

Modicon M241 Logic Controller

Funciones y variables del sistema

Guía de la biblioteca PLCSystem

Traducción del manual original

EIO0000003068.06

12/2024



Información legal

La información proporcionada en este documento contiene descripciones generales, características técnicas o recomendaciones relacionadas con productos o soluciones.

Este documento no pretende sustituir a un estudio detallado o un plan de desarrollo o esquemático específico de operaciones o sitios. No debe usarse para determinar la adecuación o la fiabilidad de los productos o las soluciones para aplicaciones de usuario específicas. Es responsabilidad del usuario realizar o solicitar a un experto profesional (integrador, especificador, etc.) que realice análisis de riesgos, evaluación y pruebas adecuados y completos de los productos o las soluciones con respecto a la aplicación o el uso específicos de dichos productos o dichas soluciones.

La marca Schneider Electric y cualquier otra marca comercial de Schneider Electric SE y sus filiales mencionadas en este documento son propiedad de Schneider Electric SE o sus filiales. Todas las otras marcas pueden ser marcas comerciales de sus respectivos propietarios.

Este documento y su contenido están protegidos por las leyes de copyright aplicables, y se proporcionan exclusivamente a título informativo. Ninguna parte de este documento puede ser reproducida o transmitida de cualquier forma o por cualquier medio (electrónico, mecánico, fotocopia, grabación u otro), para ningún propósito, sin el permiso previo por escrito de Schneider Electric.

Schneider Electric no otorga ningún derecho o licencia para el uso comercial del documento o su contenido, excepto por una licencia no exclusiva y personal para consultarla "tal cual".

Schneider Electric se reserva el derecho de realizar cambios o actualizaciones con respecto a o en el contenido de este documento o con respecto a o en el formato de dicho documento en cualquier momento sin previo aviso.

En la medida permitida por la ley aplicable, Schneider Electric y sus filiales no asumen ninguna responsabilidad u obligación por cualquier error u omisión en el contenido informativo de este documento o por el uso no previsto o el mal uso del contenido de dicho documento.

Tabla de contenido

Información de seguridad	7
Acerca de este libro	8
M241 Variables del sistema	13
Variables del sistema: definición y uso	13
Descripción de las variables de sistema	13
Utilización de las variables de sistema	14
Estructuras de <i>PLC_R</i> y <i>PLC_W</i>	15
<i>PLC_R</i> : variables de sistema de solo lectura del controlador	16
<i>PLC_W</i> : variables de sistema de lectura/escritura del controlador	19
Estructuras de <i>SERIAL_R</i> y <i>SERIAL_W</i>	19
<i>SERIAL_R</i> [0 a 1]: Variables de sistema de solo lectura de línea serie	19
<i>SERIAL_W</i> [0 a 1]: Variables de sistema de lectura/escritura de línea serie	20
Estructuras de <i>ETH_R</i> y <i>ETH_W</i>	21
<i>ETH_R</i> : variables de sistema de solo lectura del puerto Ethernet	21
<i>ETH_W</i> : variables de sistema de lectura/escritura del puerto Ethernet	24
Estructura de <i>TM3_MODULE_R</i>	24
<i>TM3_MODULE_R</i> [0...13]: variables del sistema de solo lectura de los módulos TM3	24
Estructura de <i>TM3_BUS_W</i>	25
<i>TM3_BUS_W</i> : variables de sistema de bus TM3	25
Estructura de <i>PROFIBUS_R</i>	25
<i>PROFIBUS_R</i> : variables de sistema de solo lectura de PROFIBUS	25
Estructura de <i>CART_R</i>	26
<i>CART_R_STRUCT</i> : Variables de sistema de solo lectura del cartucho	26
Funciones de sistema M241,	27
Funciones de lectura M241	27
<i>GetImmediateFastInput</i> : leer entrada de una E/S experta incrustada	27
<i>GetRtc</i> : obtener el reloj en tiempo real	28
<i>HasForcedIo</i> : indica si una entrada o una salida es forzosa	28
<i>IsFirstMastColdCycle</i> : Indicar si este ciclo es el primer ciclo MAST del arranque en frío	29
<i>IsFirstMastCycle</i> : indicar si este ciclo es el primer ciclo MAST	30
<i>IsFirstMastWarmCycle</i> : indicar si este ciclo es el primer ciclo MAST del arranque en caliente	31
Funciones de escritura de M241	32
<i>InhibitBatLed</i> : Activa o desactiva el indicador LED de la batería	32
<i>PhysicalWriteFastOutputs</i> : escribir salida rápida de una E/S experta incrustada	33

<i>SetRTCDrift</i> : Establecer el valor de compensación en el RTC	34
Funciones de usuario de M241	35
<i>FB_ControlClone</i> : clonar el controlador.....	35
<i>DataFileCopy</i> : copiar comandos de archivo.....	36
<i>ExecuteScript</i> : ejecutar comandos de script	38
Funciones de espacio en disco de M241	39
<i>FC_GetFreeDiskSpace</i> : Obtiene el espacio libre de la memoria	39
<i>FC_GetLabel</i> : Obtiene la etiqueta de memoria	40
<i>FC_GetTotalDiskSpace</i> : Obtiene el tamaño de la memoria.....	41
Funciones de lectura TM3	42
<i>TM3_GetModuleBusStatus</i> : obtener estado del bus del módulo TM3	42
<i>TM3_GetModuleFWVersion</i> : Obtener la versión de firmware del módulo TM3	43
<i>TM3_GetModuleInternalStatus</i> : obtener estado interno del módulo TM3	43
Tipos de datos de la biblioteca PLCSystem de M241	46
Tipos de datos de variables de sistema de <i>PLC_RW</i>	46
<i>PLC_R_APPLICATION_ERROR</i> : códigos de estado de error detectado de la aplicación.....	47
<i>PLC_R_BOOT_PROJECT_STATUS</i> : códigos de estado del proyecto de arranque	48
<i>PLC_R_IO_STATUS</i> : códigos de estado de E/S	48
<i>PLC_R_SDCARD_STATUS</i> : códigos de estado de la ranura de la tarjeta SD	49
<i>PLC_R_STATUS</i> : códigos de estado del controlador	49
<i>PLC_R_STOP_CAUSE</i> : códigos de causa de transición de estado RUN a otro estado.....	50
<i>PLC_R_TERMINAL_PORT_STATUS</i> : códigos de estado de la conexión del puerto del programación	51
<i>PLC_R_TM3_BUS_STATE</i> : códigos de estado del bus TM3	51
<i>PLC_W_COMMAND</i> : códigos de comando de control	51
Tipos de datos de variables de sistema de <i>DataFileCopy</i>	52
<i>DataFileCopyError</i> : códigos de error detectado	52
<i>DataFileCopyLocation</i> : códigos de ubicación	52
Tipos de datos de variables de sistema de <i>ExecScript</i>	52
<i>ExecuteScriptError</i> : códigos de error detectado	53
Tipos de datos de variables de sistema de <i>ETH_RW</i>	53
<i>ETH_R_FRAME_PROTOCOL</i> : códigos de protocolo de transmisión de tramas	53
<i>ETH_R_IP_MODE</i> : códigos de origen de la dirección IP	53
<i>ETH_R_PORT_DUPLEX_STATUS</i> : códigos de modo de transmisión.....	54
<i>ETH_R_PORT_IP_STATUS</i> : códigos de estado del puerto TCP/ IP Ethernet	54
<i>ETH_R_PORT_LINK_STATUS</i> : códigos de estado de la conexión de comunicación.....	54
<i>ETH_R_PORT_SPEED</i> : códigos de velocidad de comunicación del puerto Ethernet.....	55

<i>ETH_R_RUN_IDLE</i> : códigos de estado de ejecución e inactividad Ethernet/IP	55
Tipos de datos de variables de sistema de <i>TM3_MODULE_RW</i>	55
<i>TM3_ERR_CODE</i> : códigos de error detectado del módulo de extensión TM3	56
<i>TM3_MODULE_R_ARRAY_TYPE</i> : tipo de matriz de lectura del módulo de extensión TM3	56
<i>TM3_MODULE_STATE</i> : códigos de estado del módulo de extensión TM3	56
<i>TM3_BUS_W_IOBUSERRMOD</i> : modalidad de error del bus TM3	57
Tipos de datos de variables de sistema del cartucho	57
<i>CART_R_ARRAY_TYPE</i> : Tipo de matriz de lectura del cartucho	57
<i>CART_R_MODULE_ID</i> : Identificador del módulo de lectura del cartucho	57
<i>CART_R_STATE</i> : Estado de lectura del cartucho	58
Tipos de datos de funciones de sistema	58
<i>IMMEDIATE_ERR_TYPE</i> : <i>GetImmediateFastInput</i> leer entrada de códigos de E/S expertas incrustadas	58
<i>RTCSETDRIFT_ERROR</i> : Códigos de error detectados de la función <i>SetRTCDrift</i>	58
Apéndices	59
Representación de funciones y de bloques de funciones	60
Diferencias entre una función y un bloque de funciones	60
Usar una función o un bloque de funciones en lenguaje IL	61
Usar una función o un bloque de funciones en lenguaje ST	64
Glosario	67
Índice	74

Información de seguridad

Información importante

Lea atentamente estas instrucciones y observe el equipo para familiarizarse con el dispositivo antes de instalarlo, utilizarlo, revisarlo o realizar su mantenimiento. Los mensajes especiales que se ofrecen a continuación pueden aparecer a lo largo de la documentación o en el equipo para advertir de peligros potenciales, o para ofrecer información que aclara o simplifica los distintos procedimientos.



La inclusión de este icono en una etiqueta "Peligro" o "Advertencia" indica que existe un riesgo de descarga eléctrica, que puede provocar lesiones si no se siguen las instrucciones.



Éste es el icono de alerta de seguridad. Se utiliza para advertir de posibles riesgos de lesiones. Observe todos los mensajes que siguen a este icono para evitar posibles lesiones o incluso la muerte.

⚠ PELIGRO
PELIGRO indica una situación de peligro que, si no se evita, provocará lesiones graves o incluso la muerte.
⚠ ADVERTENCIA
ADVERTENCIA indica una situación de peligro que, si no se evita, podría provocar lesiones graves o incluso la muerte.
⚠ ATENCIÓN
ATENCIÓN indica una situación peligrosa que, si no se evita, podría provocar lesiones leves o moderadas.
AVISO
AVISO indica una situación potencialmente peligrosa que, si no se evita, puede provocar daños en el equipo.

Tenga en cuenta

La instalación, manejo, puesta en servicio y mantenimiento de equipos eléctricos deberán ser realizados sólo por personal cualificado. Schneider Electric no se hace responsable de ninguna de las consecuencias del uso de este material.

Una persona cualificada es aquella que cuenta con capacidad y conocimientos relativos a la construcción, el funcionamiento y la instalación de equipos eléctricos, y que ha sido formada en materia de seguridad para reconocer y evitar los riesgos que conllevan tales equipos.

Acerca de este libro

Presentación

Esta documentación le permitirá familiarizarse con las funciones y las variables del sistema ofrecidas en Modicon M241 Logic Controller. La biblioteca M241 PLCSystem contiene funciones y variables para obtener información y enviar comandos al sistema del controlador.

En este documento se describen las funciones y las variables de tipos de datos de la biblioteca M241 PLCSystem.

Se requieren los conocimientos siguientes:

- Información básica sobre la funcionalidad, la estructura y la configuración de M241 Logic Controller
- Programación en lenguaje FBD, LD, ST, IL o CFC
- Variables del sistema (variables globales)

Nota de validez

Para obtener más información sobre la validez del presente documento, consulte la ayuda en línea del producto.

Idiomas disponibles de este documento

Este documento está disponible en los siguientes idiomas:

- Inglés (EIO0000003065)
- Francés (EIO0000003066)
- Alemán (EIO0000003067)
- Español (EIO0000003068)
- Italiano (EIO0000003069)
- Chino (EIO0000003070)

Documentos relacionados

Título de la documentación	Número de referencia
EcoStruxure Automation Expert - Motion, EcoStruxure Machine Expert - Guía de programación	EIO0000002854 (ENG)
	EIO0000002855 (FRE)
	EIO0000002856 (GER)
	EIO0000002858 (SPA)
	EIO0000002857 (ITA)
	EIO0000002859 (CHS)
Modicon M241 Logic Controller - Guía de hardware	EIO0000003083 (ENG)
	EIO0000003084 (FRE)
	EIO0000003085 (GER)
	EIO0000003086 (SPA)
	EIO0000003087 (ITA)
	EIO0000003088 (CHS)
Modicon M241 Logic Controller - Guía de programación	EIO0000003059 (ENG)
	EIO0000003060 (FRE)
	EIO0000003061 (GER)
	EIO0000003062 (SPA)
	EIO0000003063 (ITA)
	EIO0000003064 (CHS)

Para consultar documentos en línea, visite el centro de descargas de Schneider Electric (www.se.com/ww/en/download/).

Información relacionada con el producto

▲ ADVERTENCIA

PÉRDIDA DE CONTROL

- Realice un análisis de efecto o de modalidad de fallo (FMEA), o un análisis de riesgo equivalente, de su aplicación y aplique controles preventivos y de detección antes de la implementación.
- Proporcione un estado de recuperación para los eventos o las secuencias de control no deseados.
- Proporcione rutas de control separadas o redundantes donde se necesiten.
- Proporcione los parámetros adecuados, en especial respecto a límites.
- Revise las implicaciones de los retrasos en la transmisión y tome medidas para mitigarlos.
- Revise las implicaciones de las interrupciones del enlace de comunicación y tome medidas para mitigarlas.
- Proporcione rutas independientes para las funciones de control (por ejemplo, parada de emergencia, condiciones de superación de los límites y condiciones de error) de acuerdo con su evaluación de riesgos y con los códigos y normativas aplicables.
- Aplique las regulaciones y directrices locales de seguridad y prevención de accidentes.¹
- Realice pruebas de todas las implementaciones de un sistema para verificar que funcione correctamente antes de ponerlas en servicio.

Si no se siguen estas instrucciones, pueden producirse lesiones graves, muerte o daños en el equipo.

¹ Para obtener información adicional, consulte NEMA ICS 1.1 (última edición), *Safety Guidelines for the Application, Installation, and Maintenance of Solid State Control* (Directrices de seguridad para la aplicación, la instalación y el mantenimiento del control de estado estático) y NEMA ICS 7.1 (última edición), *Safety Standards for Construction and Guide for Selection, Installation and Operation of Adjustable-Speed Drive Systems* (Estándares de seguridad para la construcción y guía para la selección, instalación y utilización de sistemas de unidades de velocidad ajustable) o su equivalente aplicable a la ubicación específica.

▲ ADVERTENCIA

FUNCIONAMIENTO IMPREVISTO DEL EQUIPO

- Utilice sólo software aprobado por Schneider Electric para este equipo.
- Actualice el programa de aplicación siempre que cambie la configuración de hardware física.

Si no se siguen estas instrucciones, pueden producirse lesiones graves, muerte o daños en el equipo.

Información sobre terminología no inclusiva o insensible

Como empresa responsable e inclusiva, Schneider Electric actualiza constantemente sus comunicaciones y productos que contienen terminología no inclusiva o insensible. Sin embargo, a pesar de estos esfuerzos, nuestro contenido aún puede contener términos que algunos clientes consideren inapropiados.

Normas y términos utilizados

Los términos técnicos, símbolos y las descripciones correspondientes de esta información o que aparecen en la parte interior o exterior de los propios productos se derivan, por lo general, de los términos y las definiciones de estándares internacionales.

En el área de sistemas de seguridad funcional, unidades y automatización general, se incluyen, pero sin limitarse a ellos, términos como *seguridad*, *función de seguridad*, *estado de seguridad*, *fallo*, *reinicio tras fallo*, *avería*, *funcionamiento incorrecto*, *error*, *mensaje de error*, *peligroso*, etc.

Entre estas normas se incluyen:

Norma	Descripción
IEC 61131-2:2007	Controladores programables, parte 2: Requisitos y ensayos de los equipos.
ISO 13849-1:2023	Seguridad de la maquinaria: Componentes de los sistemas de control relacionados con la seguridad. Principios generales del diseño.
EN 61496-1:2020	Seguridad de la maquinaria: Equipos de protección electrosensibles. Parte 1: Pruebas y requisitos generales.
ISO 12100:2010	Seguridad de las máquinas. Principios generales del diseño. Evaluación del riesgo y reducción del riesgo
EN 60204-1:2006	Seguridad de la maquinaria - Equipo eléctrico de las máquinas - Parte 1: Requisitos generales
ISO 14119:2013	Seguridad de la maquinaria. Dispositivos de bloqueo asociados con protecciones: principios de diseño y selección
ISO 13850:2015	Seguridad de la maquinaria. Parada de emergencia: principios de diseño
IEC 62061:2021	Seguridad de la maquinaria. Seguridad funcional de sistemas de control eléctricos, electrónicos y electrónicos programables relacionados con la seguridad
IEC 61508-1:2010	Seguridad funcional de los sistemas eléctricos/electrónicos/electrónicos programables relacionados con la seguridad: Requisitos generales.
IEC 61508-2:2010	Seguridad funcional de los sistemas eléctricos/electrónicos/electrónicos programables relacionados con la seguridad: Requisitos para los sistemas eléctricos/electrónicos/electrónicos programables relacionados con la seguridad.
IEC 61508-3:2010	Seguridad funcional de los sistemas eléctricos/electrónicos/electrónicos programables relacionados con la seguridad: Requisitos de software.
IEC 61784-3:2021	Redes de comunicaciones industriales - Perfiles - Parte 3: Buses de campo de seguridad funcionales - Reglas generales y definiciones de perfiles.
2006/42/EC	Directiva de maquinaria
2014/30/EU	Directiva de compatibilidad electromagnética
2014/35/EU	Directiva de baja tensión

Además, los términos utilizados en este documento se pueden usar de manera tangencial porque se obtienen de otros estándares como:

Norma	Descripción
Serie IEC 60034	Máquinas eléctricas giratorias
Serie IEC 61800	Sistemas de variadores eléctricos de velocidad ajustable
Serie IEC 61158	Comunicación digital de datos para la medición y control: bus de campo para su uso en sistemas de control industriales

Por último, el término *zona de funcionamiento* se puede utilizar junto con la descripción de peligros específicos, y se define como tal para una *zona de peligro* o *zona peligrosa* en la *Directiva de maquinaria (2006/42/EC)* y *ISO 12100:2010*.

NOTA: Los estándares mencionados anteriormente podrían o no aplicarse a los productos específicos citados en la presente documentación. Para obtener más información en relación con los diferentes estándares aplicables a los productos descritos en este documento, consulte las tablas de características de las referencias de dichos productos.

M241 Variables del sistema

Descripción general

En este capítulo:

- Se proporciona una introducción a las variables de sistema, página 13.
- Se describen las variables de sistema, página 16 incluidas en la biblioteca PLCSystem de M241.

Variables del sistema: definición y uso

Descripción general

En esta sección se definen las variables del sistema y cómo implementarlas en Modicon M241 Logic Controller.

Descripción de las variables de sistema

Introducción

En esta sección se describe el modo en que se implementan las variables de sistema. Las variables de sistema:

- Permiten acceder a información general del sistema, realizar diagnósticos del sistema y controlar acciones sencillas.
- Son variables estructuradas que cumplen las definiciones y las convenciones de nomenclatura de IEC 61131-3. Puede acceder a las variables de sistema con el nombre simbólico de IEC *PLC_GVL*. Algunas de las variables de *PLC_GVL* son de solo lectura (por ejemplo, *PLC_R*) y otras son de lectura/escritura (por ejemplo, *PLC_W*).
- Se declaran automáticamente como variables globales. Tienen alcance en todo el sistema y cualquier unidad de organización del programa (POU) puede acceder a ellas en cualquier tarea.

Convención sobre nomenclatura

Las variables de sistema se identifican mediante:

- Un nombre de estructura que representa la categoría de la variable de sistema. Por ejemplo, representa un nombre de estructura de variables de solo lectura utilizado para el diagnóstico del controlador.
- Un conjunto de nombres de componentes que identifica el objetivo de la variable. Por ejemplo, *i_wVendorID* representa el ID del proveedor del controlador.

Puede acceder a las variables de sistema escribiendo el nombre de estructura de las variables seguido del nombre del componente.

Aquí tiene un ejemplo de implementación de variables de sistema:

```
VAR
myCtr_Serial : DWORD;
myCtr_ID : DWORD;
myCtr_FramesRx : UDINT;
END_VAR
myCtr_Serial := PLC_GVL.PLC_R.i_dwSerialNumber;
myCtr_ID := PLC_GVL.PLC.R.i_wVendorID;
```

```
myCtr_FramesRx := SERIAL_R[0].i_udiFramesReceivedOK
```

NOTA: El nombre completo de la variable de sistema en el ejemplo anterior es `PLC_GVL.PLC_R`. El `PLC_GVL` está implícito al declarar una variable mediante la opción de **Accesibilidad**, pero también puede introducirse con el prefijo. Las buenas prácticas de programación suelen dictar la utilización de nombres de variables completos en las declaraciones.

Ubicación de las variables de sistema

Se definen dos variables de sistema para su uso al programar el controlador:

- Variables ubicadas
- Variables no ubicadas

Se utilizan en programas de EcoStruxure Automation Expert - Motion y EcoStruxure Machine Expert según la convención `structure_name.component_name` explicada anteriormente.

Se puede acceder directamente a las direcciones %MW de 0 a 59 999. EcoStruxure Automation Expert - Motion y EcoStruxure Machine Expert consideran que las direcciones mayores están fuera de rango y solo son accesibles mediante la convención `structure_name.component_name`.

Las variables ubicadas:

- Tienen una ubicación fija en un área de %MW estática: de %MW60000 a %MW60199 para variables de sistema de solo lectura.
- se pueden acceder mediante Modbus TCP, serie Modbus y peticiones de EtherNet/IP en los estados RUNNING y STOPPED.

Las variables no ubicadas:

- No están ubicadas físicamente en el área %MW.
- no se pueden acceder a través de ninguna petición de bus de campo o de red a menos que las localice en la tabla de reubicación y solo entonces se puede acceder a ellas en los estados RUNNING y STOPPED. La tabla de reubicación utiliza las siguientes áreas %MW dinámicas:
 - De %MW60200 a %MW61999 para variables de solo lectura
 - De %MW62200 a %MW63999 para variables de lectura/escritura

Utilización de las variables de sistema

Introducción

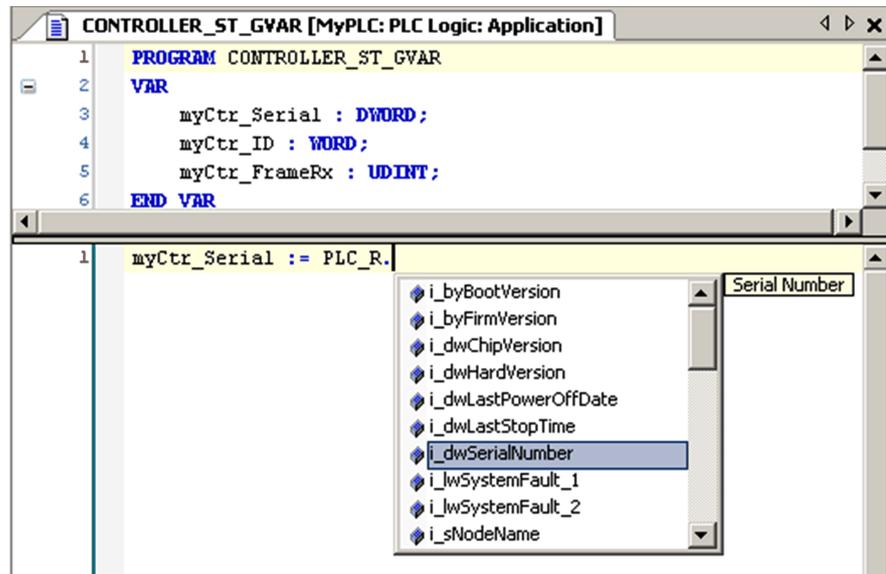
En este apartado se describen los pasos necesarios para programar y usar las variables de sistema en EcoStruxure Automation Expert - Motion y EcoStruxure Machine Expert.

Las variables de sistema son globales en lo que respecta al ámbito de la aplicación y se pueden utilizar en todas las unidades de organización de programa (POU) de la aplicación.

No es necesario declarar las variables de sistema en la Lista de variables globales (GVL). Se declaran automáticamente desde la biblioteca del sistema del controlador.

Utilización de las variables de sistema en una POU

EcoStruxure Automation Expert - Motion y EcoStruxure Machine Expert tienen una función de finalización automática. En una **POU**, empiece por especificar el nombre de estructura de la variable de sistema (*PLC_R*, *PLC_W*...) seguido de un punto. Las variables de sistema se muestran en la **Accesibilidad**. Puede seleccionar la variable que desee o especificar el nombre completo manualmente.



NOTA: En el ejemplo anterior, tras introducir el nombre de estructura *PLC_R*., EcoStruxure Automation Expert - Motion y EcoStruxure Machine Expert ofrecen un menú emergente de nombres/variables de componentes posibles.

Ejemplo

En el siguiente ejemplo se muestra el uso de algunas variables de sistema:

```

VAR
myCtr_Serial : DWORD;
myCtr_ID : WORD;
myCtr_FramesRx : UDINT;
END_VAR
myCtr_Serial := PLC_R.i_dwSerialNumber;
myCtr_ID := PLC_R.i_wVendorID;
myCtr_FramesRx := SERIAL_R[0].i_udiFramesReceivedOK;

```

Estructuras de *PLC_R* y *PLC_W*

Descripción general

En este apartado se describen las diferentes variables de sistema incluidas en las estructuras *PLC_R* y *PLC_W*.

PLC_R: variables de sistema de solo lectura del controlador

Estructura de la variable

En esta tabla se describen los parámetros de la variable de sistema *PLC_R* (tipo *PLC_R_STRUCT*):

Dirección Modbus ⁽¹⁾	Nombre de variable	Tipo	Comentario
60000	<i>i_wVendorID</i>	WORD	ID de proveedor del controlador. 101A hex = Schneider Electric
60 001	<i>i_wProductID</i>	WORD	ID de referencia del controlador. NOTA: ID de proveedor e ID de referencia son los componentes del ID del destino del controlador mostrado en la vista de la configuración de comunicación (ID del destino = 101A XXXX hex).
60002	<i>i_dwSerialNumber</i>	DWORD	Número de serie del controlador
60004	<i>i_byFirmVersion</i>	ARRAY[0..3] OF BYTE	Versión de firmware del controlador [aa.bb.cc.dd]: <ul style="list-style-type: none"> • <i>i_byFirmVersion</i>[0] = aa • ... • <i>i_byFirmVersion</i>[3] = dd
60 006	<i>i_byBootVersion</i>	ARRAY[0..3] OF BYTE	Versión de arranque del controlador [aa.bb.cc.dd]: <ul style="list-style-type: none"> • <i>i_byBootVersion</i>[0] = aa • ... • <i>i_byBootVersion</i>[3] = dd
60008	<i>i_dwHardVersion</i>	DWORD	Versión de hardware del controlador. NOTA: Parámetro reservado únicamente para uso interno. Para conocer la versión del producto (PV), consulte la etiqueta del producto.
60010	<i>i_dwChipVersion</i>	DWORD	Versión de coprocesador del controlador.
60012	<i>i_wStatus</i>	<i>PLC_R_STATUS</i> , página 49	Estado del controlador.
60013	<i>i_wBootProjectStatus</i>	<i>PLC_R_BOOT_PROJECT_STATUS</i> , página 48	Devuelve información sobre la aplicación de arranque almacenada en la memoria no volátil.
60014	<i>i_wLastStopCause</i>	<i>PLC_R_STOP_CAUSE</i> , página 50	Causa de la última transición desde el estado <i>RUN</i> a otro estado.
60015	<i>i_wLastApplicationError</i>	<i>PLC_R_APPLICATION_ERROR</i> , página 47	Causa de la última excepción del controlador.
60016	<i>i_lwSystemFault_1</i>	LWORD	El campo de bit FFFF FFFF FFFF FFFF hex indica que no se ha detectado ningún error. Un bit en nivel bajo significa que se ha detectado un error: <ul style="list-style-type: none"> • bit 0 = Error detectado de E/S expertas • bit 1 = Error detectado de TM3 • Bit 2 = Error detectado de Ethernet IF1 • bit 3 = Error detectado de Ethernet IF2 • bit 4 = Error detectado de sobrecorriente de serie 1 • Bit 5 = Error detectado de serie 2 • bit 6 = Error detectado de CAN 1 • Bit 7 = Error detectado de cartucho 1 • Bit 8 = Error detectado de cartucho 2 • bit 9 = Error detectado de TM4 • Bit 10 = Error detectado de tarjeta SD • Bit 11 = Error detectado de servidor de cortafuegos • Bit 12 = Error detectado del servidor DHCP • Bit 13 = Error detectado de servidor OPC UA

Dirección Modbus (1)	Nombre de variable	Tipo	Comentario
60020	<i>i_lwSystemFault_2</i>	LWORD	El campo de bit FFFF hex indica que no se ha detectado ningún error. Si <i>i_wIOStatus1</i> = <i>PLC_R_IO_SHORTCUT_FAULT</i> , el significado de <i>i_lwSystemFault_2</i> es: <ul style="list-style-type: none"> Bit 0 = 0: se ha detectado un cortocircuito en el grupo de salidas 0 (de Q0 a Q1) Bit 1 = 0: se ha detectado un cortocircuito en el grupo de salidas 1 (de Q2 a Q3) Bit 2 = 0: se ha detectado un cortocircuito en el grupo de salidas 2 (de Q4 a Q7) Bit 3 = 0: se ha detectado un cortocircuito en el grupo de salidas 3 (de Q8 a Q11) Bit 4 = 0: se ha detectado un cortocircuito en el grupo de salidas 4 (de Q12 a Q15)
60024	<i>i_wIOStatus1</i>	<i>PLC_R_IO_STATUS</i> , página 48	Estado de E/S experta incrustada.
60025	<i>i_wIOStatus2</i>	<i>PLC_R_IO_STATUS</i> , página 48	Estado de E/S de TM3.
60026	<i>i_wClockBatterystatus</i>	WORD	Estado de la batería del RTC: <ul style="list-style-type: none"> 0 = Se requiere cambio de batería 100 = Batería completamente cargada Otros valores (de 1 a 99) representa el porcentaje de la carga. Por ejemplo, si el valor es 75, representa que la carga de la batería es del 75%.
60028	<i>i_dwAppliSignature1</i>	DWORD	Primera DWORD de la firma de 4 DWORD (16 bytes en total). El software genera la firma de la aplicación durante la compilación.
60030	<i>i_dwAppliSignature2</i>	DWORD	Segunda DWORD de la firma de 4 DWORD (16 bytes en total). El software genera la firma de la aplicación durante la compilación.
60032	<i>i_dwAppliSignature3</i>	DWORD	Tercera DWORD de la firma de 4 DWORD (16 bytes en total). El software genera la firma de la aplicación durante la compilación.
60034	<i>i_dwAppliSignature4</i>	DWORD	Cuarta DWORD de la firma de 4 DWORD (16 bytes en total). El software genera la firma de la aplicación durante la compilación.
n/a	<i>i_sVendorName</i>	STRING(31)	Nombre del fabricante: "Schneider Electric".
n/a	<i>i_sProductRef</i>	STRING(31)	Referencia del controlador.
n/a	<i>i_sNodeName</i>	STRING(99)	Nombre de nodo en la red de software.
n/a	<i>i_dwLastStopTime</i>	DWORD	La hora de la última detención (STOP) detectada en segundos, comenzando por el 1 de enero de 1970 a las 00:00 UTC.
n/a	<i>i_dwLastPowerOffDate</i>	DWORD	La fecha y la hora del último apagado detectado en segundos, comenzando por el 1 de enero de 1970 a las 00:00 UTC. NOTA: Convierta este valor en fecha y hora mediante la función <i>SysTimeRtcConvertUtcToDate</i> . Para obtener más información sobre la conversión de fecha y hora, consulte la Guía de la biblioteca <i>Systime</i> (consulte Obtención y ajuste del reloj en tiempo real, Guía de la biblioteca de <i>SysTimeRtc</i> y <i>SysTimeCore</i>)
n/a	<i>i_uiEventsCounter</i>	UINT	Número de eventos externos detectados en las entradas configuradas para detección de eventos externos desde el último arranque en frío. Restablecimiento mediante un arranque en frío o con el comando <i>PLC_W.q_wResetCounterEvent</i> .
n/a	<i>i_wTerminalPortStatus</i>	<i>PLC_R_TERMINAL_PORT_STATUS</i> , página 51	Estado del puerto de programación USB (USB Mini-B).

Dirección Modbus ⁽¹⁾	Nombre de variable	Tipo	Comentario
n/a	<i>i_wSdCardStatus</i>	<i>PLC_R_SDCARD_STATUS</i> , página 49	Estado de la tarjeta SD.
n/a	<i>i_wUsrFreeFileHdl</i>	WORD	Número de controladores de archivo disponibles, incluidos los sockets y tubos a nivel de sistema. Un controlador de archivos es el recurso asignado por el sistema cuando se abre un archivo. NOTA: Este valor no indica el número de controladores de archivo disponibles para la aplicación.
n/a	<i>i_udiUsrFsTotalBytes</i>	UDINT	Tamaño total de la memoria del sistema de archivos del usuario (en bytes). Se trata del tamaño de la memoria no volátil del directorio <i>/usr/</i> .
n/a	<i>i_udiUsrFsFreeBytes</i>	UDINT	Tamaño de memoria libre del sistema de archivos del usuario (en bytes).
n/a	<i>i_uiTM3BusState</i>	<i>PLC_R_TM3_BUS_STATE</i> , página 51	Estado de bus TM3. <i>i_uiTM3BusState</i> puede tener los siguientes valores: <ul style="list-style-type: none"> • 1: TM3_CONF_ERROR La configuración no coincide entre la configuración física y la configuración de software. • 3: TM3_OK La configuración física coincide con la configuración del software. • 4: TM3_POWER_SUPPLY_ERROR El bus TM3 no recibe alimentación (por ejemplo, cuando el Logic Controller recibe alimentación por USB).
n/a	<i>i_ExpertIO_RunStop_Input</i>	BYTE	La ubicación de la entrada Run/Stop es: <ul style="list-style-type: none"> • 16 a FF hex si la E/S experta no se ha configurado • 0 para <i>%IX0.0</i> • 1 para <i>%IX0.1</i> • 2 para <i>%IX0.2</i> • ... etc.
n/a	<i>i_x10msClk</i>	BOOL	Bit de TimeBase de 10 ms. Esta variable alterna On/Off con un período = 10 ms. El valor se alterna cuando el Logic Controller se encuentra en el estado Stop y Run.
n/a	<i>i_x100msClk</i>	BOOL	Bit de TimeBase de 100 ms. Esta variable alterna On/Off con un período = 100 ms. El valor se alterna cuando el controlador lógico se encuentra en el estado de Parada y Ejecución.
n/a	<i>i_x1sClk</i>	BOOL	Bit de TimeBase de 1 s. Esta variable alterna On/Off con un período = 1 ms. El valor se alterna cuando el controlador lógico se encuentra en el estado de Parada y Ejecución.
(1) significa que no se puede acceder a la dirección Modbus a través de la aplicación.			
n/d significa que no hay ninguna asignación de dirección Modbus predefinida para esta variable de sistema.			

PLC_W: variables de sistema de lectura/escritura del controlador

Estructura de la variable

En esta tabla se describen los parámetros de la variable de sistema *PLC_W* (tipo *PLC_W_STRUCT*):

%MW	Nombre de variable	Tipo	Comentario
n/d	<i>q_wResetCounterEvent</i>	WORD	La transición de 0 a 1 restablece el contador de eventos (<i>PLC_R.i_uiEventsCounter</i>). Para volver a restablecer el contador, es necesario escribir 0 en esta variable antes de que pueda llevarse a cabo otra transición de 0 a 1.
n/d	<i>q_uiOpenPLCControl</i>	UINT	Cuando el valor de la variable pasa de 0 a 6699, se ejecuta el comando escrito previamente en el siguiente <i>PLC_W.q_wPLCControl</i> .
n/d	<i>q_wPLCControl</i>	<i>PLC_W_COMMAND</i> , página 51	Los comandos RUN/STOP del controlador se ejecutan cuando el valor de la variable de sistema <i>PLC_W.q_uiOpenPLCControl</i> pasa de 0 a 6699.
n/d significa que no existe ninguna asignación de %MW predefinida para esta variable de sistema.			

Estructuras de SERIAL_R y SERIAL_W

Descripción general

En este apartado se enumeran y describen las diversas variables de sistema incluidas en las estructuras *SERIAL_R* y *SERIAL_W*.

SERIAL_R[0 a 1]: Variables de sistema de solo lectura de línea serie

Introducción

SERIAL_R es una matriz de dos tipos *SERIAL_R_STRUCT*. Cada elemento de la matriz devuelve las variables de sistema de diagnóstico para la línea serie correspondiente.

En el caso de M241 Logic Controller:

- *Serial_R[0]* hace referencia a la línea serie 1
- *Serial_R[1]* hace referencia a la línea serie 2

Estructura de la variable

En esta tabla se describen los parámetros de las variables de sistema de *SERIAL_R[0...1]*:

%MW	Nombre de variable	Tipo	Comentario
Línea serie			
n/d	<i>i_udiFramesTransmittedOK</i>	UDINT	Número de tramas transmitidas correctamente.
n/d	<i>i_udiFramesReceivedOK</i>	UDINT	Número de tramas recibidas sin errores detectados.
n/d	<i>i_udiRX_MessagesError</i>	UINT	Número de tramas recibidas con errores detectados (suma de comprobación, paridad).
Específico de Modbus			
n/d	<i>i_uiSlaveExceptionCount</i>	UINT	Número de respuestas de excepción de Modbus devueltas por el controlador lógico.
n/d	<i>i_udiSlaveMsgCount</i>	UINT	Número de mensajes recibidos desde el maestro y dirigidos al controlador lógico.
n/d	<i>i_uiSlaveNoRespCount</i>	UINT	Número de peticiones de difusión de Modbus recibidas por el controlador lógico.
n/d	<i>i_uiSlaveNakCount</i>	UINT	No se utiliza
n/d	<i>i_uiSlaveBusyCount</i>	UINT	No se utiliza
n/d	<i>i_uiCharOverrunCount</i>	UINT	Número de desbordes de caracteres.
<p>n/d significa que no existe ninguna asignación de %MW predefinida para esta variable de sistema.</p> <p>No se utiliza significa que la variable no está gestionada por el sistema y que, si el valor de la variable es diferente de cero, debe considerarse externa.</p>			

Los contadores de *SERIAL_R* se restablecen en los siguientes casos:

- Descarga
- Restablecimiento del controlador.
- Comando *SERIAL_W[x].q_wResetCounter*
- Comando de restablecimiento mediante el número 8 del código de la función de petición Modbus.

SERIAL_W[0 a 1]: Variables de sistema de lectura/escritura de línea serie

Introducción

SERIAL_W es una matriz de dos tipos *SERIAL_W_STRUCT*. Cada elemento de la matriz restablece las variables de sistema *SERIAL_R* para restablecer la línea serie correspondiente.

En el caso de M241 Logic Controller:

- *Serial_W[0]* hace referencia a la línea serie 1
- *Serial_W[1]* hace referencia a la línea serie 2

Estructura de la variable

En esta tabla se describen los parámetros de la variable de sistema *SERIAL_W* [0...1]:

%MW	Nombre de variable	Tipo	Comentario
n/d	<i>q_wResetCounter</i>	WORD	La transición de 0 a 1 restablece todos los contadores de <i>SERIAL_R</i> [0...1]. Para volver a restablecer los contadores, es necesario escribir esta variable en 0 antes de que pueda llevarse a cabo otra transición de 0 a 1.
n/d significa que no existe ninguna asignación de %MW predefinida para esta variable de sistema.			

Estructuras de *ETH_R* y *ETH_W*

Descripción general

En este apartado se describen las diferentes variables de sistema incluidas en las estructuras *ETH_R* y *ETH_W*.

ETH_R: variables de sistema de solo lectura del puerto Ethernet

Estructura de la variable

En esta tabla se describen los parámetros de la variable de sistema *ETH_R* (tipo *ETH_R_STRUCT*):

%MW	Nombre de variable	Tipo	Comentario
60050	<i>i_byIPAddress</i>	ARRAY[0..3] OF BYTE	Dirección IP [aaa.bbb.ccc.ddd]: <ul style="list-style-type: none"> <i>i_byIPAddress</i>[0] = aaa ... <i>i_byIPAddress</i>[3] = ddd
60052	<i>i_bySubNetMask</i>	ARRAY[0..3] OF BYTE	Máscara de subred [aaa.bbb.ccc.ddd]: <ul style="list-style-type: none"> <i>i_bySub-netMask</i>[0] = aaa ... <i>i_bySub-netMask</i>[3] = ddd
60054	<i>i_byGateway</i>	ARRAY[0..3] OF BYTE	Dirección de pasarela [aaa.bbb.ccc.ddd]: <ul style="list-style-type: none"> <i>i_byGateway</i>[0] = aaa ... <i>i_byGateway</i>[3] = ddd
60056	<i>i_byMACAddress</i>	ARRAY[0..5] OF BYTE	Dirección MAC [aa.bb.cc.dd.ee.ff]: <ul style="list-style-type: none"> <i>i_byMACAddress</i>[0] = aa ... <i>i_byMACAddress</i>[5] = ff
60059	<i>i_sDeviceName</i>	STRING(15)	Nombre utilizado para obtener la dirección IP del servidor.
n/d	<i>i_wIpMode</i>	<i>ETH_R_IP_MODE</i> , página 53	Método utilizado para obtener una dirección IP.
n/d	<i>i_byFDRServerIPAddress</i>	ARRAY[0..3] OF BYTE	La dirección IP [aaa.bbb.ccc.ddd] del servidor DHCP o BootP: <ul style="list-style-type: none"> <i>i_byFDRServerIPAddress</i>[0] = aaa ... <i>i_byFDRServerIPAddress</i>[3] = ddd Igual a 0.0.0.0 si se usa la IP almacenada o predeterminada.
n/d	<i>i_udiOpenTcpConnections</i>	UDINT	Número de conexiones TCP abiertas.

%MW	Nombre de variable	Tipo	Comentario
n/d	<i>i_udiFramesTransmittedOK</i>	UDINT	Número de tramas transmitidas correctamente. Restablecer al encender o con el comando de restablecimiento <i>ETH_W.q_wResetCounter</i> .
n/d	<i>i_udiFramedReceivedOK</i>	UDINT	Número de tramas recibidas correctamente. Restablecer al encender o con el comando de restablecimiento <i>ETH_W.q_wResetCounter</i> .
n/d	<i>i_udiTransmitBufferErrors</i>	UDINT	Número de tramas transmitidas con errores detectados. Restablecer al encender o con el comando de restablecimiento <i>ETH_W.q_wResetCounter</i> .
n/d	<i>i_udiReceiveBufferErrors</i>	UDINT	Número de tramas recibidas con errores detectados. Restablecer al encender o con el comando de restablecimiento <i>ETH_W.q_wResetCounter</i> .
n/d	<i>i_wFrameSendingProtocol</i>	<i>ETH_R_FRAME_PROTOCOL</i> , página 53	Protocolo Ethernet configurado para el envío de tramas (IEEE 802.3 o Ethernet II).
n/d	<i>i_wPortALinkStatus</i>	<i>ETH_R_PORT_LINK_STATUS</i> , página 54	Conexión del puerto Ethernet (0 = sin conexión, 1 = conexión conectada a otro dispositivo Ethernet).
n/d	<i>i_wPortASpeed</i>	<i>ETH_R_PORT_SPEED</i> , página 55	Velocidad de red del puerto Ethernet (10 Mb/s, 100 Mb/s).
n/d	<i>i_wPortADuplexStatus</i>	<i>ETH_R_PORT_DUPLEX_STATUS</i> , página 54	Estado dúplex del puerto Ethernet (0 = semidúplex o 1 = dúplex completo).
n/d	<i>i_udiPortACollisions</i>	UDINT	Número de tramas implicadas en una o más colisiones y transmitidas a continuación correctamente. Restablecer al encender o con el comando de restablecimiento <i>ETH_W.q_wResetCounter</i> .
n/d	<i>i_byIPAddress_lf2</i>	ARRAY[0..3] OF BYTE	Dirección IP del módulo de extensión TM4.
n/d	<i>i_bySubNetMask_lf2</i>	ARRAY[0..3] OF BYTE	Máscara de subred del módulo de extensión TM4.
n/d	<i>i_byGateway_lf2</i>	ARRAY[0..3] OF BYTE	Dirección de pasarela del módulo de extensión TM4.
n/d	<i>i_byMACAddress_lf2</i>	ARRAY[0..3] OF BYTE	Dirección MAC del módulo de extensión TM4.
n/d	<i>i_sDeviceName_lf2</i>	STRING(15)	Nombre utilizado para obtener la dirección IP del módulo de extensión TM4.
n/d	<i>i_wIpMode_lf2</i>	<i>ETH_R_IP_MODE</i> , página 53	Método utilizado para obtener la dirección IP del módulo de extensión TM4.
n/d	<i>i_wPortALinkStatus_lf2</i>	<i>ETH_R_PORT_LINK_STATUS</i> , página 54	Enlace del puerto Ethernet del módulo de extensión TM4: <ul style="list-style-type: none"> • 0: Sin conexión • 1: Conexión a otro dispositivo Ethernet
n/d	<i>i_wPortASpeed_lf2</i>	<i>ETH_R_PORT_SPEED</i> , página 55	Velocidad de red del puerto Ethernet del módulo de extensión TM4 (10 Mb/s o 100 Mb/s).
n/d	<i>i_wPortADuplexStatus_lf2</i>	<i>ETH_R_PORT_DUPLEX_STATUS</i> , página 54	Estado de dúplex del puerto Ethernet del módulo de extensión TM4: <ul style="list-style-type: none"> • 0: Semi • 1: Dúplex completo
n/d	<i>i_wPortAlpStatus_lf2</i>	<i>ETH_R_PORT_IP_STATUS</i> , página 54	Estado de la pila del puerto TCP/IP Ethernet del módulo de extensión TM4.
Específico de Modbus TCP/IP			
n/d	<i>i_udiModbusMessageTransmitted</i>	UDINT	Número de mensajes Modbus transmitidos. Restablecer al encender o con el comando de restablecimiento <i>ETH_W.q_wResetCounter</i> .
n/d	<i>i_udiModbusMessageReceived</i>	UDINT	Número de mensajes Modbus recibidos. Restablecer al encender o con el comando de restablecimiento <i>ETH_W.q_wResetCounter</i> .
n/d	<i>i_udiModbusErrorMessage</i>	UDINT	Mensajes de error detectados de Modbus transmitidos y recibidos. Restablecer al encender o con el comando de restablecimiento <i>ETH_W.q_wResetCounter</i> .
Específico de EtherNet/IP			

%MW	Nombre de variable	Tipo	Comentario
n/d	<i>i_udiETHIP_IOMessagingTransmitted</i>	UDINT	Tramas EtherNet/IP Clase 1 transmitidas. Restablecer al encender o con el comando de restablecimiento <i>ETH_W.q_wResetCounter</i> .
n/d	<i>i_udiETHIP_IOMessagingReceived</i>	UDINT	Tramas EtherNet/IP Clase 1 recibidas. Restablecer al encender o con el comando de restablecimiento <i>ETH_W.q_wResetCounter</i> .
n/d	<i>i_udiUCMM_Request</i>	UDINT	Mensajes recibidos de EtherNet/IP desconectada. Restablecer al encender o con el comando de restablecimiento <i>ETH_W.q_wResetCounter</i> .
n/d	<i>i_udiUCMM_Error</i>	UDINT	Mensajes recibidos no válidos de EtherNet/IP desconectada. Restablecer al encender o con el comando de restablecimiento <i>ETH_W.q_wResetCounter</i> .
n/d	<i>i_udiClass3_Request</i>	UDINT	Solicitudes de EtherNet/IP Clase 3 recibidas. Restablecer al encender o con el comando de restablecimiento <i>ETH_W.q_wResetCounter</i> .
n/d	<i>i_udiClass3_Error</i>	UDINT	Solicitudes de EtherNet/IP Clase 3 no válidas recibidas. Restablecer al encender o con el comando de restablecimiento <i>ETH_W.q_wResetCounter</i> .
n/d	<i>i_uiAssemblyInstanceInput</i>	UINT	Número de instancia de unidad de entrada. Consulte la guía de programación correspondiente del controlador para obtener más información.
n/d	<i>i_uiAssemblyInstanceInputSize</i>	UINT	Tamaño de la instancia de unidad de entrada. Consulte la guía de programación correspondiente del controlador para obtener más información.
n/d	<i>i_uiAssemblyInstanceOutput</i>	UINT	Número de instancia de unidad de salida. Consulte la guía de programación correspondiente del controlador para obtener más información.
n/d	<i>i_uiAssemblyInstanceOutputSize</i>	UINT	Tamaño de instancia de unidad de salida. Consulte la guía de programación correspondiente del controlador para obtener más información.
n/d	<i>i_uiETHIP_ConnectionTimeouts</i>	UINT	Número de timeouts de la conexión. Restablecer al encender o con el comando de restablecimiento <i>ETH_W.q_wResetCounter</i> .
n/d	<i>i_ucEipRunIdle</i>	<i>ETH_R_RUN_IDLE</i> , página 55	Indicador de ejecución (valor=1)/inactividad (valor=0) para la conexión EtherNet/IP de Clase 1.
n/d	<i>i_byMasterIpTimeouts</i>	BYTE	Contador de eventos de timeout del maestro Ethernet Modbus TCP. Restablecer al encender o con el comando de restablecimiento <i>ETH_W.q_wResetCounter</i> .
n/d	<i>i_byMasterIpLost</i>	BYTE	Estado de la conexión del maestro Ethernet Modbus TCP: 0 = conexión correcta, 1 = conexión perdida.
n/d	<i>i_wPortIpStatus</i>	<i>ETH_R_PORT_IP_STATUS</i> , página 54	Estado de la pila del puerto TCP/IP Ethernet.
n/d significa que no existe ninguna asignación de %MW predefinida para esta variable de sistema.			

ETH_W: variables de sistema de lectura/escritura del puerto Ethernet

Estructura de la variable

En esta tabla se describen los parámetros de la variable de sistema *ETH_W* (tipo *ETH_W_STRUCT*):

%MW	Nombre de variable	Tipo	Comentario
n/d	<i>q_wResetCounter</i>	WORD	La transición de 0 a 1 restablece todos los contadores de <i>ETH_R</i> . Para volver a restablecer, es necesario escribir 0 en esta variable antes de que pueda llevarse a cabo otra transición de 0 a 1.
n/d significa que no existe ninguna asignación de %MW predefinida para esta variable de sistema.			

Estructura de *TM3_MODULE_R*

Descripción general

En esta sección se enumeran y describen las diversas variables de sistema incluidas en la estructura *TM3_MODULE_R*.

TM3_MODULE_R[0...13]: variables del sistema de solo lectura de los módulos TM3

Introducción

El *TM3_MODULE_R* es una matriz de tipo 14 *TM3_MODULE_R_STRUCT*. Cada elemento de la matriz devuelve las variables del sistema de diagnóstico para el módulo de extensión TM3 correspondiente.

En el caso de Modicon M241 Logic Controller:

- *TM3_MODULE_R[0]* hace referencia al módulo de extensión TM3 0
- ...
- *TM3_MODULE_R[13]* hace referencia al módulo de extensión TM3 13

Estructura de variables

En la siguiente tabla se describen los parámetros de la variable de sistema *TM3_MODULE_R[0...13]*:

%MW	Nombre de variable	Tipo	Comentario
n/a	<i>i_wProductID</i>	WORD	ID del módulo de extensión TM3.
n/a	<i>i_wModuleState</i>	<i>TM3_MODULE_STATE</i> , página 56	Describe el estado del módulo TM3.
n/d significa que no existe ninguna asignación de %MW predefinida para esta variable del sistema.			

Estructura de *TM3_BUS_W*

Descripción general

En esta sección se enumeran y describen las diversas variables de sistema incluidas en la estructura *TM3_BUS_W*.

TM3_BUS_W: variables de sistema de bus TM3

Estructura de la variable

En esta tabla se describen los parámetros de la variable de sistema *TM3_BUS_W* (tipo *TM3_BUS_W_STRUCT*):

Nombre de variable	Tipo	Comentario
<i>q_wIOBusErrPassiv</i>	<i>TM3_BUS_W_IOBUSERRMOD</i>	<p>Cuando está establecido en <i>ERR_ACTIVE</i> (el valor predeterminado), los errores de bus detectados en los módulos de extensión TM3 detienen los intercambios de E/S.</p> <p>Cuando está establecido en <i>ERR_PASSIVE</i>, se utiliza la gestión de errores de E/S pasivas: el controlador intenta seguir intercambiando buses de datos.</p>
<i>q_wIOBusRestart</i>	<i>TM3_BUS_W_IOBUSINIT</i>	Cuando se establece en 1, reinicia el bus de extensión de E/S. Esto solo es necesario cuando <i>q_wIOBusErrPassiv</i> está configurado en <i>ERR_ACTIVE</i> y al menos un bit de <i>TM3_MODULE_R[i].i_wModuleState</i> está configurado en <i>TM3_BUS_ERROR</i> .

Para obtener más información, consulte Descripción general de la configuración de E/S (consulte Modicon M241 Logic Controller, Guía de programación).

Estructura de *PROFIBUS_R*

PROFIBUS_R: variables de sistema de solo lectura de PROFIBUS

Estructura de la variable

En esta tabla se describen los parámetros de la variable de sistema *PROFIBUS_R* (tipo *PROFIBUS_R_STRUCT*):

%MW	Nombre de variable	Tipo	Comentario
n/a	<i>i_wPNIdentifier</i>	WORD	Código de identificación de esclavo (de 1 a 126).
n/a	<i>i_wBusAdr</i>	UINT	Dirección del esclavo PROFIBUS.
n/a	<i>i_CommState</i>	UDINT	<p>Valor que representa el estado del módulo PROFIBUS:</p> <ul style="list-style-type: none"> • 0x00: No determinable • 0x01: Sin configurar • 0x02: Parada • 0x03: Inactivo • 0x04: Funcionamiento
n/a	<i>i_CommError</i>	UDINT	Si el valor no es cero, significa que el módulo Profibus detectó un error de comunicación indicado mediante un código de error (consulte Módulos de extensión TM4 - Guía de programación).
n/a	<i>i_ErrorCount</i>	UDINT	Contador de error de comunicación.
n/d significa que no existe ninguna asignación de %MW predefinida para esta variable de sistema.			

Estructura de *CART_R*

CART_R_STRUCT: Variables de sistema de solo lectura del cartucho

Estructura de la variable

En la siguiente tabla se describen los parámetros de la variable de sistema *CART_R_STRUCT*:

%MW	Nombre de variable	Tipo	Comentario
n/d	<i>i_uiModuleId</i>	CART_R_MODULE_ID, página 57	ID del módulo
n/d	<i>i_uifirmwareVersion</i>	UINT	Versión del firmware
n/d	<i>i_udiCartState</i>	CART_R_STATE, página 58	Estado del cartucho
n/d significa que no existe ninguna asignación de %MW predefinida para esta variable de sistema.			

Funciones de sistema M241,

Descripción general

En este capítulo se describen las funciones de sistema incluidas en la biblioteca PLCSystem de M241.

Funciones de lectura M241

Descripción general

En este apartado se describen las funciones de lectura incluidas en la biblioteca PLCSystem de M241.

GetImmediateFastInput: leer entrada de una E/S experta incrustada

Descripción de la función

Esta función devuelve el valor de la entrada, que puede ser diferente del valor lógico de esa entrada. El valor se lee directamente desde el hardware en el momento de la llamada de la función. Esta función solo permite acceder a I0 hasta I7.

Representación gráfica



Representación IL y ST

Para ver la representación general en lenguaje IL o ST, consulte el capítulo *Representación de funciones y de bloques de funciones*, página 60.

Descripción de variables de E/S

En la siguiente tabla se describen las variables de entrada:

Entrada	Tipo	Comentario
<i>Block</i>	INT	No se utiliza.
<i>Input</i>	INT	Índice de entrada para leer de 0 a 7.

En la siguiente tabla se describe la variable de salida:

Salida	Tipo	Comentario
<i>GetImmediateFastInput</i>	BOOL	Valor de la entrada <Input>: FALSE/TRUE.

En la siguiente tabla se describen las variables de entrada/salida:

Entrada/salida	Tipo	Comentario
<i>Error</i>	BOOL	FALSE= funcionamiento correcto. TRUE= error de funcionamiento, la función devuelve un valor no válido.
<i>ErrID</i>	<i>IMMEDIATE_ERR_TYPE</i> , página 58	Código del error en la operación cuando Error es TRUE.

GetRtc: obtener el reloj en tiempo real

Descripción de la función

Esta función devuelve la hora del RTC en segundos en formato UNIX (tiempo transcurrido en segundos desde el 1 de enero de 1970 a las 00:00 UTC).

Representación gráfica



Representación IL y ST

Para ver la representación general en lenguaje IL o ST, consulte el capítulo *Representación de funciones y de bloques de funciones*, página 60.

Descripción de variables de E/S

En la siguiente tabla se describe la variable de E/S:

Salida	Tipo	Comentario
<i>GetRtc</i>	DINT	RTC en segundos en formato UNIX.

Ejemplo

En el siguiente ejemplo se describe cómo obtener el valor del RTC:

```
VAR
MyRTC : DINT := 0;
END_VAR
MyRTC := GetRtc();
```

HasForcedIo: indica si una entrada o una salida es forzosa

Descripción de funciones

Esta función devuelve TRUE si se fuerza una entrada o salida.

Representación gráfica



Representación IL y ST

Para ver la representación general en lenguaje IL o ST, consulte el capítulo *Representación de funciones y de bloques de funciones*, página 60.

Descripción de variables de E/S

En esta tabla se describe la variable de salida:

Salida	Tipo	Comentario
<i>HasForcedIo</i>	BOOL	TRUE si se fuerza una entrada o salida.

Ejemplo

En el ejemplo siguiente se describe cómo utilizar esta función:

```
VAR
hasIo: BOOL;
END_VAR
hasIo := SEC.HasForcedIo();
```

IsFirstMastColdCycle: Indicar si este ciclo es el primer ciclo MAST del arranque en frío

Descripción de la función

Esta función devuelve TRUE durante el primer ciclo MAST después de un arranque en frío (primer ciclo tras la descarga o el restablecimiento en frío).

Representación gráfica



Representación IL y ST

Para ver la representación general en lenguaje IL o ST, consulte el capítulo *Representación de funciones y de bloques de funciones*, página 60.

Descripción de variables de E/S

En esta tabla se describe la variable de salida:

Salida	Tipo	Comentario
<i>IsFirstMastColdCycle</i>	BOOL	TRUE durante el primer ciclo de tarea MAST después de un arranque en frío.

Ejemplo

Consulte la función *IsFirstMastCycle*, página 30.

IsFirstMastCycle: indicar si este ciclo es el primer ciclo MAST

Descripción de la función

Esta función devuelve TRUE durante el primer ciclo MAST después de un arranque.

Representación gráfica



Representación IL y ST

Para ver la representación general en lenguaje IL o ST, consulte el capítulo *Representación de funciones y de bloques de funciones*, página 60.

Descripción de variables de E/S

Salida	Tipo	Comentario
<i>IsFirstMastCycle</i>	BOOL	TRUE durante el primer ciclo de tarea MAST después de un arranque.

Ejemplo

En este ejemplo se describen las tres funciones utilizadas a la vez: *IsFirstMastCycle*, *IsFirstMastColdCycle* y *IsFirstMastWarmCycle*.

Use este ejemplo en la tarea MAST. De lo contrario, se puede ejecutar varias veces o posiblemente nunca (puede llamarse varias veces a una tarea adicional o no llamarse durante un ciclo de tarea MAST):

```

VAR
MyIsFirstMastCycle : BOOL;
MyIsFirstMastWarmCycle : BOOL;
MyIsFirstMastColdCycle : BOOL;
END_VAR
MyIsFirstMastWarmCycle := IsFirstMastWarmCycle();
MyIsFirstMastColdCycle := IsFirstMastColdCycle();
MyIsFirstMastCycle := IsFirstMastCycle();
IF (MyIsFirstMastWarmCycle) THEN
(*This is the first Mast Cycle after a Warm Start: all
variables are set to their initialization values except the
Retain variables*)
(*=> initialize the needed variables so that your
application runs as expected in this case*)
END_IF;
IF (MyIsFirstMastColdCycle) THEN
(*This is the first Mast Cycle after a Cold Start: all
variables are set to their initialization values including
the Retain Variables*)
(*=> initialize the needed variables so that your
application runs as expected in this case*)
END_IF;
IF (MyIsFirstMastCycle) THEN
(*This is the first Mast Cycle after a Start, i.e. after a
Warm or Cold Start as well as STOP/RUN commands*)
(*=> initialize the needed variables so that your
application runs as expected in this case*)
END_IF;
    
```

IsFirstMastWarmCycle: indicar si este ciclo es el primer ciclo MAST del arranque en caliente

Descripción de la función

Esta función devuelve TRUE durante el primer ciclo MAST después de un arranque en caliente.

Representación gráfica



Representación IL y ST

Para ver la representación general en lenguaje IL o ST, consulte el capítulo *Representación de funciones y de bloques de funciones*, página 60.

Descripción de variables de E/S

En esta tabla se describe la variable de salida:

Salida	Tipo	Comentario
<i>IsFirstMastWarmCycle</i>	BOOL	TRUE durante el primer ciclo de tarea MAST después de un arranque en caliente.

Ejemplo

Consulte la función *IsFirstMastCycle*, página 30.

Funciones de escritura de M241

Descripción general

En este capítulo se describen las funciones de escritura incluidas en la biblioteca M241 PLCSystem.

InhibitBatLed: Activa o desactiva el indicador LED de la batería

Descripción de la función

Esta función activa o desactiva la visualización del indicador LED de la batería, independientemente de su nivel de carga.

Representación gráfica



Representación IL y ST

Para ver la representación general en lenguaje IL o ST, consulte el capítulo *Representación de funciones y de bloques de funciones*, página 60.

Descripción de variables de E/S

En la siguiente tabla se describe la variable de entrada:

Entrada	Tipo	Comentario
<i>Inhibit</i>	BOOL	Si es <i>TRUE</i> , desactiva la visualización del indicador LED de la batería. Si es <i>FALSE</i> , activa habilita la visualización del indicador LED de la batería.

En la siguiente tabla se describe la variable de salida:

Salida	Tipo	Comentario
<i>InhibitBatLed</i>	INT	Un valor de 0 indica que no se ha detectado ningún error al ejecutar el bloque de funciones. Un valor distinto de cero indica que se ha detectado un error.

Ejemplo

En este ejemplo se describe cómo desactivar la visualización del indicador LED de la batería:

```
(* Disable Battery LED Information *)
SEC.InhibitBatLed(TRUE);
```

PhysicalWriteFastOutputs: escribir salida rápida de una E/S experta incrustada

Descripción de la función

Esta función escribe un estado en las salidas de Q0 a Q3 en el momento de la llamada de la función.

Representación gráfica



Representación IL y ST

Para ver la representación general en lenguaje IL o ST, consulte el capítulo *Representación de funciones y de bloques de funciones*, página 60.

Descripción de variables de E/S

En la siguiente tabla se describen las variables de entrada:

Entrada	Tipo	Comentario
Q0Value	BOOL	Valor solicitado para la salida 0.
Q1Value	BOOL	Valor solicitado para la salida 1.
Q2Value	BOOL	Valor solicitado para la salida 2.
Q3Value	BOOL	Valor solicitado para la salida 3.

En la siguiente tabla se describe la variable de salida:

Salida	Tipo	Comentario
PhysicalWriteFastOutputs	WORD	Valor de salida de la función.

NOTA: Solo son importantes los cuatro primeros bits del valor de salida, y se utilizan como campo de bit para indicar si se escribe la salida.

Si el bit correspondiente a la salida es 1, la salida se escribe correctamente.

Si el bit correspondiente a la salida es 0, la salida no se escribe porque ya la utiliza una función experta.

Si el bit correspondiente a la salida es 1111 bin, las cuatro salidas se escriben correctamente.

Si el bit correspondiente a la salida es 1110 bin, Q0 no se escribe porque lo utiliza un generador de frecuencias.

NOTA: Los valores se aplican al principio y al final de un ciclo de procesamiento. La función aplica un valor dentro del ciclo.

NOTA: Si se asigna una variable a más de una de las salidas incrustadas, la última (ordenada de Q0 a Q3) establece el valor en la variable al final de la ejecución del bloque de funciones.

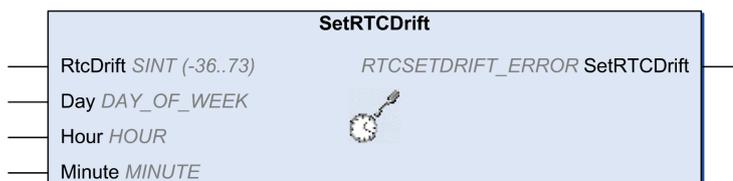
SetRTCDrift: Establecer el valor de compensación en el RTC

Descripción de la función

Esta función acelera o ralentiza la frecuencia del RTC para dar control a la aplicación para compensar el RTC, en función del entorno de funcionamiento (temperatura, etc.). El valor de compensación se indica en segundos por semana. Puede ser positivo (acelerar) o negativo (ralentizar).

NOTA: La función *SetRTCDrift* solo debe llamarse una vez. Con cada nueva llamada, se reemplaza el valor de compensación por el nuevo. El valor se mantendrá en el hardware del controlador mientras el RTC reciba alimentación de la red eléctrica o por medio de la batería. Si se retiran la batería y la fuente de alimentación, el valor de compensación del RTC no estará disponible.

Representación gráfica



Representación IL y ST

Para ver la representación general en lenguaje IL o ST, consulte el capítulo *Representación de funciones y de bloques de funciones*, página 60.

Descripción de variables de E/S

En esta tabla se describen los parámetros de entrada:

Entradas	Tipo	Comentario
<i>RtcDrift</i>	<i>SINT</i> (-36..73)	Corrección en segundos por semana (de -36 a +73).

NOTA: Los parámetros *Day*, *Hour* y *Minute* se utilizan únicamente para garantizar la compatibilidad con versiones anteriores.

NOTA: Si el valor introducido para *RtcDrift* sobrepasa el límite, el firmware del controlador definirá el valor en el máximo.

En esta tabla se describe la variable de salida:

Salida	Tipo	Comentario
<i>SetRTCDrift</i>	<i>RTCSETDRIFT_ERROR</i> , página 58	Devuelve <i>RTC_OK</i> (00 hex) si el comando es correcto; en caso contrario, devuelve el código ID del error detectado.

Ejemplo

En este ejemplo, la función únicamente se invoca durante el primer ciclo de la tarea MAST. Acelera el RTC 4 segundos a la semana (18 segundos al mes).

```

VAR
MyRTCDrift : SINT (-36..+73) := 0;
MyDay : sec.DAY_OF_WEEK;
MyHour : sec.HOUR;
MyMinute : sec.MINUTE;
END_VAR
IF IsFirstMastCycle() THEN
MyRTCDrift := 4;
MyDay := 0;
MyHour := 0;
MyMinute := 0;
SetRTCDrift(MyRTCDrift, MyDay, MyHour, MyMinute);
END_IF
    
```

Funciones de usuario de M241

Descripción general

En esta sección se describen las funciones *FB_ControlClone*, *DataFileCopy* y *ExecuteScript* incluidas en la biblioteca M241 PLCSystem.

FB_ControlClone: clonar el controlador

Descripción del bloque de funciones

La clonación es posible de forma predeterminada mediante la tarjeta SD o el **Asistente del controlador**. Cuando los derechos de usuario están habilitados y se deniega el derecho de visualización **ExternalCmd** para el grupo **ExternalMedia**, no se permite la función de clonación. En este caso, el bloque de funciones habilitará la funcionalidad de clonación una vez cuando se vuelva a encender el controlador.

NOTA: Puede elegir si incluir los derechos de usuario en el clon en la página **Gestión de clones** del servidor web (consulte Modicon M241 Logic Controller, Guía de programación).

En esta tabla se muestra cómo configurar el bloque de funciones y los derechos de usuario:

Ajuste del bloque de funciones	Con derechos de usuario habilitados	Con derechos de usuario deshabilitados
<i>xEnable</i> = 1	Clonación permitida	Clonación permitida
<i>xEnable</i> = 0	Clonación no permitida	Clonación no permitida

Representación gráfica



Representación IL y ST

Para ver la representación general en lenguaje IL o ST, consulte el capítulo *Representación de funciones y de bloques de funciones*, página 60.

Descripción de variables de E/S

En la siguiente tabla se describen las variables de entrada:

Entrada	Tipo	Comentario
<i>xEnable</i>	BOOL	Si está ajustada en <i>TRUE</i> , habilita la funcionalidad de clonación una vez. Si está ajustada en <i>FALSE</i> , deshabilita la funcionalidad de clonación.

En la siguiente tabla se describen las variables de salida:

Salida	Tipo	Comentario
<i>xError</i>	UDINT	Un valor de 0 indica que no se ha detectado ningún error al ejecutar el bloque de funciones. Un valor distinto de cero indica que se ha detectado un error.

DataFileCopy: copiar comandos de archivo

Descripción del bloque de funciones

Este bloque de funciones copia los datos de la memoria en un archivo y viceversa. El archivo se encuentra en el sistema de archivos interno o en un sistema de archivos externo (tarjeta SD).

El bloque de funciones *DataFileCopy* puede:

- Leer datos de un archivo formateado.
- Copiar los datos de la memoria en un archivo formateado. Para obtener más información, consulte Non-VolatileMemory Organization (consulte Modicon M241 Logic Controller - Guía de programación).

Representación gráfica



Representación IL y ST

Para ver la representación general en lenguaje IL o ST, consulte el capítulo *Representación de funciones y de bloques de funciones*, página 60.

Descripción de variables de E/S

En esta tabla se describen las variables de entrada:

Entrada	Tipo	Comentario
<i>xExecute</i>	BOOL	<p>En el flanco ascendente, inicia la ejecución del bloque de funciones.</p> <p>En el flanco descendente, restablece las salidas del bloque de funciones cuando finaliza cualquier ejecución en curso.</p> <p>NOTA: Con el flanco descendente, la función continúa hasta que concluye su ejecución y actualiza sus salidas. Las salidas se conservan durante un ciclo y se restablecen.</p>
<i>sFileName</i>	STRING	<p>Nombre de archivo sin extensión (la extensión <i>.DTA</i> se añade automáticamente). Utilice solamente caracteres alfanuméricos (mayúsculas, minúsculas o números).</p>
<i>xRead</i>	BOOL	<p>TRUE: copiar datos del archivo identificado por <i>sFileName</i> en la memoria interna del controlador.</p> <p>FALSE: copiar datos de la memoria interna del controlador en el archivo identificado por <i>sFileName</i>.</p>
<i>xSecure</i>	BOOL	<p>TRUE: la dirección MAC siempre se almacena en el archivo. Solo un controlador con la misma dirección MAC podrá leer el archivo.</p> <p>FALSE: otro controlador con el mismo tipo de memoria puede leer el archivo.</p>
<i>iLocation</i>	INT	<p>0: la ubicación del archivo en el sistema de archivos interno es <i>/usr/DTA</i>.</p> <p>1: la ubicación del archivo en el sistema de archivos externo (tarjeta SD) es <i>/usr/DTA</i>.</p> <p>NOTA: Si el archivo no existe todavía en el directorio, se creará.</p>
<i>uiSize</i>	UINT	<p>Indica el tamaño en bytes. El máximo son 65 534 bytes.</p> <p>Utilice solamente direcciones de variables conformes con IEC 61131-3 (variables, matrices, estructuras); por ejemplo:</p> <pre>Variable : int; uiSize := SIZEOF (Variable);</pre>
<i>dwAdd</i>	DWORD	<p>Indica la dirección de la memoria desde la que leerá o en la que escribirá la función.</p> <p>Utilice solamente direcciones de variables conformes con IEC 61131-3 (variables, matrices, estructuras); por ejemplo:</p> <pre>Variable : int; dwAdd := ADR (Variable);</pre>

⚠ ADVERTENCIA
FUNCIONAMIENTO IMPREVISTO DEL EQUIPO
<p>Compruebe que el tamaño de la ubicación de la memoria sea el adecuado y que el tipo de archivo sea el correcto antes de copiar el archivo en la memoria.</p> <p>Si no se siguen estas instrucciones, pueden producirse lesiones graves, muerte o daños en el equipo.</p>

En esta tabla se describen las variables de salida:

Salida	Tipo	Comentario
<i>xDone</i>	BOOL	TRUE = indica que la acción se ha llevado a cabo correctamente.
<i>xBusy</i>	BOOL	TRUE = indica que el bloque de funciones está en ejecución.
<i>xError</i>	BOOL	TRUE = indica que se ha detectado un error y el bloque de funciones ha cancelado la acción.
<i>eError</i>	<i>DataFileCopyError</i> , página 52	Indica el tipo de error de copia de archivo de datos que se ha detectado.

NOTA: Si modifica los datos de la memoria (variables, matrices, estructuras) que se utilizan para escribir el archivo, se producirá un error de integridad de CRC.

Ejemplo

En este ejemplo se describe cómo copiar comandos de archivo:

```

VAR
LocalArray : ARRAY [0..29] OF BYTE;
myFileName: STRING := 'exportfile';
EXEC_FLAG: BOOL;
DataFileCopy: DataFileCopy;
END_VAR
DataFileCopy(
xExecute:= EXEC_FLAG,
sFileName:= myFileName,
xRead:= FALSE,
xSecure:= FALSE,
iLocation:= DFCL_INTERNAL,
uiSize:= SIZEOF(LocalArray),
dwAdd:= ADR(LocalArray),
xDone=> ,
xBusy=> ,
xError=> ,
eError=> );

```

ExecuteScript: ejecutar comandos de script

Descripción del bloque de funciones

Esta función puede ejecutar los siguientes comandos de script de la tarjeta SD:

- *Download*
- *Upload*
- *SetNodeName*
- *Delete*
- *Reboot*
- *ChangeModbusPort*

Para obtener información sobre el formato de archivo de script requerido, consulte Archivos de script para tarjetas SD (consulte Modicon M241 Logic Controller, Guía de programación).

Representación gráfica



Representación IL y ST

Para ver la representación general en lenguaje IL o ST, consulte el capítulo *Representación de funciones y de bloques de funciones*, página 60.

Descripción de variables de E/S

En esta tabla se describen las variables de entrada:

Entrada	Tipo	Comentario
<i>xExecute</i>	BOOL	<p>Cuando se detecta un flanco ascendente, inicia la ejecución del bloque de funciones.</p> <p>Cuando se detecta un flanco descendente, restablece las salidas del bloque de funciones cuando finaliza cualquier ejecución en curso.</p> <p>NOTA: Con el flanco descendente, la función continúa hasta que concluye su ejecución y actualiza sus salidas. Las salidas se conservan durante un ciclo y se restablecen.</p>
<i>sCmd</i>	STRING	<p>Sintaxis de comando de script de la tarjeta SD.</p> <p>No se permiten ejecuciones de comando simultáneas: si se ejecuta un comando desde otro bloque de funciones o desde un script de la tarjeta SD, el bloque de funciones pone el comando en cola y no lo ejecuta inmediatamente.</p> <p>NOTA: Se considera que un script de tarjeta SD ejecutado desde una tarjeta SD se estará ejecutando hasta que se extraiga la tarjeta SD.</p>

En esta tabla se describen las variables de salida:

Salida	Tipo	Comentario
<i>xDone</i>	BOOL	TRUE indica que la acción se ha completado correctamente.
<i>xBusy</i>	BOOL	TRUE indica que el bloque de funciones se está ejecutando.
<i>xError</i>	BOOL	TRUE indica la detección de errores; el bloque de funciones anula la acción.
<i>eError</i>	ExecuteScriptError, página 53	Indica el tipo de error detectado en la ejecución del script.

Ejemplo

En este ejemplo se describe cómo ejecutar un comando de script *Upload*:

```

VAR
EXEC_FLAG: BOOL;
ExecuteScript: ExecuteScript;
END_VAR
ExecuteScript(
xExecute:= EXEC_FLAG,
sCmd:= 'Upload "/usr/Syslog/*"',
xDone=> ,
xBusy=> ,
xError=> ,
eError=> );
    
```

Funciones de espacio en disco de M241

Descripción general

En esta sección se describen las funciones de espacio en disco incluidas en esta biblioteca.

FC_GetFreeDiskSpace: Obtiene el espacio libre de la memoria

Descripción de la función

Esta función recupera la cantidad de espacio libre de memoria de un soporte de memoria (disco de usuario, disco de sistema, tarjeta SD) en bytes.

El nombre del soporte de memoria se transfiere:

- Disco de usuario = "/usr"
- Disco de sistema = "/sys"
- Tarjeta SD = "/sd0"

No se puede acceder al espacio libre en la memoria de un dispositivo remoto. Si se especifica como parámetro un dispositivo remoto, la función devuelve "-1".

Representación gráfica



Representación IL y ST

Para ver la representación general en lenguaje IL o ST, consulte el capítulo *Representación de funciones y de bloques de funciones*, página 60.

Descripción de variables de E/S

En esta tabla se describen las variables de entrada:

Input	Tipo de datos	Descripción
<i>i_sVolumeName</i>	STRING[80]	Nombre del dispositivo a cuyo espacio libre en la memoria se debe acceder
<i>iq_ulifFreeDiskSpace</i>	ULINT	Espacio libre en la memoria en bytes

En esta tabla se describen las variables de salida:

Salida	Tipo de datos	Descripción
<i>FC_GetFreeDiskSpace</i>	DINT	0: La cantidad de espacio de memoria libre se recuperó correctamente. -1: Error al intentar acceder a la cantidad de memoria libre. Por ejemplo, se ha seleccionado un dispositivo no válido o un dispositivo remoto. -318: Parámetro no válido (<i>i_sVolumeName</i>)

FC_GetLabel: Obtiene la etiqueta de memoria

Descripción de la función

Esta función recupera la etiqueta de un soporte de memoria. Si un dispositivo no tiene etiqueta, se devuelve una cadena vacía.

El nombre del soporte de memoria (disco de usuario, disco de sistema, tarjeta SD) se transfiere:

- Disco de usuario = "/usr"
- Disco de sistema = "/sys"
- Tarjeta SD = "/sd0"

Representación gráfica



Representación IL y ST

Para ver la representación general en lenguaje IL o ST, consulte el capítulo *Representación de funciones y de bloques de funciones*, página 60.

Descripción de variables de E/S

En esta tabla se describen las variables de entrada:

Input	Tipo de datos	Descripción
<i>i_sVolumeName</i>	STRING[80]	Nombre del dispositivo a cuya etiqueta se debe acceder
<i>iq_sLabel</i>	STRING[11]	Etiqueta del dispositivo

En esta tabla se describen las variables de salida:

Salida	Tipo de datos	Descripción
<i>FC_GetLabel</i>	DINT	0: La etiqueta se recuperó correctamente. -1: Error al acceder a la etiqueta. -318: Parámetro no válido

FC_GetTotalDiskSpace: Obtiene el tamaño de la memoria

Descripción de la función

Esta función permite recuperar el tamaño de un soporte de memoria (disco de usuario, disco de sistema, tarjeta SD) en bytes.

El nombre del soporte de memoria se transfiere:

- Disco de usuario = "/usr"
- Disco de sistema = "/sys"
- Tarjeta SD = "/sd0"

No se puede acceder al tamaño de un dispositivo remoto. Si se especifica como parámetro un dispositivo remoto, la función devuelve "-1".

Representación gráfica



Representación IL y ST

Para ver la representación general en lenguaje IL o ST, consulte el capítulo *Representación de funciones y de bloques de funciones*, página 60.

Descripción de variables de E/S

En esta tabla se describen las variables de entrada:

Input	Tipo de datos	Descripción
<i>i_sVolumeName</i>	STRING[80]	Nombre del dispositivo a cuyo tamaño de memoria se debe acceder
<i>iq_uliTotalDiskSpace</i>	ULINT	El tamaño del soporte de la memoria en bytes

En esta tabla se describen las variables de salida:

Salida	Tipo de datos	Descripción
<i>FC_GetTotalDiskSpace</i>	DINT	0: El tamaño se recuperó correctamente. -1: Error al leer el tamaño. -318: Al menos uno de los parámetros no es válido.

Funciones de lectura TM3

Descripción general

En este apartado se describen las funciones de lectura TM3 incluidas en la biblioteca M241 PLCSystem.

TM3_GetModuleBusStatus: obtener estado del bus del módulo TM3

Descripción de la función

Esta función devuelve el estado del bus del módulo. El índice del módulo se proporciona como parámetro de entrada.

Representación gráfica



Representación IL y ST

Para ver la representación general en lenguaje IL o ST, consulte el capítulo *Representación de funciones y de bloques de funciones*, página 60.

Descripción de variables de E/S

En la siguiente tabla se describe la variable de entrada:

Entrada	Tipo	Comentario
<i>ModuleIndex</i>	BYTE	Índice del módulo (0 para la primera expansión, 1 para la segunda, etc.).

En la siguiente tabla se describe la variable de salida:

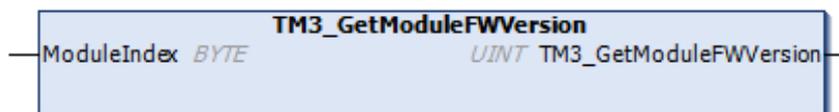
Salida	Tipo	Comentario
<i>TM3_GetModuleBusStatus</i>	<i>TM3_ERR_CODE</i> , página 56	Devuelve <i>TM3_OK</i> (00 hex) si el comando es correcto; en caso contrario, devuelve el código ID del error detectado.

TM3_GetModuleFWVersion: Obtener la versión de firmware del módulo TM3

Descripción de la función

Esta función devuelve la versión del firmware de un módulo TM3 especificado.

Representación gráfica



Representación IL y ST

Para ver la representación general en lenguaje IL o ST, consulte el capítulo *Representación de funciones y de bloques de funciones*, página 60.

Descripción de variables de E/S

En la siguiente tabla se describen las variables de entrada:

Entrada	Tipo	Comentario
<i>ModuleIndex</i>	BYTE	Índice del módulo (0 para la primera expansión, 1 para la segunda, etc.).

En la siguiente tabla se describe la variable de salida:

Salida	Tipo	Comentario
<i>TM3_GetModuleFWVersion</i>	UINT	Devuelve la versión del firmware del módulo, o <i>FFFF hex</i> si no se puede leer la información. Por ejemplo, <i>001A hex</i> indica la versión de firmware 26.

TM3_GetModuleInternalStatus: obtener estado interno del módulo TM3

Descripción de la función

Esta función lee selectivamente el estado del canal de E/S de un módulo TM3 analógico o de temperatura, que se indica mediante *ModuleIndex*. El bloque de

funciones escribe el estado de cada canal solicitado a partir de la ubicación de la memoria señalada por *pStatusBuffer*.

NOTA: Este bloque de funciones está diseñado para usarse con módulos de E/S analógicos y de temperatura. Para obtener información de estado de los módulos de E/S digitales, consulte *TM3_GetModuleBusStatus*, página 42.

NOTA: Es posible actualizar el valor de los bytes de diagnóstico ejecutando la función *TM3_GetModuleInternalStatus* solo si el parámetro **Estado habilitado** de la ficha **Configuración de E/S** está desactivado.

Representación gráfica



Representación IL y ST

Para ver la representación general en lenguaje IL o ST, consulte el capítulo *Representación de funciones y de bloques de funciones*, página 60.

Descripción de variables de E/S

Cada canal de E/S analógico o de temperatura del módulo solicitado requiere un byte de memoria. Si no hay suficiente memoria asignada al búfer para el número de estados de canal del módulo de E/S solicitado, es posible que la función sobrescriba la memoria asignada para otros fines o quizás intente sobrescribir un área restringida de memoria.

⚠ ADVERTENCIA

FUNCIONAMIENTO NO DESEADO DEL EQUIPO

Asegúrese de que *pStatusBuffer* apunte a un área de memoria que haya sido suficientemente asignada para el número de canales que se van a leer.

Si no se siguen estas instrucciones, pueden producirse lesiones graves, muerte o daños en el equipo.

En la siguiente tabla se describen las variables de entrada:

Entrada	Tipo	Comentario
<i>ModuleIndex</i>	BYTE	Índice del módulo de extensión (0 para el módulo más próximo al controlador, 1 para el segundo más próximo y así sucesivamente)
<i>StatusOffset</i>	BYTE	Offset del primer estado que se debe leer en la tabla de estado.
<i>StatusSize</i>	BYTE	Número de bytes que se deben leer en la tabla de estado.
<i>pStatusBuffer</i>	POINTER TO BYTE	Búfer que contiene la tabla de estado de lectura (IBStatusIWx/IBStatusQWx).

En la siguiente tabla se describe la variable de salida:

Salida	Tipo	Comentario
TM3_GetModuleInternalStatus	TM3_ERR_CODE, página 56	Devuelve TM3_NO_ERR (00 hex) si el comando es correcto. En caso contrario, devuelve el código ID del error. A los efectos de este bloque de funciones, cualquier valor devuelto distinto de cero indica que el módulo no es compatible con la solicitud de estado o que el módulo tiene otros problemas de comunicación.

Ejemplo

En los siguientes ejemplos se describe cómo obtener el estado interno del módulo:

```
VAR
TM3AQ2_Channel_0_Output_Status: BYTE;
END_VAR
TM3AQ2 is on position 1
Status of channel 0 is at offset 0
We read 1 channel
TM3_GetModuleInternalStatus(1, 0, 1, ADR(TM3AQ2_Channel_0_
Output_Status));
status of channel 0 is in TM3AQ2_Channel_0_Output_Status
```

Módulo TM3AQ2 (2 salidas)

Obtención del estado de la primera salida QW0

- *StatusOffset* = 0 (0 entradas x 2)
- *StatusSize* = 1 (1 estado para leer)
- *pStatusBuffer* debe tener al menos 1 byte

```
VAR
TM3AM6_Channels_1_2_Input_Status: ARRAY[1..2] OF BYTE;
END_VAR
TM3AM6 is on position 1
Status of channel 1 is at offset 1
We read 2 consecutive channels
TM3_GetModuleInternalStatus(1, 1, 2, ADR(TM3AM6_Channels_1_
2_Input_Status));
status of channel 1 is in TM3AM6_Channels_1_2_Input_Status
[1]
status of channel 2 is in TM3AM6_Channels_1_2_Input_Status
[2]
```

Módulo TM3AM6 (4 entradas, 2 salidas)

Obtención del estado de la entrada IW1 e IW2 (IW0 es la primera)

- *StatusOffset* = 1 (1 para omitir el estado IW0)
- *StatusSize* = 2 (2 estados para leer)
- *pStatusBuffer* debe tener al menos 2 bytes

Tipos de datos de la biblioteca PLCSystem de M241

Descripción general

En este capítulo se describen los tipos de datos de la biblioteca PLCSystem de M241.

Hay dos tipos de datos disponibles:

- Los tipos de datos de variables de sistema los utilizan las variables de sistema, página 13 de la biblioteca PLCSystem de M241 (*PLC_R*, *PLC_W*...).
- Los tipos de datos de funciones de sistema los utilizan las funciones de sistema, página 27 de lectura/escritura de la biblioteca PLCSystem de M241.

Tipos de datos de variables de sistema de *PLC_RW*

Descripción general

En esta sección se enumeran y describen los tipos de datos de las variables de sistema incluidos en las estructuras *PLC_R* y *PLC_W*.

PLC_R_APPLICATION_ERROR: códigos de estado de error detectado de la aplicación

Descripción del tipo enumerado

El tipo de datos de enumeración *PLC_R_APPLICATION_ERROR* contiene los siguientes valores:

Enumerador	Valor	Comentario	Qué hacer
<i>PLC_R_APP_ERR_UNKNOWN</i>	FFFF hex	Error no definido detectado.	Póngase en contacto con su representante de servicio de Schneider Electric.
<i>PLC_R_APP_ERR_NOEXCEPTION</i>	0000 hex	No se ha detectado ningún error.	–
<i>PLC_R_APP_ERR_WATCHDOG</i>	0010 hex	Ha caducado el watchdog de la tarea.	Compruebe la aplicación. Es necesario un restablecimiento para acceder a la modalidad Run (ejecución).
<i>PLC_R_APP_ERR_HARDWAREWATCHDOG</i>	0011 hex	Ha caducado el watchdog de sistema.	Si el problema se reproduce, verifique que no haya puertos de comunicación configurados y desconectados. De lo contrario, actualice el firmware. Si el problema persiste, póngase en contacto con su representante de servicio de Schneider Electric.
<i>PLC_R_APP_ERR_IO_CONFIG_ERROR</i>	0012 hex	Se han detectado parámetros de configuración de E/S incorrectos.	La aplicación puede estar dañada. Para resolver este problema, utilice uno de los siguientes métodos: <ol style="list-style-type: none"> 1. Compilar > Limpiar todo 2. Exporte/Importe la aplicación. 3. Actualice el software a la versión más reciente.
<i>PLC_R_APP_ERR_UNRESOLVED_EXTREFS</i>	0018 hex	Funciones no definidas detectadas.	Elimine las funciones no resueltas de la aplicación.
<i>PLC_R_APP_ERR_IEC_TASK_CONFIG_ERROR</i>	0025 hex	Se han detectado parámetros de configuración de tareas incorrectos.	La aplicación puede estar dañada. Para resolver este problema, utilice uno de los siguientes métodos: <ol style="list-style-type: none"> 1. Compilar > Limpiar todo 2. Exporte/Importe la aplicación. 3. Actualice el software a la versión más reciente.
<i>PLC_R_APP_ERR_ILLEGAL_INSTRUCTION</i>	0050 hex	Instrucción no definida detectada.	Para resolver el problema, depure la aplicación.
<i>PLC_R_APP_ERR_ACCESS_VIOLATION</i>	0051 hex	Intento de acceso al área de memoria reservada.	Para resolver el problema, depure la aplicación.
<i>PLC_R_APP_ERR_DIVIDE_BY_ZERO</i>	0102 hex	Detectada división de entero por cero.	Para resolver el problema, depure la aplicación.
<i>PLC_R_APP_ERR_PROCESSORLOAD_WATCHDOG</i>	0105 hex	Las tareas de la aplicación han sobrecargado el procesador.	Reduzca la carga de trabajo de la aplicación mejorando la arquitectura de la aplicación. Aumente la duración del ciclo de tarea. Reduzca la frecuencia de evento.
<i>PLC_R_APP_ERR_DIVIDE_REAL_BY_ZERO</i>	0152 hex	Detectada división de real por cero.	Para resolver el problema, depure la aplicación.

Enumerador	Valor	Comentario	Qué hacer
<i>PLC_R_APP_ERR_EXPIO_EVENTS_COUNT_EXCEEDED</i>	4E20 hex	Se han detectado demasiados eventos en E/S expertas.	Reduzca el número de tareas de evento.
<i>PLC_R_APP_ERR_APPLICATION_VERSION_MISMATCH</i>	4E21 hex	Discrepancia detectada en la versión de la aplicación.	La versión de la aplicación en el controlador lógico no coincide con la versión en el software. Consulte las Aplicaciones que encontrará en la EcoStruxure Automation Expert - Motion / EcoStruxure Machine Expert Guía de programación.

PLC_R_BOOT_PROJECT_STATUS: códigos de estado del proyecto de arranque

Descripción del tipo enumerado

El tipo de datos de enumeración *PLC_R_BOOT_PROJECT_STATUS* contiene los siguientes valores:

Enumerador	Valor	Comentario
<i>PLC_R_NO_BOOT_PROJECT</i>	0000 hex	El proyecto de arranque no existe en la memoria no volátil.
<i>PLC_R_BOOT_PROJECT_CREATION_IN_PROGRESS</i>	0001 hex	Se está creando el proyecto de arranque.
<i>PLC_R_DIFFERENT_BOOT_PROJECT</i>	0002 hex	El proyecto de inicio en la memoria no volátil es distinto del cargado en la memoria.
<i>PLC_R_VALID_BOOT_PROJECT</i>	FFFF hex	El proyecto de arranque de la memoria no volátil es el mismo que el proyecto cargado en la memoria.

PLC_R_IO_STATUS: códigos de estado de E/S

Descripción del tipo enumerado

El tipo de datos de enumeración *PLC_R_IO_STATUS* contiene los siguientes valores:

Enumerador	Valor	Comentario
<i>PLC_R_IO_OK</i>	FFFF hex	Las entradas/salidas están operativas.
<i>PLC_R_IO_NO_INIT</i>	0001 hex	Las entradas/salidas no se han inicializado.
<i>PLC_R_IO_CONF_FAULT</i>	0002 hex	Se han detectado parámetros de configuración de E/S incorrectos.
<i>PLC_R_IO_SHORTCUT_FAULT</i>	0003 hex	Se ha detectado un cortocircuito de las entradas/salidas.
<i>PLC_R_IO_POWER_SUPPLY_FAULT</i>	0004 hex	Se ha detectado un error de alimentación en las entradas/salidas.

PLC_R_SDCARD_STATUS: códigos de estado de la ranura de la tarjeta SD

Descripción del tipo enumerado

El tipo de datos de enumeración *PLC_R_SDCARD_STATUS* contiene los siguientes valores:

Enumerador	Valor	Comentario
<i>NO_SDCARD</i>	0000 hex	No se ha detectado ninguna tarjeta SD en la ranura o la ranura no está conectada.
<i>SDCARD_READONLY</i>	0001 hex	La tarjeta SD se encuentra en la modalidad de solo lectura.
<i>SDCARD_READWRITE</i>	0002 hex	La tarjeta SD se encuentra en la modalidad de lectura/escritura.
<i>SDCARD_ERROR</i>	0003 hex	Error detectado en la tarjeta SD. El archivo FwLog.txt contiene más detalles sobre el error.

PLC_R_STATUS: códigos de estado del controlador

Descripción del tipo enumerado

El tipo de datos de la enumeración *PLC_R_STATUS* contiene los siguientes valores:

Enumerador	Valor	Comentario
<i>PLC_R_EMPTY</i>	0000 hex	El controlador no contiene ninguna aplicación.
<i>PLC_R_STOPPED</i>	0001 hex	El controlador se ha detenido.
<i>PLC_R_RUNNING</i>	0002 hex	El controlador está en ejecución.
<i>PLC_R_HALT</i>	0004 hex	El controlador se encuentra en estado HALT (consulte el diagrama de estado del controlador en la guía de programación [consulte Modicon M241 Logic Controller, Guía de programación]).
<i>PLC_R_BREAKPOINT</i>	0008 hex	El controlador se ha detenido en un punto de interrupción.

PLC_R_STOP_CAUSE: códigos de causa de transición de estado RUN a otro estado

Descripción del tipo enumerado

El tipo de datos de enumeración *PLC_R_STOP_CAUSE* contiene los siguientes valores:

Enumerador	Valor	Comentario	Qué hacer
<i>PLC_R_STOP_REASON_UNKNOWN</i>	00 hex	El valor inicial o la causa de la detención no se pueden determinar.	Póngase en contacto con su representante local de Schneider Electric.
<i>PLC_R_STOP_REASON_HW_WATCHDOG</i>	01 hex	Detenido tras un timeout de watchdog de hardware.	Póngase en contacto con su representante local de Schneider Electric.
<i>PLC_R_STOP_REASON_RESET</i>	02 hex	Detenido tras restablecimiento.	Vea las opciones de restablecimiento en el diagrama de estado del controlador.
<i>PLC_R_STOP_REASON_EXCEPTION</i>	03 hex	Detenido tras excepción.	Compruebe su aplicación y corríjala si es necesario. Consulte Watchdogs del sistema y de tareas (consulte Modicon M241 Logic Controller, Guía de programación). Es necesario un restablecimiento para acceder a la modalidad Run (ejecución).
<i>PLC_R_STOP_REASON_USER</i>	04 hex	Detenido tras una petición de usuario.	Consulte Comando de parada en Comandos de transiciones de estado (consulte Modicon M241 Logic Controller, Guía de programación).
<i>PLC_R_STOP_REASON_IECPROGRAM</i>	05 hex	Detenido después de una petición de comando de programa (por ejemplo: comando de control con parámetro <i>PLC_W.q_wPLCControl:=PLC_W_COMMAND.PLC_W_STOP;</i>).	–
<i>PLC_R_STOP_REASON_DELETE</i>	06 hex	Detenido tras un comando de eliminación de aplicación.	Consulte la ficha Aplicaciones del Editor de dispositivos del controlador (consulte Modicon M241 Logic Controller, Guía de programación).
<i>PLC_R_STOP_REASON_DEBUGGING</i>	07 hex	Detenido tras entrar en la modalidad de depuración.	–
<i>PLC_R_STOP_FROM_NETWORK_REQUEST</i>	0A hex	Detenido tras una petición de la red, el servidor web del controlador o el comando <i>PLC_W</i> .	–
<i>PLC_R_STOP_FROM_INPUT</i>	0B hex	Detención requerida por una entrada de controlador.	–
<i>PLC_R_STOP_FROM_RUN_STOP_SWITCH</i>	0C hex	Detención requerida por el interruptor del controlador.	–
<i>PLC_R_STOP_REASON_RETAIN_MISMATCH</i>	0D hex	Detenido tras una prueba de comprobación de contexto incorrecta durante el reinicio.	Algunas variables retenidas en la memoria no volátil no existen en la aplicación que se está ejecutando. Compruebe su aplicación, corríjala si es necesario y restablezca la aplicación de arranque.

Enumerador	Valor	Comentario	Qué hacer
<i>PLC_R_STOP_REASON_BOOT_APPLI_MISMATCH</i>	0E hex	Detenido tras una comparación incorrecta entre la aplicación de arranque y la aplicación que se encontraba en la memoria antes del reinicio.	Cree una aplicación de arranque válida.
<i>PLC_R_STOP_REASON_POWERFAIL</i>	0F hex	Detenido tras una interrupción de la alimentación.	–

Para obtener más información sobre los motivos por los que se ha detenido el controlador, consulte la descripción de los estados del controlador.

***PLC_R_TERMINAL_PORT_STATUS*: códigos de estado de la conexión del puerto del programación**

Descripción del tipo enumerado

El tipo de datos de enumeración *PLC_R_TERMINAL_PORT_STATUS* contiene los siguientes valores:

Enumerador	Valor	Comentario
<i>TERMINAL_NOT_CONNECTED</i>	00 hex	No hay ningún PC conectado al puerto de programación.
<i>TERMINAL_CONNECTION_IN_PROGRESS</i>	01 hex	La conexión está en curso.
<i>TERMINAL_CONNECTED</i>	02 hex	El PC está conectado al puerto de programación.
<i>TERMINAL_ERROR</i>	0F hex	Error detectado durante la conexión.

***PLC_R_TM3_BUS_STATE*: códigos de estado del bus TM3**

Descripción del tipo enumerado

El tipo de datos de enumeración *PLC_R_TM3_BUS_STATE* contiene los siguientes valores:

Enumerador	Valor	Comentario
<i>TM3_CONF_ERROR</i>	01 hex	Error detectado debido a una discrepancia entre la configuración física y la configuración en el software.
<i>TM3_OK</i>	03 hex	La configuración física y la configuración del software coinciden.
<i>TM3_POWER_SUPPLY_ERROR</i>	04 hex	Error detectado en la fuente de alimentación.

***PLC_W_COMMAND*: códigos de comando de control**

Descripción del tipo enumerado

El tipo de datos de enumeración *PLC_W_COMMAND* contiene los siguientes valores:

Enumerador	Valor	Comentario
<i>PLC_W_STOP</i>	0001 hex	Comando para detener el controlador.
<i>PLC_W_RUN</i>	0002 hex	Comando para ejecutar el controlador.
<i>PLC_W_RESET_COLD</i>	0004 hex	Comando para iniciar un restablecimiento en frío del controlador.
<i>PLC_W_RESET_WARM</i>	0008 hex	Comando para iniciar un restablecimiento en caliente del controlador.

Tipos de datos de variables de sistema de *DataFileCopy*

Descripción general

En esta sección se describen los tipos de datos de variables de sistema incluidos en las estructuras *DataFileCopy*.

DataFileCopyError: códigos de error detectado

Descripción del tipo enumerado

El tipo de datos de enumeración *DataFileCopyError* contiene los siguientes valores:

Enumerador	Valor	Descripción
<i>ERR_NO_ERR</i>	00 hex	No se ha detectado ningún error.
<i>ERR_FILE_NOT_FOUND</i>	01 hex	El archivo no existe.
<i>ERR_FILE_ACCESS_REFUSED</i>	02 hex	No se puede abrir el archivo.
<i>ERR_INCORRECT_SIZE</i>	03 hex	El tamaño de la petición es diferente del tamaño que se ha leído en el archivo.
<i>ERR_CRC_ERR</i>	04 hex	El CRC es incorrecto y se supone que el archivo está dañado.
<i>ERR_INCORRECT_MAC</i>	05 hex	El controlador que intenta leer del archivo no tiene la misma dirección MAC que la que se presenta en el archivo.

DataFileCopyLocation: códigos de ubicación

Descripción del tipo enumerado

El tipo de datos de enumeración *DataFileCopyLocation* contiene los siguientes valores:

Enumerador	Valor	Descripción
<i>DFCL_INTERNAL</i>	00 hex	El archivo de datos con la extensión DTA se encuentra en el directorio <i>/usr/Dta</i> .
<i>DFCL_EXTERNAL</i>	01 hex	El archivo de datos con la extensión DTA se encuentra en el directorio <i>/sd0/usr/Dta</i> .
<i>DFCL_TBD</i>	02 hex	No utilizado.

Tipos de datos de variables de sistema de *ExecScript*

Descripción general

En esta sección se describen los tipos de datos de variables de sistema incluidos en las estructuras *ExecScript*.

ExecuteScriptError: códigos de error detectado

Descripción del tipo enumerado

El tipo de datos de enumeración *ExecuteScriptError* contiene los siguientes valores:

Enumerador	Valor	Descripción
<i>CMD_OK</i>	00 hex	No se ha detectado ningún error.
<i>ERR_CMD_UNKNOWN</i>	01 hex	El comando no es válido.
<i>ERR_SD_CARD_MISSING</i>	02 hex	La tarjeta SD no está presente.
<i>ERR_SEE_FWLOG</i>	03 hex	Se ha detectado un error durante la ejecución del comando, consulte <i>FwLog.txt</i> . Para obtener más información, consulte Tipo de archivo (consulte Modicon M241 Logic Controller, Guía de programación).
<i>ERR_ONLY_ONE_COMMAND_ALLOWED</i>	04 hex	Se han intentado ejecutar diversos scripts simultáneamente.
<i>CMD_BEING_EXECUTED</i>	05 hex	Ya hay un script en curso.

Tipos de datos de variables de sistema de *ETH_RW*

Descripción general

En esta sección se enumeran y describen los tipos de datos de las variables de sistema incluidos en las estructuras *ETH_R* y *ETH_W*.

ETH_R_FRAME_PROTOCOL: códigos de protocolo de transmisión de tramas

Descripción del tipo enumerado

El tipo de datos de enumeración *ETH_R_FRAME_PROTOCOL* contiene los siguientes valores:

Enumerador	Valor	Comentario
<i>ETH_R_802_3</i>	00 hex	El protocolo utilizado para la transmisión de tramas es IEEE 802.3.
<i>ETH_R_ETHERNET_II</i>	01 hex	El protocolo utilizado para la transmisión de tramas es Ethernet II.

ETH_R_IP_MODE: códigos de origen de la dirección IP

Descripción del tipo enumerado

El tipo de datos de enumeración *ETH_R_IP_MODE* contiene los siguientes valores:

Enumerador	Valor	Comentario
<i>ETH_R_STORED</i>	00 hex	Se utiliza la dirección IP almacenada.
<i>ETH_R_BOOTP</i>	01 hex	Se usa el protocolo Bootstrap (BOOTP) para obtener una dirección IP.
<i>ETH_R_DHCP</i>	02 hex	Se usa el protocolo DHCP para obtener una dirección IP.
<i>ETH_DEFAULT_IP</i>	FF hex	Se utiliza la dirección IP predeterminada.

ETH_R_PORT_DUPLEX_STATUS: códigos de modo de transmisión

Descripción del tipo enumerado

El tipo de datos de enumeración *ETH_R_PORT_DUPLEX_STATUS* contiene los siguientes valores:

Enumerador	Valor	Comentario
<i>ETH_R_PORT_HALF_DUPLEX</i>	00 hex	Se utiliza la modalidad de transmisión semidúplex.
<i>ETH_R_FULL_DUPLEX</i>	01 hex	Se utiliza la modalidad de transmisión de dúplex completo.
<i>ETH_R_PORT_NA_DUPLEX</i>	03 hex	No se utiliza ninguna modalidad de transmisión de dúplex.

ETH_R_PORT_IP_STATUS: códigos de estado del puerto TCP/IP Ethernet

Descripción del tipo enumerado

El tipo de datos de enumeración *ETH_R_PORT_IP_STATUS* contiene los siguientes valores:

Enumerador	Valor	Comentario
<i>WAIT_FOR_PARAMS</i>	00 hex	En espera de parámetros.
<i>WAIT_FOR_CONF</i>	01 hex	En espera de configuración.
<i>DATA_EXCHANGE</i>	02 hex	Listo para el intercambio de datos.
<i>ETH_ERROR</i>	03 hex	Error detectado en el puerto TCP/IP Ethernet (cable desconectado, configuración no válida, etc.).
<i>DUPLICATE_IP</i>	04 hex	Dirección IP ya usada por otro equipo.

ETH_R_PORT_LINK_STATUS: códigos de estado de la conexión de comunicación

Descripción del tipo enumerado

El tipo de datos de enumeración *ETH_R_PORT_LINK_STATUS* contiene los siguientes valores:

Enumerador	Valor	Comentario
<i>ETH_R_LINK_DOWN</i>	00 hex	Enlace de comunicación a otro dispositivo no disponible.
<i>ETH_R_LINK_UP</i>	01 hex	Enlace de comunicación a otro dispositivo disponible.

ETH_R_PORT_SPEED: códigos de velocidad de comunicación del puerto Ethernet

Descripción del tipo enumerado

El tipo de datos de enumeración *ETH_R_PORT_SPEED* contiene los siguientes valores:

Enumerador	Valor	Comentario
<i>ETH_R_SPEED_NA</i>	0 dec	La velocidad de la red no está disponible.
<i>ETH_R_SPEED_10_MB</i>	10 dec	La velocidad de la red es de 10 megabits por segundo.
<i>ETH_R_100_MB</i>	100 dec	La velocidad de la red es de 100 megabits por segundo.

ETH_R_RUN_IDLE: códigos de estado de ejecución e inactividad Ethernet/IP

Descripción del tipo enumerado

El tipo de datos de enumeración *ETH_R_RUN_IDLE* contiene los siguientes valores:

Enumerador	Valor	Comentario
<i>IDLE</i>	00 hex	La conexión EtherNet/IP está inactiva.
<i>RUN</i>	01 hex	La conexión EtherNet/IP está funcionando.

Tipos de datos de variables de sistema de *TM3_MODULE_RW*

Descripción general

En esta sección se enumeran y describen los tipos de datos de las variables de sistema incluidos en las estructuras *TM3_MODULE_R* y *TM3_MODULE_W*.

TM3_ERR_CODE: códigos de error detectado del módulo de extensión TM3

Descripción del tipo enumerado

El tipo de datos de enumeración *TM3_ERR_CODE* contiene los siguientes valores:

Enumerador	Valor	Comentario
<i>TM3_NO_ERR</i>	00 hex	El último intercambio de bus con el módulo de extensión se ha realizado correctamente.
<i>TM3_ERR_FAILED</i>	01 hex	Error detectado debido a que el último intercambio de bus con el módulo de extensión no se ha realizado correctamente.
<i>TM3_ERR_PARAMETER</i>	02 hex	Error de parámetro detectado en el último intercambio de bus con el módulo.
<i>TM3_ERR_COK</i>	03 hex	Error de hardware temporal o permanente detectado en uno de los módulos de extensión TM3.
<i>TM3_ERR_BUS</i>	04 hex	Error de bus detectado en el último intercambio de bus con el módulo de extensión.

TM3_MODULE_R_ARRAY_TYPE: tipo de matriz de lectura del módulo de extensión TM3

Descripción

TM3_MODULE_R_ARRAY_TYPE es una matriz de 0 a 13 *TM3_MODULE_R_STRUCT*.

TM3_MODULE_STATE: códigos de estado del módulo de extensión TM3

Descripción del tipo enumerado

El tipo de datos de enumeración *TM3_MODULE_STATE* contiene los siguientes valores:

Enumerador	Valor	Comentario
<i>TM3_EMPTY</i>	00 hex	No hay módulo.
<i>TM3_CONF_ERROR</i>	01 hex	El módulo de extensión física no coincide con el configurado en el software.
<i>TM3_BUS_ERROR</i>	02 hex	Error de bus detectado en el último intercambio con el módulo.
<i>TM3_OK</i>	03 hex	El último intercambio de bus con este módulo se ha realizado correctamente.
<i>TM3_MISSING_OPT_MOD</i>	05 hex	El módulo opcional no está físicamente presente.

TM3_BUS_W_IOBUSERRMOD: modalidad de error del bus TM3

Descripción del tipo enumerado

El tipo de datos de enumeración *TM3_BUS_W_IOBUSERRMOD* contiene los siguientes valores:

Enumerador	Valor	Comentario
<i>IOBUS_ERR_ACTIVE</i>	00 hex	Modalidad activa. El controlador lógico detiene todos los intercambios de E/S en el bus TM3 cuando detecta un error permanente. Consulte Descripción general de la configuración de E/S (consulte Modicon M241 Logic Controller, Guía de programación).
<i>IOBUS_ERR_PASSIVE</i>	01 hex	Modalidad pasiva. Los intercambios de E/S continúan en el bus TM3 aunque se detecte un error.

Tipos de datos de variables de sistema del cartucho

Descripción general

En esta sección se enumeran y se describen los tipos de datos de variables de sistema incluidos en la estructura *Cartridge*.

CART_R_ARRAY_TYPE: Tipo de matriz de lectura del cartucho

Descripción

CART_R_ARRAY_TYPE es una matriz de 0 a 1 de tipo *CART_R_STRUCT*.

CART_R_MODULE_ID: Identificador del módulo de lectura del cartucho

Descripción del tipo enumerado

El tipo de datos de enumeración *CART_R_MODULE_ID* contiene los siguientes valores:

Enumerador	Valor	Descripción
<i>CART_R_MODULE_ID</i>	40 hex	TMC4AI2
<i>CART_R_MODULE_ID</i>	41 hex	TMC4AQ2
<i>CART_R_MODULE_ID</i>	42 hex	TMC4TI2
<i>CART_R_MODULE_ID</i>	48 hex	TMC4HOIS01
<i>CART_R_MODULE_ID</i>	49 hex	TMC4PACK01
<i>CART_R_MODULE_ID</i>	FF hex	Ninguno

CART_R_STATE: Estado de lectura del cartucho

Descripción del tipo enumerado

El tipo de datos de enumeración *CART_R_STATE* contiene los siguientes valores:

Enumerador	Valor	Comentario
<i>CONFIGURED</i>	00 hex	El cartucho está configurado.
<i>INITIALIZED_NOT_CONFIGURED</i>	01 hex	El cartucho se ha inicializado, pero no se ha configurado.
<i>NOT_INITIALIZED</i>	02 hex	El cartucho no se ha inicializado.

Tipos de datos de funciones de sistema

Descripción general

En este apartado se describen los diferentes tipos de datos de funciones de sistema de la biblioteca PLCSystem de M241.

IMMEDIATE_ERR_TYPE: GetImmediateFastInput leer entrada de códigos de E/S expertas incrustadas

Descripción del tipo enumerado

El tipo de datos de enumeración contiene los siguientes valores:

Enumerador	Tipo	Comentario
<i>IMMEDIATE_NO_ERROR</i>	WORD	No se han detectado errores.
<i>IMMEDIATE_UNKNOWN</i>	WORD	La referencia de la función <i>Immediate</i> es incorrecta o no está configurada.
<i>IMMEDIATE_UNKNOWN_PARAMETER</i>	WORD	Una referencia de parámetro es incorrecta.

RTCSETDRIFT_ERROR: Códigos de error detectados de la función SetRTCDrift

Descripción del tipo enumerado

El tipo de datos de enumeración *RTCSETDRIFT_ERROR* contiene los siguientes valores:

Enumerador	Valor	Comentario
<i>RTC_OK</i>	00 hex	Desviación del RTC correctamente configurada.
<i>RTC_BAD_DAY</i>	01 hex	No utilizado.
<i>RTC_BAD_HOUR</i>	02 hex	No utilizado.
<i>RTC_BAD_MINUTE</i>	03 hex	No utilizado.
<i>RTC_BAD_DRIFT</i>	04 hex	Parámetro de desviación de RTC fuera de rango.
<i>RTC_INTERNAL_ERROR</i>	05 hex	Ajustes de desviación de RTC rechazados en el error interno detectado.

Apéndices

Contenido de esta parte

Representación de funciones y de bloques de funciones	60
---	----

Descripción general

Este apéndice contiene extractos de la guía de programación que ayudan a obtener una comprensión técnica de la documentación de la biblioteca.

Representación de funciones y de bloques de funciones

Contenido de este capítulo

Diferencias entre una función y un bloque de funciones	60
Usar una función o un bloque de funciones en lenguaje IL	61
Usar una función o un bloque de funciones en lenguaje ST	64

Descripción general

Cada función se puede representar en los lenguajes siguientes:

- IL: Lista de instrucciones
- ST: Texto estructurado
- LD: Diagrama de contactos
- FBD: Diagrama de bloques de funciones
- CFC: Diagrama de función continua

En este capítulo se proporcionan funciones y ejemplos de representación de bloques de funciones y se describe cómo utilizarlas en lenguajes IL y ST.

Diferencias entre una función y un bloque de funciones

Función

Una función:

- es una POU (Unidad de organización de programa) que devuelve un resultado inmediato.
- Se le llama directamente por su nombre (y no a través de una instancia).
- No tiene instancias creadas.
- Se puede utilizar como un operando en otras expresiones.

Ejemplos: operadores booleanos (AND), cálculos, conversión (BYTE_TO_INT)

Bloque de funciones

Un bloque de funciones:

- Es una POU (Unidad de organización de programa) que devuelve una o más salidas.
- Debe ser convocado por una instancia (copia del bloque de funciones con un nombre y variables específicos).
- Todas las instancias tienen un estado persistente (salidas y variables internas) de una llamada a otra desde un bloque de funciones o programa.

Ejemplos: temporizadores, contadores

En el ejemplo, `Timer_ON` es una instancia del bloque de funciones `TON`:

```

1  PROGRAM MyProgram_ST
2  VAR
3      Timer_ON: TON; // Function Block Instance
4      Timer_RunCd: BOOL;
5      Timer_PresetValue: TIME := T#5S;
6      Timer_Output: BOOL;
7      Timer_ElapsedTime: TIME;
8  END_VAR

1  Timer_ON(
2      IN:=Timer_RunCd,
3      PT:=Timer_PresetValue,
4      Q=>Timer_Output,
5      ET=>Timer_ElapsedTime);
    
```

Usar una función o un bloque de funciones en lenguaje IL

Información general

En esta sección se describe cómo implementar una función y un bloque de funciones en lenguaje IL.

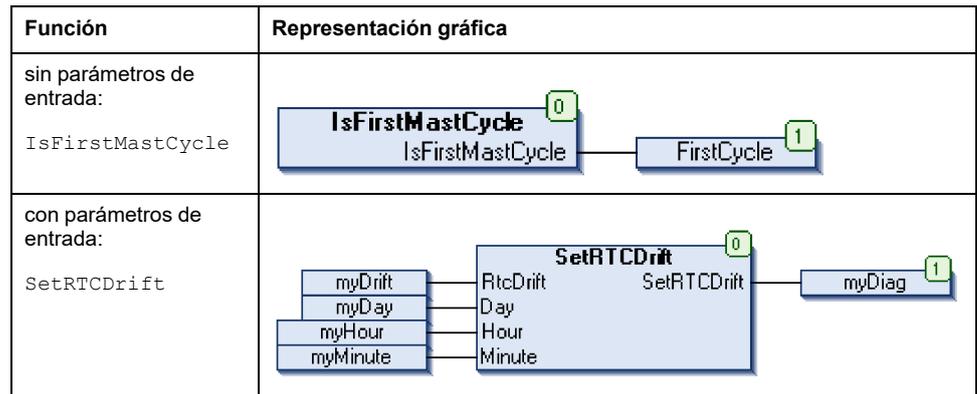
Las funciones `IsFirstMastCycle` y `SetRTCDrift` y el bloque de funciones `TON` se usan como ejemplos para mostrar implementaciones.

Usar una función en lenguaje IL

En este procedimiento se describe cómo insertar una función en lenguaje IL:

Paso	Acción
1	Abra o cree una POU nueva en el lenguaje de Lista de instrucciones (IL). NOTA: Aquí no se detalla el procedimiento para crear una POU. Para obtener más información, consulte <code>Añadir y llamar POU</code> que se puede encontrar en la <code>Guía de programaciónEcoStruxure Automation Expert - Motion / EcoStruxure Machine Expert</code> .
2	Cree las variables que necesite la función.
3	Si la función tiene 1 o más entradas, empiece a cargar la primera entrada usando la instrucción <code>LD</code> .
4	Inserte una nueva línea abajo y: <ul style="list-style-type: none"> escriba el nombre de la función en la columna de operadores (campo izquierdo); o utilice la opción Accesibilidad para seleccionar la función (seleccione Insertar módulo en el menú contextual).
5	Si la función tiene más de una entrada y se usa <code>Accesibilidad</code> , se crea automáticamente el número necesario de líneas con <code>???</code> en los campos de la derecha. Sustituya los <code>???</code> por el valor o la variable adecuada que corresponda al orden de las entradas.
6	Inserte una línea nueva para almacenar el resultado de la función en la variable correspondiente: escriba la instrucción <code>ST</code> en la columna del operador (campo de la izquierda) y el nombre de la variable en el campo de la derecha.

Para ilustrar el procedimiento, considere las funciones `IsFirstMastCycle` (sin parámetro de entrada) y `SetRTCDrift` (con parámetros de entrada) que se representan gráficamente a continuación:



En lenguaje IL, el nombre de la función se usa directamente en la columna de operadores:

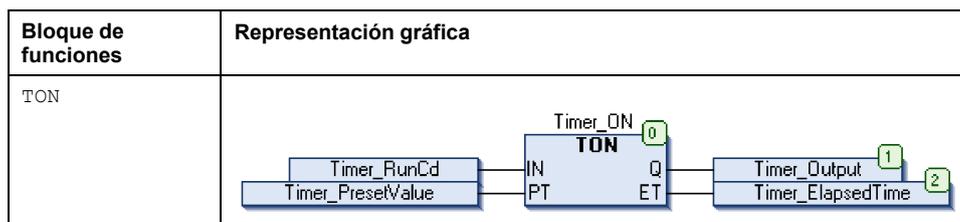
Función	Representación en el Editor IL de POU
Ejemplo en IL de una función sin parámetros de entrada: <code>IsFirstMastCycle</code>	<pre> 1 PROGRAM MyProgram_IL 2 VAR 3 FirstCycle: BOOL; 4 END_VAR 5 </pre> <hr/> <pre> 1 IsFirstMast Cycle ST FirstCycle </pre>
Ejemplo en IL de una función con parámetros de entrada: <code>SetRTCDrift</code>	<pre> 1 PROGRAM MyProgram_IL 2 VAR 3 myDrift: SINT (-29..29) := 5; 4 myDay: DAY_OF_WEEK := SUNDAY; 5 myHour: HOUR := 12; 6 myMinute: MINUTE; 7 myDiag: RTCSETDRIFT_ERROR; 8 END_VAR 9 </pre> <hr/> <pre> 1 LD myDrift SetRTCDri ft myDay myHour myMinute ST myDiag </pre>

Usar un bloque de funciones en lenguaje IL

En este procedimiento se describe cómo insertar un bloque de funciones en lenguaje IL:

Paso	Acción
1	Abra o cree una POU nueva en el lenguaje de Lista de instrucciones (IL). NOTA: Aquí no se detalla el procedimiento para crear una POU. Para obtener más información, consulte Añadir y llamar POU que se puede encontrar en la Guía de programaciónEcoStruxure Automation Expert - Motion / EcoStruxure Machine Expert .
2	Cree las variables que necesita el bloque de funciones, incluido el nombre de instancia.
3	Se llama a los bloques de funciones usando una instrucción CAL: <ul style="list-style-type: none"> Utilice la opción de Accesibilidad para seleccionar el bloque de funciones (haga clic con el botón derecho y seleccione Insertar módulo en el menú contextual). La instrucción CAL y la E/S necesaria se crean automáticamente. Cada parámetro (E/S) es una instrucción: <ul style="list-style-type: none"> Los valores de las entradas se establecen con ":=". Los valores de las salidas se establecen con "=>".
4	En el campo de la derecha CAL, sustituya ??? por el nombre de la instancia.
5	Sustituya otros ??? por una variable apropiada o un valor inmediato.

Para ilustrar el procedimiento, considere este ejemplo con el bloque de funciones TON que se representa gráficamente a continuación:



En lenguaje IL, el nombre del bloque de funciones se usa directamente en la columna de operadores:

Bloque de funciones	Representación en el Editor IL de POU
TON	<pre> 1 PROGRAM MyProgram_IL 2 VAR 3 Timer_ON: TON; // Function Block instance declaration 4 Timer_RunCd: BOOL; 5 Timer_PresetValue: TIME := T#5S; 6 Timer_Output: BOOL; 7 Timer_ElapsedTime: TIME; 8 END_VAR 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000 </pre>

Usar una función o un bloque de funciones en lenguaje ST

Información general

En esta sección se describe el modo de implementar una función y un bloque de funciones en lenguaje ST.

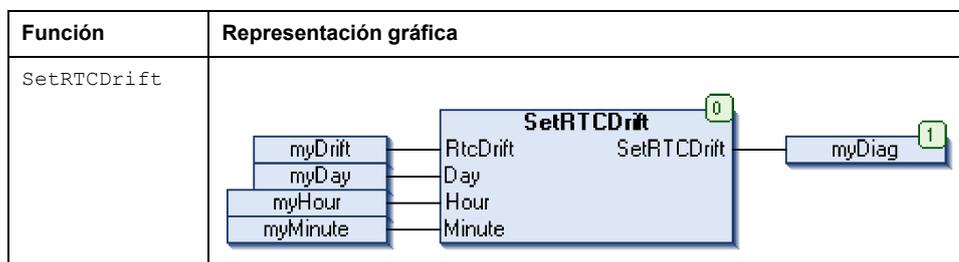
La función `SetRTCDrift` y el bloque de funciones `TON` se usan como ejemplos para mostrar implementaciones.

Usar una función en lenguaje ST

En este procedimiento se describe cómo insertar una función en lenguaje ST:

Paso	Acción
1	Abra o cree una POU nueva en el lenguaje de Texto estructurado (ST). NOTA: Aquí no se detalla el procedimiento para crear una POU. Para obtener más información, consulte Añadir y llamar POU que se puede encontrar en la Guía de programación EcoStruxure Automation Expert - Motion / EcoStruxure Machine Expert .
2	Cree las variables que necesite la función.
3	Utilice la sintaxis general en el Editor POU ST para el lenguaje ST de una función. La sintaxis general es: <code>FunctionResult := FunctionName (VarInput1, VarInput2, .. VarInputx);</code>

Para ilustrar el procedimiento, considere la función `SetRTCDrift` que se representa gráficamente a continuación:



El lenguaje ST de esta función es este:

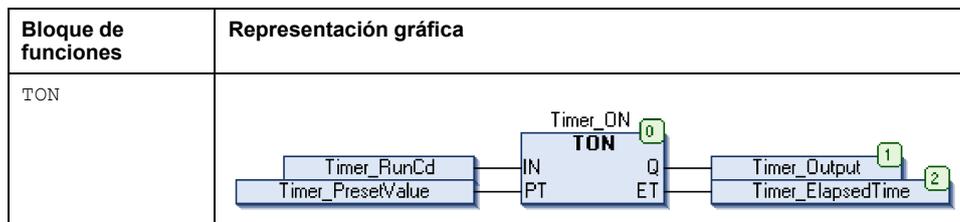
Función	Representación en el editor POU ST
SetRTCDrift	<pre>PROGRAM MyProgram_ST VAR myDrift: SINT (-36..+73) := 5; myDay: sec.DAY_OF_WEEK := SUNDAY; myHour: sec.HOUR := 12; myMinute: sec.MINUTE; myRTCAdjust: sec.RTCDRIFT_ERROR; END VAR myRTCAdjust := SetRTCDrift(myDrift, myDay, myHour, myMinute);</pre>

Usar un bloque de funciones en lenguaje ST

En este procedimiento se describe cómo insertar un bloque de funciones en lenguaje ST:

Paso	Acción
1	Abra o cree una POU nueva en el lenguaje de Texto estructurado (ST). NOTA: Aquí no se detalla el procedimiento para crear una POU. Para obtener más información, consulte Añadir y llamar POU que se puede encontrar en la Guía de programaciónEcoStruxure Automation Expert - Motion / EcoStruxure Machine Expert .
2	Cree las variables de entrada y salida y la instancia requeridas para el bloque de funciones: <ul style="list-style-type: none"> Las variables de entrada son los parámetros de entrada requeridos por el bloque de funciones Las variables de salida reciben el valor devuelto por el bloque de funciones
3	Utilice la sintaxis general en el Editor POU ST para el lenguaje ST de un bloque de funciones. La sintaxis general es: FunctionBlock_InstanceName (Input1:=VarInput1, Input2:=VarInput2, ... Ouput1=>VarOutput1, Ouput2=>VarOutput2, ...);

Para ilustrar el procedimiento, considere este ejemplo con el bloque de funciones TON que se representa gráficamente a continuación:



En esta tabla se muestran ejemplos de una llamada de bloque de funciones en lenguaje ST:

Bloque de funciones	Representación en el editor POU ST
TON	<pre> 1 PROGRAM MyProgram_ST 2 VAR 3 Timer_ON: TON; // Function Block Instance 4 Timer_RunCd: BOOL; 5 Timer_PresetValue: TIME := T#5S; 6 Timer_Output: BOOL; 7 Timer_ElapsedTime: TIME; 8 END_VAR 1 Timer_ON(2 IN:=Timer_RunCd, 3 PT:=Timer_PresetValue, 4 Q=>Timer_Output, 5 ET=>Timer_ElapsedTime); </pre>

Glosario

A

Aplicación de arranque:

(*aplicación de arranque*) El archivo binario que contiene la aplicación. Normalmente está guardada en el controlador y permite que este arranque en la aplicación generada por el usuario.

aplicación:

Un programa que incluye datos de configuración, símbolos y documentación.

ARRAY:

La disposición sistemática de objetos de datos de un solo tipo en forma de tabla definida en la memoria del controlador lógico. La sintaxis es la siguiente: ARRAY [<dimensión>] OF <Tipo>

Ejemplo 1: ARRAY [1..2] OF BOOL es una tabla de una dimensión compuesta por dos elementos de tipo BOOL.

Ejemplo 2: ARRAY [1..10, 1..20] OF INT es una tabla de dos dimensiones compuesta por 10 x 20 elementos de tipo INT.

B

bloque de funciones:

Una unidad de programación que dispone de una o varias entradas y devuelve una o varias salidas. Los FBs se llaman mediante una instancia (copia del bloque de funciones con nombre y variables dedicados), y todas las instancias tienen un estado persistente (salidas y variables internas) de una llamada a otra.

Ejemplos: temporizadores, contadores

BOOL:

(*booleano*) Un tipo de datos básico en informática. Una variable BOOL puede tener uno de estos valores: 0 (FALSE) o 1 (TRUE). Un bit extraído de una palabra es de tipo BOOL, por ejemplo, %MW10.4 es un quinto bit con un número de palabra de memoria 10.

BOOTP:

(*protocolo bootstrap*) Un protocolo de red UDP que puede utilizar un cliente de red para obtener de forma automática una dirección IP (y tal vez otros datos) de un servidor. El cliente se identifica ante el servidor utilizando la dirección MAC del cliente. El servidor, que mantiene una tabla preconfigurada de direcciones MAC de los dispositivos cliente y las direcciones IP asociadas, envía al cliente su dirección IP predefinida. BOOTP se utilizaba originariamente como un método que permitía iniciar los hosts sin disco de forma remota por una red. El proceso BOOTP asigna un arrendamiento infinito de una dirección IP. El servicio BOOTP utiliza los puertos UDP 67 y 68.

byte:

Un tipo que está codificado en un formato de 8 bits que, en el formato hexadecimal, va de 00 hex a FF hex.

C

cadena:

Una variable que es una serie de caracteres ASCII.

CAN:

(*red de área del controlador*) Un protocolo (ISO 11898) para redes de bus serie diseñado para la interconexión de dispositivos inteligentes (de varios fabricantes) en sistemas inteligentes para aplicaciones industriales en tiempo real. CAN, diseñado originariamente para utilizarlo en automóviles, se emplea actualmente en una amplia variedad de entornos industriales de control automático.

CFC:

(*diagrama de función continua*) Un lenguaje de programación (una ampliación del estándar IEC 61131-3) basado en el lenguaje de diagrama de bloque de funciones (FBD) y que funciona como un diagrama de flujo. Sin embargo, no se utiliza ninguna red y es posible un posicionamiento libre de elementos gráficos, lo que permite bucles de realimentación. En cada bloque, las entradas se sitúan a la izquierda y las salidas, a la derecha. Las salidas del bloque se pueden conectar a las entradas de otros bloques para formar expresiones complejas.

configuración:

Organización e interconexión de los componentes de hardware en un sistema y los parámetros del hardware y software que determina las características operativas del sistema.

CRC:

(*comprobación de redundancia cíclica*) Método que se emplea para determinar la validez de la transmisión de la comunicación. La transmisión contiene un campo de bits que constituye una suma de comprobación. El mensaje se usa para que el transmisor calcule la suma de comprobación según el contenido del mensaje. A continuación, los nodos receptores recalculan el campo de la misma manera. Toda discrepancia en el valor de los dos cálculos CRC indica que el mensaje transmitido y el mensaje recibido son diferentes.

D**DHCP:**

(*protocolo de configuración dinámica del host*) Una ampliación avanzada de BOOTP. DHCP es más avanzado, pero tanto DHCP como BOOTP son habituales. (DHCP puede manejar las solicitudes de clientes BOOTP).

diagrama de bloques de funciones:

Uno de los cinco lenguajes para lógica o control que cumplen con el estándar IEC 61131-3 para sistemas de control. El diagrama de bloques de funciones es un lenguaje de programación orientado gráficamente. Funciona con una lista de redes en la que cada red contiene una estructura gráfica de cuadros y líneas de conexión que representa una expresión lógica o aritmética, la llamada de un bloque de funciones, un salto o una instrucción de retorno.

dirección MAC:

(*dirección de control de acceso a medios*) Un número único de 48 bits asociado a una parte específica del hardware. La dirección MAC se programa en cada tarjeta de red o dispositivo cuando se fabrica.

DWORD:

(*palabra doble*) Con codificación en formato de 32 bits.

E**ejecución:**

Un comando que hace que el controlador explore el programa de la aplicación, lea las entradas físicas y escriba en las salidas físicas según la solución de la lógica del programa.

elemento:

El nombre abreviado de ARRAY.

E/S:

(*entrada/salida*)

EtherNet/IP:

(*protocolo industrial de Ethernet*) Un protocolo de comunicaciones abiertas para fabricar soluciones de automatización en sistemas industriales. EtherNet/IP se incluye en una familia de redes que implementan el protocolo industrial común en sus capas superiores. La organización de apoyo (ODVA) especifica EtherNet/IP para cumplir la adaptabilidad y la independencia de los medios.

Ethernet:

Una tecnología de capas física y de conexión de datos para LANs, también conocida como IEEE 802.3.

F**FB:**

(*bloque de funciones*) Un práctico mecanismo de programación que consolida un grupo de instrucciones de programación para realizar una acción específica y normalizada, por ejemplo, el control de velocidad, el control de intervalo o el conteo. Un bloque de funciones se puede componer de datos de configuración, un conjunto de parámetros de funcionamiento internos o externos y, normalmente, una o diversas entradas y salidas de datos.

firmware:

Representa el BIOS, los parámetros de datos y las instrucciones de programación que constituyen el sistema operativo en un controlador. El firmware se almacena en la memoria no volátil del controlador.

función:

Una unidad de programación que dispone de una entrada y devuelve un resultado inmediato. No obstante, a diferencia de los FBs, se llama directamente por su nombre (y no mediante una instancia), no tiene un estado persistente desde una llamada hasta la siguiente y se puede utilizar como un operando en otras expresiones de programación.

Ejemplos: operadores booleanos (AND), cálculos, conversiones (BYTE_TO_INT)

G**GVL:**

(*lista de variables globales*) Gestiona las variables globales dentro de un proyecto EcoStruxure Machine Expert.

H**hex:**

(*hexadecimal*)

I**ID:**

(*identificador/identificación*)

IEC 61131-3:

Tercera parte de un estándar de tres partes de la IEC para los equipos de automatización industriales. IEC 61131-3 se ocupa de los lenguajes de programación del controlador y define dos estándares de lenguajes de programación gráficos y dos textuales. Los lenguajes de programación gráficos son un diagrama de contactos y un diagrama de bloque de funciones. Los lenguajes de programación textuales incluyen texto estructurado y lista de instrucciones.

IEC:

(*International Electrotechnical Commission*) Una organización de estándares internacional sin ánimo de lucro y no gubernamental que prepara y publica estándares internacionales para todas las tecnologías eléctricas, electrónicas y relacionadas.

IEEE 802.3:

Una recopilación de estándares de IEEE que definen la capa física y la subcapa de control de acceso a medios de la capa de conexión de datos, de Ethernet cableado.

IL:

(*lista de instrucciones*) Un programa escrito en lenguaje que se compone de una serie de instrucciones basadas en texto y ejecutadas secuencialmente por el controlador. Cada instrucción incluye un número de línea, un código de instrucción y un operando (consulte IEC 61131-3).

INT:

(*entero*) Un número entero con codificación de 16 bits.

IP:

(*protocolo de Internet*) Parte de la familia de protocolos TCP/IP que hace un seguimiento de las direcciones de Internet de los dispositivos, encamina los mensajes salientes y reconoce los mensajes entrantes.

L

LD:

(*diagrama de contactos*) Una representación gráfica de instrucciones de un programa de controlador con símbolos para contactos, bobinas y bloques en una serie de escalones ejecutados de forma secuencial por un controlador (consulte IEC 61131-3).

LED:

(*diodo electroluminiscente*) Un indicador que se ilumina con una carga eléctrica de nivel bajo.

LWORD:

(*palabra larga*) Un tipo de datos con codificación en formato de 64 bits.

M

MAST:

Una tarea del procesador que se ejecuta en el software de programación. La tarea MAST consta de dos secciones:

- **IN:** las entradas se copian en la sección IN antes de ejecutar la tarea MAST.
- **OUT:** las salidas se copian en la sección OUT después de ejecutar la tarea MAST.

Modbus:

El protocolo de comunicaciones que permite las comunicaciones entre muchos dispositivos conectados a la misma red.

%MW:

Según el estándar IEC, %MW representa un registro de palabra de memoria (por ejemplo, un objeto de lenguaje del tipo palabra de memoria).

N**NVM:**

(Memoria no volátil) Memoria no volátil que se puede sobrescribir. Se almacena en una memoria EEPROM especial que se puede borrar y volver a programar.

P**PCI:**

(*interconexión de componentes periféricos*) Un bus estándar en el sector para la conexión de periféricos.

PLC:

(*controlador lógico programable*) Un ordenador industrial que se usa para automatizar procesos industriales, de fabricación y otros procesos electromecánicos. Los PLCs se diferencian de los ordenadores comunes en que están diseñados de forma que tienen varias matrices de entrada y salida, y que disponen de especificaciones más sólidas contra los golpes, las vibraciones, la temperatura, las interferencias eléctricas, etc.

POU:

(*unidad de organización de programas*) Una declaración variable en el código fuente y el conjunto de instrucciones correspondiente. Las POU facilitan la reutilización modular de programas de software, funciones y bloques de funciones. Una vez declaradas, cada una de las POU está disponible para las otras.

programa :

El componente de una aplicación consistente en código fuente compilado capaz de poder ser instalado en la memoria de un controlador lógico.

protocolo:

Una convención o una definición de norma que controla o habilita la conexión, la comunicación y la transferencia de datos entre dos sistemas o dispositivos informáticos.

R**red de control:**

Red que contiene logic controllers, sistemas SCADA, PC, HMI, conmutadores, etc.

Se admiten dos tipos de topología:

- Plana: todos los módulos y dispositivos de esta red pertenecen a la misma subred.
- Dos niveles: la red se divide en una red operativa y en una red de controladores.

Estas dos redes pueden ser físicamente independientes, pero normalmente están conectadas mediante un dispositivo de enrutamiento.

red:

Un sistema de dispositivos interconectados que comparten una ruta de datos común y un protocolo de comunicaciones.

RTC:

(reloj de tiempo real) Un reloj calendario de fecha/hora con respaldo de batería que funciona de forma continua aunque el controlador no reciba alimentación, mientras dure la batería.

S

%:

Según el estándar IEC, % es un prefijo que identifica direcciones de memoria interna en el controlador lógico que se utilizan para almacenar el valor de las variables del programa, constantes, E/S, etc.

SINT:

(entero con signo) Un valor de 15 bits más signo.

STOP:

Comando que hace que el controlador detenga la ejecución de un programa de aplicación.

ST:

(texto estructurado) Un lenguaje que incluye instrucciones complejas y anidadas (por ejemplo, bucles de repetición, ejecuciones condicionales o funciones). ST cumple con IEC 61131-3.

T

tarea:

Grupo de secciones y subrutinas ejecutadas cíclica o periódicamente si se trata de la tarea MAST, o periódicamente si se trata de la tarea FAST.

Una tarea siempre tiene un nivel de prioridad y tiene asociadas entradas y salidas del controlador. Estas E/S se actualizan en función de la tarea.

Un controlador puede tener diversas tareas.

TCP:

(protocolo de control de transmisión) Un protocolo de capas de transporte basado en conexiones que proporciona una transmisión de datos simultánea y bidireccional. TCP forma parte del conjunto de protocolos TCP/IP.

U

UDINT:

(entero doble sin signo) Codificado en 32 bits.

UINT:

(entero sin signo) Codificado en 16 bits.

V

variable del sistema:

Una variable que proporciona datos del controlador e información de diagnóstico, y permite enviar comandos al controlador.

variable no ubicada:

Una variable que no tiene dirección (consulte *variable ubicada*).

variable:

Una unidad de memoria direccionada y modificada por un programa.

W**watchdog:**

Un watchdog es un cronómetro especial utilizado para garantizar que los programas no superen su tiempo de exploración asignado. El cronómetro watchdog suele configurarse con un valor superior al tiempo de exploración y se resetea a 0 cuando termina cada ciclo de exploración. Si el cronómetro watchdog alcanza el valor predeterminado, por ejemplo, porque el programa queda atrapado en un bucle infinito, se declara un error y el programa se detiene.

WORD:

Un tipo codificado en formato de 16 bits.

Schneider Electric
35 rue Joseph Monier
92500 Rueil Malmaison
France

+ 33 (0) 1 41 29 70 00

www.se.com

Debido a que las normas, especificaciones y diseños cambian periódicamente, solicite la confirmación de la información dada en esta publicación.

© 2024 Schneider Electric. Reservados todos los derechos.

EIO0000003068.06