llamadas de función del sistema | 74 |

La finalidad de este capítulo es informarle de las instrucciones dedicadas disponibles en Logix para que usted no desperdicie tiempo en el desarrollo de soluciones que ya existen.

Funciones del sistema Logix

En Logix, el equivalente de la mayoría de las funciones del sistema S7 serán las instrucciones GSV (Get System Value) y SSV (Set System Value). Estas instrucciones tienen acceso a una jerarquía de objetos (clases, instancias y atributos) incorporados en los controladores Logix. Si usted programa las instrucciones GSV y SSV, los menús desplegables le guiarán en la selección de parámetros.

Instrucción SSV



Al aprender los aspectos básicos de las instrucciones GSV y SSV, el nuevo usuario de Logix puede descubrir que el acceso al sistema operativo es más fácil que con los SFC S7.

Copiar

Se usa para copiar estructuras de datos complejos, matrices de instancias de tipos de datos de usuario.

| S7 | Comentario | Logix | Comentario |
|----------------|--|---------------------------------|--|
| SFC20 BLKMOV | Con BLKMOV, las direcciones deben definirse al momento de compilar. | COP (instrucción) | Si se usa COP para hacer copias entre matrices, el comienzo del bloque (de fuente o destino) puede incluir un índice de matriz para direccionar el elemento cuyo valor se evalúa durante la ejecución. |
| SFC81 UBLKMOV | Versión no interrumpible, para asegurar que los datos de fuente no puedan cambiarse durante la copia. | CPS (instrucción) | Versión no interrumpible, para asegurar que los datos de fuente no puedan cambiarse durante la copia. |
| SFC14 DPRD_DAT | Si el dispositivo Profibus DP tiene área de datos de comunicaciones > 4 bytes, el SFC asegurará lecturas coherentes. | CPS (ControlNet y Ethernet /IP) | No se requiere para DeviceNet |
| SFC15 DPWR_DAT | Si el dispositivo Profibus DP tiene área de datos de comunicaciones > 4 bytes, el SFC asegurará escrituras coherentes. | CPS (ControlNet y Ethernet /IP) | No se requiere para DeviceNet |

Establecimiento y lectura de fecha y hora

Los controladores de ambos sistemas tienen un reloj en tiempo real que puede leerse o establecerse.

| S7 | Comentario | Logix | Comentario |
|---------------|--|---------------------------|--|
| SFC0 SET_CLK | Valores pasados en una instancia de tipo DT (DateTime) | SSV (Set System Value) | Clase SSV – WallClockTime Atributo SSV – DateTime SSV de fuente – especificar elemento[0] de DINT[7] |
| SFC1 READ_CLK | Valores devueltos en una instancia de tipo DT (DateTime) | GSV (Get System Value) | Clase GSV – WallClockTime Atributo GSV – DateTime GSV de destino – elemento[0] de DINT[7] |

Leer hora del sistema

Los controladores de estos dos sistemas tienen un reloj de sistema que comienza a la hora que se inicia el controlador. En el sistema S7, la hora está en milisegundos, mientras que en Logix está en microsegundos.

| S7 | Comentario | Logix | Comentario |
|----------------|--|---------------------------|---|
| SFC64 TIME_TCK | Retorna la hora del sistema en el rango de 0...2.31 ms | GSV (Get System Value) | Retorna la hora del sistema en el rango de 0...2.63 µs Clase GSV – CST Atributo GSV – CurrentValue GSV de destino – especificar elemento[0] de DINT[2] DINT[0] – 32 bits inferiores DINT[1] – 32 bits superiores |

Manejo de interrupciones

Las interrupciones pueden ser habilitadas e inhabilitadas por el programa del usuario mediante llamadas a las funciones del sistema.

| S7 | Comentario | Logix | Comentario |
|----------------|--|--|--|
| SFC39 DIS_IRT | Inhabilita las interrupciones manejadas por un OB especificado. Las solicitudes de interrupción se pierden. | SSV Inhibe la tarea especificada. | Clase SSV – Tarea Instancia SSV – Nombre de tarea Atributo SSV – InhibitTask SSV de fuente – Variable DINT establecida en 1 |
| SFC39 EN_IRT | Habilita las interrupciones manejadas por un OB especificado. | SSV Habilita la tarea especificada. | Clase SSV – Tarea Instancia SSV – Nombre de tarea Atributo SSV – InhibitTask SSV de fuente – Variable DINT establecida en 0 |
| SFC41 DIS_AIRT | Inhabilita las interrupciones manejadas por un OB especificado. Las solicitudes de interrupción se retardan. | UID | Inhabilita la interrupción de la tarea actual por una tarea de mayor prioridad |
| SFC42 EN_AIRT | Habilita las interrupciones manejadas por un OB especificado. Todas las interrupciones retardadas por SFC41 se ejecutan. | UIE | Habilita las interrupciones de la tarea actual. |

Errores

Estas llamadas del sistema devuelven campos de bits en el caso del S7 o un número entero en el caso de Logix, los cuales representan códigos de error.

| S7 | Comentario | Logix | Comentario |
|----------------|--|---|---|
| SFC38 READ_ERR | Lee y borra bits de error. El tipo de error que se va a indagar puede seleccionarse mediante un campo de filtrado. | GSV (Use SSV para restablecer contadores o fallos) | Clase GSV – FaultLog Atributo GSV: MajorEvents – núm. de eventos mayores MinorEvents – núm. de eventos menores MajorFaultBits – fallo mayor actual MinorFaultBits – fallo menor actual GSV de destino – INT o DINT para recibir datos |

Estado – Controlador

La llamada SFC (S7) y GSV (Logix) devolverá datos sobre el controlador.
 Nota: SFC51 requiere cierto aprendizaje antes de poder usarse. GSV en este sentido es más accesible.

| S7 | Comentario | Logix | Comentario |
|---------------|--|-------|---|
| SFC51 RDSYSST | <p>Los parámetros de entrada especifican la clase de información que se va a leer, y posiblemente un número de instancia si hay varios objetos.</p> <p>Los parámetros de salida son un puntero a una lista con la información devuelta, y el número y tamaño de los elementos en la lista.</p> | GSV | <p>Módulos con una conexión directa: Examine el miembro 'Fault' o 'ChannelFault', si están presentes. Módulos con una conexión de rack optimizada: Examine el miembro 'SlotStatusBits' de los datos de entrada del adaptador o el miembro 'Fault' de la tarjeta como se indica anteriormente. Para todas las demás tarjetas: Ejecute GSV:</p> <p>Clase – Módulo Instancia – ModuleName Atributo – Entrystatus</p> |

Estado – Módulo

La llamada SFC (S7) y GSV (Logix) devolverá datos sobre los módulos instalados.

| S7 | Comentario | Logix | Comentario |
|---------------|--|-------|--|
| SFC51 RDSYSST | <p>Los parámetros de entrada especifican la clase de información que se va a leer, y posiblemente un número de instancia si hay varios objetos.</p> <p>Los parámetros de salida son un puntero a una lista con la información devuelta, y el número y tamaño de los elementos en la lista.</p> | GSV | <p>Clase GSV – Módulo Atributo GSV: EntryStatus (relación del módulo objeto con el módulo) FaultCode FaultInfo ForceStatus LEDStatus Modo (SSV también) GSV de destino – depende del atributo seleccionado</p> |

Usted puede monitorear información de fallo en los tags Logix creados cuando se inserta el módulo en la configuración de E/S. De manera similar, con STEP 7, si va a la configuración de hardware y cambia a “Open ONLINE”, aparecerá la información de fallos de los módulos.

Estado – para OB y tareas

| S7 | Comentario | Logix | Comentario |
|-----------|---|-----------|--|
| OB Header | <p>Los datos de estado de los OB se almacenan en variables temporales que son generadas automáticamente por el encabezado de OB. Se puede acceder a ellas directamente mediante el código OB y transferirse a áreas de datos estáticos si se requiere acceso desde fuera del OB. Vea un ejemplo a continuación.</p> | GSV / SSV | <p>Clase GSV – Tarea Instancia GSV – Nombre de tarea Atributo GSV: DisableUpdateOutputs (al final de la tarea) EnableTimeOut InhibitTask Instancia LastScanTime (microsegundos) MaxIntervaln (entre ejecuciones sucesivas de la tarea) OverlapCount (activado durante la ejecución) Prioridad Régimen (período en microsegundos) StartTime (valor de WallClockTime cuando se inició la última tarea) Estado (3 bits de estado) Temporizador de control (watchdog) (microsegundos) GSV de fuente / destino – depende del atributo seleccionado</p> |

Temporizadores

| S7 | Comentario | Logix | Comentario |
|----------|---|--|---|
| SFB4 TON | Temporizador de retardo a la conexión | TON (LD) TONR (ST y FBD) | Temporizador de retardo a la conexión |
| | | RTO (LD) RTOR (LD y ST) | Temporizador de retardo a la conexión retentivo |
| SFB5 TOF | Temporizador de retardo a la desconexión | TOF (LD) TOFR (ST y FBD) | Temporizador de retardo a la desconexión |
| SFB3 TP | Genera un impulso que se ejecutará incondicionalmente | Bit del acumulador de una instrucción TON de ejecución libre | |

Rutinas de conversión

| S7 | Comentario | Logix | Comentario |
|-----------------------------------|------------------------------|----------------------|--|
| Funciones de la biblioteca | | Instrucciones | |
| FC16 I_STRNG | Número entero a cadena | DTOS | INT puede usarse como tag de fuente en lugar de DINT |
| FC5 DI_STRNG | Número entero doble a cadena | DTOS | DINT a cadena |
| FC30 R_STRG | Número real a cadena | RTOS | Número real a cadena |
| FC38 STRG_I | Cadena a número entero | DTOS | |
| FC37 STRG_DI | Cadena a número entero doble | STOD | Cadena a DINT |
| FC39 STRG_R | Cadena a número real | STOR | Cadena a número real |

Rutinas de manejo de cadenas

| S7 | Comentario | Logix | Comentario |
|---------------|--|------------------------|---|
| | Funciones de la biblioteca | | Instrucciones |
| FC10 EQ_STRNG | Compara cadenas para determinar igualdad | EQU | Comparar cadenas para determinar igualdad |
| FC13 GE_STRNG | Compara cadenas para determinar >= | GEQ (LD) >= (ST) | Compara cadenas para determinar >= |
| FC15 GT_STRNG | Compara cadenas para determinar > | GRT (LD) | Compara cadenas para determinar > |
| FC19 LE_STRNG | Compara cadenas para determinar <= | LEQ (LD) <= (ST) | Compara cadenas para determinar <= |
| FC24 LT_STRNG | Compara cadenas para determinar < | LES (LD) < (ST) | Compara cadenas para determinar < |
| FC29 NE_STRNG | Compara cadenas para determinar <> | NEQ (LD) <> (ST) | Compara cadenas para determinar <> |
| FC21 LEN | Longitud de la cadena | .LEN | Propiedad de cualquier instancia de cadena |
| FC26 MID | Regresa una sección central de la cadena | MID | Regresa una sección central de la cadena |
| FC2 CONCAT | Concatena dos cadenas | CONCAT | Concatena dos cadenas |
| | Puede realizarse con FC31 REPLACE | DELETE | Elimina una sección de una cadena |
| FC17 INSERT | Inserta cadena de fuente en cadena de destino | INSERT | Inserta cadena de fuente en cadena de destino |
| FC31 REPLACE | Reemplaza n caracteres de la cadena de destino por la cadena de fuente | Use DELETE / INSERT | |
| FC11 FIND | Encuentra una cadena en otra cadena | FIND | Encuentra una cadena en otra cadena |

No existe un equivalente en STEP 7 para las instrucciones de puerto en serie ASCII de Logix, ni en el conjunto de instrucciones ni en la biblioteca de funciones. Éstas tendrían que programarse en STL si se requiere.

Ejemplos de llamadas de función del sistema

Estos ejemplos se ofrecen principalmente para ilustrar el uso de las instrucciones GSV/SSV.

Puesta en hora del reloj

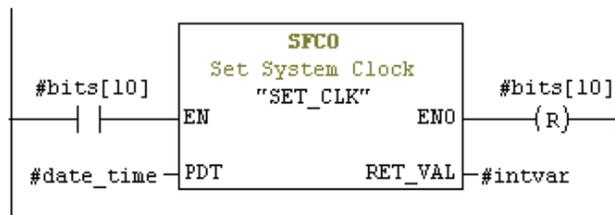
STEP 7

Esta llamada a SFC0 pone en hora el reloj. La hora y fecha se introducen en formato #fecha_hora.

La fecha y hora se almacenan en formato de 8 bits a continuación de #fecha_hora en formato BCD.

Network 14 : Title:

```
set the clock to the value stored in "date_time"
```



0 – año

1 – mes

2 – día

3 – hora

4 – minuto

5 – segundo

6 – los 2 dígitos más significativos de milisegundos

7 – el dígito menos significativo de milisegundos y el día de la semana

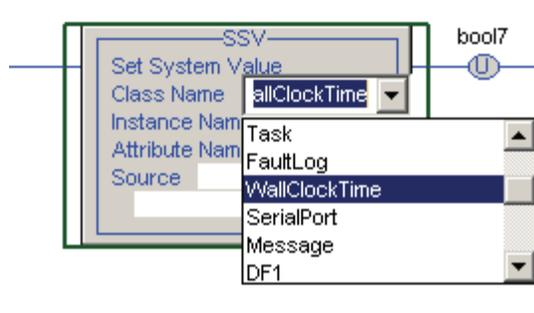
Logix

Los valores de fecha y hora se almacenan en los siete DINT a continuación de #fecha_hora.

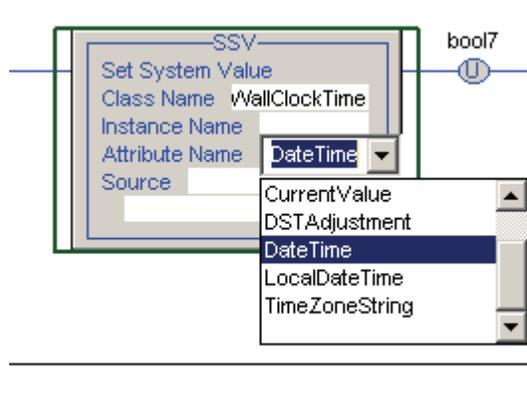


- 0 – año
- 1 – mes
- 2 – día
- 3 – hora
- 6 – minuto
- 5 – segundo
- 6 – microsegundo

La captura de pantalla de Logix muestra la estructura de datos asociada con GSV y SSV. Seleccione la clase desde un menú desplegable como sigue.



Seleccione el atributo desde el menú desplegable como sigue.



Finalmente, seleccione el tag que será la fuente (SSV) o el destino (GSV) de los datos.

Inhabilitación de interrupciones

STEP 7

Network 2 : Title:

```
Disable interrupts for the Interrupt Execution (ie Periodic) 0B35
```

```
CALL "DIS_IRT"           SFC39           -- Disable New Int  
MODE   :=B#16#2  
OB_NR  :=35  
RET VAL:=#intVar
```

Logix

Este ejemplo muestra SSV en texto estructurado.

Si usted tecléa “gsv” y seguidamente “alt-A”, aparecerá la siguiente pantalla de selección de parámetros.

```
// disable task_0.2s  
ssv();
```



Cuando haya introducido los parámetros, haga clic en “OK” para completar los parámetros.

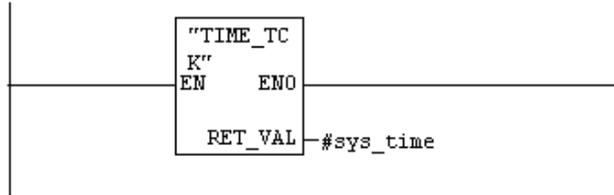
```
// disable task_0.2s  
ssv(Task,task_02s,InhibitTask,disable);
```

Leer hora del sistema

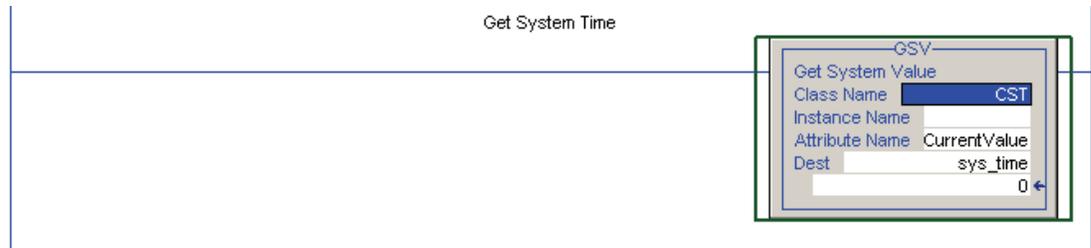
STEP 7

Network 15: Title:

```
read system time
```



Logix

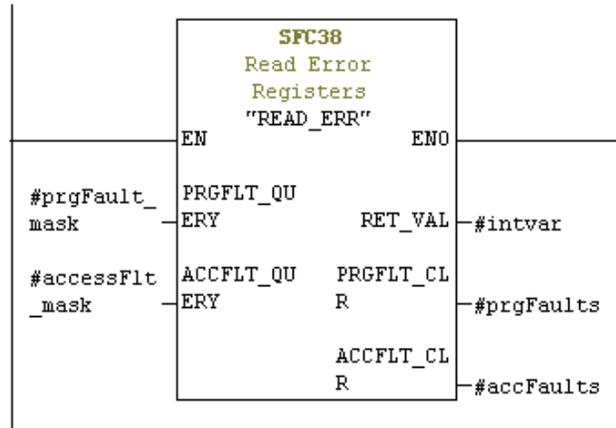


Obtener fallos

STEP 7

Network 16 : Title:

Get programming faults and I/O access faults



El patrón de bits en los parámetros de entrada actúa como filtro para seleccionar los fallos que se van a indagar. Los fallos retornados son los fallos **enmascarados** – el enmascarado evita que detengan al controlador o llamen a un OB de fallo.

Logix



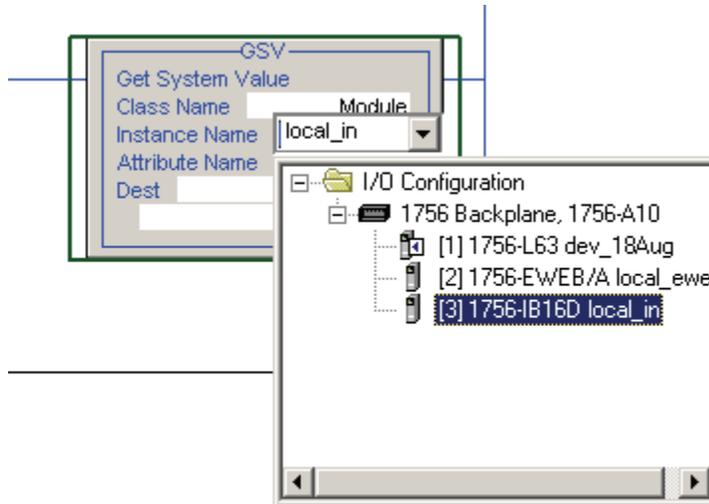
Información del módulo

La manera más fácil es inspeccionar los tags de perfil del dispositivo módulo, los cuales contienen información de fallo/diagnóstico.

Tag de tarjeta de entradas analógicas de termopar 1756-IT6I2

| Name | Alias For | Base Tag | Data Type |
|----------------------------|-----------|----------|--------------------------|
| Local:4:C | | | AB:1756_AI6_Float:C:0 |
| Local:4:I | | | AB:1756_AI6_CJ_Float:I:0 |
| Local:4:I.ChannelFaults | | | INT |
| Local:4:I.Ch0Fault | | | BOOL |
| Local:4:I.Ch1Fault | | | BOOL |
| Local:4:I.Ch2Fault | | | BOOL |
| Local:4:I.Ch3Fault | | | BOOL |
| Local:4:I.Ch4Fault | | | BOOL |
| Local:4:I.Ch5Fault | | | BOOL |
| Local:4:I.ModuleFaults | | | INT |
| Local:4:I.AnalogGroupFault | | | BOOL |
| Local:4:I.InGroupFault | | | BOOL |
| Local:4:I.Calibrating | | | BOOL |
| Local:4:I.CalFault | | | BOOL |
| Local:4:I.CJUnderrange | | | BOOL |
| Local:4:I.CJOverrange | | | BOOL |
| Local:4:I.Ch0Status | | | SINT |
| Local:4:I.Ch0CalFault | | | BOOL |
| Local:4:I.Ch0Underrange | | | BOOL |
| Local:4:I.Ch0Overrange | | | BOOL |
| Local:4:I.Ch0RateAlarm | | | BOOL |
| Local:4:I.Ch0LAlarm | | | BOOL |
| Local:4:I.Ch0HAlarm | | | BOOL |
| Local:4:I.Ch0LLAlarm | | | BOOL |
| Local:4:I.Ch0HHAlarm | | | BOOL |

Otra manera es usar la instrucción GSV para leer objetos del módulo. La captura de pantalla siguiente muestra cómo usar GSV para obtener información respecto al módulo de entradas digitales 1756-IB16D.



Obtener tiempo de escán

STEP 7

Ésta es una captura de pantalla del encabezado de variables temporales de OB1.

| Contents Of: 'Environment\Interface\TEMP' | | | | |
|---|-------------|---------|---|--|
| Name | Data Type | Address | Comment | |
| OB1_SCAN_1 | Byte | 1.0 | 1 (Cold restart scan 1 of OB 1), 3 (Scan 2-n of OB 1) | |
| OB1_PRIORITY | Byte | 2.0 | Priority of OB Execution | |
| OB1_OB_NUMBR | Byte | 3.0 | 1 (Organization block 1, OB1) | |
| OB1_RESERVED_1 | Byte | 4.0 | Reserved for system | |
| OB1_RESERVED_2 | Byte | 5.0 | Reserved for system | |
| OB1_PREV_CYCLE | Int | 6.0 | Cycle time of previous OB1 scan (milliseconds) | |
| OB1_MIN_CYCLE | Int | 8.0 | Minimum cycle time of OB1 (milliseconds) | |
| OB1_MAX_CYCLE | Int | 10.0 | Maximum cycle time of OB1 (milliseconds) | |
| OB1_DATE_TIME | Date_And... | 12.0 | Date and time OB1 started | |

#OB1_PREV_CYCLE es el tiempo de escán. Como variable temporal, deja de existir cuando concluye la ejecución de OB1. Para almacenar el tiempo de escán, copie #OB1_PREV_CYCLE a una ubicación de la memoria estática.

Logix

El tiempo de ejecución puede recuperarse para cada tarea Logix.



Con S7 usted puede obtener directamente el tiempo de escán para OB1 desde B1_PREV_CYCLE. Sin embargo, para OB periódicos, no existe un equivalente a #OB1_PREV_CYCLE. Para obtener el tiempo de ejecución de los OB periódicos, usted necesitará insertar llamadas a SFC64 TIME_TCK al inicio y al final del OB, y restar las horas del reloj del sistema devueltas por el SFC.

Conversión de estructuras de programa típicas

Introducción

El objetivo de esta sección es demostrar cómo algunas tareas de programación típicas en STEP 7 pueden ejecutarse en el software RSLogix 5000. La descripción se basa principalmente en fragmentos de código, pero también se proporcionan algunos ejemplos completos.

| Tema | Página |
|--|--------|
| Ejemplos de código de conversión | 83 |
| Otros temas relacionados con la programación | 120 |
| Un ejemplo mayor – módulo de control | 121 |

También se describen algunos asuntos relacionados con la programación, tal como el alcance y la visibilidad de las variables, y la secuenciación de las secciones de código.

Ejemplos de código de conversión

Estos ejemplos muestran código de conversión.

Traducción de lógica de escalera

Esta sección describe algunos ejemplos de comparación entre STEP 7 LAD y Logix LD.

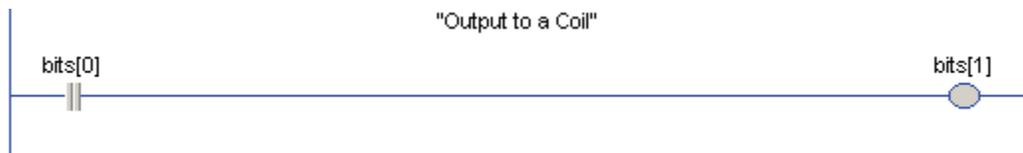
Escritura a una bobina

STEP 7

Output to a coil



LOGIX



Establecer y restablecer

STEP 7

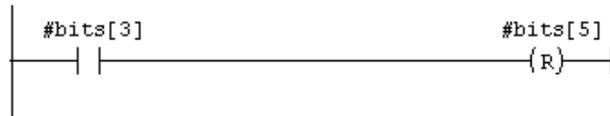
Network 3 : Title:

```
set bit
```



Network 4 : Title:

```
reset bit
```



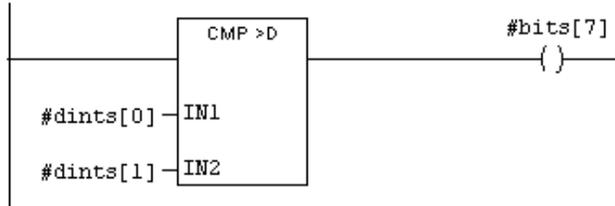
LOGIX



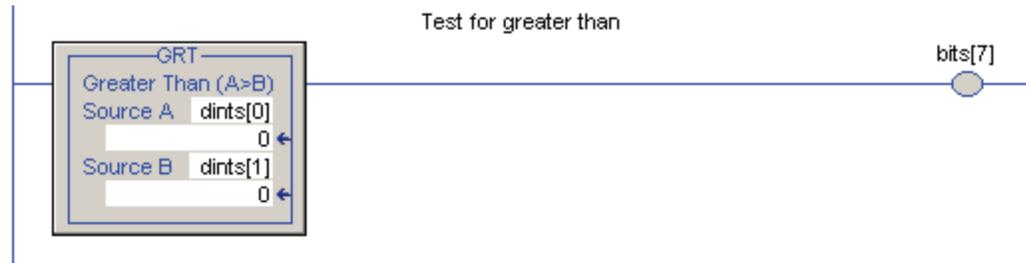
Probar si es mayor que

STEP 7

test for greater than



LOGIX

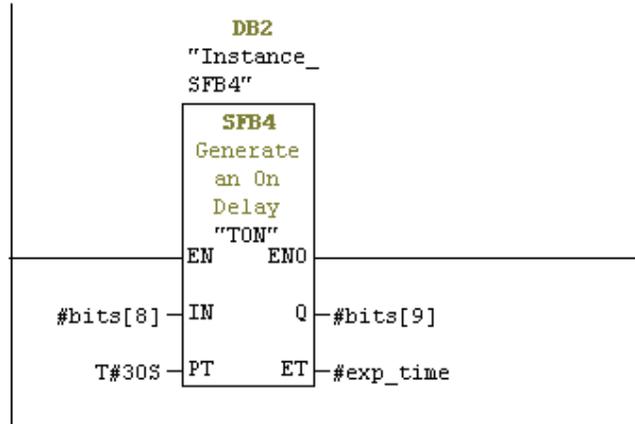


Al igual que antes, use la instrucción CMP si la expresión es más compleja que simplemente una comparación entre dos números.

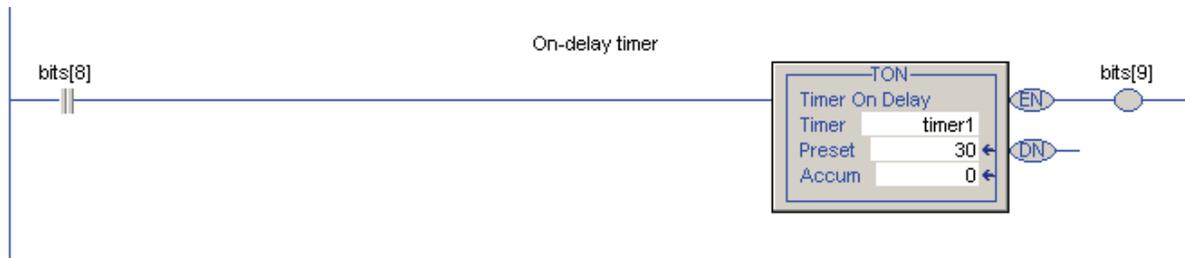
Retardo de temporizador a la conexión

STEP 7

On delay timer



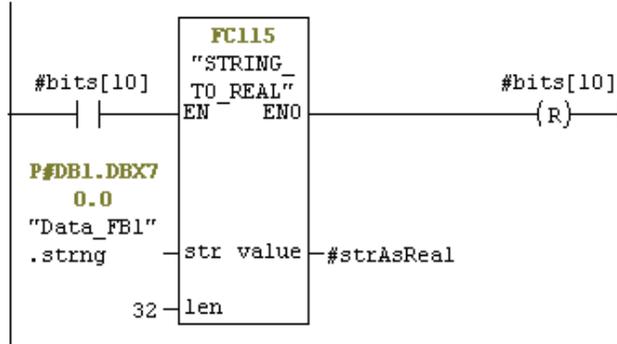
LOGIX



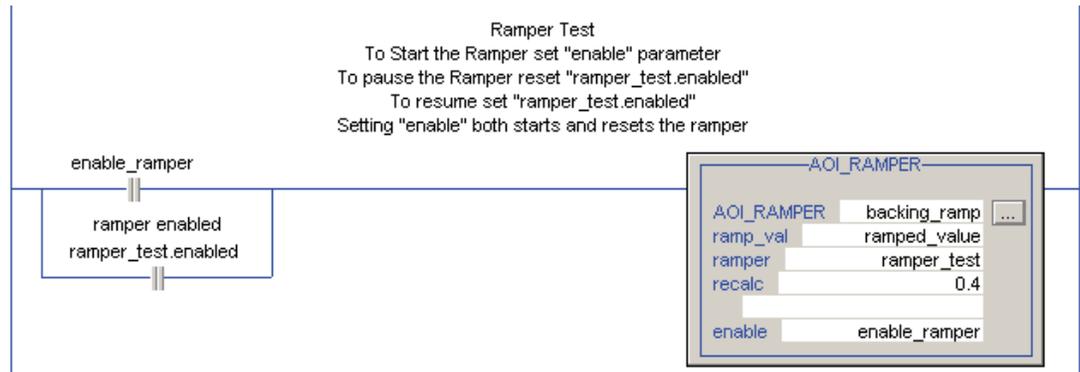
Llamada a función de usuario

STEP 7

user-function call



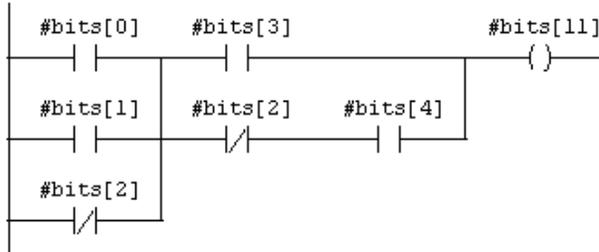
LOGIX



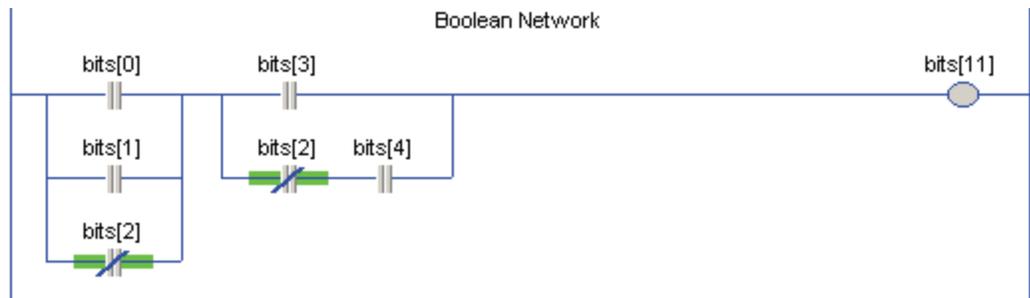
Red booleana

STEP 7

boolean network



LOGIX

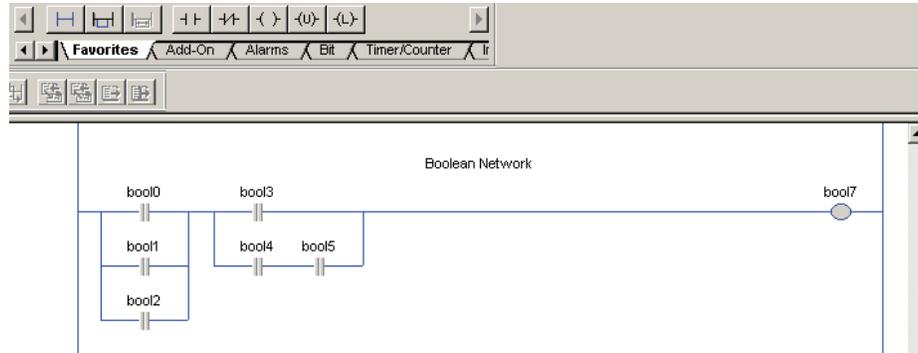


Hay suficiente similitud entre STEP 7 LAD y Logix LD para hacer que la traducción a nivel de rutinas sea bastante sencilla.

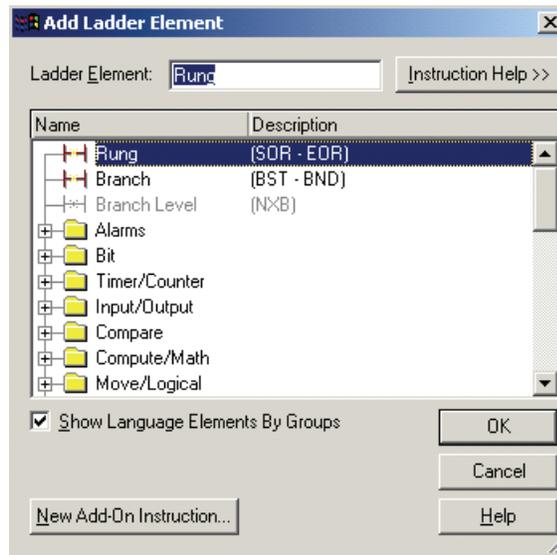
El editor de Logix LD

Hay por lo menos siete maneras de seleccionar instrucciones LD. A continuación se describen dos métodos que son bastante similares a la manera en que esto se realiza en STEP 7.

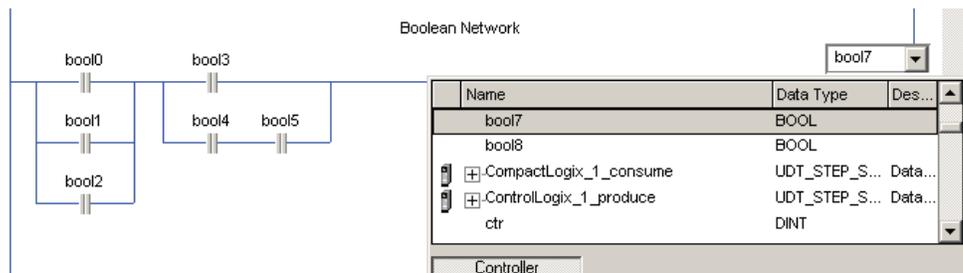
Usted puede seleccionar desde una paleta ubicada en la parte superior de la hoja de cálculo LD.



Si tecllea Alt+Insert, aparecerá este diálogo emergente de selección.



Al configurar instrucciones, están disponibles los menús desplegables para permitirle seleccionar el tag a introducir.



Saltos y toma de decisiones

STEP 7 – Secuencia de saltos convencional

El siguiente ejemplo de tarea se explica en el comentario de red. Se muestran dos versiones S7 porque ambas se usan con frecuencia.

Network 1: Multi-way selection

```
if #input is 5 set #target to 8
else if #input is 6 set #target to 10
else if #input is 7 set #target to 16
else set #input to 0
```

```

L   #input
L   5
==I
JCN _001

L   8
T   #target

JU  end

_001: L   #input
L   6
==I
JCN _002

L   10
T   #target

JU  end

_002: L   #input
L   7
==I
JCN _003

L   16
T   #target

JU  end

_003: L   0
T   #target

end: NOP 0
```

El valor de #input se compara con el conjunto de constantes hasta que se encuentra el valor de comparación. Seguidamente se emprende la acción y la comparación se detiene. Se ejecuta una acción predeterminada si #input no es igual a ningún valor en el conjunto.

STEP 7 – Lista de saltos

En este ejemplo la tarea es igual pero se usa una lista de saltos. Esto es similar a una tabla de saltos de microprocesador, y transfiere la ejecución a una etiqueta dependiendo del valor de una variable.

Network 2 : Title:

```
if #input is 5 set #target to 8
else if #input is 6 set #target to 10
else if #input is 7 set #target to 16
else set #input to 0
```

```

L   #input
L   5
-I
JL  rng
JU  d5
JU  d6
JU  d7
rng: L   0
    T   #target
    JU  cont

d5:  L   8
    T   #target
    JU  cont

d6:  L   10
    T   #target
    JU  cont

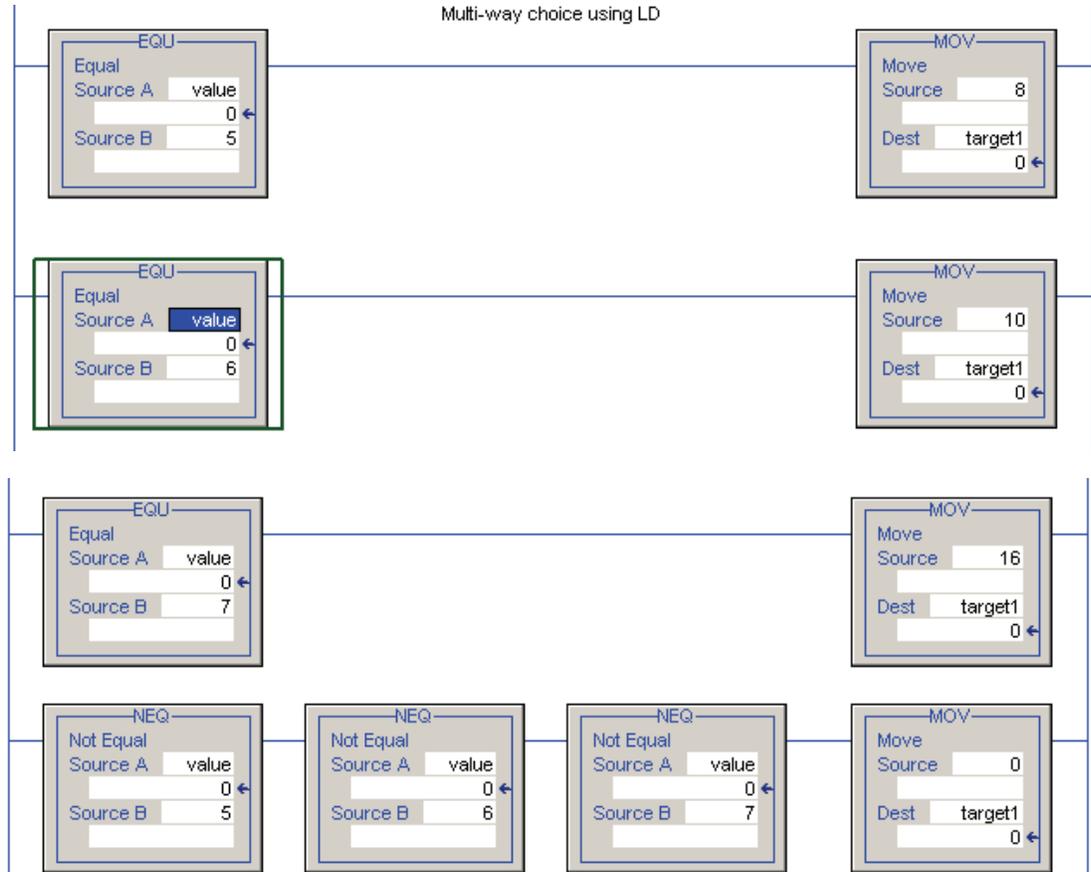
d7:  L   16
    T   #target

cont: NOP 0
```

Esto es más fácil de leer que la secuencia de saltos convencional, y es más eficiente porque sólo se ejecuta el código en la etiqueta de destino.

Logix – Lógica de escalera

Esto muestra múltiples maneras de selección usando LD.



Logix – Texto estructurado If...Then...Else

Cualquier persona familiarizada con un lenguaje de programación de las familias Basic/Pascal/C entenderá esto sin dificultad.

```
// multi-way choice using Structured Text  
  
if (value = 5) then target := 8;  
elsif (value = 6) then target := 10;  
elsif (value = 7) then target := 16;  
else target := 0;  
end_if;
```

Los corchetes alrededor de la condición “if” no son obligatorios.

Instrucción CASE de texto estructurado Logix

Existe otra variante en ST que realiza la misma tarea. Es suficientemente compacta y limpia, por lo que hay poca necesidad de comentarios adicionales.

```
// multi-way choice using Structured Text CASE  
  
case value of  
  5: target := 8;  
  6: target := 10;  
  7: target := 16;  
else target :=0;  
end_case;
```

Todas las soluciones funcionan, pero ésta es la solución Logix preferida. Es compacta y suficientemente clara, por lo que no se requiere documentación adicional.

Matrices

STEP 7 y Logix permiten la creación de matrices de objetos simples o complejos en la memoria. Logix cuenta con soporte de alto nivel para acceder a las matrices. Sin embargo, en STEP 7 se requiere programación de bajo nivel.

Creación de matriz STEP 7

La siguiente captura de pantalla muestra dos matrices creadas en un bloque de datos de instancia. Simple_array es una matriz de 10 elementos. UDT_array es una matriz de 10 estructuras de tipo test_UDT1, donde test_UDT1 es un tipo de datos de usuario que contiene algunos otros tipos no mostrados.

| Contents Of: 'Environment\Interface\STAT' | | | | |
|---|-----------------|---------|---------------|--|
| Name | Data Type | Address | Initial Value | |
| input | Int | 6.0 | 0 | |
| target | Int | 8.0 | 0 | |
| simple_array | Array [0..9...] | 10.0 | | |
| UDT_array | Array [0..9...] | 50.0 | | |
| state | Int | 170.0 | 0 | |
| error | Bool | 172.0 | FALSE | |

Creación de matriz Logix

Es exactamente igual que en Logix.

| Name | Alias For | Base Tag | Data Type | Style | Description |
|--------------|-----------|----------|---------------|---------|---------------------------------|
| target | | | DINT | Decimal | |
| value | | | DINT | Decimal | |
| simple_array | | | DINT[10] | Decimal | |
| UDT_array | | | test_UDT1[10] | | For testing Step7->Controllo... |
| index | | | DINT | Decimal | |

Sintaxis de declaración de matriz

STEP 7 usa la sintaxis de declaración ARRAY[0...15] OF REAL.

Logix usa REAL[15].

STEP 7 tiene una sintaxis especial para cadenas. STRING[32] es una cadena de 32 caracteres en STEP 7, mientras que en Logix, STRING[32] es una matriz de treinta y dos cadenas y cada una de ellas contiene 82 caracteres.

Acceso a matriz en STEP 7

En este ejemplo se ejecuta una tarea simple en las dos matrices `simple_array[]` y `UDT_array[]`. La tarea se describe en el comentario de red.

En STEP 7, no es posible obtener acceso a las matrices usando la notación `array[]`. En lugar de ello, tiene que usar operaciones de bajo nivel con punteros. En el fragmento siguiente, la función "GET_INDEXED_REFERENCE" hace que la tarea sea mucho más fácil al retornar un puntero dirigido al elemento de matriz al cual se va a acceder.

Network 3 : Title:

```

array operations
-----
if (simple_array[2] = simple_array[5]) then
    UDT_array[8].boolean1 := 1;
else
    UDT_array[8].boolean1 := 0;
end_if;

// 1. compare simple_array[2] with simple_array[5]
CALL "GET_INDEXED_REFERENCE"          FC111
    refArray := "Data_test".simple_array P#DB1.DBX10.0
    index    := 2
    byteIncr := 4
    startIndex:=FALSE
    retVal   := #ptr1                    [BOOL]

CALL "GET_INDEXED_REFERENCE"          FC111
    refArray := "Data_test".simple_array P#DB1.DBX10.0
    index    := 5
    byteIncr := 4
    startIndex:=FALSE
    retVal   := #ptr2

OPN "Data_test"                        DB1
L   DID [#ptr1]
L   DID [#ptr2]
==D
=   #compare

// 2. get pointer to UDT_array
CALL "GET_INDEXED_REFERENCE"          FC111
    refArray := "Data_test".UDT_array  P#DB1.DBX44.0
    index    := 8
    byteIncr := 12                       [INT]
    startIndex:=FALSE
    retVal   := #ptr1

L   #ptr1
LAR1

// 3. set or reset the bit
A   #compare
=   DIX [AR1,P#0.0]
    
```

En este caso, el código de texto de estructura Logix se usó como comentario de red, lo cual demuestra lo intuitivo que es el código Logix.

STEP 7 – Lazo a través de elementos de matriz

El objetivo de este ejemplo es borrar el campo Float en cada estructura en UDT_array[]. No es difícil, pero se requiere confianza en el uso de punteros.

```
Array Operations
-----
Clear all float elements at UDT offset P#6.0 in array UDT_array
```

```
// transfer pointer to UDT_array to AR1
    L    P##UDT_array
    LAR1
// initialise counter
    L    0
    T    #ctr

// end if #ctr > 9
loop: L    #ctr
    L    9
    >I
    JC    end2
// clear the float field at offset p#6.0
    L    0.000000e+000
    T    DID [AR1,P#6.0]
// increment AR1 by size of the UDT
    +AR1 P#12.0
// increment counter
    L    #ctr
    INC  1
    T    #ctr
// loop back
    JU    loop

end2: NOP  0
```

Logix – Operaciones de matriz en texto estructurado

El siguiente fragmento ST realiza las tareas descritas en las dos secciones anteriores.

```
// array access in ST
if (simple_array[2] = simple_array[5]) then
    UDT_array[8].boolean1 := 1;
else
    UDT_array[8].boolean1 := 0;
end_if;

// clearing array elements
if (simple_array[0] = 5) then
    index := 0;
    while (index <= 9) do
        UDT_array[index].reall := 0.0;
        index := index + 1;
    end_while;
end_if;
```

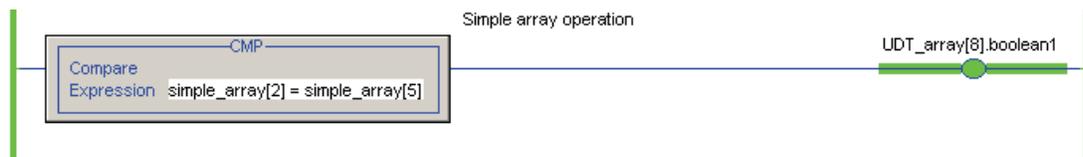
No se necesitan comentarios adicionales para describir cómo funciona esto.

Si usted se encuentra conmutando variables booleanas con las instrucciones If...Then...Else, considere escribir una ecuación booleana en lugar de ello.

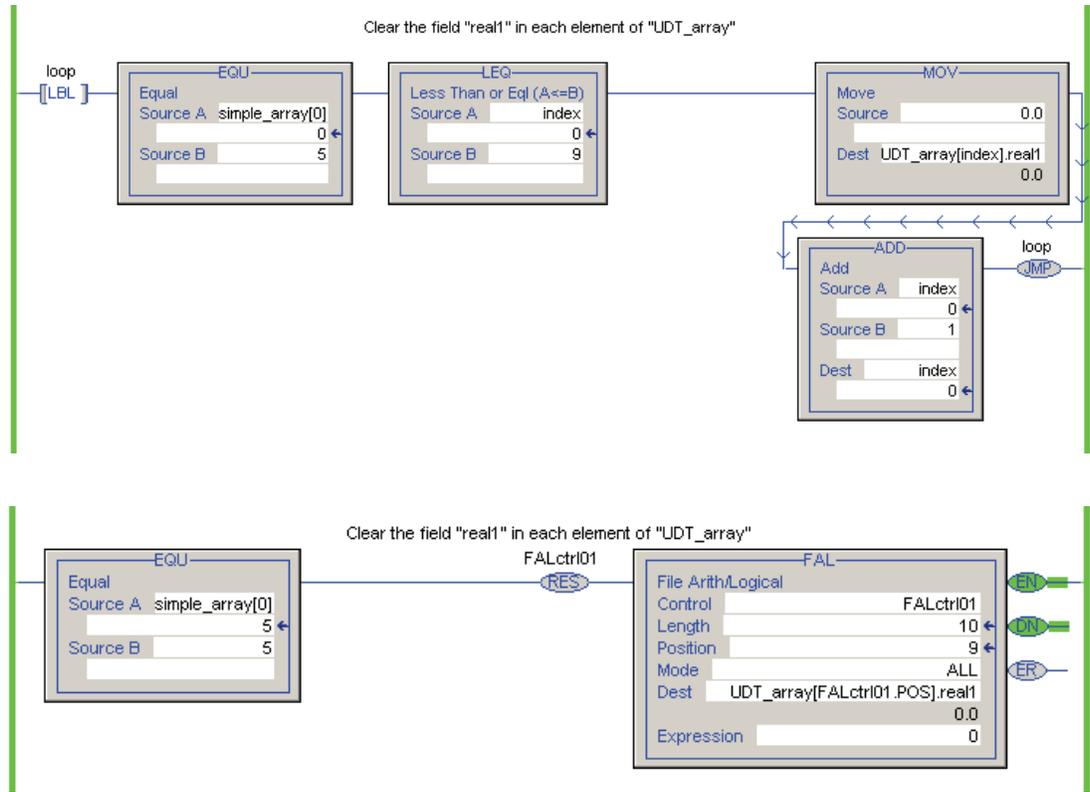
```
// array access in ST
UDT_array[8].boolean1 := simple_array[2] = simple_array[5];
```

Logix – Operaciones de matriz en diagrama de lógica de escalera

Los ejemplos de la sección anterior pueden escribirse en LD usando la instrucción CMP (Compare) como sigue.



La segunda (borrar el campo real en la matriz de UDT) puede realizarse de cualquiera de estas maneras.



El primer método para borrar los elementos de matriz es una traducción del lazo While del código ST. El segundo usa la instrucción FAL avanzada para operaciones de matriz.

Tipos de datos de usuario

En STEP 7 y en Logix, configurar tipos de datos de usuario (UDT) es muy similar.

Más abajo verá un UDT en STEP 7.

| Address | Name | Type | Initial value | Comment |
|---------|----------|------------|---------------|---------|
| 0.0 | | STRUCT | | |
| +0.0 | boolean1 | BOOL | FALSE | |
| +0.1 | boolean2 | BOOL | FALSE | |
| +2.0 | dint1 | DINT | L#0 | |
| +6.0 | real1 | REAL | 0.000000e+000 | |
| +10.0 | spare | WORD | W#16#0 | |
| =12.0 | | END_STRUCT | | |

Más abajo verá un UDT en Logix.

Name:

Description:

Members: Data Type Size: 16 byte(s)

| Name | Data Type | Style | Description |
|----------|-----------|---------|-------------|
| boolean1 | BOOL | Decimal | |
| boolean2 | BOOL | Decimal | |
| dint1 | DINT | Decimal | |
| real1 | REAL | Float | |
| spare | DINT | Decimal | |
| 16#010 | | | |

En ambos sistemas, los UDT pueden usarse para declarar y definir variables.

La siguiente declaración incluye un UDT en STEP 7.

| Contents Of: 'Environment\Interface\STAT' | | | | |
|---|-----------------------|---------|---------------|--|
| Name | Data Type | Address | Initial Value | |
| input | Int | 6.0 | 0 | |
| target | Int | 8.0 | 0 | |
| simple_array | Array [0..9] Of Dint | 10.0 | | |
| UDT_array | Array [0..9] Of UDT 1 | 50.0 | | |

La siguiente declaración incluye un UDT en Logix.

| Name | Alias For | Base Tag | Data Type | Style |
|---------------------|------------------|------------------|----------------|---------|
| Limit_Switch_1 | Local:3:I.Data.0 | Local:3:I.Data.0 | BOOL | Decimal |
| Local:3:C | | | AB:1756_DI:C:0 | |
| Local:3:I | | | AB:1756_DI:I:0 | |
| conveyor_1 | | | UDT1 | |
| conveyor_1.boolean1 | | | BOOL | Decimal |
| conveyor_1.boolean2 | | | BOOL | Decimal |
| conveyor_1.dint1 | | | DINT | Decimal |
| conveyor_1.real1 | | | REAL | Float |
| conveyor_1.spare | | | DINT | Decimal |

A continuación se ve una diferencia menor entre los dos sistemas:

En STEP 7 usted puede declarar una variable de tipo “struct”.

| Contents Of: 'Environment\Interface\STAT' | | | | |
|---|-----------------------|---------|---------------|--|
| Name | Data Type | Address | Initial Value | |
| target | Int | 8.0 | 0 | |
| simple_array | Array [0..9] Of Dint | 10.0 | | |
| UDT_array | Array [0..9] Of UDT 1 | 50.0 | | |
| state | Int | 170.0 | 0 | |
| error | Bool | 172.0 | FALSE | |
| transition01 | Bool | 172.1 | FALSE | |
| transition12 | Bool | 172.2 | FALSE | |
| transition13 | Bool | 172.3 | FALSE | |
| transition24 | Bool | 172.4 | FALSE | |
| transition43 | Bool | 172.5 | FALSE | |
| transition31 | Bool | 172.6 | FALSE | |
| str | String[46] | 174.0 | " | |
| table | Struct | 222.0 | | |

Observe la “tabla” de entrada de tipo Struct. El interior de la “tabla” puede ser un conjunto (ordenado o no) de cualquier combinación de tipos.

En Logix, esto se haría definiendo “Struct” como un UDT que contiene la estructura de datos deseada y seguidamente declarando la “tabla” de tipo Struct.

| Name | Alias For | Base Tag | Data Type | Style |
|----------------------|------------------|------------------|----------------|---------|
| Limit_Switch_1 | Local:3:I.Data.0 | Local:3:I.Data.0 | BOOL | Decimal |
| + Local:3:C | | | AB:1756_DI:C:0 | |
| + Local:3:I | | | AB:1756_DI:I:0 | |
| - conveyor_1 | | | UDT1 | |
| conveyor_1.boolean1 | | | BOOL | Decimal |
| conveyor_1.boolean2 | | | BOOL | Decimal |
| + conveyor_1.dint1 | | | DINT | Decimal |
| conveyor_1.real1 | | | REAL | Float |
| + conveyor_1.spare | | | DINT | Decimal |
| - table | | | Struct | |
| table.status_bit1 | | | BOOL | Decimal |
| table.status_bit2 | | | BOOL | Decimal |
| table.status_bit3 | | | BOOL | Decimal |
| + table.dwell_timer1 | | | TIMER | |
| + table.dwell_timer2 | | | TIMER | |
| table.speed | | | REAL | Float |

Punteros y matrices

Un programa STEP 7 puede tener punteros dirigidos a cualquier objeto de datos. También se permite acceso indirecto a los bloques de datos, pero no hay punteros dirigidos a funciones (excepto de manera restringida por la instrucción JL [Jump List]). El puntero de datos es poco usual por el hecho de que es un puntero dirigido a un bit. Es por ello que su valor es ocho veces el de un puntero normal dirigido a un byte. Esto indica la importancia de los bits en la programación de sistemas de control.

En Logix no hay punteros. Las matrices realizan la misma función que los punteros, pero son más simples y seguros.

¿Podrá el programador de S7 ejecutar la misma gama de tareas en Logix sin punteros? En programación de computadoras, los punteros dirigidos a datos se usan principalmente con tres finalidades:

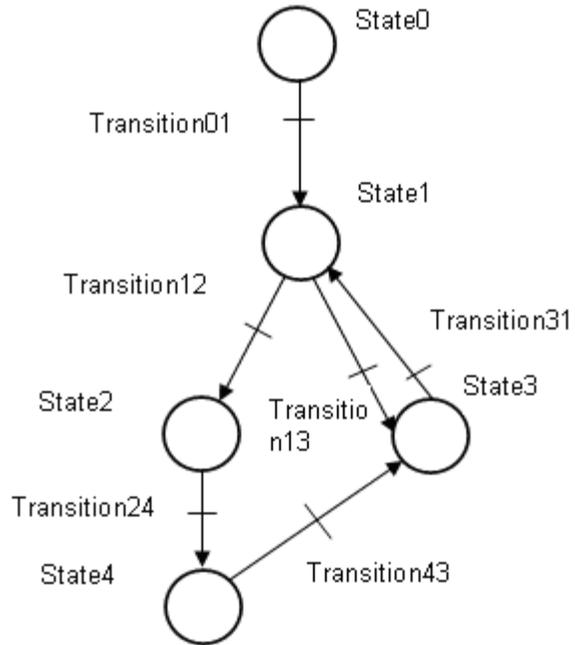
- Operaciones en ítems de datos ordenados secuencialmente (matrices de objetos, cadenas).
- Asignación, acceso y eliminación de objetos asignados dinámicamente.
- Paso de referencias a objetos como parámetros en llamadas de funciones.

En Logix, la primera finalidad está cubierta por las matrices. La segunda finalidad no es relevante en el software de control porque no tenemos objetos asignados dinámicamente. La tercera está cubierta por los parámetros “inout” tanto en los bloques de funciones de STEP 7 como en las instrucciones Add-On de Logix.

Por lo tanto, la ausencia de punteros explícitos no representa una limitación para los programadores de Logix. Los programadores de STEP 7 también descubrirán que la codificación con matrices puede realizarse más rápidamente en texto estructurado mediante matrices que en STL mediante punteros.

Máquina de estados

La máquina de estados es una construcción importante en el software de sistemas de control porque simplifica considerablemente la tarea de programación de control secuencial.



Máquina de estados STEP 7

STEP 7 ofrece un diagrama de función secuencial gráfico como opción adicional a la aplicación básica. Si el SFC gráfico no está disponible, la lista de instrucciones cubrirá las necesidades.

Network 4: Title:

```
state machine
```

```

L      #state
JL     rngl
JU     st0
JU     st1
JU     st2
rngl: SET
S      #error
BEU

st0: L   1
A     #transition01
JC    next
JU    ovr

st1: L   2
A     #transition12
JC    next

L     3
A     #transition13
JC    next
JU    ovr

st2: L   4
A     #transition24
JC    next
JU    ovr

st3: L   0
A     #transition31
JC    next
JU    ovr

st4: L   3
A     #transition43
JC    next
JU    ovr

next: T   #state

ovr: NOP 0

```

La variable #state contiene el número de estado. La instrucción Jump List hace que la ejecución salte a la etiqueta correspondiente al valor de #state. Si una condición de transición de dicho estado es verdadera, el nuevo valor de estado se carga en el acumulador y la ejecución salta a la etiqueta “next”, donde el nuevo número de estado se transfiere a la variable #state.

Logix – Máquina de estados en texto estructurado

Ésta es la misma máquina de estados en texto estructurado, usando la instrucción CASE. Al igual que con los otros ejemplos de ST, sería difícil escribir una descripción más clara que el código.

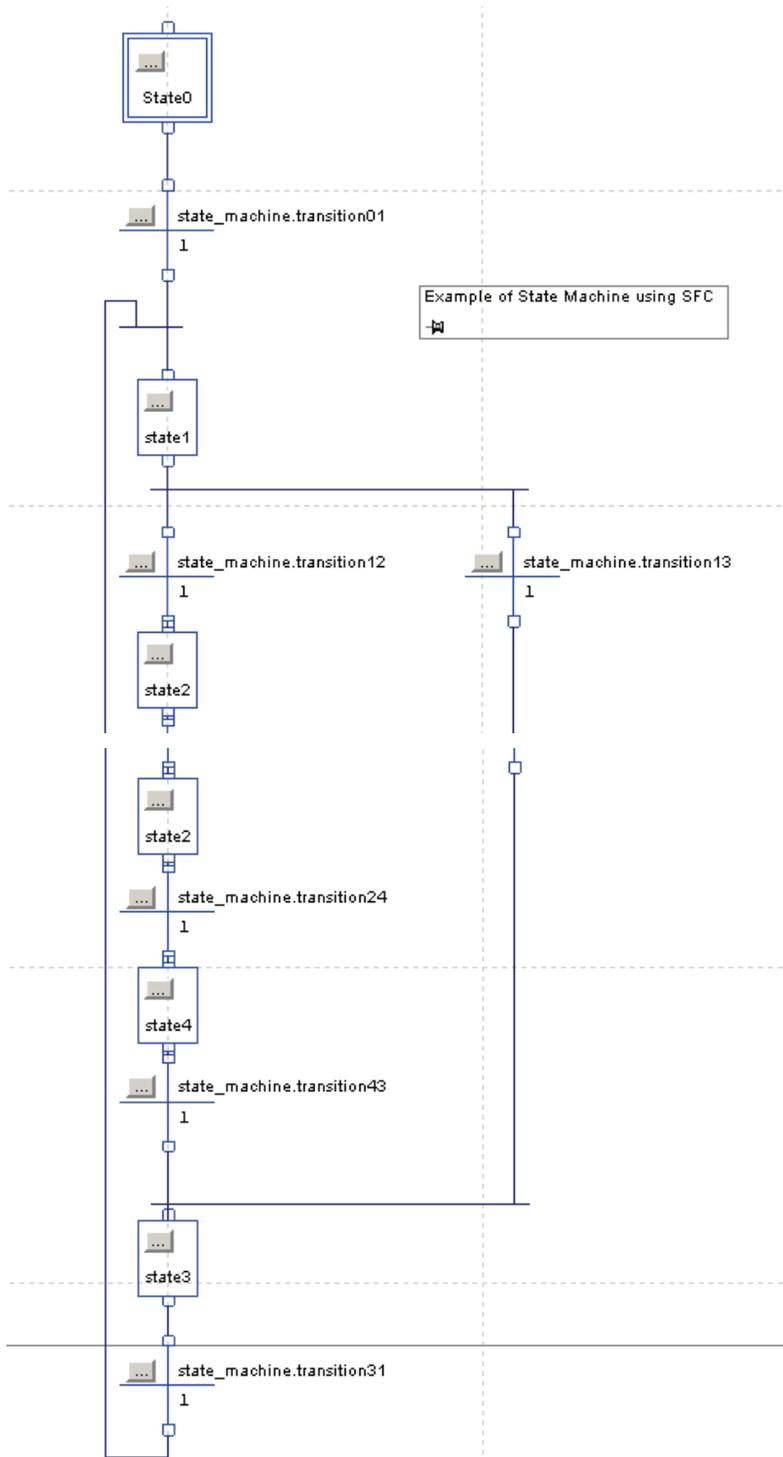
```
// implementation of State Machine using CASE in ST

case state_machine.state of
0:  if state_machine.transition01 then
      state_machine.state := 1;
    end_if;
1:  if state_machine.transition12 then
      state_machine.state := 2;
    elsif state_machine.transition13 then
      state_machine.state := 3;
    end_if;
2:  if state_machine.transition24 then
      state_machine.state := 4;
    end_if;
3:  if state_machine.transition31 then
      state_machine.state := 1;
    end_if;
4:  if state_machine.transition43 then
      state_machine.state := 3;
    end_if;
end_case;
```

Máquina de estados Logix en diagrama de función secuencial

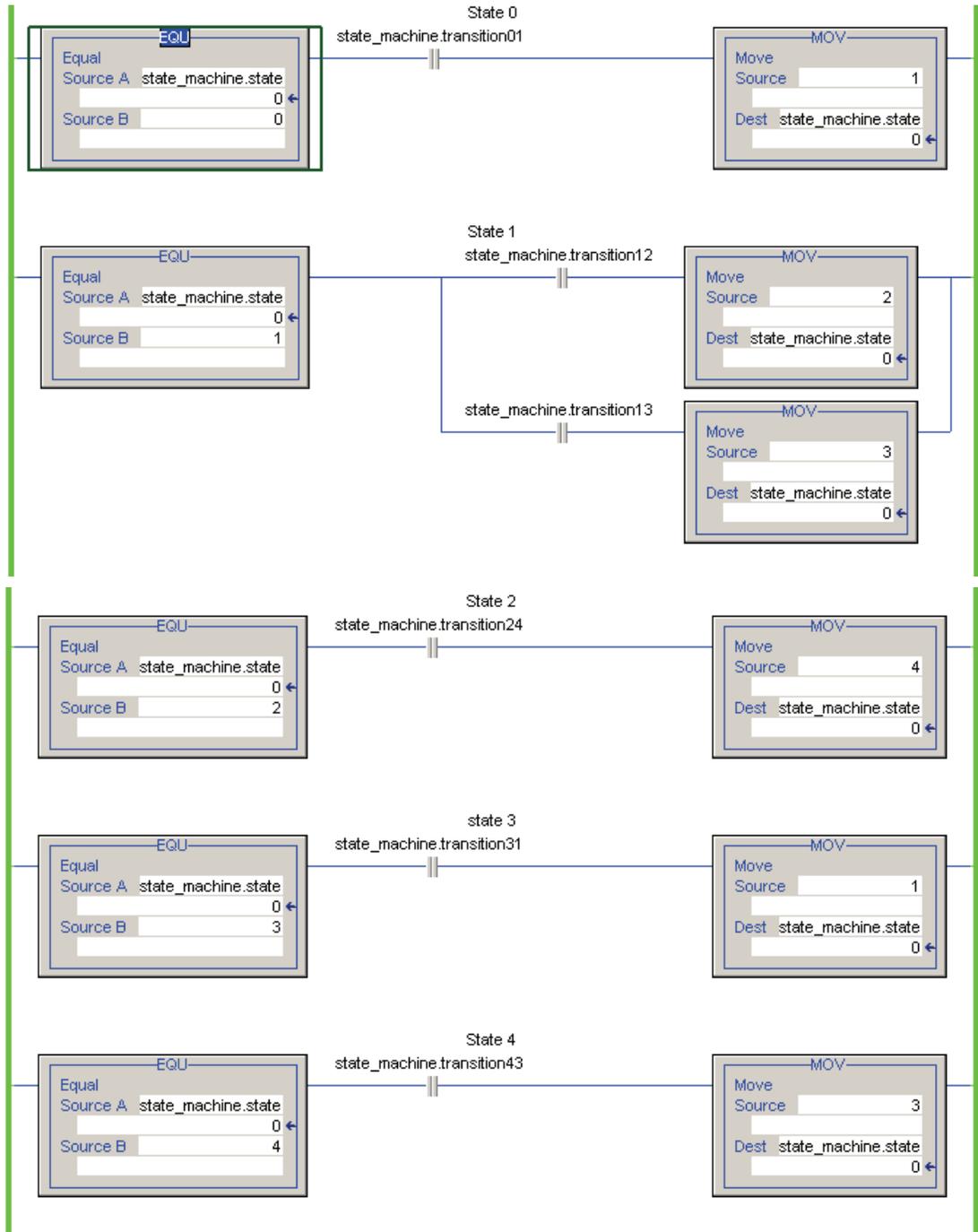
Logix proporciona un SFC gráfico como uno de sus suites de lenguajes estándar. A continuación se presenta la máquina de estados en SFC.

Implementación de la máquina de estados usando el diagrama SFC



Máquina de estados en diagrama de lógica de escalera

La captura de pantalla siguiente muestra cómo puede implementarse la máquina de estados en LD.



Cadenas

Definición de cadena en STEP 7

| Contents Of: 'Environment\Interface\STAT' | | | | |
|---|------------|---------|-----------------------------|--|
| Name | Data Type | Address | Initial Value | |
| transition12 | Bool | 172.2 | FALSE | |
| transition13 | Bool | 172.3 | FALSE | |
| transition24 | Bool | 172.4 | FALSE | |
| transition43 | Bool | 172.5 | FALSE | |
| transition31 | Bool | 172.6 | FALSE | |
| str | String[46] | 174.0 | 'This is an example string' | |
| table | Struct | 222.0 | | |

El encabezado de datos muestra cómo están definidas las cadenas. La longitud de la cadena se introduce entre corchetes [] después del tipo de datos de cadena. El valor inicial de la cadena se escribe en la columna “Initial Value”.

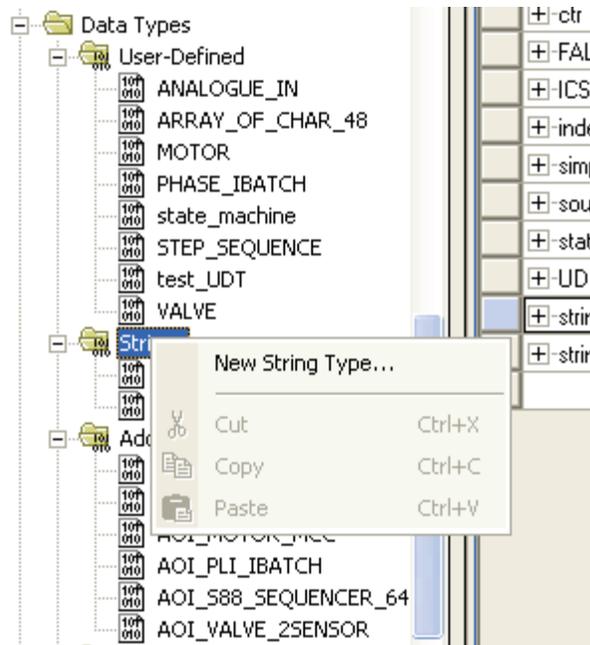
Es posible crear una matriz de cadenas, pero a cada una no se le podría dar un valor inicial. Una definición alternativa para evitar este problema se muestra en la entrada “table” en el encabezado de datos. “Table” es una estructura. El contenido de la estructura, no mostrado, son cinco instancias de string[46], y a cada uno se le ha dado un valor inicial.

Definición de cadenas en Logix

El extracto de la tabla de configuración de tags siguiente muestra cómo se definen las cadenas en Logix.

| | | |
|-------------------|--|--------------|
| +UDT_array | | test_UDT[10] |
| +string_of_82char | | STRING |
| +string_of_40char | | STRING 40 |

Si usted desea crear una cadena de una longitud diferente a la de la opción predeterminada de 82 caracteres, haga clic con el botón derecho del mouse en “strings” en su árbol de proyectos (como de muestra a continuación).



Seguidamente configure las propiedades como se indica a continuación.

Name:

Description:

Maximum Characters:

Members: Data Type Size: 52 byte(s)

| | Name | Data Type | Style | Description |
|-------------------------------------|------|-----------|---------|-------------|
| <input checked="" type="checkbox"/> | LEN | DINT | Decimal | |
| <input type="checkbox"/> | DATA | SINT[48] | ASCII | |

Después de hacerlo, podrá definir las instancias de su nuevo tipo.

| | | | | |
|--------------------|--|--|-----------|--|
| + string_of_8zchar | | | STRING | |
| + string_of_48char | | | STRING_48 | |

Con las instancias de tipo STRING o STRING_48, hay un campo LEN que se actualiza automáticamente cuando se introduce una constante de cadena o cuando la cadena es manipulada por instrucciones ASCII o STRING.

Variables temporales de STEP 7

Una de las categorías de variable en STEP 7 es la variable temporal. Éstas pueden crearse en cualquier bloque de organización, función o bloque de funciones.

Las variables temporales se usan para almacenamiento temporal local de valores intermedios y para punteros. Sólo existen mientras su bloque se está ejecutando y sus valores se pierden cuando el bloque termina.

Logix no tiene variables temporales. Todo almacenamiento es estático; es decir, los valores se retienen entre ejecuciones de código.

Si utiliza instrucciones Add-On, notará que es posible crear variables locales para una instrucción Add-On. Estas variables pueden usarse de la misma manera que las variables temporales.

Funciones

Si el programador de STEP 7 utiliza la lista de instrucciones, quizás tenga que desarrollar rutinas de bajo nivel cuya escritura es laboriosa y requieren pruebas cuidadosas. Las funciones son importantes porque el desarrollo de esas rutinas necesita hacerse solo una vez, tras lo cual, tanto el originador de la función como los otros programadores pueden hacer lo mismo en una fracción del tiempo.

Esta sección describe cómo pueden implementarse las funciones en Logix.

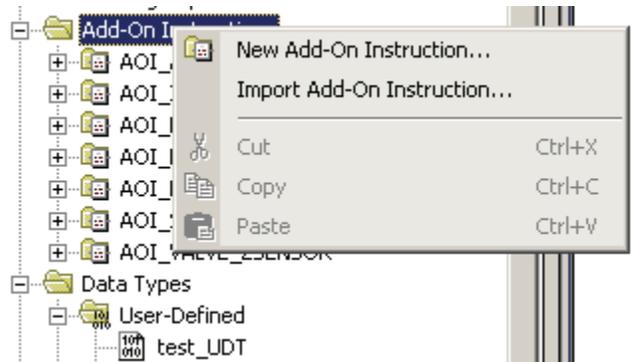
Función como instrucción Add-On en Logix

Las funciones y bloques de funciones de STEP 7 son similares en su estructura a la instrucción Add-On de Logix. La instrucción Add-On tiene los mismos tipos de parámetros que el FB (Input, Output e InOut), y tiene su propia área de datos. Una vez codificada y probada, una instrucción Add-On puede usarse desde cualquier lugar del programa y es suficientemente autónoma para exportarse a otros proyectos o colocarse en una biblioteca de código.

Ejemplo – una función de rampa

Este ejemplo toma una variable real y utiliza una rampa lineal desde su valor actual hasta un nuevo valor a una tasa especificada.

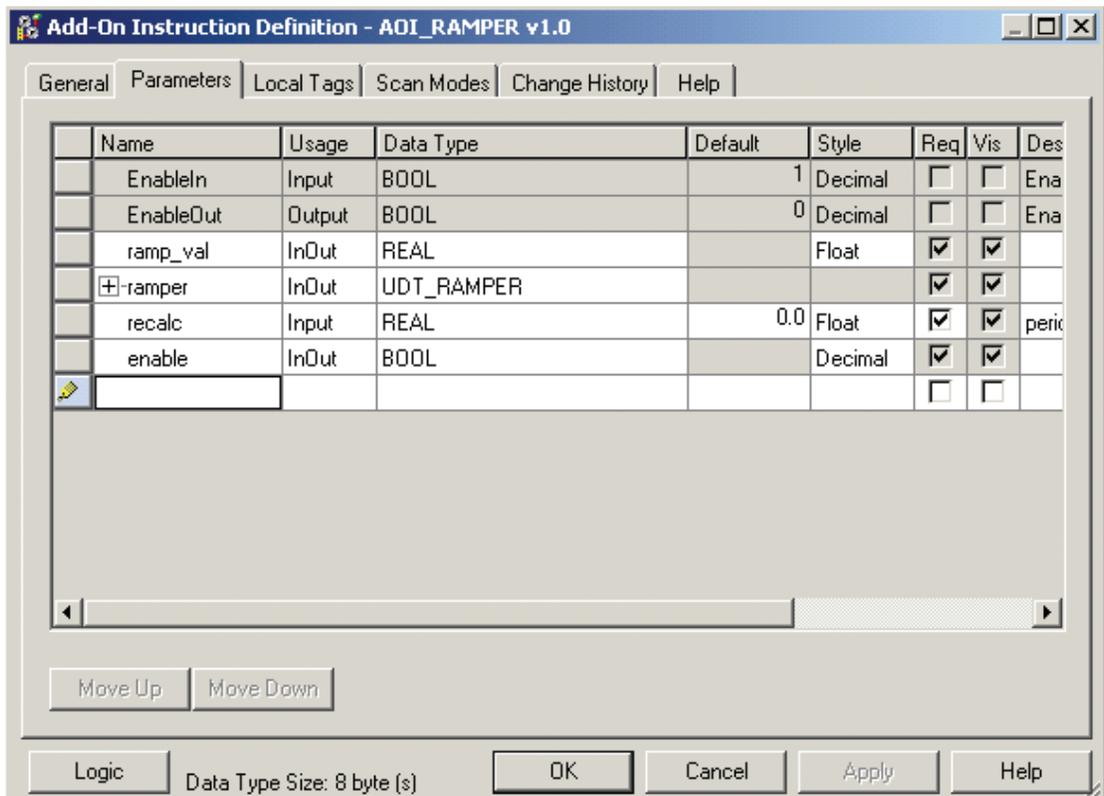
Vaya a la bifurcación Add-On Instructions de su árbol de proyectos y haga clic con el botón derecho del mouse en Add-On Instruction.



Aparece este formulario.

Introduzca el nombre de la instrucción Add-On y especifique el lenguaje en el cual se escribirá la sección de código.

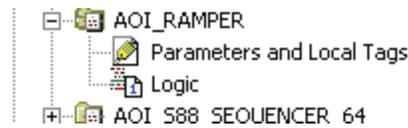
Seleccione la ficha Parameters.



Como en STEP 7, los parámetros Input son valores del programa a la instrucción Add-On, mientras que los parámetros Output son valores de la instrucción Add-On al programa, y los parámetros InOut son para variables que serán modificadas por la instrucción Add-On. Si tiene alguna estructura de datos, seleccione el tipo InOut ya que éstas se pasan por referencia y esto es más eficiente.

| Members: | | Data Type Size: 28 byte(s) | | |
|----------------|-----------|----------------------------|-------------------------------|--|
| Name | Data Type | Style | Description | |
| initial_output | REAL | Float | saved initial output | |
| increment | REAL | Float | calculated increment | |
| RAMP_RATE_ABS | REAL | Float | per second - (set always +ve) | |
| RAMP_TARGET | REAL | Float | final value - (set) | |
| change | REAL | Float | calculated change over ramp | |
| counter | DINT | Decimal | internal counter | |
| complete | BOOL | Decimal | ramminn is complete | |

En el árbol de proyectos para AOI_RAMPER hay una sección de lógica.



Ábrala para ver el código de esta instrucción Add-On.

```

// Ramps a real variable from its current value to a new value at a
// specified rate.

// Parameters:
//   ramp_val   - variable to be ramped
//   ramper     - instance of UDT UDT_RAMPER
//   recalcalc  - code recalculation period (s)
//   enable     - start signal

// To use - set the target value in ramper.RAMP_TARGET_ABS
//         - set the ramp rate in ramper.RAMP_RATE_ABS
//         - to Start the Ramper set "enable" parameter
//         - to pause the Ramper reset "ramper.enabled"
//         - to resume set "ramper.enabled"
//         - setting "enable" both starts and resets the ramper

// on completion, the UDT field "complete" is set and the UDT field
// "enabled" is reset

// when enable is set, initialise
if (enable & (enable xor ramper._enable)) then
    ramper.initial_output := ramp_val;
    ramper.change := ramper.RAMP_TARGET - ramp_val;
    ramper.increment := ramper.change / abs(ramper.change)
                        * ramper.RAMP_RATE_ABS * recalcalc;

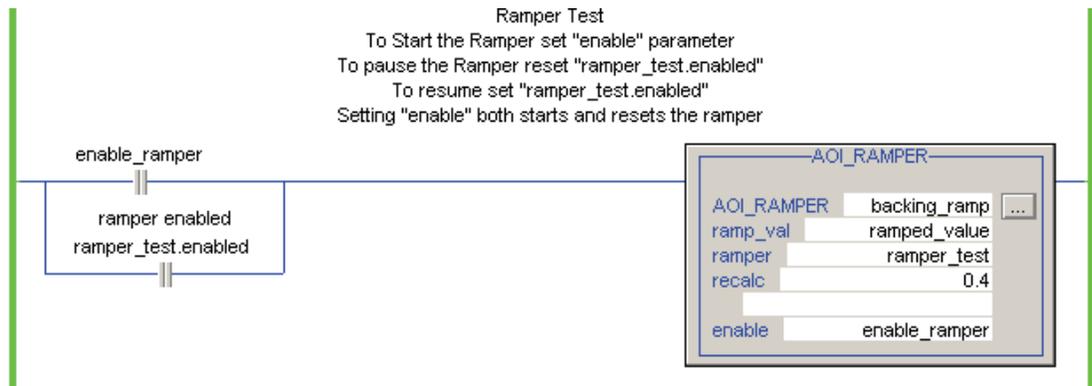
    ramper.counter := 0;
    ramper.complete := 0;
    enable := 0;
    ramper.enabled := 1;
end_if;

ramper._enable := enable;

// ramp calculations
if (ramper.enabled) then
    ramp_val := ramper.initial_output + (ramper.counter
                                        * ramper.increment);

    ramper.counter := ramper.counter + 1;
    if (abs(ramper.counter * ramper.increment)
        > abs(ramper.change)) then
        ramp_val := ramper.RAMP_TARGET;
        ramper.complete := 1;
        ramper.enabled := 0;
    end_if;
end_if;
    
```

La instrucción Add-On puede llamarse desde cualquier rutina.



Tenga presente que con las instrucciones Add-On necesitará crear un tag de tipo Add-On Instruction en un área de datos visible para la rutina. Esto se llama tag de respaldo.

Antes de escribir una instrucción Add-On, revise la ayuda para instrucciones en el software RSLogix 5000. Es posible que encuentre que hay una instrucción ya existente que cubrirá las necesidades. Esto se ilustrará en la siguiente sección.

Block Copy, COP y CPS

En STEP 7 es común usar la función del sistema SFC20 "BLKMOV" para copiar un bloque de datos entre diferentes ubicaciones.

```

CALL "BLKMOV"
SRCBLK := "Data_EMI".stepMsgs.step5
RET_VAL := #intVar
DSTBLK := "Data_EMI".actualStep
  
```

La instrucción copia la cadena desde la quinta ubicación en una matriz de cadenas a una cadena de destino.

A menudo deseamos copiar el elemento i-ésimo de una matriz, donde "i" puede variar a medida que se ejecuta el programa. "BLKMOV" no puede hacer esto.

El programador de STEP 7 escribiría una función para cumplir con este requisito.

```
// copy step number descriptor to SCADA display area (EM faceplate)
CALL "INDEXED_COPY"
indexSource := "Data_EMs".EM1.stepNumber
sourceRef   := "Data_EM1".stepMsgs
indexDest   := 1
destRef     := "Data_EM1".actualStep
recordLength:=8           // 32 bytes
```

En este caso, la copia se realiza entre dos matrices y los índices son definidos por indexSource e indexDest.

En Logix, la instrucción COP incorporada guardará todo el trabajo.



Puesto que las especificaciones de fuente y destino pueden incluir índices de matrices variables, COP cubrirá las necesidades. Es el equivalente de "INDEXED_COPY".

La instrucción CPS es igual que la instrucción COP, con una diferencia.



La instrucción no puede interrumpirse. Por lo tanto, los datos de fuente y destino permanecerán constantes durante la ejecución. Si desea mover datos que pueden cambiar, use CPS.

Como ejemplos cabe mencionar:

- copiar datos de entrada a un búfer, desde donde el programa operará con los datos.
- copiar tags consumidos a un búfer, desde donde el programa operará con los datos.

Expresiones matemáticas

En esta sección se describe cómo el programador de S7 puede realizar cálculos matemáticos en Logix. Se usará como ejemplo la expresión “ $\sqrt{\cos(x)^2 + \sin(x)^2}$ ”. El resultado de esta expresión siempre es exactamente 1; por lo tanto, es fácil verificar que obtiene la respuesta correcta.

STEP 7 – STL

El código matemático en STEP 7 STL es eficiente, pero quizás no suficientemente claro para una persona que no esté familiarizada con STL.

Network 1: Title:

```
calculate:
```

```
(SIN(x)^2 + COS(x)^2) ^0.5
```

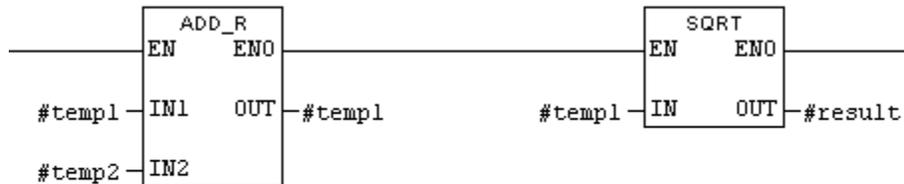
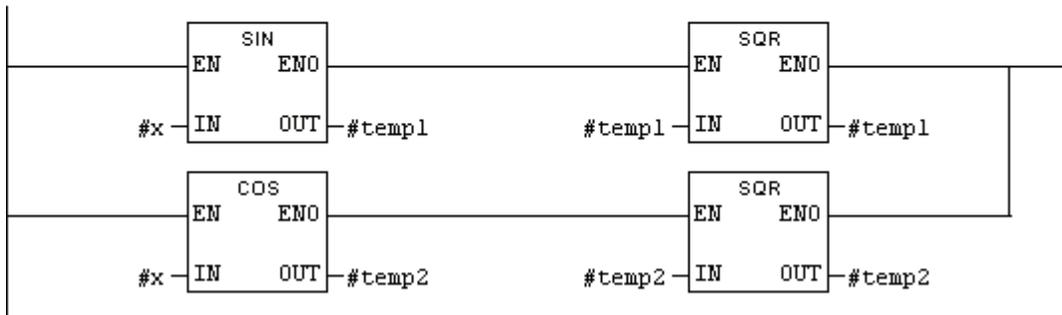
```
L      #x  
SIN  
SQR  
L      #x  
COS  
SQR  
+R  
SQRT  
T      #result
```

STEP 7 – LAD

La evaluación matemática en LAD sigue un patrón convencional de funciones combinadas.

Network 1: Title:

```
calculate:
(SIN(x)^2 + COS(x)^2) ^0.5
```



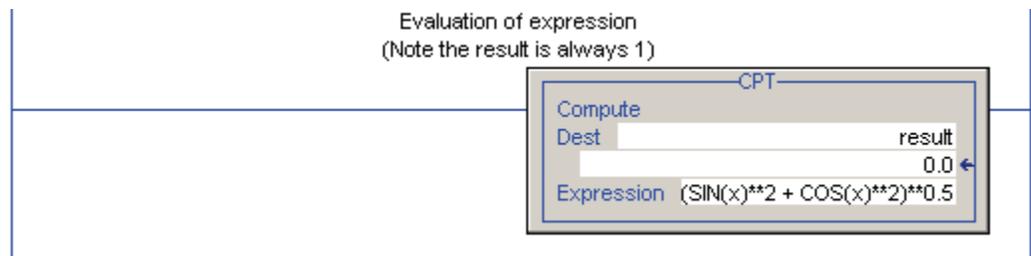
TEMP: REAL

Logix – ST

La expresión se introduce de la misma manera que con cualquier otro lenguaje de alto nivel.

```
// evaluation of mathematical expression in Structured Text
result := (SIN(x)**2 + COS(x)**2)**0.5;
```

Logix – LD



La instrucción CPT permite que la expresión se introduzca en forma de alto nivel, lo cual la mayoría de las personas podrá entender más fácilmente que una red (renglón) de instrucciones separadas.

STEP 7 – Función de usuario

Este bloque de funciones se ha escrito para que haga prácticamente lo mismo que la instrucción CPT de Logix.

```
CALL "CPT" , "Data_CPT"           FB119 / DB119
str  := "test_formulae".test_string15 P#DB16.DBX476.0
x    := 3.300000e+001
a    := 0.000000e+000
b    := 0.000000e+000
c    := 0.000000e+000
result:=#realVar
```

```
-- ((cos x)^2 + (sin x)^2) ^0.5 --
```

| IN | | OUT | |
|-------------|--|-----|-------------|
| +3.300e+001 | | | |
| +0.000e+000 | | | |
| +0.000e+000 | | | |
| +0.000e+000 | | | |
| | | | +1.000e+000 |

Ésta Lee y evalúa una cadena de expresiones almacenadas en un bloque de datos. Si se compara con la CPT de Logix, tiene una limitación: la expresión se escribe en notación polaca inversa, lo cual no será lo ideal para todos.

Los problemas principales de escribir un bloque de funciones como éste son que requiere tiempo y no es lo adecuado para programadores sin experiencia. Con Logix, la instrucción CPT está disponible para todos tan pronto se instala el software RSLogix 5000.

Verificación de tipo

Tanto con STEP 7 como con Logix, los parámetros para las funciones, bloques de funciones, instrucciones e instrucciones Add-On son estrictamente verificados por sus compiladores en cuanto al tipo.

Existen diferencias con las expresiones matemáticas.

Logix distingue entre valores numéricos y booleanos. El compilador rechazará las expresiones que combinen de manera ilógica valores numéricos y booleanos. Cuando encuentra expresiones de tipo numéricas mixtas, hará conversiones para producir un resultado del tipo de la variable de resultado declarada. Por lo tanto, interpretará * como multiplicación de números enteros si el resultado debe ser un número entero y como multiplicación real si el resultado debe ser un número real.

En STEP 7 debe especificarse el tipo de operación aritmética. Existen, por ejemplo *I (multiplique dos números enteros de 16 bits) *D (multiplique dos números enteros de 32 bits) y *R (multiplique dos números reales). El programador deberá asegurarse de que los dos números que son los operandos de una instrucción *R sean números reales. Si no lo son, el compilador no presentará ningún mensaje de queja, pero el resultado no tendrá sentido.

Conclusión

Los métodos Logix para programar expresiones matemáticas son más claros, y al separar el código matemático del resto de la lógica se simplificará la prueba y la validación.

Otros temas relacionados con la programación

Alcance de variables

Ésta es un área donde Logix difiere considerablemente de STEP 7.

Reglas para STEP 7

- Las variables temporales son invisibles fuera del bloque en el cual se declaran.
- Las variables estáticas globales son visibles en todo el programa.
- Las variables estáticas declaradas como datos de instancia para un bloque de funciones tienen un estado especial en el FB, pero no se puede obtener acceso a ellas desde otras partes del programa.

Reglas para Logix

La ejecución en Logix se divide en tareas. Cada tarea puede tener varios programas y cada programa puede tener varias rutinas. Cada programa puede tener su propia sección de tags.

- Los tags bajo el control del controlador son visibles en todas las rutinas de todos los programas.
- Los tags bajo el control del programa son visibles solamente en las rutinas del programa en el cual están definidos. Esto significa que si una rutina de un programa comparte datos con una rutina de otro programa, debe usar datos bajo el control del controlador.
- Los tags locales de la instrucción Add-On sólo son visibles para la lógica de dicha instrucción Add-On.

OB, tareas y secuenciación

Los bloques de organización, las tareas y la secuenciación se describen en el [Capítulo 2](#).

Un ejemplo mayor – módulo de control

Este ejemplo reúne algunos de los diferentes temas ilustrados en las secciones anteriores. El término “módulo de control” (CM) viene del influyente estándar de control de lotes S88. S88 ha fomentado que el diseño de software de controlador sea más “orientado a objetos”. Este módulo de control es para una válvula binaria. La instrucción Add-On es apta para este tipo de programación.

Componentes del CM

Son los siguientes:

- Un UDT llamado UDT_VALVE.
- Una instrucción Add-On llamada AOI_VALVE_2SENSOR
- Un nuevo programa bajo “task_02s” llamado “valves_callup”, que contiene una sección de tags del programa y una rutina.

Tipo datos de usuario Valve

El UDT se muestra a continuación.

Name:

Description:

Members: Data Type Size: 24 byte(s)

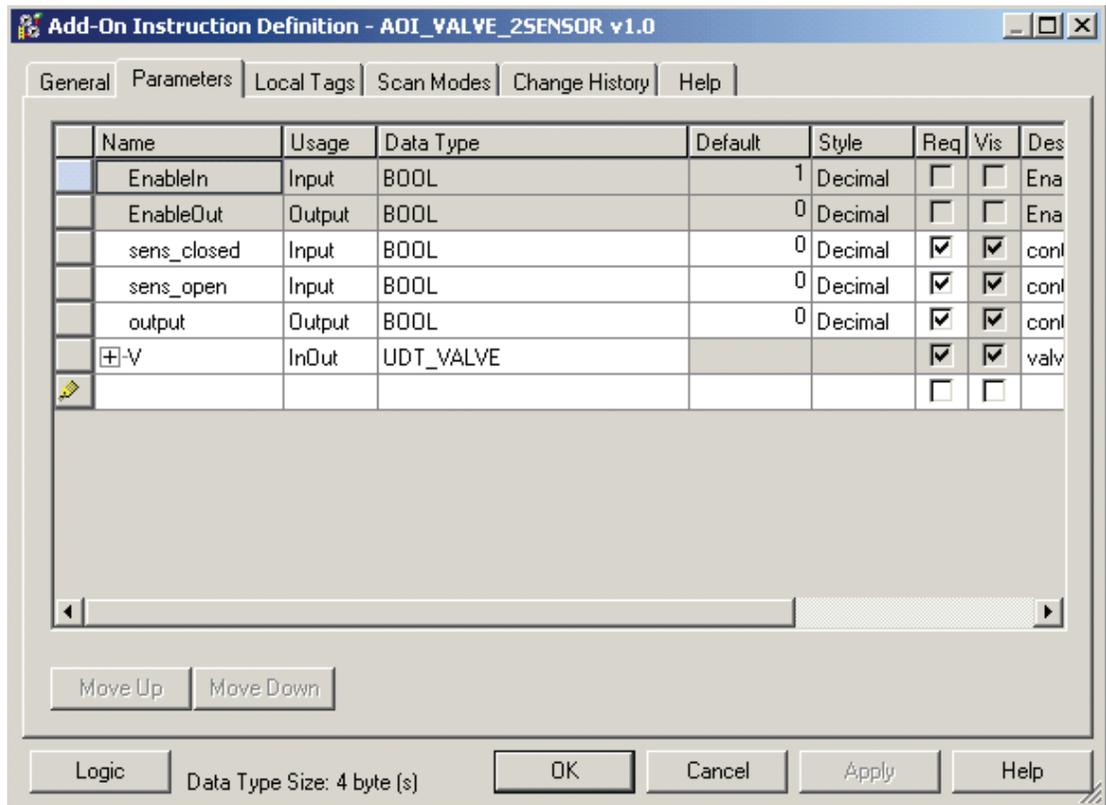
| | Name | Data Type | Style | Description |
|--|----------------|-----------|---------|---|
| | opening_preset | DINT | Decimal | max preset for opening |
| | closing_preset | DINT | Decimal | max preset for closing |
| | state | DINT | Decimal | state of valve (for internal logic) |
| | state_saved | DINT | Decimal | for evaluation of edge |
| | timecount | DINT | Decimal | valve timer |
| | auto | BOOL | Decimal | auto mode (set from SCADA) |
| | manual | BOOL | Decimal | manual mode (set from SCADA) |
| | closed | BOOL | Decimal | state of valve |
| | open | BOOL | Decimal | state of valve |
| | fault_closing | BOOL | Decimal | closed sensor feedback not received |
| | fault_opening | BOOL | Decimal | open sensor feedback not received |
| | fault_sensors | BOOL | Decimal | sensors and logical state of valve do not agree |
| | acquired | BOOL | Decimal | acquired by EM |
| | interlocked | BOOL | Decimal | interlocked - de-energise |
| | fail_open | BOOL | Decimal | property - fails open |

Crear el UDT debe ser el primer paso; éste incluye todos los datos necesarios para modelar la válvula.

La instrucción Add-On

Parámetros de la instrucción Add-On

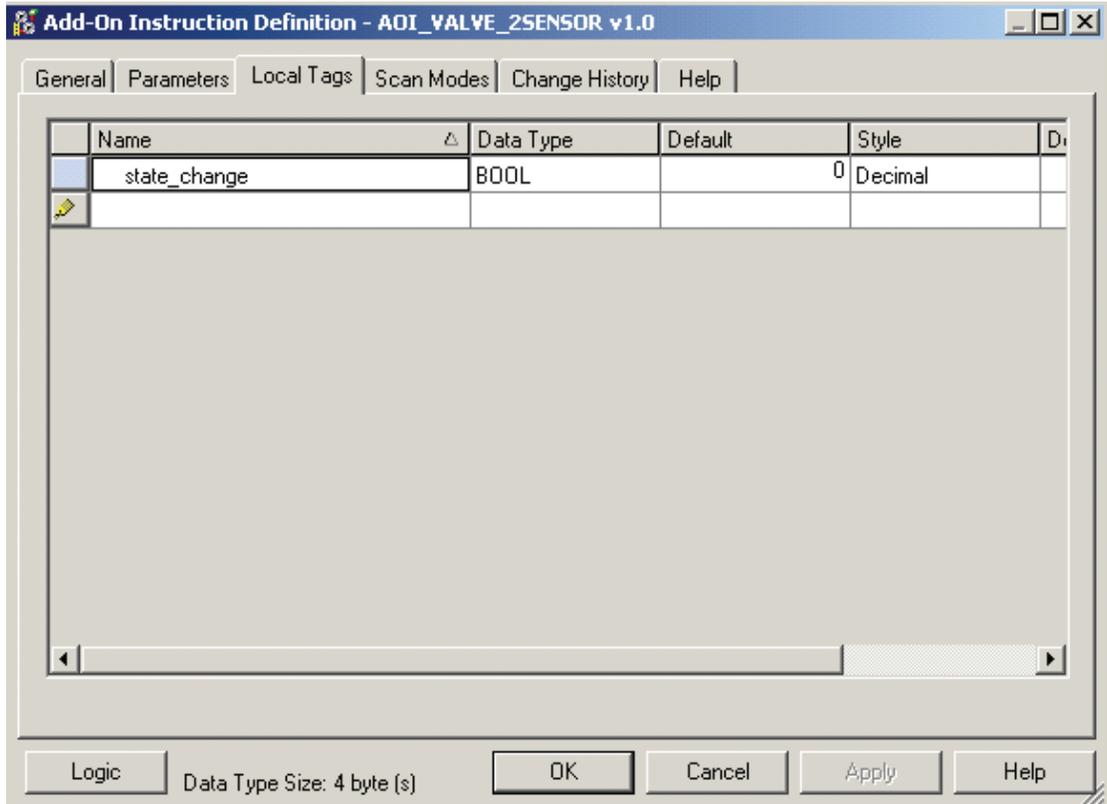
La captura de pantalla muestra la pantalla de configuración de parámetros.



Los parámetros que se han añadido son las E/S para la válvula y un objeto de tipo “UDT_VALVE”. “V” debe ser un parámetro InOut.

Datos locales de la instrucción Add-On

La captura de pantalla siguiente muestra la página de configuración para los datos locales de la instrucción Add-On.



Lógica de la instrucción Add-On

La siguiente captura de pantalla muestra la lógica para esta instrucción Add-On.

```

// Control Module Valve_2sensor
// -----
// Implements logic for a valve with an open and a closed sensor and one output

// See UDT Valve for data structure.

// Note the open/close command V.open_command must be set or reset externally
// and then left until the next activation is required. Do not continuously
// hold the flag set or reset.

// increment timer counter
V.timecount := V.timecount + 1;

// evaluate change of state (state machine)
state_change := V.state <> V.state_saved;
V.state_saved := V.state;

// set output
output := (V.fail_open xor V.open_command) and not
           (V.interlocked or V.faulted);

// valve is faulted
V.faulted := V.fault_opening or V.fault_closing or V.fault_sensors;

// action on fault or interlock
if V.faulted or V.interlocked then
  if V.fail_open then
    V.state := 3;
    V.open_command := 1;
  else
    V.state := 0;
    V.open_command := 0;
  end_if;
end_if;

// state machine:
// the state machine does not set outputs - it monitors inputs
// to set status and faults.
case V.state of
  // state 0 - valve is closed - wait for open command
  0: V.closed := 1;
     V.open := 0;
     if (V.open_Command) then
       V.state := 1;
     // fault sensors
     else
       V.fault_sensors := (not sens_closed) or (sens_open);
     end_if;
  // state 1 -
  1: V.state := 2;
  // state 2 - waiting for open sensor
  2: if (sens_open & not sens_closed) then
       V.state := 3;

```

```

        // possible close command while waiting to open
        elsif not V.open_command then
            V.state := 0;
        // fault opening
        else
            V.fault_opening := (V.timecount > V.opening_preset);
        end_if;
// state 3 - open - wait for close command
3: V.closed := 0;
   V.open := 1;
   if (not V.open_command) then
       V.state := 4;
   // fault sensors
   else
       V.fault_sensors := (sens_closed) or (not sens_open);
   end_if;
// state 4 -
4: V.state := 5;
// state 5 - wait for closed sensor
5: if (sens_closed & not sens_open) then
    V.state := 0;
    // possible open command while waiting to close
    elsif V.open_command then
        V.state := 3;
    // fault closing
    else
        V.fault_closing := (V.timecount > V.closing_preset);
    end_if;
else;
end_case;
// end state machine

// reset timer if change of state
if (state_change) then V.timecount := 0;
end_if;

// external fault reset
if (V.clear_faults) then
    V.fault_opening := 0;
    V.fault_closing := 0;
    V.fault_sensors := 0;
    V.clear_faults := 0;
end_if;

```

Los tags referidos en esta lógica son todos parámetros o tags locales. Esto significa que la instrucción Add-On podría usarse en cualquier programa (siempre que la válvula UDT también esté presente).

Llamada

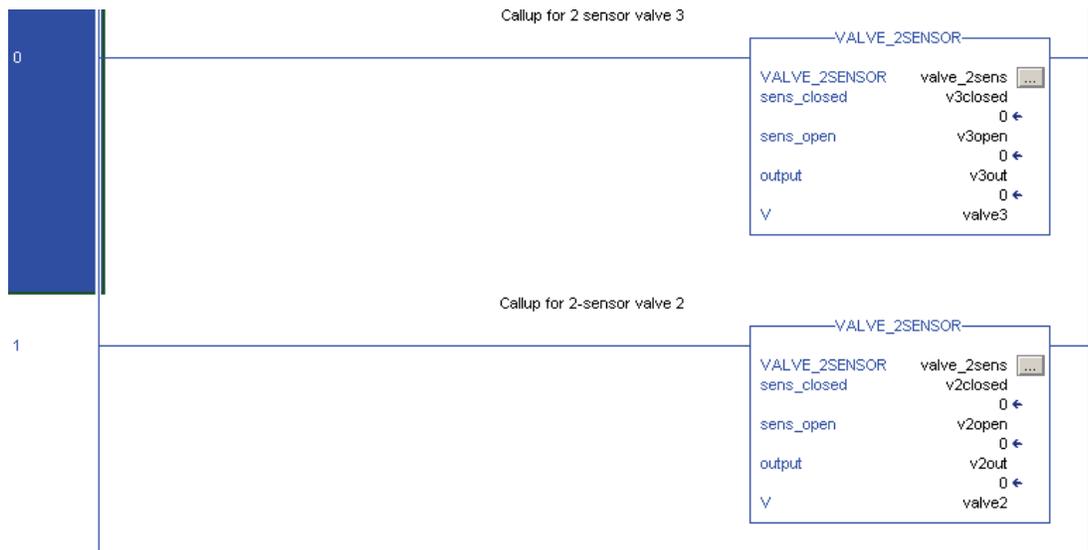
Tanto el código de llamada como las instancias de la válvula UDT están ubicados en el programa “valves_callup”, el cual se ejecuta bajo task_02s. La frecuencia con la cual se ejecuta el código de llamada depende de la aplicación y del tamaño de la válvula.

La siguiente captura de pantalla muestra las instancias de datos.

| Name | Alias For | Base Tag | Data Type |
|--------------|-----------|----------|---------------|
| +valve_2sens | | | valve_2sensor |
| +valve1 | | | Valve |
| +valve2 | | | Valve |
| +valve3 | | | Valve |
| +valve4 | | | Valve |
| +valve5 | | | Valve |

Añada una instancia del tipo Valve para cada válvula física. El primer tag es el “tag de respaldo” requerido para la instrucción Add-On.

La siguiente captura de pantalla muestra el código de llamada.



Llame a la instrucción Add-On una vez por cada válvula. Los parámetros son los tags de E/S para el solenoide y los sensores de la válvula, y la instancia de UDT "valve".

Los tags de E/S sólo aparecerán en la llamada a la instrucción Add-On. No se usarán en ningún otro lugar en el programa. Esto, aparte de ser más ordenado desde el punto de vista de estructura de software, evita los problemas que surgen de la actualización asíncrona de E/S.

Recuerde que con los controladores Logix, las E/S se escanean de manera asíncrona.

Errores comunes que se cometen al convertir a Logix

Introducción

El objetivo de esta sección es señalar algunos de los errores de diseño y programación que los usuarios de S7 cometen a menudo al convertir aplicaciones a Logix. Estos errores se han identificado mediante el examen de programas convertidos de STEP 7 a Logix.

| Tema | Página |
|--|--------|
| No seleccionar el hardware apropiado | 129 |
| Subestimar el impacto de la secuenciación de tareas | 130 |
| Realizar traducción en lugar de conversión | 130 |
| No usar los lenguajes Logix más apropiados | 130 |
| Implementación de tipos de datos incorrectos – DINT frente a INT | 131 |
| Código de usuario que emula instrucciones existentes | 132 |
| Uso incorrecto de COP, MOV y CPS | 133 |
| Uso incorrecto de CPT | 133 |
| No manejar las cadenas de una manera óptima | 133 |
| Uso frecuente de saltos | 133 |
| No usar tag de alias | 133 |

Los errores de programación se clasifican en estas dos categorías:

- Programación que conduce a una reducción de la eficiencia del controlador.
- Programación que conduce a un sistema de control difícil de entender, mantener y desarrollar.

En la mayoría de los casos, una codificación eficiente mejorará también la facilidad de lectura y la modularidad de su programa. Adicionalmente, el mejorar la estructura del programa también debe hacerlo más eficiente.

No seleccionar el hardware apropiado

Este capítulo se concentra principalmente en el software. Recuerde, sin embargo, que la selección correcta de hardware es un requisito para una operación satisfactoria. Es posible que el número de controladores y racks no sea igual que para el sistema S7 equivalente.

Lea el [Capítulo 1](#) y el [Apéndice A](#) para obtener más información sobre el hardware. Puede encontrar más información en los [Apéndices A y B](#).

Subestimar el impacto de la secuenciación de tareas

En el área de secuenciación e interrupciones no hay mucha diferencia en la capacidad de los dos sistemas. Sin embargo, en el entorno Logix, la secuenciación se fomenta de manera más activa.

Es bastante común que los programadores de STEP 7 desatiendan la secuenciación al trabajar con los controladores Logix. Consulte el [Capítulo 2](#) para obtener información más detallada sobre la secuenciación en Logix.

Realizar traducción en lugar de conversión

Es un error común traducir línea por línea un programa STEP 7 a Logix.

En lugar de ello, se requiere un proceso más completo, el cual se describe como conversión. Éste abarca selección de lenguajes, secuenciación y selección de rutinas de código.

Al convertir en lugar de traducir los programas STEP 7, usted aprovechará mejor la capacidad de su sistema Logix.

No usar los lenguajes Logix más apropiados

Los programadores a menudo no consideran el uso de otros lenguajes, aparte del de lógica de escalera.

Lea el [Capítulo 2](#) para obtener información sobre cómo elegir un lenguaje Logix, y el [Capítulo 4](#) para obtener ejemplos de código STEP 7 traducido a Logix.

Implementación de tipos de datos incorrectos – DINT frente a INT

Generalmente es aconsejable usar DINT en lugar de INT.

El siguiente ejemplo muestra una adición de dos DINT comparada con la adición de dos INT.

Añadir DINT

```
// add two DINTs and assign to a third DINT
for index := 0 to 999 do
    result_DINT := operandA_DINT + operandB_DINT;
end_for;
```

Añadir INT

```
// add two INTs and assign to a third INT
for index := 0 to 999 do
    result_INT := operandA_INT + operandB_INT;
end_for;
```

Resultados de temporización

La tabla muestra los tiempos relativos (un número menor es más rápido). Estos números se proporcionan sólo para fines de comparación con otros números en la tabla. No deben compararse con entradas en otras tablas.

| Método | Tiempos relativos |
|-----------------------------|-------------------|
| Añadir DINT con ST For Loop | 53 |
| Añadir INT con ST For Loop | 100 |

Para fines de comparación, la misma prueba se realizó con un controlador S7. En este caso, los resultados fueron idénticos para DINT e INT.

La conclusión es que debe usarse DINT para todo trabajo con números enteros en Logix. Sólo use INT o SINT si está interconectándose con un sistema externo que requiere el uso de INT o SINT.

Código de usuario que emula instrucciones existentes

Los programadores a menudo escriben código de usuario cuando una instrucción existente puede cubrir las necesidades. Como ejemplo, compare la copia de una matriz mediante código de usuario con la instrucción COP.

Código de usuario

```
for index := 0 to 99 do
    target_DINT[index] := source_DINT[index];
end_for;
```

Instrucción COP

```
cop(source_DINT[0], target_DINT[0], 100);
```

A continuación se proporcionan los tiempos relativos de los dos métodos. Nuevamente, estos números se proporcionan sólo para fines de comparación con otros números en la tabla. No deben compararse con entradas en otras tablas.

| Método | Tiempo relativo |
|--|-----------------|
| Copiar matriz de DINT con texto estructurado | 100 |
| Copiar matriz de DINT con COP | 18 |

Para realizar operaciones como copia de matrices, se usan las funciones de biblioteca de STEP 7 escritas en la lista de instrucciones. Si la función de biblioteca no hace lo que se requiere, puede escribirse una nueva. Las funciones escritas pueden ser casi tan eficientes como las que proporciona STEP 7.

Sin embargo, en Logix es imposible que un programador escriba una función de copia tan eficiente como la incorporada en COP. La lección para los programadores de S7 es revisar atentamente la ayuda para instrucciones en el software RSLogix 5000 antes de hacerlo por sí mismos.

Uso incorrecto de COP, MOV y CPS

MOV copia un valor simple (inmediato o de tag) a un tag simple tipo DINT, INT, SINT o REAL. COP puede hacer lo mismo que MOV (la fuente no puede ser un valor inmediato), pero su uso más importante es copiar tipos de datos complejos.

Sería un error de programación menor usar COP para copiar tipos de datos simples.

Un error que se comete a menudo es usar múltiples MOV para copiar una estructura de datos cuando se podría usar una sola instrucción COP.

Si sus datos de fuente pudieran cambiar durante la operación de copia debido a actualizaciones de E/S asíncronas, utilice CPS. Esta instrucción no puede ser interrumpida; por lo tanto, los datos de fuente permanecerán constantes durante la copia.

Uso incorrecto de CPT

En Logix, la instrucción CPT puede usarse para evaluar expresiones. La expresión se introduce en uno de los campos de la instrucción. Esto es muy conveniente.

Sin embargo, la instrucción CPT sólo debe usarse si se requiriera más de una instrucción aritmética para evaluar la expresión. Si una sola instrucción fuera suficiente, ésta será más rápida que CPT.

Puede leer más acerca de la instrucción CPT en el [Capítulo 4](#).

No manejar las cadenas de una manera óptima

Si desea definir un nuevo tipo de cadena como, por ejemplo, con un número diferente de caracteres que los 82 caracteres de la opción predeterminada, sería un error crear un nuevo 'tipo de datos de usuario'. En lugar de ello, cree un nuevo tipo de datos de cadena. La ventaja de hacerlo de esta manera es que el campo 'LEN' se actualizará automáticamente a medida que cambie la longitud de la cadena.

Uso frecuente de saltos

En Logix, los saltos sólo pueden ocurrir en la lógica de escalera. Se recomienda usar la instrucción JMP con moderación. Los saltos en la lógica de escalera generalmente dificultan la lectura del programa.

No usar tag de alias

Recuerde crear tags de alias para los tags de E/S que el software RSLogix 5000 crea para usted. Estos facilitarán la lectura del programa. Vea el [Capítulo 2](#).

Notas:

Glosario de S7 a Logix

Introducción

Este capítulo proporciona un glosario de términos de S7 y sus equivalentes en Logix.

Terminología de hardware

| Término de S7 | Definición | Término de Logix | Definición |
|------------------------------|---|--|---|
| Procesador de comunicaciones | Módulo de comunicaciones | Puente | |
| Controlador | El controlador | Controlador | |
| CPU | Unidad central de procesamiento | CPU o controlador | |
| CPU a prueba de fallos | La CPU 315F-2 DP implementa la versión PROFISAFE de DP | GuardLogix | L61S, L62S, L63S |
| Ethernet industrial | Versión Siemens de Ethernet | EtherNet/IP ControlNet | Ambas tienen la misma funcionalidad (o mejor) que Ethernet industrial |
| MPI | Interface multipuntos – un bus en serie | Serie | Protocolos DF1 o DH485 |
| Controlador programable | | Controlador o PAC | |
| PROFIBUS DP | Bus de campo de uso común | EtherNet/IP ControlNet DeviceNet | |
| PROFIBUS PA | Variación de Profibus que se especializa en la automatización de procesos | Igual que Profibus DP | |
| PROFINET | Profibus por Ethernet | EtherNet/IP | |
| PROFISAFE | Versión a prueba de fallos de PROFIBUS DP | GuardLogix | |
| S7-200 | Controladores de la gama inferior | MicroLogix | |
| S7-300 | Controladores de gama media | CompactLogix | |
| S7-400 | Controladores de la gama superior | ControlLogix | |
| SIMATIC | Denominación comercial de productos de automatización Siemens | Logix | |

Terminología de software

| Término de S7 | Definición | Término Logix más cercano | Definición |
|--------------------------|---|--|---|
| Acumulador | Usado en STL | N/A | En los lenguajes Logix no hay necesidad de acceder a estructuras de bajo nivel de la CPU |
| AR1, AR2 | Registros de punteros | N/A | En los lenguajes Logix no hay necesidad de acceder a estructuras de bajo nivel de la CPU |
| Matriz | Sintaxis ARRAY[0...7] OF REAL | Matriz | Sintaxis REAL[8] El indexado siempre comienza en 0 |
| Memoria de bits | Direcciones M... | N/A | Se usan tags |
| Transferencia en bloques | Copia bloque de datos. SFC20 BLK_MOV | COP | Instrucción (use MOV para una variable simple) |
| BOOL | | BOOL | |
| BYTE | Palabra de 8 bits | SINT | Uso no recomendado (es más lento que DINT) excepto cuando es necesario (por ejemplo, caracteres de la cadena) |
| CFC | Lenguaje de control de procesos opcional | FBD | Lenguaje de bloque de funciones estándar. |
| CHAR | Byte como carácter | SINT | |
| Cycle_Execution | OB1 – Se ejecuta continuamente | Tarea continua | Se ejecuta continuamente |
| Bloque de datos | Unidad de memoria de datos estáticos | Base de datos de tags bajo el control del controlador. o base de datos de tags bajo de control del programa | Global visible dentro del programa al cual está vinculada la base de datos |
| DINT | Número entero doble | DINT | Número entero doble |
| DWORD | Palabra de 32 bits | DINT | |
| FBD | Diagrama de bloques de funciones | FBD | Diagrama de bloques de funciones |
| Función | Unidad del programa con memoria temporal pero no memoria estática | Rutina AOI (instrucción Add-On) | Estos dos podrían corresponder a una función |
| Bloque de funciones | Unidad del programa con memoria temporal y memoria estática | Rutina AOI (instrucción Add-On) Programa | Todos estos podrían corresponder a un bloque de funciones |
| GRAPH | Lenguaje gráfico opcional | Diagrama de función secuencial | Lenguaje gráfico estándar |
| HW Config | Configuración de hardware – componente de STEP 7 | Configuración de E/S | Bifurcación de Controller Organiser |
| INT | Número entero | INT | Uso no recomendado (es más lento que DINT) |
| Interrupt_Execution | Ejecuta periódicamente un OB | Tarea periódica | Ejecuta periódicamente una tarea |
| LAD | Lógica de escalera | LD | Lógica de escalera |
| Biblioteca | Funciones del sistema | GSV, SSV | Instrucciones – Get System Value Set System Value |

| Término de S7 | Definición | Término Logix más cercano | Definición |
|------------------------|---|----------------------------------|--|
| NetPro | Configurador de red | N/A | Parte de la bifurcación I/O Configuration del Controller Organiser. |
| Bloque de organización | Unidad del programa llamada por el sistema operativo | Tarea | Unidad del programa llamada por el sistema operativo |
| Puntero | Puntero de datos usado en STL | N/A | Use matrices |
| REAL | Número de punto flotante (coma flotante) de 32 bits | REAL | Número de punto flotante (coma flotante) de 32 bits |
| SCL | Lenguaje de alto nivel opcional | Texto estructurado | Lenguaje estándar |
| Simatic Manager | Componente de STEP 7 | Controller Organiser | Componente de RSLogix 5000 |
| STEP 7 | Software de desarrollo y monitoreo de S7 | RSLogix 5000 | Software de desarrollo y monitoreo de Logix |
| STL | Lista de instrucciones | N/A | Use texto estructurado, lógica de escalera o diagrama de función secuencial. |
| STRING | Secuencia de caracteres. Longitud predeterminada de 254 | STRING | Secuencia de SINT. Longitud predeterminada de 82. El objeto de cadena también contiene su longitud como property.LEN |
| STRUCT | Recolección de datos sin tipo | N/A | En Logix una estructura es una instancia de tipo (UDT) |
| Símbolo | Nombre de la dirección de la memoria de datos | Tag | Tag define la estructura de la variable y reserva la memoria |
| Memoria temporal | Memoria creada en la pila en tiempo de ejecución | N/A | Use tags |
| WORD | Palabra de 16 bits | INT | |
| UDT | Tipo de datos de usuario | UDT | Tipo de datos de usuario |

Notas:

Componentes de S7 300 y S7 400 y equivalentes de RA

Introducción

Este apéndice presenta una lista de los productos Siemens y sus equivalentes de Rockwell Automation.

| Tema | Página |
|---|---------------|
| CPU S7 300 compactas | 140 |
| CPU S7 300 estándar | 140 |
| CPU S7 300 de alta tecnología | 141 |
| CPU S7 300 a prueba de fallos | 141 |
| Módulos de entradas digitales S7 300 | 142 |
| Módulos de salidas digitales S7 300 | 142 |
| Módulos de salidas de relés S7 300 | 143 |
| Módulos digitales combinados S7 300 | 143 |
| Módulos de entradas analógicas S7 300 | 144 |
| Módulos de salidas analógicas S7 300 | 145 |
| Módulos analógicos combinados S7 300 | 145 |
| Módulos de salidas analógicas S7 300 | 146 |
| Controladores redundantes y a prueba de fallo | 146 |
| Módulos de entradas digitales | 147 |
| Módulos de salida digital | 147 |
| Módulos de entradas analógicas | 147 |
| Módulos de salidas analógicas | 148 |

CPU S7 300 compactas

| Número de catálogo Siemens | Referencia corta de Siemens | Memoria | Puertos de com. | | | Tamaño máximo de MMC | E/S incorporadas | | | | Solución de RA |
|-------------------------------|-----------------------------|-------------------|-----------------|----|-----------|----------------------|------------------|----|----|----|-------------------------------|
| | | | MPI | DP | En serie | | DI | DO | AI | AO | |
| 6ES7 312-5BE0x-xxxx | S7-312C | 32 K | S | N | N | 4 MB | 10 | 6 | | | 1769-L31 + Compact I/O ML1500 |
| 6ES7 313-5BF0x-xxxx | S7-313C | 64 K Sí No No | S | N | N | 8 MB | 24 | 16 | 4 | 2 | 1769-L31 + Compact I/O ML1500 |
| 6ES7 313-6BF0x-xxxx | S7-313C- PTP | 64 K | S | N | RS422/485 | 8 MB | 16 | 16 | | | 1769-L31 + Compact I/O ML1500 |
| 6ES7 313-6CF0x-xxxx | S7-313C- DP | 64 K | S | S | N | 8 MB | 16 | 16 | | | 1769-L31 + Compact I/O ML1500 |
| 6ES7 314-6BG0x-xxxx | S7-314C- PTP | 96 K | S | N | RS422/485 | 8 MB | 24 | 16 | 4 | 2 | 1769-L31 + Compact I/O ML1500 |
| 6ES7 314-6CG0x-xxxx | S7-314C- DP | 96K Sí Sí No 8 MB | S | S | N | 8 MB | 24 | 16 | 4 | 2 | 1769-L31 + Compact I/O ML1500 |

CPU S7 300 estándar

| Número de catálogo Siemens | Referencia corta de Siemens | Memoria | Puertos de com. | | | Tamaño de memoria de carga máxima (RAM) | Solución de RA |
|-------------------------------|-----------------------------|---------|-----------------|----|----|---|-----------------------|
| | | | MPI | DP | PN | | |
| 6ES7 312-1AE1x-xxxx | S7-312 | 32 K | S | N | N | 4 MB | 1769-L31 |
| 6ES7 314-1AG1x-xxxx | S7-314 | 96 K | S | N | N | 8 MB | 1769-L31 |
| 6ES7 315-2AG1x-xxxx | S7-315-2 DP | 128 K | S | S | N | 8 MB | 1769-L3xE o 1769-L3xC |
| 6ES7 315-2EH1x-xxxx | S7-315-2 PN/DP | 256 K | S | S | S | 8 MB | 1769-L3xE o 1769-L3xC |

| Número de catálogo Siemens | Referencia corta de Siemens | Memoria | Puertos de com. | | | Tamaño de memoria de carga máxima (RAM) | Solución de RA |
|-------------------------------|-----------------------------|---------|-----------------|----|----|---|-----------------------|
| | | | MPI | DP | PN | | |
| 6ES7 317-2AJ1x-xxxx | S7-317-2 DP | 512 K | S | S | N | 8 MB | 1769-L3xE o 1769-L3xC |
| 6ES7 317-2EK1x-xxxx | S7-317-2 PN/DP | 1 MB | S | S | S | 8 MB | 1769-L3xE o 1769-L3xC |
| 6ES7 319-3E0x-xxxx | S7-319-3 PN/DP | 1.4 MB | S | S | S | 8 MB | 1769-L3xE o 1769-L3xC |

CPU S7 300 de alta ecnología

| Número de catálogo Siemens | Referencia corta de Siemens | Memoria | Puertos de com. | | | Tamaño de memoria de carga máxima (RAM) | Solución de RA |
|-------------------------------|-----------------------------|---------|-----------------|----|----|---|----------------|
| | | | MPI | DP | PN | | |
| 6ES7 315-6TG1x-xxxx | S7-315T-2 DP | 128 K | S | S | S | 4 u 8 MB | 1768-L43 |
| 6ES7 317-6TJ1x-xxxx | S7-317T-2 DP | 512 K | S | S | S | 4 u 8 MB | 1768-L43 |

CPU S7 300 a prueba de fallos

| Número de catálogo Siemens | Referencia corta de Siemens | Memoria | Puertos de com. | | | Tamaño de memoria de carga máxima (RAM) | Solución ControlLogix de RA |
|-------------------------------|-----------------------------|---------|-----------------|----|----|---|-----------------------------|
| | | | MPI | DP | PN | | |
| 6ES7 315-6FF1x-xxxx | S7-315F-2 DP | 192 K | S | S | N | 8 MB | GuardLogix o SmartGuard 600 |
| 6ES7 315-2FH1x-xxxx | S7-315F-2 PN/DP | 256 K | S | S | S | 8 MB | GuardLogix o SmartGuard 600 |
| 6ES7 317-6FF0x-xxxx | S7-317F-2 DP | 1 MB | S | S | N | 8 MB | GuardLogix o SmartGuard 600 |
| 6ES7 317-2FK1x-xxxx | S7-317F-2 PN/DP | 1 MB | S | S | S | 8 MB | GuardLogix o SmartGuard 600 |

Módulos de entradas digitales S7 300

| Número de catálogo Siemens | Conector frontal | Puntos | Rango | Solución de RA | Comentarios |
|----------------------------|------------------|--------|-----------------|-------------------------|-------------------------------|
| 6ES7 321-1BH0x-xxxx | 20 pines | 16 | 24 VCC | 1769-IQ16 1769-IQ16F | |
| 6ES7 321-1BH5x-xxxx | 20 pines | 16 | 24 VCC | 1769-IQ16 1769-IQ16F | |
| 6ES7 321-1BL0x-xxxx | 40 pines | 32 | 24 VCC | 1769-IQ32 1769-IQ32T | |
| 6ES7 321-1CH0x-xxxx | 40 pines | 16 | 24 ... 48 V | n/a | |
| 6ES7 321-1CH2x-xxxx | 20 pines | 16 | 48 ... 125 VCC | n/a | |
| 6ES7 321-1BH1x-xxxx | 20 pines | 16 | 24 VCC | 1769-IQ16 1769-IQ16F | |
| 6ES7 321-7BH0x-xxxx | 20 pines | 16 | 24 VCC | 1769-IQ16 1769-IQ16F | |
| 6ES7 321-1FH0x-xxxx | 20 pines | 16 | 120 ... 230 VCA | 1769-IA16 | 1769-IA16 sólo admite 120 VCA |
| 6ES7 321-1FF0x-xxxx | 20 pines | 8 | 120 ... 230 VCA | 1769-IM12 | 1769-IM12 sólo admite 230 VCA |
| 6ES7 321-1FF1x-xxxx | 40 pines | 8 | 120 ... 230 VCA | 1769-IA8I | 1769-IA8I sólo admite 120 VCA |
| 6ES7 321-1EL0x-xxxx | 40 pines | 32 | 120 VCA | n/a | |
| n/a | | 16 | 5 VCC TTL | 1769-IG16 | |

Módulos de salidas digitales S7 300

| Número de catálogo Siemens | Conector frontal | Puntos | Rango | Corriente de salida | Solución de RA | Comentarios |
|----------------------------|------------------|--------|-------------|---------------------|-------------------------|--------------------------------|
| 6ES7 332-1FH0x-xxxx | 20 pines | 16 | 120/230 VCA | 0.5 A | 1769-OA16 | |
| 6ES7 332-1FF0x-xxxx | 20 pines | 8 | 120/230 VCA | 2 A | 1769-OA8 | S7-300 tiene fusible por grupo |
| 6ES7 332-5FF0x-xxxx | 40 pines | 8 | 120/230 VCA | 2 A | 1769-OA8 | S7-300 viene en grupos de 1 |
| 6ES7 322-1BH0x-xxxx | 20 pines | 16 | 24 VCC | 0.5 A | 1769-OB16 1769-OB16P | |
| 6ES7 322-1BH1x-xxxx | 20 pines | 16 | 24 VCC | 0.5 A | n/a | Alta velocidad |
| 6ES7 322-1BL0x-xxxx | 40 pines | 32 | 24 VCC | 0.5 A | 1769-OB32 1769-OB32T | |

| | | | | | | |
|---------------------|------------|----|----------------|-------|------------|--|
| 6ES7 322-1BF0x-xxxx | 20 pines | 8 | 24 VCC | 2 A | 1769-OB8 | |
| 6ES7 322-8BF0x-xxxx | 20 pines | 8 | 24 VCC | 0.5 A | 1769-OB8 | |
| 6ES7 332-1FL0x-xxxx | 2x20 pines | 32 | 120 VCA | 1 A | n/a | |
| 6ES7 332-5GH0x-xxxx | 40 pines | 16 | 24/48 V | 0.5 A | n/a | |
| 6ES7 332-1CF0x-xxxx | 20 pines | 8 | 48 ... 125 VCC | | n/a | |
| n/a | | 16 | 5 VCC TTL | | 1769-OG16 | |
| n/a | | 16 | 24 VCC | | 1769-OV16 | |
| n/a | | 32 | 24 VCC | | 1769-OV32T | |
| n/a | | 16 | 24 VCC | | 1769-OB16P | |

Módulos de salidas de relés S7 300

| Número de catálogo Siemens | Conector frontal | Puntos | Corriente de salida | Solución de RA | Comentarios |
|----------------------------|------------------|--------|---------------------|----------------|---|
| 6ES7 322-1HH0x-xxxx | 20 pines | 16 | 2 A | 1769-OW16 | |
| 6ES7 322-1HF0x-xxxx | 20 pines | 8 | 5 A | 1769-OW8 | |
| 6ES7 322-1HF1x-xxxx | 40 pines | 8 | 5 A | 1769-OW8I | |
| 6ES7 322-5HF0x-xxxx | 40 pines | 8 | 8 A | 1769-OW8I | El módulo S7-300 viene con filtro RC y protección contra sobrevoltaje |

Módulos digitales combinados S7 300

| Número de catálogo Siemens | Conector frontal | Puntos | Rango de entradas | Corriente de salida | Solución de RA | Comentarios |
|----------------------------|------------------|---------|-------------------|---------------------|----------------|--|
| 6ES7 323-1BH0x-xxxx | 20 pines | 8 / 8 | 24 VCC | 24 VCC / 0.5 A | 1769-IQ6XOW4 | Compact I/O tiene menos E/S y las salidas son de relés |
| 6ES7 323-1BL0x-xxxx | 40 pines | 16 / 16 | 24 VCC | 24 VCC / 0.5 A | n/a | |
| 6ES7 327-1BH0x-xxxx | 20 pines | 8 / 8 | 24 VCC | 24 VCC / 0.5 A | n/a | 8 entradas; 8 entradas o salidas (configurables) |

Módulos de entradas analógicas S7 300

| Número de catálogo Siemens | Conector frontal | Puntos | Resolución (bits) | Tipo | Solución Compact I/O | Comentarios |
|----------------------------|------------------|--------|-------------------|--|--------------------------|--|
| 6ES7 331-1KF0x-xxxx | 40 | 8 | 13 | Voltaje, corriente, resistencia Temperatura | 1769sc-IF8U 1769-IF8U | |
| 6ES7 331-7KF0x-xxxx | 20 | 8 | 9 / 12 / 14 | Voltaje, corriente, resistencia Temperatura | 1769sc-IF8U 1769-IF8U | |
| 6ES7 331-7KB0x-xxxx | 20 | 2 | 9 / 12 / 14 | Voltaje, corriente, resistencia Temperatura | 1769sc-IF8U 1769-IF4 | |
| 6ES7 331-7NF0x-xxxx | 40 | 8 | 16 | Voltaje Voltaje | 1769-IF8 | |
| 6ES7 331-7NF1x-xxxx | 40 | 8 | 16 | Voltaje Voltaje | 1769-IF8 | Incluye interrupción de hardware al final del ciclo a dif, del 6ES7 331-7NF0x-xxxx |
| 6ES7 331-7HF0x-xxxx | 20 | 8 | 14 | Voltaje Voltaje | 1769-IF8 | |
| 6ES7 331-7PF0x-xxxx | 40 | 8 | | RTD Resistencia | 1769-IR6 | |
| 6ES7 331-7PF1x-xxxx | 40 | 8 | | Termopar | 1769-IT6 | |
| n/a | | | | | 1769-IF4I | |

Módulos de salidas analógicas S7 300

| Número de catálogo Siemens | Conector frontal | Puntos | Resolución (bits) | Tipo | Solución de RA | Comentarios |
|----------------------------|------------------|--------|-------------------|----------------------|--------------------------|-------------|
| 6ES7 332-5HD0x-xxxx | 40 | 4 | 12 | Voltaje Corriente | 1769-OF4VI 1769-OF4CI | |
| 6ES7 332-7ND0x-xxxx | 20 | 4 | 16 | Voltaje Corriente | 1769-OF4VI 1769-OF4CI | |
| 6ES7 332-5HB0x-xxxx | 20 | 2 | 12 | Voltaje Corriente | 1769-OF2 | |
| 6ES7 332-5HF0x-xxxx | 20 | 8 | 12 | Voltaje Corriente | 1769-OF8V 1769-OF8C | |

Módulos analógicos combinados S7 300

| Número de catálogo Siemens | Conector frontal | Puntos | Resolución (bits) | Tipo | Solución de RA | Comentarios |
|----------------------------|------------------|--------|-------------------|--|----------------|------------------------------|
| 6ES7 334-0KE0x-xxxx | 20 | 4 / 2 | 12 | Voltaje Corriente Pt 100 | | Salidas de voltaje solamente |
| 6ES7 334-0CE0x-xxxx | 20 | 4 / 2 | 8 | Voltaje y corriente (entradas y salidas) | 1769-IF4XOF2 | |

Controladores estándar S7 400

| Número de catálogo Siemens | Referencia corta de Siemens | Tamaño de memoria de trabajo | | Puertos de com. | | Tamaño de memoria de carga máxima (RAM) | Solución ControlLogix de RA |
|----------------------------|-----------------------------|------------------------------|------------|-----------------|-----------|---|-----------------------------|
| | | | MPI | DP | PN | | |
| 6ES7 412-1XF04-0AB0 | CPU 412-1 | 144 KB | S | S | N | 64 MB | 1756-L61 |
| 6ES7 412-2GX04-0AB0 | CPU 412-2 | 256 KB | S | S | N | 64 MB | 1756-L61 |
| 6ES7 414-2GX04-0AB0 | CPU 414-2 | 512 KB | S | S | N | 64 MB | 1756-L62 |
| 6ES7 414-3XJ04-0AB0 | CPU 414-3 | 1.4 MB | S | S | N | 64 MB | 1756-L63 |
| 6ES7 414-3EM05-0AB0 | CPU 414-3 PN/DP | 2.8 MB | S | S | S | 64 MB | 1756-L63 |
| 6ES7 416-3XK04-0AB0 | CPU 416-2 | 2.8 MB | S | S | N | 64 MB | 1756-L63 |
| 6ES7 416-3XL04-0AB0 | CPU 416-3 | 5.6 MB | S | S | N | 64 MB | 1756-L64 |
| 6ES7 416-3ER05-0AB0 | CPU 416-3 PN/DP | 11.2 MB | S | S | S | 64 MB | 1756-L64 |
| 6ES7 417-4XL04-0AB0 | CPU 417-4 | 20 MB | S | S | N | 64 MB | 1756-L64 |

Controladores redundantes y a prueba de fallo

| Número de catálogo Siemens | Referencia corta de Siemens | Tamaño de memoria de trabajo | | Puertos de com. | | | Tamaño de memoria de carga máxima (RAM) | Solución ControlLogix de RA |
|----------------------------|-----------------------------|------------------------------|------------|-----------------|-----------|----------------------|---|-----------------------------|
| | | | MPI | DP | PN | Puertos sínc. | | |
| 6ES7 414-4HJ04-0AB0 | CPU 414-4H | 1.4 MB | S | S | N | S | 64 MB | 1756-L63 |
| 6ES7 417-4HL04-0AB0 | CPU 417-4H | 20 MB | S | S | N | S | 64 MB | 1756-L64 |
| 6ES7 416-2FK04-0AB0 | CPU-416F-2 | 2.6 MB | S | S | N | N | 64 MB | 1756-L61S |

Módulos de entradas digitales

| Número de catálogo Siemens | Conector frontal | Puntos | Rango | Solución de RA | Comentarios |
|--|------------------|--------|--------------|----------------|-------------|
| 6ES7 421-7BH01-0AB0 (Interrupción/ diagnóstico) | 48 pines | 16 | 24 VCC | 1756-IB16D | |
| 6ES7 421-1BL01-0AA0 | 48 pines | 32 | 24 VCC | 1756-IB32 | |
| 6ES7 421-1EL00-0AA0 | 48 pines | 32 | 120 VCA/CC | 1756-IA32 | |
| 6ES7 421-1FH20-0AA0 | 48 pines | 16 | 230 VCA/CC | 1756-IM161 | |
| 6ES7 421-7DH00 0AB0 (interrupción/diagnóstico) | 48 pines | 32 | 24-60 VCA/CC | | |

Módulos de salida digital

| Número de catálogo Siemens | Conector frontal | Puntos | Rango | Corriente | Solución de RA | Comentarios |
|--------------------------------------|------------------|--------|--------------------------|-----------|-------------------------|-------------|
| 6ES7 422-1FH00-0AA0 | 48 pines | 16 | 230 VCA | 2 A | 1756-OA16 | |
| 6ES7 422-1HH00-0AA0 | 48 pines | 16 | 60 VCC 230 VCA (relé) | 5 A | 1756-OW16I | |
| 6ES7 422 1BH11-0AA0 | 48 pines | 16 | 24 VCC | 2 A | 1756-OB16E | |
| 6ES7 422-1BL00-0AA0 | 48 pines | 32 | 24 VCC | 0.5 A | 1756-OB32 | |
| 6ES7 422-7BL00-0AB0 (diagnóstico) | 48 pines | 32 | 24 VCC | 0.5 A | 1756-OB16D 1756-OB32 | |

Módulos de entradas analógicas

| Número de catálogo Siemens | Conector frontal | Canales | Resolución (bits) | Tipo | Solución de RA | Comentarios |
|----------------------------|------------------|---------|-------------------|---|------------------------|---|
| 6ES7 431-0HH0-0AB0 | 48 pines | 16 | 13 | Voltaje Corriente | 1756-IF16 | 16 bits |
| 6ES7 431-1KF00-0AB0 | 48 pines | 8 | 13 | Voltaje Corriente Impedancia | 1756-IF8 | 16 bits 4 entradas diferenciales |
| 6ES7 431-1KF10-0AB0 | 48 pines | 8 | 14-16 | Voltaje Corriente Termopar Termorresistencia Impedancia | 1756-IR6I 1756-IT6I | 6 RTD 6 termopar Ambas de 16 bits |

| | | | | | | |
|---------------------------------------|----------|----|----|---|------------------------|---------------------|
| 6ES7 431-1FK20-0AB0 | 48 pines | 8 | 14 | Voltaje Corriente Impedancia | 1756-IF16 | 16 bits |
| 6ES7 431-7QH00-0AB0 (interrupción) | 48 pines | 16 | 16 | Voltaje Corriente Termopar Termorresistencia Impedancia | 1756-IR6I 1756-IT6I | 6 RTD 6 termopar |
| 6ES7 431-7KF00-0AB0 | 48 pines | 8 | 16 | Voltaje Corriente Termopar | 1756-IT6I | 6 canales |
| 6ES7 431-7KF01-0AB0 | 48 pines | 8 | 16 | Termorresistencia | 1756-IR6I | 5 canales |

Módulos de salidas analógicas

| Número de catálogo Siemens | Conector frontal | Canales | Resolución (bits) | Tipo | Solución de RA | Comentarios |
|-------------------------------|------------------|---------|-------------------|----------------------|----------------|-------------|
| 6ES7 432-1HF00-0AB0 | 48 pines | 8 | 13 | Voltaje Corriente | 1756-OF8 | 15 bits |

Tabla de referencias cruzadas de HMI Siemens

Use este apéndice para comparar paneles de Rockwell Automation con tipos específicos de paneles Siemens.

| Tema | Página |
|--|--------|
| Micropaneles SIMATIC y equivalentes de Rockwell Automation | 149 |
| Paneles SIMATIC serie 7x y equivalentes de Rockwell Automation | 151 |
| Paneles SIMATIC serie 17x y equivalentes de Rockwell Automation | 152 |
| Paneles SIMATIC serie 27x y equivalentes de Rockwell Automation | 154 |
| Multipaneles SIMATIC serie 27x y equivalentes de Rockwell Automation | 156 |
| Multipaneles SIMATIC serie 37x y equivalentes de Rockwell Automation | 158 |

Micropaneles SIMATIC y equivalentes de Rockwell Automation

| Micropaneles SIMATIC | | | | | Solución de Rockwell Automation | | |
|----------------------------|---|---|--------|---|---|---|--|
| Numero de catálogo Siemens | Referencia corta | Descripción | Mem. | Opción de com. | Número de catálogo de Rockwell Automation | Nombre | Descripción |
| 6AV6640-OBA11-0AX0 | SIMATIC OP 73MICRO | Pantalla de 3 pulg. STN monocroma, 160x48 pixeles, teclado, 24 VCC solamente | 128 KB | 1xRS485, compatible con S7-200, sin puerto de impresora | 2711P-K4M5D | PanelView Plus 400, escala de grises, con teclado | Pantalla de 3.8 pulg. STN con escala de grises, de 32 niveles, 320 x 240 pixeles, comunicación RS-232, teclado, 24 VCC, Flash de 64 MB, capacidades de impresión USB |
| 6AV6545-OAA15-2AX0 | SIMATIC TP070 Se retiró del mercado en abril de 2007 | Pantalla de 5.7 pulg. STN, modo azul (4 niveles), 320x240 pixeles, táctil, 24 VCC solamente | 128 KB | 1xRS485, compatible con S7-200, sin puerto de impresora | 2711P-T6M5D | PanelView Plus 600, escala de grises, táctil | Pantalla de 5.5 pulg. STN, con escala de grises de 32 niveles, 320 x 240 pixeles, comunicación RS-232, táctil, 24 VCC, capacidades de impresión USB |
| 6AV6640-OCA01-0AX0 | SIMATIC TP 170MICRO Se retiró del mercado en abril de 2007 | Pantalla de 5.7 pulg. STN, modo azul (4 niveles), 320x240 pixeles, táctil, 24 VCC solamente, funcionalidad limitada para aplicación | 256 KB | 1xRS485, compatible con S7-200, sin puerto de impresora | 2711P-T6M5D | PanelView Plus 600, escala de grises, táctil | Pantalla de 5.5 pulg. STN con escala de grises de 32 niveles, 320x240 pixeles, comunicación RS-232, táctil, 24 VCC, capacidades de impresión USB |

| Micropaneles SIMATIC | | | | | Solución de Rockwell Automation | | |
|----------------------------|-------------------------------|---|--------|---|---|--|--|
| Numero de catálogo Siemens | Referencia corta | Descripción | Mem. | Opción de com. | Número de catálogo de Rockwell Automation | Nombre | Descripción |
| 6AV6640-OCA11-0AX0 | SIMATIC TP 177MICRO | Pantalla de 5.7 pulg. STN, modo azul (4 niveles), 320x240 pixeles, táctil, 24 VCC solamente | 256 KB | 1xRS485, compatible con S7-200, sin puerto de impresora | 2711P-T6MSD | PanelView Plus 600, escala de grises, táctil | Pantalla de 5.5 pulg. STN con escala de grises de 32 niveles, 320x240 pixeles, comunicación RS-232, táctil, 24 VCC, capacidades de impresión USB |
| 6AV6610-OAA01-1CA8 | Software WINCC FLEXIBLE MICRO | Software de configuración y programación para micropaneles Simatic solamente | N/A | N/A | 9701-VWSTMENE | Software RStudio Machine Edition | Software de configuración RStudio for Machine Edition para desarrollo y prueba de aplicaciones de HMI a nivel de máquina |

Paneles SIMATIC serie 7x y equivalentes de Rockwell Automation

| Paneles SIMATIC serie 7x | | | | | Solución de Rockwell Automation | | |
|----------------------------|---------------------------------|---|--------|--|---|---|---|
| Numero de catálogo Siemens | Referencia corta | Descripción | Mem. | Opciones de com. | Número de catálogo de Rockwell Automation | Nombre | Descripción |
| 6AV6641-0AA11-0AX0 | SIMATIC OP73 | Pantalla de 3 pulg. STN monocroma, 160x48 pixeles, teclado, 24 VCC solamente | 256 KB | 1x RS485, compatible con S7-200, S7-300/400, sin puerto de impresora | 2711P-K4M5D | PanelView Plus 400, escala de grises, con teclado | Pantalla de 3.8 pulg. STN con escala de grises de 32 niveles, 320x240 pixeles, comunicación RS-232, teclado, 24 VCC, Flash de 64 MB, capacidades de impresión USB |
| 6AV6641-0BA11-0AX0 | SIMATIC OP77A | Pantalla de 4.5 pulg. STN monocroma, 160x64 pixeles, teclado, 24 VCC solamente | 256 KB | 1xRS422, 1xRS485, S7-200, S7-300/400, sin puerto de impresora | 2711P-K4M5D | PanelView Plus 400, escala de grises, con teclado | Pantalla de 3.8 pulg. STN con escala de grises de 32 niveles, 320x240 pixeles, comunicación RS-232, teclado, 24 VCC, Flash de 64 MB, capacidades de impresión USB |
| 6AV6641-0CA01-0AX0 | SIMATIC OP77B | Pantalla de 4.5 pulg. STN monocroma, 160x64 pixeles, teclado, 24 VCC solamente | 1 MB | 1xRS232, 1xRS422, 1xRS485, USB, S7-200, S7-300/400, puerto de impresora disponible | 2711P-K4M5D | PanelView Plus 400, escala de grises, con teclado | Pantalla de 3.8 pulg. STN con escala de grises de 32 niveles, 320x240 pixeles, comunicación RS-232, teclado, 24 VCC, Flash de 64 MB, capacidades de impresión USB |
| 6AV6621-0AA01-0AA0 | Software WINCC FLEXIBLE COMPACT | Software de configuración y programación para Simatic OP77, OP/TP170 y micropaneles | N/A | N/A | 9701-VWSTMENE | Software RStudio for Machine Edition | Software de configuración RStudio Studio Machine Edition para desarrollo y prueba de aplicaciones de HMI a nivel de máquina |

Paneles SIMATIC serie 17x y equivalentes de Rockwell Automation

| Paneles SIMATIC serie 17x | | | | | Solución de Rockwell Automation | | |
|----------------------------|--|--|--------|--|---|---|--|
| Numero de catálogo Siemens | Referencia corta | Descripción | Mem. | Opciones de com. | Número de catálogo de Rockwell Automation | Nombre | Descripción |
| 6AV6545-OBA15-2AX0 | SIMATIC TP170A, modo azul. Se retiró del mercado en abril de 2007 | Pantalla de 5.7 pulg. STN, modo azul (4 niveles), 320x240 pixeles, táctil, 24 VCC solamente | 320 KB | 1xRS232, 1xRS422, 1xRS485, controladores S5, S7-200, S7-300/400 y de otros fabricantes, sin puerto de impresora | 2711P-T6M20D | PanelView Plus 600, escala de grises, táctil | Pantalla de 5.5 pulg. STN con escala de grises de 32 niveles, 320x240 pixeles, comunicación RS-232 y EtherNet/IP, táctil, 24 VCC, Flash de 64 MB, capacidades de impresión USB |
| 6AV6545-OB15-2AX0 | SIMATIC TP170B, modo azul. Se retiró del mercado en abril de 2007 | Pantalla de 5.7 pulg. STN, modo azul (4 niveles), 320x240 pixeles, táctil, 24 VCC solamente | 768 KB | 2xRS232, 1xRS422, 1xRS485, controladores S5, S7-200, S7-300/400 y de otros fabricantes, puerto de impresora disponible | 2711P-T6M20D | PanelView Plus 600, escala de grises, táctil | Pantalla de 5.5 pulg. STN con escala de grises de 32 niveles, 320x240 pixeles, comunicación EtherNet/IP, RS-232, táctil, 24 VCC, Flash de 64 MB, capacidades de impresión USB |
| 6AV6545-OB15-2AX0 | SIMATIC TP170B de colores. Se retiró del mercado en abril de 2007 | Pantalla de 5.7 pulg. STN de colores (256 colores), 320x240 pixeles, táctil. 24 VCC solamente | 768 KB | 2xRS232, 1xRS422, 1xRS485, controladores S5, S7-200, S7-300/400 y de otros fabricantes, puerto de impresora disponible | 2711P-T6C20D | PanelView Plus 600, de colores, táctil | Pantalla de 5.5 pulg. TFT de colores, 320x240 pixeles, profundidad de color de 18 bits, comunicación EtherNet/IP, RS-232, táctil, 24 VCC, Flash de 64 MB, capacidades de impresión USB |
| 6AV6542-OB15-2AX0 | SIMATIC OP170B, modo azul. Se retiró del mercado en abril de 2007 | Pantalla de 5.7 pulg. STN, modo azul (4 niveles), 320x240 pixeles, teclado y pantalla táctil, 24 VCC solamente | 768 KB | 2xRS232, 1xRS422, 1xRS485, controladores S5, S7-200, S7-300/400 y de otros fabricantes, puerto de impresora disponible | 2711P-B6M20D | PanelView Plus 600, escala de grises, pantalla táctil y teclado | Pantalla de 5.5 pulg. STN con escala de grises de 32 niveles, 320x240 pixeles, comunicación EtherNet/IP, RS-232, pantalla táctil y teclado, 24 VCC, Flash de 64 MB, capacidades de impresión USB |
| 6AV6642-ODC01-1AX0 | SIMATIC OP177B, modo azul | Pantalla de 5.7 pulg. STN, modo azul (4 niveles), 320x240 pixeles, teclado y pantalla táctil, 24 VCC solamente | 2 MB | 1xRS422, 1xRS485, USB, Ethernet, controladores S5, S7-200, S7-300/400 y de otros fabricantes, puerto de impresora disponible | 2711P-B6M20D | PanelView Plus 600, escala de grises, pantalla táctil y teclado | Pantalla de 5.5 pulg. STN con escala de grises de 32 niveles, 320x240 pixeles, comunicación EtherNet/IP, RS-232, pantalla táctil y teclado, 24 VCC, Flash de 64 MB, capacidades de impresión USB |

| Paneles SIMATIC serie 17x | | | | | Solución de Rockwell Automation | | |
|----------------------------|--|---|--------|--|---|---|--|
| Numero de catálogo Siemens | Referencia corta | Descripción | Mem. | Opciones de com. | Número de catálogo de Rockwell Automation | Nombre | Descripción |
| 6AV6642-OAA11-0AX0 | SIMATIC TP177A, modo azul | Pantalla de 5.7 pulg. STN, modo azul (4 niveles), 320x240 pixeles, táctil, 24 VCC solamente | 512 KB | 1xRS422, 1xRS485, compatible con S7-200, S7-300/400, sin puerto de impresora | 2711P-T6M20D | PanelView Plus 600, escala de grises, táctil | Pantalla de 5.5 pulg. STN con escala de grises de 32 niveles, 320 x 240 pixeles, comunicaciones EtherNet/IP, RS-232, táctil, 24 VCC, Flash de 64 MB, capacidades de impresión USB |
| 6AV6642-OBA01-1AX0 | SIMATIC TP177B de colores | Pantalla de 5.7 pulg. STN de colores (256 colores), 320x240 pixeles, táctil. 24 VCC solamente | 2 MB | 1xRS422, 1xRS485, USB, Ethernet, controladores S5, S7-200, S7-300/400 y de otros fabricantes, puerto de impresora disponible | 2711P-T6C20D | PanelView Plus 600 de colores, táctil | Pantalla de 5.5 pulg. TFT de colores, 320 x 240 pixeles, profundidad de color de 18 bits, comunicaciones EtherNet/IP, RS-232, táctil, 24 VCC, Flash de 64 MB, capacidades de impresión USB |
| 6AV6642-OBC01-1AX0 | SIMATIC TP177B, modo azul | Pantalla de 5.7 pulg. STN, modo azul (4 niveles), 320x240 pixeles, táctil, 24 VCC solamente | 2 MB | 1xRS422, 1xRS485, USB, S5, controladores S7-200, S7-300/400 y de otros fabricantes, puerto de impresora disponible | 2711P-T6M20D | PanelView Plus 600, escala de grises, táctil | Pantalla de 5.5 pulg. STN con escala de grises de 32 niveles, 320 x 240 pixeles, comunicaciones EtherNet/IP, RS-232, táctil, 24 VCC, Flash de 64 MB, capacidades de impresión USB |
| 6AV6642-8BA10-0AA0 | SIMATIC TP177B de colores, de acero inoxidable | Pantalla de 5.7 pulg. STN, de colores (256 colores), 320x240 pixeles, táctil, 24 VCC solamente, bisel de acero inoxidable | 2 MB | 1xRS422, 1xRS485, USB, Ethernet, controladores S5, S7-200, S7-300/400 y de otros fabricantes, puerto de impresora disponible | 2711P-T6C20D | PanelView Plus 600, de colores, táctil | Pantalla de 5.5 pulg. TFT de colores, 320x240 pixeles, profundidad de color de 18 bits, comunicación EtherNet/IP, RS-232, táctil, 24 VCC, Flash de 64 MB, capacidades de impresión USB |
| 6AV6642-ODA01-1AX0 | SIMATIC OP177B, de colores | Pantalla de 5.7 pulg. STN, de colores (256 colores), 320x240 pixeles, teclado y pantalla táctil, 24 VCC solamente | 2 MB | 1xRS422, 1xRS485, USB, Ethernet, controladores S5, S7-200, S7-300/400 y de otros fabricantes, puerto de impresora disponible | 2711P-B6C20D | PanelView Plus 600, de colores, pantalla táctil y teclado | Pantalla de 5.5 pulg. TFT de colores, 320x240 pixeles, profundidad de color de 18 bits, comunicación EtherNet/IP, RS-232, táctil y teclado, 24 VCC, Flash de 64 MB, capacidades de impresión USB |
| 6AV6621-OAA01-0AA0 | Software WINCC FLEXIBLE COMPACT | Software de configuración y programación para Simatic OP77, OP/TP170, y micropaneles | N/A | N/A | 9701-VWSTMENE | Software RSView Studio Machine Edition | Software de configuración RSView Studio Machine Edition para desarrollo y prueba de aplicaciones de HMI a nivel de máquina |

Paneles SIMATIC serie 27x y equivalentes de Rockwell Automation

| Paneles SIMATIC serie 27x | | | | | Solución de Rockwell Automation | | |
|----------------------------|--|--|------|---|---|---|---|
| Numero de catálogo Siemens | Referencia corta | Descripción | Mem. | Opciones de com. | Número de catálogo de Rockwell Automation | Nombre | Descripción |
| 6AV6545-OCA10-OAX0 | SIMATIC TP270, 6 pulg., de colores. Se retiró del mercado en octubre de 2006 | Pantalla de 5.7 pulg. STN, de colores (256 colores), 320x240 pixeles, táctil, 24 VCC solamente | 2 MB | 2xRS232, 1xRS422, 1xRS485, USB, controladores S5, S7-200, S7-300/400 y de otros fabricantes, puerto de impresora disponible | 2711P-T6C20D | PanelView Plus 600, de colores, táctil | Pantalla de 5.5 pulg. TFT de colores, 320x240 pixeles, profundidad de color de 18 bits, comunicación EtherNet/IP, RS-232, táctil, 24 VCC, Flash de 64 MB, capacidades de impresión USB |
| 6AV6545-OCC10-OAX0 | SIMATIC TP270, 10 pulg., de colores. Se retiró del mercado en octubre de 2006 | Pantalla de 10.4 pulg. STN, de colores (256 colores), 640x480 pixeles, táctil, 24 VCC solamente | 2 MB | 2xRS232, 1xRS422, 1xRS485, USB, controladores S5, S7-200, S7-300/400 y de otros fabricantes, puerto de impresora disponible | 2711P-T10C4D1 | PanelView Plus 1000, de colores, táctil | Pantalla de 10.4 pulg. TFT, 640x480 pixeles, colores de 18 bits, EtherNet/IP y RS-232, táctil, 24 VCC, Flash de 64 MB, capacidades de impresión USB |
| 6AV6542-OCA10-OAX0 | SIMATIC OP270, 6 pulg., de colores. Se retiró del mercado en octubre de 2006 | Pantalla de 5.7 pulg. STN, de colores (256 colores), 320x240 pixeles, teclado, 24 VCC solamente | 2 MB | 2xRS232, 1xRS422, 1xRS485, USB, controladores S5, S7-200, S7-300/400 y de otros fabricantes, puerto de impresora disponible | 2711P-K6C20D | PanelView Plus 600, de colores | Pantalla de 5.5 pulg. TFT de colores, 320x240 pixeles, profundidad de color de 18 bits, comunicación EtherNet/IP, RS-232, teclado, 24 VCC, Flash de 64 MB, capacidades de impresión USB |
| 6AV6542-OCC10-OAX0 | SIMATIC OP270, 10 pulg., de colores. Se retiró del mercado en octubre de 2006 | Pantalla de 10.4 pulg. STN, de colores (256 colores), 640x480 pixeles, teclado, 24 VCC solamente | 2 MB | 2xRS232, 1xRS422, 1xRS485, USB, controladores S5, S7-200, S7-300/400 y de otros fabricantes, puertode impresora disponible | 2711P-K10C4D1 | PanelView Plus 1000, de colores, teclado. | Pantalla de 10.4 pulg. TFT, 640x480 pixeles, colores de 18 bits, EtherNet/IP y RS-232, teclado, 24 VCC, Flash de 64 MB, capacidades de impresión USB |

| Paneles SIMATIC serie 27x | | | | | Solución de Rockwell Automation | | |
|----------------------------|-------------------------------------|---|------|--|---|--|---|
| Numero de catálogo Siemens | Referencia corta | Descripción | Mem. | Opciones de com. | Número de catálogo de Rockwell Automation | Nombre | Descripción |
| 6AV6643-0AA01-1AX0 | SIMATIC TP 277, 6 pulg., de colores | Pantalla de 5.7 pulg. STN, de colores (256 colores), 320x240 pixeles, táctil, 24 VCC solamente | 4 MB | 1xRS422, 1xRS485, USB, Ethernet: controladores S5, S7-200, S7-300/400 y de otros fabricantes, puerto de impresora disponible | 2711P-T6C20D | PanelView Plus 600, de colores, táctil | Pantalla de 5.5 pulg. TFT de colores, 320x240 pixeles, profundidad de color de 18 bits, comunicación EtherNet/IP, RS-232, táctil, 24 VCC, Flash de 64 MB, capacidades de impresión USB |
| 6AV6643-0BA01-1AX0 | SIMATIC OP 277, 6 pulg., de colores | Pantalla de 5.7 pulg. STN, de colores (256 colores), 320x240 pixeles, teclado, 24 VCC solamente | 4 MB | 1xRS422, 1xRS485, USB, Ethernet, controladores S5, S7-200, S7-300/400 y de otros fabricantes, puerto de impresora disponible | 2711P-K6C20D | PanelView Plus 600, de colores | Pantalla de 5.5 pulg. TFT de colores, 320x240 pixeles, profundidad de color de 18 bits, comunicación EtherNet/IP, RS-232, teclado, 24 VCC, Flash de 64 MB, capacidades de impresión USB |
| 6AV6622-0BA01-0AA0 | Software WINCC FLEXIBLE STANDARD | Software de configuración y programación para Simatic OP/TP/MP270, MP370, OP77, OP/TP170 y micropaneles | N/A | N/A | 9701-VWSTMENE | Software RSView Studio Machine Edition | Software de configuración RSView Studio Machine Edition para desarrollo y prueba de aplicaciones de HMI a nivel de máquina |

Multipaneles SIMATIC serie 27x y equivalentes de Rockwell Automation

| Multipaneles SIMATIC serie 27x | | | | | Solución de Rockwell Automation | | |
|--------------------------------|---|---|------|--|---|---|---|
| Numero de catálogo Siemens | Referencia corta | Descripción | Mem. | Opciones de com. | Número de catálogo de Rockwell Automation | Nombre | Descripción |
| 6AV6542-OAG10-OAXO | SIMATIC MP270B, teclado, 10 pulg. Se retiró del mercado en octubre de 2006 | Pantalla de 10.4 pulg. TFT, de colores (64 K colores), 640x480 pixeles, teclado, 24 VCC solamente | 5 MB | 2xRS422, 1xRS485, USB, Ethernet, controladores S5, S7-200, S7-300/400 y de otros fabricantes, puerto de impresora disponible | 2711P-K10C4D1 | PanelView Plus 1000, de colores, teclado. | Pantalla de 10.4 pulg. TFT, 640x480 pixeles, colores de 18 bits, EtherNet/IP y RS-232, teclado, 24 VCC, Flash de 64 MB, capacidades de impresión USB |
| 6AV6545-OAG10-OAXO | SIMATIC MP270B, pantalla táctil de 10 pulg. Se retiró del mercado en octubre de 2006 | Pantalla de 10.4 pulg. TFT, de colores (64 K colores), 640x480 pixeles, táctil, 24 VCC solamente | 5 MB | 2xRS422, 1xRS485, USB, Ethernet, controladores S5, S7-200, S7-300/400 y de otros fabricantes, puerto de impresora disponible | 2711P-T10C4D1 | PanelView Plus 1000, de colores, táctil | Pantalla de 10.4 pulg. TFT, 640x480 pixeles, colores de 18 bits, EtherNet/IP y RS-232, táctil, 24 VCC, Flash de 64 MB, capacidades de impresión USB |
| 6AV6545-OAH10-OAXO | SIMATIC MP270B, pantalla táctil de 6 pulg. Se retiró del mercado en octubre de 2006 | Pantalla de 5.7 pulg. TFT, de colores (64 K colores), 320x240 pixeles, táctil, 24 VCC solamente | 5 MB | 2xRS422, 1xRS485, USB, Ethernet, controladores S5, S7-200, S7-300/400 y de otros fabricantes, puerto de impresora disponible | 2711P-K6C20D | PanelView Plus 600, de colores | Pantalla de 5.5 pulg. TFT de colores, 320x240 pixeles, profundidad de color de 18 bits, comunicación EtherNet/IP, RS-232, teclado, 24 VCC, Flash de 64 MB, capacidades de impresión USB |
| 6AV6643-OCB01-1AXO | SIMATIC MP277, pantalla táctil de 8 pulg. | Pantalla de 7.5 pulg. TFT, de colores (64 K colores), 640x480 pixeles, táctil, 24 VCC solamente | 6 MB | 1xRS422, 1xRS485, 2xUSB, Ethernet, controladores S5, S7-200, S7-300/400 y de otros fabricantes, puerto de impresora disponible | 2711P-T7C4D1 | PanelView Plus 700, de colores, táctil | Pantalla de 6.5 pulg. TFT, 640x480 pixeles, colores de 18 bits, EtherNet/IP y RS-232, táctil, 24 VCC, Flash de 64 MB, capacidades de impresión USB |

| Multipaneles SIMATIC serie 27x | | | | | Solución de Rockwell Automation | | |
|--------------------------------|---|---|------|--|---|---|--|
| Numero de catálogo Siemens | Referencia corta | Descripción | Mem. | Opciones de com. | Número de catálogo de Rockwell Automation | Nombre | Descripción |
| 6AV6643-OC01-1AX0 | SIMATIC MP 277, pantalla táctil de 10 pulg. | Pantalla de 10.4 pulg. TFT, de colores (64 K colores), 640x480 pixeles, táctil, 24 VCC solamente | 6 MB | 1xRS422, 1xRS485, 2xUSB, Ethernet: controladores S5, S7-200, S7-300/400 y de otros fabricantes, puerto de impresora disponible | 2711P-T10C4D1 | PanelView Plus 1000, de colores, táctil | Pantalla de 10.4 pulg. TFT, 640x480 pixeles, colores de 18 bits, EtherNet/IP y RS-232, táctil, 24 VCC, Flash de 64 MB, capacidades de impresión USB |
| — | SIMATIC MP 277, pantalla táctil, 10 pulg., acero inoxidable | Pantalla de 10.4 pulg. TFT, de colores (64 K colores), 640x480 pixeles, táctil, 24 VCC solamente, bisel de acero inoxidable, IP66 | 6 MB | 1xRS422, 1xRS485, 2xUSB, Ethernet, controladores S5, S7-200, S7-300/400 y de otros fabricantes, puerto de impresora disponible | 2711P-T10C4D1 | PanelView Plus 1000, de colores, táctil | Pantalla de 10.4 pulg. TFT, 640x480 pixeles, colores de 18 bits, EtherNet/IP y RS-232, táctil, 24 VCC, Flash de 64 MB, capacidades de impresión USB |
| 6AV6643-0DB01-1AX0 | SIMATIC MP 277, teclado, 8 pulg. | Pantalla de 7.5 pulg. TFT, de colores (64 K colores), 640x480 pixeles, teclado, 24 VCC solamente | 6 MB | 1xRS422, 1xRS485, 2xUSB, Ethernet, controladores S5, S7-200, S7-300/400 y de otros fabricantes, puerto de impresora disponible | 2711P-K7C4D1 | PanelView Plus 700, de colores, teclado. | Pantalla de 6.5 pulg. TFT, 640x480 pixeles, colores de 18 bits, EtherNet/IP y RS-232, teclado, 24 VCC, Flash de 64 MB, capacidades de impresión USB |
| 6AV6643-0DD01-1AX0 | SIMATIC MP 277, teclado, 10 pulg. | Pantalla de 10.5 pulg. TFT, de colores (64 K colores), 640x480 pixeles, teclado, 24 VCC solamente | 6 MB | 1xRS422, 1xRS485, 2xUSB, Ethernet, controladores S5, S7-200, S7-300/400 y de otros fabricantes, puerto de impresora disponible | 2711P-K10C4D1 | PanelView Plus 1000, de colores, teclado. | Pantalla de 10.4 pulg. TFT, 640x480 pixeles, colores de 18 bits, EtherNet/IP y RS-232, teclado, 24 VCC, Flash de 64 MB, capacidades de impresión USB |
| 6AV6622-0BA01-0AA0 | Software WINCC FLEXIBLE STANDARD | Software de configuración y programación para Simatic OP/TP/MP270, MP370, OP77, OP/TP170 y micropaneles | N/A | N/A | 9701-VWSTMENE | Software RSView Studio Machine Edition | Software de configuración RSView Studio Machine Edition para desarrollo y prueba de aplicaciones de HMI a nivel de máquina |

Multipaneles SIMATIC serie 37x y equivalentes de Rockwell Automation

| Multipaneles SIMATIC serie 37x | | | | | Solución de Rockwell Automation | | |
|--------------------------------|--|---|---------|--|---|---|--|
| Numero de catálogo Siemens | Referencia corta | Descripción | Mem. | Opciones de com. | Número de catálogo de Rockwell Automation | Nombre | Descripción |
| 6AV6542-ODA10-0AX0 | SIMATIC MP370, teclado, 12 pulg. | Pantalla de 12.1 pulg. TFT, de colores (256 colores), 800x600 pixeles, teclado, 24 VCC solamente | 12.5 MB | 1xTTY, 2xRS232, 1xRS422, 1xRS485, 1xUSB, Ethernet, controladores S5, S7-200, S7-300/400 y de otros fabricantes, puerto de impresora disponible | 2711P-K12C4D1 | PanelView Plus 1250, de colores, teclado. | Pantalla de 12.1 pulg. TFT, 800x600 pixeles, colores de 18 bits, EtherNet/IP y RS-232, teclado, 24 VCC, Flash de 64 MB, capacidades de impresión USB |
| 6AV6545-ODA10-0AX0 | SIMATIC MP370, pantalla táctil, 12 pulg. | Pantalla de 12.1 pulg. TFT, de colores (256 colores), 800x600 pixeles, táctil, 24 VCC solamente | 12.5 MB | 1xTTY, 2xRS232, 1xRS422, 1xRS485, 1xUSB, Ethernet, controladores S5, S7-200, S7-300/400 y de otros fabricantes, puerto de impresora disponible | 2711P-T12C4D1 | PanelView Plus 1250, de colores, táctil | Pantalla de 12.1 pulg. TFT, 800x600 pixeles, colores de 18 bits, EtherNet/IP y RS-232, táctil, 24 VCC, Flash de 64 MB, capacidades de impresión USB |
| 6AV6545-ODB10-0AX0 | SIMATIC MP370, pantalla táctil, 15 pulg. | Pantalla de 15.1 pulg. TFT, de colores (256 colores), 1024x768 pixeles, táctil, 24 VCC solamente | 12.5 MB | 1xTTY, 2xRS232, 1xRS422, 1xRS485, 1xUSB, Ethernet, controladores S5, S7-200, S7-300/400 y de otros fabricantes, puerto de impresora disponible | 2711P-T15C4D1 | PanelView Plus 1500, de colores, táctil | Pantalla de 15 pulg. TFT, 1024x768 pixeles, colores de 18 bits, EtherNet/IP y RS-232, táctil, 24 VCC, Flash de 64 MB, capacidades de impresión USB |
| 6AV6545-8DB10-0AAA | SIMATIC MP379, pantalla táctil, 15 pulg., acero inoxidable | Pantalla de 15.1 pulg. TFT, de colores (256 K colores), 1024x768 pixeles, táctil, 24 VCC solamente, bisel de acero inoxidable, IP66 | 12.5 MB | 1xTTY, 2xRS232, 1xRS422, 1xRS485, 1xUSB, Ethernet, controladores S5, S7-200, S7-300/400 y de otros fabricantes, puerto de impresora disponible | 2711P-T15C4D1 | PanelView Plus 1500, de colores, táctil | Pantalla de 15 pulg. TFT, 1024x768 pixeles, colores de 18 bits, EtherNet/IP y RS-232, táctil, 24 VCC, Flash de 64 MB, capacidades de impresión USB |

| Multipaneles SIMATIC serie 37x | | | | | Solución de Rockwell Automation | | |
|--------------------------------|--|---|---------|--|---|---|--|
| Numero de catálogo Siemens | Referencia corta | Descripción | Mem. | Opciones de com. | Número de catálogo de Rockwell Automation | Nombre | Descripción |
| 6AV6 644-0AA01-2AX0 | SIMATIC MP377, pantalla táctil, 12.1 pulg. | Pantalla de 12.1 pulg. TFT, 65,536 colores, 800x600 pixeles, táctil, 24 VCC solamente | 12.5 MB | 1xTTY, 2xRS232, 1xRS422, 1xRS485, 2xUSB, 2xEthernet, controladores S5, S7-200, S7-300/400 y de otros fabricantes, puerto de impresora disponible | 2711P-T12C4D1 | PanelView Plus 1250, de colores, táctil | Pantalla de 12.1 pulg. TFT, 800x600 pixeles, colores de 18 bits, EtherNet/IP y RS-232, táctil, 24 VCC, Flash de 64 MB, capacidades de impresión USB |
| 6AV6 644-0BA01-2AX0 | SIMATIC MP377, teclado, 12.1 pulg. | Pantalla de 12.1 pulg. TFT, 65,536 colores, 800x600 pixeles, teclado, 24 VCC solamente | 12.5 MB | 1xTTY, 2xRS232, 1xRS422, 1xRS485, 2xUSB, 2xEthernet, controladores S5, S7-200, S7-300/400 y de otros fabricantes, puerto de impresora disponible | 2711P-K12C4D1 | PanelView Plus 1250, de colores, teclado. | Pantalla de 12.1 pulg. TFT, 800x600 pixeles, colores de 18 bits, EtherNet/IP y RS-232, teclado, 24 VCC, Flash de 64 MB, capacidades de impresión USB |
| 6AV6 644-0AB01-2AX0 | SIMATIC MP377, pantalla táctil, 15 pulg. | Pantalla de 15 pulg. TFT, 65,536 colores, 1024x768 pixeles, táctil, 24 VCC solamente | 12.5 MB | 1xTTY, 2xRS232, 1xRS422, 1xRS485, 2xUSB, 2xEthernet, controladores S5, S7-200, S7-300/400 y de otros fabricantes, puerto de impresora disponible | 2711P-T15C4D1 | PanelView Plus 1500, de colores, táctil | Pantalla de 15 pulg. TFT, 1024x768 pixeles, colores de 18 bits, EtherNet/IP y RS-232, táctil, 24 VCC, Flash de 64 MB, capacidades de impresión USB |
| 6AV6 644-0BA01-2AX0 | SIMATIC MP377, pantalla táctil, 19 pulg. | Pantalla de 19 pulg. TFT, 65,536 colores, 1280x1024 pixeles, táctil, 24 VCC solamente | 12.5 MB | 1xTTY, 2xRS232, 1xRS422, 1xRS485, 2xUSB, 2xEthernet, controladores S5, S7-200, S7-300/400 y de otros fabricantes, puerto de impresora disponible | 2711P-T15C4D1 | PanelView Plus 1500, de colores, táctil | Pantalla de 15 pulg. TFT, 1024x768 pixeles, colores de 18 bits, EtherNet/IP y RS-232, táctil, 24 VCC, Flash de 64 MB, capacidades de impresión USB |
| 6AV6622-0BA01-0AA0 | Software WINCC FLEXIBLE STANDARD | Software de configuración y programación para Simatic OP/TP/MP270, MP370, OP77, OP/TP170 y micropaneles | N/A | N/A | 9701-VWSTMENE | Software RStudio Machine Edition | Software de configuración RStudio Machine Edition para desarrollo y prueba de aplicaciones de HMI a nivel de máquina |

Notas:

Servicio de asistencia técnica de Rockwell Automation

Rockwell Automation proporciona información técnica en la web para ayudarle a usar nuestros productos. En <http://support.rockwellautomation.com>, encontrará manuales técnicos, una base de conocimientos con respuestas a preguntas frecuentes, notas técnicas y de aplicación, ejemplo de códigos y vínculos a Service Packs de software, además de la función MySupport que usted puede personalizar para aprovechar al máximo estas herramientas.

Con el fin de brindar un nivel adicional de asistencia técnica por teléfono para instalación, configuración y resolución de problemas, ofrecemos los programas de asistencia TechConnect. Para obtener más información, comuníquese con el distribuidor local o con el representante de Rockwell Automation, o visite <http://support.rockwellautomation.com>.

Asistencia para la instalación

Si se le presenta un problema durante las primeras 24 horas de instalación, revise la información contenida en este manual. También puede comunicarse con un número especial de asistencia al cliente a fin de obtener ayuda inicial para la puesta en marcha de su producto.

| | |
|-------------------------|--|
| Estados Unidos | +1-440-646-3434 Lunes a viernes de 08:00 a 17:00, hora oficial del Este de EE.UU. |
| Fuera de Estados Unidos | Para cualquier asunto relacionado con asistencia técnica, comuníquese con el representante local de Rockwell Automation. |

Procedimiento de devolución de productos nuevos

Rockwell Automation prueba todos sus productos para asegurarse de que estén en perfecto estado de funcionamiento cuando se envían de la fábrica. Sin embargo, si su producto no funciona y necesita devolverlo, siga estos procedimientos.

| | |
|-------------------------|--|
| Estados Unidos | Comuníquese con el distribuidor. Deberá proporcionar al distribuidor un número de caso de asistencia al cliente (llame al número de teléfono anterior para obtener uno) a fin de completar el proceso de devolución. |
| Fuera de Estados Unidos | Comuníquese con el representante local de Rockwell Automation para obtener información sobre el procedimiento de devolución. |

www.rockwellautomation.com

Oficinas corporativas de soluciones de potencia, control e información

Américas: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204-2496 USA, Tel: (1) 414.382.2000, Fax: (1) 414.382.4444

Europa/Medio Oriente/África: Rockwell Automation, Vorstlaan/Boulevard du Souverain 36, 1170 Bruselas, Bélgica, Tel: (32) 2 663 0600, Fax: (32) 2 663 0640

Asia-Pacífico: Rockwell Automation, Level 14, Core F, Cyberport 3, 100 Cyberport Road, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846

Argentina: Rockwell Automation S.A., Alem 1050, 5° Piso, CP 1001AAS, Capital Federal, Buenos Aires, Tel.: (54) 11.5554.4000, Fax: (54) 11.5554.4040, www.rockwellautomation.com.ar

Chile: Rockwell Automation Chile S.A., Luis Thayer Ojeda 166, Piso 6, Providencia, Santiago, Tel.: (56) 2.290.0700, Fax: (56) 2.290.0707, www.rockwellautomation.cl

Colombia: Rockwell Automation S.A., Edf. North Point, Carrera 7 N° 156 – 78 Piso 18, PBX: (57) 1.649.96.00 Fax: (57) 649.96.15, www.rockwellautomation.com.co

España: Rockwell Automation S.A., Doctor Trueta 113-119, 08005 Barcelona, Tel.: (34) 932.959.000, Fax: (34) 932.959.001, www.rockwellautomation.es

México: Rockwell Automation S.A. de C.V., Bosques de Cierulos N° 160, Col. Bosques de Las Lomas, C.P. 11700 México, D.F., Tel.: (52) 55.5246.2000, Fax: (52) 55.5251.1169, www.rockwellautomation.com.mx

Perú: Rockwell Automation S.A., Av Victor Andrés Belaunde N°147, Torre 12, Of. 102 – San Isidro Lima, Perú, Tel.: (511) 441.59.00, Fax: (511) 222.29.87, www.rockwellautomation.com.pe

Puerto Rico: Rockwell Automation Inc., Calle 1, Metro Office # 6, Suite 304, Metro Office Park, Guaynabo, Puerto Rico 00968, Tel.: (1) 787.300.6200, Fax: (1) 787.706.3939, www.rockwellautomation.com.pr

Venezuela: Rockwell Automation S.A., Edf. Allen-Bradley, Av. González Rincones, Zona Industrial La Trinidad, Caracas 1080, Tel.: (58) 212.949.0611, Fax: (58) 212.943.3955, www.rockwellautomation.com.ve