

Modicon M251 Logic Controller

Funciones y variables del sistema Guía de la biblioteca PLCSystem

12/2019



La información que se ofrece en esta documentación contiene descripciones de carácter general y/o características técnicas sobre el rendimiento de los productos incluidos en ella. La presente documentación no tiene como objeto sustituir dichos productos para aplicaciones de usuario específicas, ni debe emplearse para determinar su idoneidad o fiabilidad. Los usuarios o integradores tienen la responsabilidad de llevar a cabo un análisis de riesgos adecuado y completo, así como la evaluación y las pruebas de los productos en relación con la aplicación o el uso de dichos productos en cuestión. Ni Schneider Electric ni ninguna de sus filiales o asociados asumirán responsabilidad alguna por el uso inapropiado de la información contenida en este documento. Si tiene sugerencias de mejoras o modificaciones o ha hallado errores en esta publicación, le rogamos que nos lo notifique.

Usted se compromete a no reproducir, salvo para su propio uso personal, no comercial, la totalidad o parte de este documento en ningún soporte sin el permiso de Schneider Electric, por escrito. También se compromete a no establecer ningún vínculo de hipertexto a este documento o su contenido. Schneider Electric no otorga ningún derecho o licencia para el uso personal y no comercial del documento o de su contenido, salvo para una licencia no exclusiva para consultarla "tal cual", bajo su propia responsabilidad. Todos los demás derechos están reservados.

Al instalar y utilizar este producto es necesario tener en cuenta todas las regulaciones sobre seguridad correspondientes, ya sean regionales, locales o estatales. Por razones de seguridad y para garantizar que se siguen los consejos de la documentación del sistema, las reparaciones solo podrá realizarlas el fabricante.

Cuando se utilicen dispositivos para aplicaciones con requisitos técnicos de seguridad, siga las instrucciones pertinentes.

Si con nuestros productos de hardware no se utiliza el software de Schneider Electric u otro software aprobado, pueden producirse lesiones, daños o un funcionamiento incorrecto del equipo.

Si no se tiene en cuenta esta información, se pueden causar daños personales o en el equipo.

© 2019 Schneider Electric. Reservados todos los derechos.

Tabla de materias



	Información de seguridad	7
	Acerca de este libro	9
Capítulo 1	M251 Variables del sistema	13
1.1	Variables del sistema: definición y uso	14
	Descripción de las variables de sistema	15
	Utilización de variables de sistema	17
1.2	Estructuras PLC_R y PLC_W	19
	PLC_R: Variables de sistema de sólo lectura del controlador	20
	PLC_W: Variables de sistema de lectura/escritura del controlador	24
1.3	Estructuras SERIAL_R y SERIAL_W	25
	SERIAL_R[0 . . . 1]: Variables del sistema de solo lectura de la línea serie	26
	SERIAL_W[0 . . . 1]: Variables del sistema de lectura/escritura de línea serie	27
1.4	Estructuras ETH_R y ETH_W	28
	ETH_R: Variables de sistema de solo lectura del puerto Ethernet	29
	ETH_W: Variables de sistema de lectura/escritura del puerto Ethernet	35
1.5	Estructura de TM3_MODULE_R	36
	TM3_MODULE_R[0 . . . 13]: Variables de sistema de solo lectura de los módulos TM3	36
1.6	TM3_BUS_W Structure	37
	TM3_BUS_W: Variables de sistema del bus TM3	37
1.7	Estructura PROFIBUS_R	38
	PROFIBUS_R: Variables de sistema de solo lectura PROFIBUS	38
Capítulo 2	Funciones de sistema M251,	39
2.1	Funciones de lectura M251	40
	GetRtc: Obtener reloj de tiempo real	41
	IsFirstMastColdCycle: indica si este ciclo es el primer ciclo MAST del arranque en frío	42
	IsFirstMastCycle: indica si este ciclo es el primer ciclo MAST	43
	IsFirstMastWarmCycle: indica si este ciclo es el primer ciclo MAST del arranque en caliente	45
2.2	Funciones de escritura de M251	46
	SetRTCDrift: Definir valor de compensación en el RTC	46

2.3	Funciones de usuario de M251	48
	FB_ControlClone: Clonar el controlador	49
	DataFileCopy: Comandos para copiar archivos	50
	ExecuteScript: Ejecutar comandos de script	53
2.4	Funciones de espacio en disco de M251.....	55
	FC_GetFreeDiskSpace: Obtiene el espacio de memoria libre.....	56
	FC_GetLabel: Obtiene la etiqueta de la memoria	57
	FC_GetTotalDiskSpace: Obtiene el tamaño de la memoria.....	58
2.5	Funciones de lectura TM3	59
	TM3_GetModuleBusStatus: Obtener estado del bus del módulo TM3	60
	TM3_GetModuleFWVersion: Obtener versión del firmware del módulo TM3	61
	TM3_GetModuleInternalStatus: Obtener estado interno del módulo TM3	62
Capítulo 3	Tipos de datos de la biblioteca PLCSystem de M251 . . .	65
3.1	Tipos de datos de variables de sistema de PLC_RW	66
	PLC_R_APPLICATION_ERROR: Códigos de estado del error detectado de la aplicación	67
	PLC_R_BOOT_PROJECT_STATUS: Códigos de estado del proyecto de inicio	69
	PLC_R_IO_STATUS: Códigos de estado de E/S	70
	PLC_R_SDCARD_STATUS: Códigos de estado de slots para tarjeta SD	71
	PLC_R_STATUS: Códigos de estado del controlador.....	72
	PLC_R_STOP_CAUSE: Códigos de causa de transición de estado RUN a otro estado	73
	PLC_R_TERMINAL_PORT_STATUS: Códigos de estado de la conexión del puerto de programación	75
	PLC_R_TM3_BUS_STATE: Códigos de estado del bus TM3	76
	PLC_W_COMMAND: Códigos de comando de control	77
3.2	Tipos de datos de variables del sistema de DataFileCopy	78
	DataFileCopyError: Códigos de errores detectados	79
	DataFileCopyLocation: Códigos de ubicación	80
3.3	Tipos de datos de variables del sistema de ExecScript.....	81
	ExecuteScriptError: Códigos de errores detectados	81

3.4	Tipos de datos de variables de sistema de ETH_RW	82
	ETH_R_FRAME_PROTOCOL: Códigos de protocolo de transmisión de tramas	83
	ETH_R_IP_MODE: Códigos de origen de la dirección IP	84
	ETH_R_PORT_DUPLEX_STATUS: Códigos de modalidad de transmisión	85
	ETH_R_PORT_IP_STATUS: Códigos de estado del puerto TCP/IP Ethernet	86
	ETH_R_PORT_LINK_STATUS: Códigos de estado de enlaces de comunicación	87
	ETH_R_PORT_SPEEDETH_R_PORT_SPEED: Códigos de velocidad de comunicación del puerto Ethernet	88
	ETH_R_RUN_IDLE: Códigos de estado de ejecución e inactividad de Ethernet/IP	89
3.5	Tipos de datos de variables de sistema de TM3_MODULE_RW	90
	TM3_ERR_CODE: Códigos de error detectados en el módulo de ampliación TM3	91
	TM3_MODULE_R_ARRAY_TYPE: Tipo de matriz de lectura del módulo de ampliación TM3	92
	TM3_MODULE_STATE: Códigos de estado del módulo de ampliación TM3	93
	TM3_BUS_W_IOBUSERRMOD: Modalidad de error de bus TM3	94
3.6	Tipos de datos de funciones de sistema	95
	RTCSETDRIFT_ERROR: Códigos de error de función SetRTCdrift detectados	95
	Apéndices	97
	Apéndice A Representación de funciones y de bloques de funciones	99
	Diferencias entre una función y un bloque de funciones	100
	Cómo utilizar una función o un bloque de funciones en lenguaje IL	101
	Cómo utilizar una función o un bloque de funciones en lenguaje ST	105
	Glosario	109
	Índice	117

Información de seguridad



Información importante

AVISO

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 LO SIGUIENTE:

La instalación, el manejo, las revisiones y el 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

Objeto

Este documento le permite familiarizarse con las funciones y variables del sistema de Modicon M251 Logic Controller. La biblioteca M251 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 M251 PLCSystem.

Se requieren los conocimientos siguientes:

- Información básica sobre la funcionalidad, la estructura y la configuración de M251 Logic Controller
- Programación en lenguaje FBD, LD, ST, IL o CFC
- Variables del sistema (variables globales)

Campo de aplicación

Este documento se ha actualizado para la publicación de EcoStruxure™ Machine Expert V1.2.

Documentos relacionados

Título de la documentación	Número de referencia
Guía de programación de EcoStruxure Machine Expert	<i>EIO0000002854 (ENG);</i> <i>EIO0000002855 (FRE);</i> <i>EIO0000002856 (GER);</i> <i>EIO0000002858 (SPA);</i> <i>EIO0000002857 (ITA);</i> <i>EIO0000002859 (CHS)</i>
Modicon M251 Logic Controller - Guía de hardware	<i>EIO0000003101 (ENG);</i> <i>EIO0000003102 (FRE);</i> <i>EIO0000003103 (GER);</i> <i>EIO0000003104 (SPA);</i> <i>EIO0000003105 (ITA);</i> <i>EIO0000003106 (CHS)</i>
Guía de programación de Modicon M251 Logic Controller	<i>EIO0000003089 (ENG);</i> <i>EIO0000003090 (FRE);</i> <i>EIO0000003091 (GER);</i> <i>EIO0000003092 (SPA);</i> <i>EIO0000003093 (ITA);</i> <i>EIO0000003094 (CHS)</i>

Puede descargar estas publicaciones técnicas y otra información técnica de nuestro sitio web <https://www.se.com/ww/en/download/> .

ADVERTENCIA

PÉRDIDA DE CONTROL

- El diseñador del esquema de control debe tener en cuenta las posibles modalidades de fallo de rutas de control y, para ciertas funciones de control críticas, proporcionar los medios para lograr un estado seguro durante y después de un fallo de ruta. Algunas funciones de control críticas son, por ejemplo, la parada de emergencia y la parada de sobrecarrera, un corte de alimentación o un reinicio.
- Para las funciones críticas de control deben proporcionarse rutas de control separadas o redundantes.
- Las rutas de control del sistema pueden incluir enlaces de comunicación. Deben tenerse en cuenta las implicaciones de retardos de transmisión imprevistos o fallos del enlace.
- Tenga en cuenta todas las reglamentaciones para la prevención de accidentes y las normativas de seguridad locales.¹
- Cada instalación de este equipo debe probarse de forma individual y exhaustiva antes de entrar en servicio.

El incumplimiento de estas instrucciones puede causar la muerte, lesiones serias o daño al 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 solo software aprobado por Schneider Electric para este equipo.
- Actualice el programa de aplicación siempre que cambie la configuración de hardware física.

El incumplimiento de estas instrucciones puede causar la muerte, lesiones serias o daño al equipo.

Capítulo 1

M251 Variables del sistema

Descripción general

En este capítulo:

- Se proporciona una introducción a las variables de sistema (*véase página 14*).
- Se describen las variables de sistema (*véase página 20*) incluidas en la biblioteca PLCSystem de M251.

Contenido de este capítulo

Este capítulo contiene las siguientes secciones:

Sección	Apartado	Página
1.1	Variables del sistema: definición y uso	14
1.2	Estructuras PLC_R y PLC_W	19
1.3	Estructuras SERIAL_R y SERIAL_W	25
1.4	Estructuras ETH_R y ETH_W	28
1.5	Estructura de TM3_MODULE_R	36
1.6	TM3_BUS_W Structure	37
1.7	Estructura PROFIBUS_R	38

Sección 1.1

Variables del sistema: definición y uso

Descripción general

En esta sección se definen las variables del sistema y cómo implementarlas en Modicon M251 Logic Controller.

Contenido de esta sección

Esta sección contiene los siguientes apartados:

Apartado	Página
Descripción de las variables de sistema	15
Utilización de variables de sistema	17

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 con las definiciones y las convenciones sobre nomenclatura de la 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, `PLC_R` representa un nombre de estructura de variables de solo lectura usado 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 del ejemplo anterior es `PLC_GVL.PLC.R`. `PLC_GVL` es implícito al declarar una variable con **Accesibilidad**, pero también puede especificarse por completo. Las buenas prácticas de programación suelen dictar la utilización de nombres de variables completos en las declaraciones.

Ubicación de variables de sistema

Se definen dos tipos de variables de sistema que se pueden utilizar al programar el controlador:

- variables ubicadas
- variables no ubicadas

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.
- Se utilizan en programas de 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 Machine Expert considera a las direcciones mayores que esto fuera de rango y solo son accesibles mediante la convención `structure_name.component_name`.

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 áreas %MW dinámicas siguientes:
 - De %MW60200 a %MW61999 para variables de solo lectura
 - De %MW62200 a %MW63999 para variables de lectura/escritura
- Se utilizan en programas de 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 Machine Expert considera a las direcciones mayores que esto fuera de rango y solo son accesibles mediante la convención `structure_name.component_name`.

Utilización de variables de sistema

Introducción

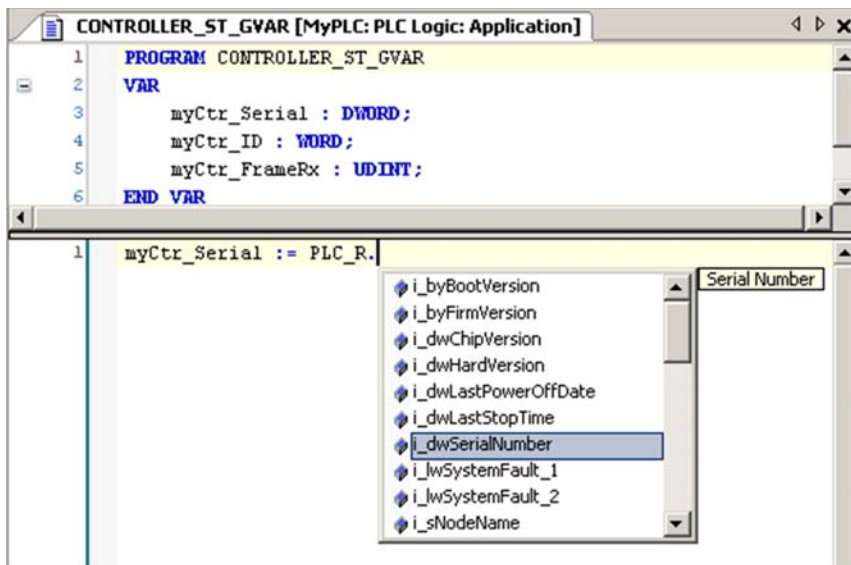
En este apartado se describen los pasos necesarios para programar y usar las variables de sistema en EcoStruxure Machine Expert.

Las variables de sistema son de ámbito global y pueden usarse en todas las unidades de organización del 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 de sistema del controlador.

Utilización de variables de sistema en un POU

EcoStruxure Machine Expert tiene una función de autocompletado. En un **POU**, empiece por especificar el nombre de estructura de la variable de sistema (`PLC_R`, `PLC_W...`) seguido de un punto. Aparecerán las variables de sistema en **Accesibilidad**. Puede seleccionar la variable que desea o especificar el nombre completo manualmente.



NOTA: En el ejemplo anterior, tras introducir el nombre de estructura `PLC_R.`, EcoStruxure Machine Expert ofrece un menú desplegable de nombres/variables de componentes posibles.

Ejemplo

En el ejemplo siguiente 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;
```

Sección 1.2

Estructuras 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.

Contenido de esta sección

Esta sección contiene los siguientes apartados:

Apartado	Página
PLC_R: Variables de sistema de sólo lectura del controlador	20
PLC_W: Variables de sistema de lectura/escritura del controlador	24

PLC_R: Variables de sistema de sólo lectura del controlador

Estructura de variables

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

Dirección Modbus ⁽¹⁾	Nombre de variable	Tipo	Comentario
60000	i_wVendorID	WORD	ID de proveedor del controlador. 101A hex = Schneider Electric
60001	i_wProductID	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_dwSerialNumber	DWORD	Número de serie del controlador
60004	i_byFirmVersion	ARRAY[0..3] OF BYTE	Controlador Firmware Versión [aa.bb.cc.dd]: <ul style="list-style-type: none"> ● i_byFirmVersion[0]= aa ● ... ● i_byFirmVersion[3]= dd
60006	i_byBootVersion	ARRAY[0..3] OF BYTE	Versión de inicio del controlador [aa.bb.cc.dd]: <ul style="list-style-type: none"> ● i_byBootVersion[0]= aa ● ... ● i_byBootVersion[3]= dd
60008	i_dwHardVersion	DWORD	Versión de hardware del controlador.
60010	i_dwChipVersion	DWORD	Versión del coprocesador del controlador.
60012	i_wStatus	PLC_R_STATUS <i>(véase página 72)</i>	Estado del controlador.
60013	i_wBootProjectStatus	PLC_R_BOOT_PROJECT_STATUS <i>(véase página 69)</i>	Devuelve información sobre la aplicación de arranque almacenada en la memoria FLASH.
60014	i_wLastStopCause	PLC_R_STOP_CAUSE <i>(véase página 73)</i>	Causa de la última transición desde el estado RUN a otro estado.
60015	i_wLastApplicationError	PLC_R_APPLICATION_ERROR <i>(véase página 67)</i>	Causa de la excepción del último controlador.

Dirección Modbus (1)	Nombre de variable	Tipo	Comentario
60016	i_lwSystemFault_1	LWORD	<p>El campo de bit FFFF FFFF FFFF FFFF hex no indica ningún error detectado.</p> <p>Un bit en nivel bajo significa que se ha detectado un error:</p> <ul style="list-style-type: none"> ● bit 0 = Reservado ● 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 = Reservado ● bit 6 = Error detectado de CAN 1 ● bit 7 = Reservado ● bit 8 = Reservado ● bit 9 = Error detectado de TM4 ● bit 10 = Error detectado de tarjeta SD ● bit 11 = Error detectado de servidor de seguridad ● bit 12 = Se ha detectado un error del servidor DHCP ● bit 13 = Se ha detectado un error del servidor OPC UA
60024	i_wIOStatus1	PLC_R_IO_STATUS <i>(véase página 70)</i>	Reservado
60025	i_wIOStatus2	PLC_R_IO_STATUS <i>(véase página 70)</i>	Estado de E/S de TM3.
60026	i_wClockBatterystatus	WORD	<p>Estado de la batería del RTC:</p> <ul style="list-style-type: none"> ● 0 = Se requiere cambio de batería ● 100 = Batería completamente cargada <p>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%.</p>
60028	i_dwAppliSignature1	DWORD	<p>Primera DWORD de la firma de 4 DWORD (16 bytes en total).</p> <p>El software genera la firma de la aplicación durante la compilación.</p>

Dirección Modbus ⁽¹⁾	Nombre de variable	Tipo	Comentario
60030	i_dwAppliSignature2	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_dwAppliSignature3	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_dwAppliSignature4	DWORD	Cuarta DWORD de la firma de 4 DWORD (16 bytes en total). El software genera la firma de la aplicación durante la compilación.
⁽¹⁾ No accesible a través de la aplicación.			

n/a	i_sVendorName	STRING (31)	Nombre del fabricante: "Schneider Electric".
n/a	i_sProductRef	STRING (31)	Referencia del controlador.
n/a	i_sNodeName	STRING (99)	Nombre de nodo en la red de EcoStruxure Machine Expert.
n/a	i_dwLastStopTime	DWORD	La hora de la última detención detectada en segundos, comenzando por el 1 de enero de 1970 a las 00:00 UTC.
n/a	i_dwLastPowerOffDate	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 <code>SysTimeRtcConvertUtcToDate</code> . Para obtener más información sobre la conversión a fecha y hora, consulte la Guía de la biblioteca <code>SysTime</code> (véase <i>EcoStruxure Machine Expert, Obtención y ajuste del reloj en tiempo real, Guía de la biblioteca SysTimeRtc y SysTimeCore</i>).
n/a	i_uiEventsCounter	UINT	Reservado
n/a	i_wTerminalPortStatus	PLC_R_TERMINAL_PORT_STATUS (véase página 75)	Estado del puerto de programación USB (Mini-B USB).
n/a	i_wSdCardStatus	PLC_R_SDCARD_STATUS (véase página 71)	Estado de la tarjeta SD.

n/a	i_wUsrFreeFileHdl	WORD	Número de controladores de archivos disponibles. Un controlador de archivos es el recurso asignado por el sistema cuando se abre un archivo.
n/a	i_udiUsrFsTotalBytes	UDINT	Tamaño total de la memoria del sistema de archivos del usuario (en bytes). Es el tamaño de la memoria flash para el directorio "/usr/".
n/a	i_udiUsrFsFreeBytes	UDINT	Tamaño de memoria libre del sistema de archivos (en bytes).
n/a	i_uiTM3BusState	PLC_R_TM3_BUS_STATE (véase página 76)	Estado de bus TM3. i_uiTM3BusState puede tener los siguientes valores: <ul style="list-style-type: none"> ● 1: TM3_CONF_ERROR Discrepancia en la configuración entre la configuración física y la configuración de EcoStruxure Machine Expert. ● 3: TM3_OK La configuración física coincide con la configuración de EcoStruxure Machine Expert. ● 4: TM3_POWER_SUPPLY_ERROR El bus TM3 no recibe alimentación (por ejemplo, cuando el Logic Controller recibe alimentación por USB).
n/a	i_ExpertIO_RunStop_Input	BYTE	Reservado
n/a	i_x10msClk	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_x100msClk	BOOL	Bit de TimeBase de 100 ms. Esta variable alterna On/Off con un período = 100 ms. El valor se alterna cuando el Logic Controller se encuentra en el estado Stop y Run.
n/a	i_x1sClk	BOOL	Bit de TimeBase de 1 s. Esta variable alterna On/Off con un período = 1 s. El valor se alterna cuando el Logic Controller se encuentra en el estado Stop y Run.

NOTA: n/a significa que no existe una asignación de dirección Modbus predefinida para esta variable de sistema.

PLC_W: Variables de sistema de lectura/escritura del controlador

Estructura de variables

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

%MW	Nombre de variable	Tipo	Comentario
n/a	q_wResetCounterEvent	WORD	La transición de 0 a 1 restablece el contador de eventos (PLC_R.i_uiEventsCounter). Para volver a restablecer el contador, es necesario escribir esta variable en 0 antes de que pueda llevarse a cabo otra transición de 0 a 1.
n/a	q_uiOpenPLCControl	UINT	Cuando el valor pasa de 0 a 6.699 se ejecuta el comando previamente escrito en el PLC_W.q_wPLCControl siguiente.
n/a	q_wPLCControl	PLC_W_COMMAND (véase página 77)	El comando RUN/STOP del controlador se ha ejecutado cuando el valor de la variable del sistema PLC_W.q_uiOpenPLCControl ha pasado de 0 a 6699.

NOTA: n/d significa que no existe ninguna asignación de %MW predefinida para esta variable de sistema.

Sección 1.3

Estructuras SERIAL_R y SERIAL_W

Descripción general

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

Contenido de esta sección

Esta sección contiene los siguientes apartados:

Apartado	Página
SERIAL_R[0...1]: Variables del sistema de solo lectura de la línea serie	26
SERIAL_W[0...1]: Variables del sistema de lectura/escritura de línea serie	27

SERIAL_R[0...1]: Variables del sistema de solo lectura de la línea serie

Introducción

SERIAL_R es una matriz de tipo 2 SERIAL_R_STRUCT. Cada elemento de la matriz devuelve las variables de sistema de diagnóstico para la línea serie correspondiente.

En el caso de M251 Logic Controller:

- Serial_R[0] hace referencia a la línea serie
- Serial_R[1] está reservado

Estructura de variables

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

%MW	Nombre de variable	Tipo	Comentario
Línea serie			
n/a	i_udiFramesTransmittedOK	UDINT	Número de tramas transmitidas correctamente.
n/a	i_udiFramesReceivedOK	UDINT	Número de tramas recibidas sin errores detectados.
n/a	i_udiRX_MessagesError	UDINT	Número de tramas recibidas con errores detectados (suma de comprobación, paridad).
Modbus Specific			
n/a	i_uiSlaveExceptionCount	UINT	Número de respuestas de excepción de Modbus devueltas por el controlador lógico.
n/a	i_udiSlaveMsgCount	UINT	Número de mensajes recibidos desde el maestro y dirigidos al controlador lógico.
n/a	i_uiSlaveNoRespCount	UINT	Número de peticiones de difusión de Modbus recibidas por el controlador lógico.
n/a	i_uiSlaveNakCount	UINT	No utilizado
n/a	i_uiSlaveBusyCount	UINT	No utilizado
n/a	i_uiCharOverrunCount	UINT	Número de desbordos de caracteres.
n/a significa que no existe una asignación de %MW predefinida para esta variable de sistema. Sin utilizar 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.			

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...1]: Variables del sistema de lectura/escritura de línea serie

Introducción

SERIAL_W es una matriz de tipo 2 SERIAL_W_STRUCT. Cada elemento de la matriz restablece las variables del sistema SERIAL_R para restablecer la línea serie correspondiente.

En el caso de M251 Logic Controller:

- Serial_W[0] hace referencia a la línea serie
- Serial_W[1] está reservado

Estructura de variables

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

%MW	Nombre de variable	Tipo	Comentario
n/a	q_wResetCounter	WORD	La transición de 0 a 1 restablece todos los contadores de SERIAL_R[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.

NOTA: n/a significa que no existe una asignación de %MW predefinida para esta variable de sistema.

Sección 1.4

Estructuras 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.

Contenido de esta sección

Esta sección contiene los siguientes apartados:

Apartado	Página
ETH_R: Variables de sistema de solo lectura del puerto Ethernet	29
ETH_W: Variables de sistema de lectura/escritura del puerto Ethernet	35

ETH_R: Variables de sistema de solo lectura del puerto Ethernet

Estructura de variables

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

%MW	Nombre de variable	Tipo	Comentario
60050	i_byIPAddress	ARRAY[0..3] OF BYTE	Dirección IP de la interfaz Ethernet o Ethernet_1 [aaa.bbb.ccc.ddd]: <ul style="list-style-type: none"> ● i_byIPAddress[0]= aaa ● ... ● i_byIPAddress[3]= ddd
60052	i_bySubNetMask	ARRAY[0..3] OF BYTE	Máscara de subred de la interfaz Ethernet o Ethernet_1 [aaa.bbb.ccc.ddd]: <ul style="list-style-type: none"> ● i_bySub-netMask[0]= aaa ● ... ● i_bySub-netMask[3]= ddd
60054	i_byGateway	ARRAY[0..3] OF BYTE	Dirección de pasarela de la interfaz Ethernet o Ethernet_1 [aaa.bbb.ccc.ddd]: <ul style="list-style-type: none"> ● i_byGateway[0]= aaa ● ... ● i_byGateway[3]= ddd
60056	i_byMACAddress	ARRAY[0..5] OF BYTE	Dirección MAC de la interfaz Ethernet o Ethernet_1 [aa.bb.cc.dd.ee.ff]: <ul style="list-style-type: none"> ● i_byMACAddress[0]= aa ● ... ● i_byMACAddress[5]= ff
60059	i_sDeviceName	STRING(15)	Nombre utilizado para obtener la dirección IP del servidor.
n/a	i_wIpMode	ETH_R_IP_MODE <i>(véase página 84)</i>	Método utilizado para obtener una dirección IP.
n/a	i_byFDRServerIPAddress	ARRAY[0..3] OF BYTE	La dirección IP [aaa.bbb.ccc.ddd] del servidor DHCP o BootP: <ul style="list-style-type: none"> ● i_byFDRServerIPAddress[0]= aaa ● ... ● i_byFDRServerIPAddress[3]= ddd Igual a 0.0.0.0 si se usa la IP almacenada o predeterminada.
n/a	i_udiOpenTcpConnections	UDINT	Número de conexiones TCP abiertas.
n/d significa que no existe ninguna asignación de %MW predefinida para esta variable de sistema.			

%MW	Nombre de variable	Tipo	Comentario
n/a	i_udiFramesTransmittedOK	UDINT	Número de tramas transmitidas correctamente. Reseteo al encender o con el comando de reseteo ETH_W.q_wResetCounter.
n/a	i_udiFramedReceivedOK	UDINT	Número de tramas recibidas correctamente. Reseteo al encender o con el comando de reseteo ETH_W.q_wResetCounter.
n/a	i_udiTransmitBufferErrors	UDINT	Número de tramas transmitidas con errores detectados. Reseteo al encender o con el comando de reseteo ETH_W.q_wResetCounter.
n/a	i_udiReceiveBufferErrors	UDINT	Número de tramas recibidas con errores detectados. Reseteo al encender o con el comando de reseteo ETH_W.q_wResetCounter.
n/a	i_wFrameSendingProtocol	ETH_R_FRAME_PROTOCOL <i>(véase página 83)</i>	Protocolo Ethernet configurado para el envío de tramas (IEEE 802.3 o Ethernet II).
n/a	i_wPortALinkStatus	ETH_R_PORT_LINK_STATUS <i>(véase página 87)</i>	Conexión del puerto Ethernet (0 = sin conexión, 1 = conexión conectada a otro dispositivo Ethernet).
n/a	i_wPortASpeed	ETH_R_PORT_SPEED <i>(véase página 88)</i>	Velocidad de red del puerto Ethernet (10Mb/s, 100Mb/s).
n/a	i_wPortADuplexStatus	ETH_R_PORT_DUPLEX_STATUS <i>(véase página 85)</i>	Estado dúplex del puerto Ethernet (0 = semidúplex o 1 = dúplex completo).
n/a	i_udiPortACollisions	UDINT	Número de tramas implicadas en una o más colisiones y enviadas a continuación correctamente. Reseteo al encender o con el comando de reseteo ETH_W.q_wResetCounter.
n/d significa que no existe ninguna asignación de %MW predefinida para esta variable de sistema.			

%MW	Nombre de variable	Tipo	Comentario
n/a	i_byIPAddress_If2	ARRAY[0..3] OF BYTE	Dirección IP de la interfaz Ethernet o Ethernet_2 [aaa.bbb.ccc.ddd]: <ul style="list-style-type: none"> ● i_byIPAddress[0]= aaa ● ... ● i_byIPAddress[3]= ddd
n/a	i_bySubNetMask_If2	ARRAY[0..3] OF BYTE	Máscara de subred de la interfaz Ethernet o Ethernet_2 [aaa.bbb.ccc.ddd]: <ul style="list-style-type: none"> ● i_bySub-netMask[0]= aaa ● ... ● i_bySub-netMask[3]= ddd
n/a	i_byGateway_If2	ARRAY[0..3] OF BYTE	Dirección de pasarela de la interfaz Ethernet o Ethernet_2 [aaa.bbb.ccc.ddd]: <ul style="list-style-type: none"> ● i_byGateway[0]= aaa ● ... ● i_byGateway[3]= ddd
n/a	i_byMACAddress_If2	ARRAY[0..5] OF BYTE	Dirección MAC de la interfaz Ethernet o Ethernet_2 [aa.bb.cc.dd.ee.ff]: <ul style="list-style-type: none"> ● i_byMACAddress[0]= aa ● ... ● i_byMACAddress[5]= ff
n/a	i_sDeviceName_If2	STRING(15)	Nombre utilizado para obtener la dirección IP del servidor.
n/a	i_wIpMode_If2	ETH_R_IP_MODE <i>(véase página 84)</i>	Método utilizado para obtener una dirección IP.
n/a	i_wPortALinkStatus_If2	ETH_R_PORT_LINK_STATUS <i>(véase página 87)</i>	Conexión del puerto Ethernet (0 = sin conexión, 1 = conexión conectada a otro dispositivo Ethernet).
n/a	i_wPortASpeed_If2	ETH_R_PORT_SPEED <i>(véase página 88)</i>	Velocidad de red del puerto Ethernet (10 Mb/s o 100 Mb/s).
n/a	i_wPortADuplexStatus_If2	ETH_R_PORT_DUPLEX_STATUS <i>(véase página 85)</i>	Estado dúplex del puerto Ethernet: <ul style="list-style-type: none"> ● 0: Semidúplex ● 1: Dúplex completo
n/a	i_wPortAIpStatus_If2	ETH_R_PORT_IP_STATUS <i>(véase página 86)</i>	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.			

%MW	Nombre de variable	Tipo	Comentario
Específico de Modbus TCP/IP			
n/a	i_udiModbusMessageTransmitted	UDINT	Número de mensajes Modbus transmitidos. Reseteo al encender o con el comando de reseteo ETH_W.q_wResetCounter.
n/a	i_udiModbusMessageReceived	UDINT	Número de mensajes Modbus recibidos. Reseteo al encender o con el comando de reseteo ETH_W.q_wResetCounter.
n/a	i_udiModbusErrorMessage	UDINT	Mensajes de error detectados de Modbus enviados y recibidos. Reseteo al encender o con el comando de reseteo ETH_W.q_wResetCounter.
n/d significa que no existe ninguna asignación de %MW predefinida para esta variable de sistema.			

%MW	Nombre de variable	Tipo	Comentario
EtherNet/IP Specific			
n/a	i_udiETHIP_IOMessagingTransmitted	UDINT	Tramas EtherNet/IP Clase 1 transmitidas. Reseteo al encender o con el comando de reseteo ETH_W.q_wResetCounter.
n/a	i_udiETHIP_IOMessagingReceived	UDINT	Tramas EtherNet/IP Clase 1 recibidas. Reseteo al encender o con el comando de reseteo ETH_W.q_wResetCounter.
n/a	i_udiUCMM_Request	UDINT	Mensajes recibidos de EtherNet/IP desconectada. Reseteo al encender o con el comando de reseteo ETH_W.q_wResetCounter.
n/a	i_udiUCMM_Error	UDINT	Mensajes recibidos no válidos de EtherNet/IP desconectada. Reseteo al encender o con el comando de reseteo ETH_W.q_wResetCounter.
n/a	i_udiClass3_Request	UDINT	Solicitudes de EtherNet/IP Clase 3 recibidas. Reseteo al encender o con el comando de reseteo ETH_W.q_wResetCounter.
n/d significa que no existe ninguna asignación de %MW predefinida para esta variable de sistema. Sin utilizar 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.			

%MW	Nombre de variable	Tipo	Comentario
n/a	i_udiClass3_Error	UDINT	Solicitudes de EtherNet/IP Clase 3 no válidas recibidas. Reseteo al encender o con el comando de reseteo ETH_W.q_wResetCounter.
n/a	i_uiAssemblyInstanceInput	UINT	Número de instancia de ensamblado de entrada. Consulte la guía de programación correspondiente del controlador para obtener más información.
n/a	i_uiAssemblyInstanceInputSize	UINT	Tamaño de la instancia de ensamblado de entrada. Consulte la guía de programación correspondiente del controlador para obtener más información.
n/a	i_uiAssemblyInstanceOutput	UINT	Número de instancia de ensamblado de salida. Consulte la guía de programación correspondiente del controlador para obtener más información.
n/a	i_uiAssemblyInstanceOutputSize	UINT	Tamaño de instancia de ensamblado de salida. Consulte la guía de programación correspondiente del controlador para obtener más información.
n/a	i_uiETHIP_ConnectionTimeouts	UINT	Número de timeouts de la conexión. Reseteo al encender o con el comando de reseteo ETH_W.q_wResetCounter.
n/a	i_ucEipRunIdle	ETH_R_R UN_IDLE <i>(véase página 89)</i>	Indicador de ejecución (valor=1) / inactividad (valor=0) para la conexión EtherNet/IP de Clase 1.
n/a	i_byMasterIpTimeouts	BYTE	Contador de eventos de timeout del maestro Ethernet Modbus TCP. Reseteo al encender o con el comando de reseteo ETH_W.q_wResetCounter.
n/a	i_byMasterIpLost	BYTE	Estado de enlace del maestro Ethernet Modbus TCP: 0 = conexión correcta, 1 = conexión perdida.
n/a	i_wPortAIPStatus	ETH_R_P ORT_IP_S TATUS <i>(véase página 86)</i>	Estado de la pila del puerto TCP/IP Ethernet.
<p>n/d significa que no existe ninguna asignación de %MW predefinida para esta variable de sistema. Sin utilizar 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>			

NOTA: 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 variables

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

%MW	Nombre de variable	Tipo	Comentario
n/a	q_wResetCounter	WORD	La transición de 0 a 1 restablece todos los contadores de ETH_R. Para volver a restablecer, es necesario escribir esta variable en 0 antes de que pueda llevarse a cabo otra transición de 0 a 1.

NOTA: n/d significa que no existe ninguna asignación de %MW predefinida para esta variable de sistema.

Sección 1.5

Estructura de TM3_MODULE_R

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

Introducción

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

En el caso de Modicon M251 Logic Controller:

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

Estructura de variables

En la tabla siguiente se describen los parámetros de la variable de sistema

TM3_MODULE_R[0...13]:

%MW	Nombre de variable	Tipo	Comentario
n/d	i_wProductID	WORD	ID del módulo de ampliación TM3.
n/a	i_wModuleState	TM3_MODULE_STATE <i>(véase página 93)</i>	Describe el estado del módulo de TM3.

NOTA: n/a significa que no existe una asignación de %MW predefinida para esta variable de sistema.

Sección 1.6

TM3_BUS_W Structure

TM3_BUS_W: Variables de sistema del bus TM3

Estructura de variables

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

Nombre de variable	Tipo	Comentario
q_wIOBusErrPassiv	TM3_BUS_W_IOBUSERRMOD	Cuando está establecido en ERR_ACTIVE (valor predeterminado), los errores de bus detectados en los módulos de expansión TM3 detienen todos los intercambios de E/S. Cuando está establecido en ERR_PASSIVE, se utiliza la gestión de errores de E/S pasiva: el controlador intenta seguir intercambiando buses de datos.
q_wIOBusRestart	TM3_BUS_W_IOBUSINIT	Cuando está establecido en 1, reinicia el bus de ampliación de E/S. Esto sólo es necesario cuando q_wIOBusErrPassiv está establecido en ERR_ACTIVE y como mínimo un bit de TM3_MODULE_R[i].i_wModuleState está establecido en TM3_BUS_ERROR.

Para obtener más información, consulte Descripción general de la configuración de E/S (véase *Modicon M251 Logic Controller, Guía de programación*).

Sección 1.7

Estructura PROFIBUS_R

PROFIBUS_R: Variables de sistema de solo lectura PROFIBUS

Estructura de variables

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/d	i_wPNOIdentifier	WORD	Código de identificación del esclavo.
n/a	i_wBusAdr	UINT	Dirección del esclavo PROFIBUS.
n/a	i_CommState	UDINT	Valor que representa el estado del módulo PROFIBUS: <ul style="list-style-type: none"> ● 0x00: Desconocido ● 0x01: No configurado ● 0x02: Detenido ● 0x03: Inactivo ● 0x04: En funcionamiento
n/a	i_CommError	UDINT	Código de error de comunicación.
n/a	i_ErrorCount	UDINT	Contador de error de comunicación.

NOTA: n/a significa que no existe ninguna asignación de %MW predefinida para esta variable de sistema.

Capítulo 2

Funciones de sistema M251,

Descripción general

En este capítulo se describen las funciones de sistema incluidas en la biblioteca PLCSystem de M251.

Contenido de este capítulo

Este capítulo contiene las siguientes secciones:

Sección	Apartado	Página
2.1	Funciones de lectura M251	40
2.2	Funciones de escritura de M251	46
2.3	Funciones de usuario de M251	48
2.4	Funciones de espacio en disco de M251	55
2.5	Funciones de lectura TM3	59

Sección 2.1

Funciones de lectura M251

Descripción general

En este apartado se describen las funciones de lectura incluidas en la biblioteca PLCSystem de M251.

Contenido de esta sección

Esta sección contiene los siguientes apartados:

Apartado	Página
GetRtc: Obtener reloj de tiempo real	41
IsFirstMastColdCycle: indica si este ciclo es el primer ciclo MAST del arranque en frío	42
IsFirstMastCycle: indica si este ciclo es el primer ciclo MAST	43
IsFirstMastWarmCycle: indica si este ciclo es el primer ciclo MAST del arranque en caliente	45

GetRtc: Obtener reloj de 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 (véase página 99)*.

Descripción de variables de E/S

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

Salida	Tipo	Comentario
GetRtc	DINT	RTC en segundos en formato UNIX.

Ejemplo

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

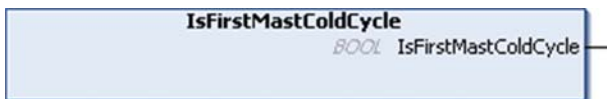
```
VAR
  MyRTC : DINT := 0;
END_VAR
MyRTC := GetRtc();
```

IsFirstMastColdCycle: indica 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 reset 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* (véase página 99).

Descripción de variables de E/S

En esta tabla se describe la variable de salida:

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

Ejemplo

Consulte la función IsFirstMastCycle (véase página 43).

IsFirstMastCycle: indica 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* (véase página 99).

Descripción de variables de E/S

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

Ejemplo

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

Use este ejemplo en la tarea MAST. De lo contrario, se puede ejecutar varias veces o posiblemente nunca (una tarea adicional puede llamarse varias veces 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
(*Este es el primer ciclo MAST después de un arranque en caliente: todas
las variables se establecen en sus valores de inicialización excepto las
variables Retain*)

```

```
(*=> inicia las variables necesarias para que la aplicación se ejecute
como se esperaba en este caso*)
END_IF;
IF (MyIsFirstMastColdCycle) THEN
(*Este es el primer ciclo MAST después de un arranque en frío: todas las
variables se establecen en sus valores de inicialización, incluso las
variables Retain*)
(*=> inicia las variables necesarias para que la aplicación se ejecute
como se esperaba en este caso*)
END_IF;
IF (MyIsFirstMastCycle) THEN
(*Este es el primer ciclo MAST después de un arranque, es decir, después
de un arranque en caliente o en frío así, como los comandos STOP/RUN*)
(*=> inicia las variables necesarias para que la aplicación se ejecute
como se esperaba en este caso*)
END_IF;
```

IsFirstMastWarmCycle: indica 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* (véase página 99).

Descripción de variables de E/S

En la tabla se describe la variable de salida:

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

Ejemplo

Consulte la función IsFirstMastCycle (véase página 43).

Sección 2.2

Funciones de escritura de M251

SetRTCDrift: Definir 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 debe llamarse sólo una vez. Cada nueva llamada sustituye el valor de compensación por el nuevo. El valor se conserva en el hardware del controlador mientras el RTC se alimenta mediante la alimentación principal o la batería. Si se eliminan la batería y la alimentación principal, 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 (véase página 99)*.

Descripción de variables de E/S

En esta tabla se describen los parámetros de entrada:

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

NOTA: Los parámetros Día, Hora y Minuto 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
<code>SetRTCDrift</code>	<code>RTCSETDRIFT_ERROR</code> <i>(véase página 95)</i>	Devuelve <code>RTC_OK</code> (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 en 4 segundos a la semana (18 segundos al mes).

```

VAR
  MyRTCDrift : SINT (-36...+73) := 0;
  MyDay : DAY_OF_WEEK;
  MyHour : HOUR;
  MyMinute : MINUTE;
END_VAR

IF IsFirstMastCycle() THEN
  MyRTCDrift := 4;
  MyDay := 0;
  MyHour := 0;
  MyMinute := 0;
  SetRTCDrift(MyRTCDrift, MyDay, MyHour, MyMinute);
END_IF

```

Sección 2.3

Funciones de usuario de M251

Descripción general

En esta sección se describen las funciones `FB_Control_Clone`, `DataFileCopy` y `ExecuteScript` incluidas en la biblioteca M251 PLCSystem.

Contenido de esta sección

Esta sección contiene los siguientes apartados:

Apartado	Página
<code>FB_ControlClone</code> : Clonar el controlador	49
<code>DataFileCopy</code> : Comandos para copiar archivos	50
<code>ExecuteScript</code> : Ejecutar comandos de script	53

FB_ControlClone: Clonar el controlador

Descripción de bloques de funciones

Es posible clonar el controlador mediante la tarjeta SD o el **Asistente del controlador**. Si están habilitados los derechos de usuario, la función de clonación no se permitirá y el bloque de funciones habilitará la funcionalidad de clonación una vez al encender el siguiente controlador.

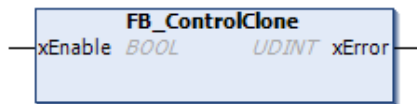
NOTA: Puede elegir qué derechos de usuario se incluyen en el clon de la página **Gestión de clonado** del Servidor web (*véase Modicon M251 Logic Controller, Guía de programación*).

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
xEnable = 1	Clonado permitido	Clonado también permitido
xEnable = 0	Clonado no permitido	Clonado no permitido

FB_ControlClone también afecta a la función **Leer del controlador** con el **Asistente del controlador**.

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* (*véase página 99*).

Descripción de variables de E/S

En la siguiente tabla se describen las variables de entrada:

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

En la tabla siguiente se describen las variables de salida:

Salida	Tipo	Comentario
xError	BOOL	TRUE indica que se ha detectado un error y que el bloque de funciones ha cancelado la acción.

DataFileCopy: Comandos para copiar archivos

Descripción de bloques de funciones

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

El bloque de funciones `DataFileCopy` puede:

- leer datos de un archivo formateado, o
- Copiar los datos de la memoria a un archivo formateado. Para obtener más información, consulte Flash Memory Organization (véase *Modicon M251 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* (véase [página 99](#)).

Descripción de variables de E/S

En esta tabla se describen las variables de entrada:

Entrada	Tipo	Comentario
xExecute	BOOL	En el flanco ascendente, inicia la ejecución del bloque de funciones. En el flanco descendente, restablece las salidas del bloque de funciones cuando finaliza cualquier ejecución en curso. 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.
sFileName	STRING	Nombre de archivo sin extensión (la extensión <i>.DTA</i> se incluye automáticamente). Utilice solamente caracteres alfanuméricos (mayúsculas, minúsculas o números).
xRead	BOOL	TRUE: copiar datos del archivo identificado por sFileName a la memoria interna del controlador. FALSE: copiar datos de la memoria interna del controlador al archivo identificado por sFileName.
xSecure	BOOL	TRUE: La dirección MAC se almacena siempre en el archivo. Sólo un controlador con la misma dirección MAC podrá leer el archivo. FALSE: Otro controlador con el mismo tipo de memoria puede leer el archivo.
iLocation	INT	0: la ubicación del archivo en el sistema interno de archivos es <i>/usr/DTA</i> . 1: la ubicación del archivo en el sistema externo de archivos (tarjeta SD) es <i>/usr/DTA</i> . NOTA: Si el archivo no existe todavía en el directorio, se creará.
uiSize	UINT	Indica el tamaño en bytes. El máximo son 65.534 bytes. Utilice solamente direcciones de variables conforme a IEC 61131-3 (variables, matrices, estructuras), por ejemplo: <code>Variable : int;</code> <code>uiSize := SIZEOF (Variable);</code>
dwAdd	DWORD	Indica la dirección de la memoria desde la que leerá o en la que escribirá la función. Utilice solamente direcciones de variables conforme a IEC 61131-3 (variables, matrices, estructuras), por ejemplo: <code>Variable : int;</code> <code>dwAdd := ADR (Variable);</code>

ADVERTENCIA

FUNCIONAMIENTO IMPREVISTO DEL EQUIPO

Compruebe que el tamaño de la ubicación de la memoria es el adecuado y que el tipo de archivo es el correcto antes de copiar el archivo a la memoria.

El incumplimiento de estas instrucciones puede causar la muerte, lesiones serias o daño al equipo.

En esta tabla se describen las variables de salida:

Salida	Tipo	Comentario
xDone	BOOL	TRUE indica que la acción se ha llevado a cabo correctamente.
xBusy	BOOL	TRUE indica que el bloque de funciones se está ejecutando.
xError	BOOL	TRUE indica que se ha detectado un error y que el bloque de funciones ha anulado la acción.
eError	DataFileCopyError (véase página 79)	Indica el tipo de error de copia de archivo de datos que se ha detectado.

NOTA: Si escribe en la variable de memoria dentro del área de escritura del archivo, se producirá un error de integridad de CRC.

Ejemplo

En este ejemplo se describe cómo copiar comandos de archivos:

```

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 de bloques de funciones

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

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

Para obtener información acerca del formato de archivo de script, consulte Archivos de script para tarjetas SD.

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* (véase [página 99](#)).

Descripción de variables de E/S

En esta tabla se describen las variables de entrada:

Entrada	Tipo	Comentario
xExecute	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>
sCmd	STRING	<p>Sintaxis de comando de la secuencia de comandos de la tarjeta SD. No se permiten ejecuciones de comando simultáneas: si se ejecuta un comando desde otro bloque de funciones o desde una secuencia de comandos de la tarjeta SD, el bloque de funciones pone el comando a la cola y no lo ejecuta inmediatamente.</p> <p>NOTA: Se considera que una secuencia de comandos de tarjeta SD ejecutada 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
xDone	BOOL	TRUE indica que la acción se ha completado correctamente.
xBusy	BOOL	TRUE indica que se está ejecutando el bloque de funciones.
xError	BOOL	TRUE indica que se ha detectado un error; el bloque de funciones interrumpe la acción.
eError	ExecuteScriptError (véase página 81)	Indica el tipo de error detectado de la secuencia de comandos de ejecución.

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=> );

```

Sección 2.4

Funciones de espacio en disco de M251

Descripción general

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

Contenido de esta sección

Esta sección contiene los siguientes apartados:

Apartado	Página
FC_GetFreeDiskSpace: Obtiene el espacio de memoria libre	56
FC_GetLabel: Obtiene la etiqueta de la memoria	57
FC_GetTotalDiskSpace: Obtiene el tamaño de la memoria	58

FC_GetFreeDiskSpace: Obtiene el espacio de memoria libre

Descripción de la función

Esta función recupera el espacio libre en la memoria de un soporte de memoria (disco flash, disco RAM, tarjeta SD) en bytes. El nombre del soporte de memoria se transfiere:

- Disco flash = "ide0:"
- Disco RAM = "ram0:"
- 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 (véase página 99)*.

Descripción de variables de E/S

En esta tabla se describen las variables de entrada:

Entrada	Tipo de datos	Descripción
i_sVolumeName	STRING[80]	Nombre del dispositivo a cuyo espacio libre en la memoria se debe acceder
iq_uliFreeDiskSpace	ULINT	Espacio libre en la memoria en bytes

En esta tabla se describen las variables de salida:

Salida	Tipo de datos	Descripción
FC_GetFreeDiskSpace	DINT	0: La cantidad de espacio libre en la memoria 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_sVolumeName)

FC_GetLabel: Obtiene la etiqueta de la 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.

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 (véase página 99)*.

Descripción de variables de E/S

En esta tabla se describen las variables de entrada:

Entrada	Tipo de datos	Descripción
i_sVolumeName	STRING[80]	Nombre del dispositivo a cuya etiqueta se debe acceder
iq_sLabel	STRING[11]	Etiqueta del dispositivo

En esta tabla se describen las variables de salida:

Salida	Tipo de datos	Descripción
FC_GetLabel	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 recupera el tamaño de un soporte de memoria (disco flash, disco RAM, tarjeta SD) en bytes.

El nombre del soporte de memoria se transfiere:

- Disco flash = "ide0:"
- Disco RAM = "ram0:"
- 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 (véase página 99)*.

Descripción de variables de E/S

En esta tabla se describen las variables de entrada:

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

En esta tabla se describen las variables de salida:

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

Sección 2.5

Funciones de lectura TM3

Descripción general

En este apartado se describen las funciones de lectura TM3 incluidas en la biblioteca M251 PLCSystem.

Contenido de esta sección

Esta sección contiene los siguientes apartados:

Apartado	Página
TM3_GetModuleBusStatus: Obtener estado del bus del módulo TM3	60
TM3_GetModuleFWVersion: Obtener versión del firmware del módulo TM3	61
TM3_GetModuleInternalStatus: Obtener estado interno del módulo TM3	62

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 (véase página 99)*.

Descripción de variables de E/S

En la tabla siguiente se describe la variable de entrada:

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

En la tabla siguiente se describe la variable de salida:

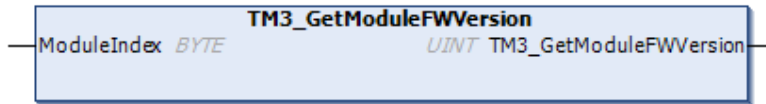
Salida	Tipo	Comentario
TM3_GetModuleBusStatus	TM3_ERR_CODE <i>(véase página 91)</i>	Devuelve TM3_OK (00 hex) si el comando es correcto. En caso contrario, devuelve el código ID del error detectado.

TM3_GetModuleFWVersion: Obtener versión del 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 (véase página 99)*.

Descripción de variables de E/S

En la siguiente tabla se describen las variables de entrada:

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

En la tabla siguiente se describe la variable de salida:

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

TM3_GetModuleInternalStatus: Obtener estado interno del módulo TM3

Descripción de la función

Esta función llena `pStatusBuffer` con la tabla de estado del módulo `ModuleIndex`.

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 (véase página 99)*.

Descripción de variables de E/S

⚠ ADVERTENCIA
FUNCIONAMIENTO IMPREVISTO DEL EQUIPO
Asegúrese de que el <code>pStatusBuffer</code> esté asignado.
El incumplimiento de estas instrucciones puede causar la muerte, lesiones serias o daño al equipo.

En la siguiente tabla se describen las variables de entrada:

Entrada	Tipo	Comentario
<code>ModuleIndex</code>	BYTE	Índice del módulo (0 para la primera expansión, 1 para la segunda, etc.).
<code>StatusOffset</code>	BYTE	Offset del primer estado que se debe leer en la tabla de estado.
<code>StatusSize</code>	BYTE	Número de bytes que se deben leer en la tabla de estado.
<code>pStatusBuffer</code>	POINTER TO BYTE	Búfer que contiene la tabla de estado de lectura.

En la tabla siguiente se describe la variable de salida:

Salida	Tipo	Comentario
TM3_GetModuleInternalStatus	TM3_ERR_CODE <i>(véase página 91)</i>	Devuelve TM3_OK (00 hex) si el comando es correcto; en caso contrario, devuelve el código ID del error.

Ejemplo

En el siguiente ejemplo se describe cómo obtener el estado interno del módulo:

```
VAR  
AMM3HT_Channel1_Input_Status: BYTE;  
END_VAR  
TM3_GetModuleInternalStatus(0, 1, 1, ADR(AMM3HT_Channel1_Input_Status));
```

Capítulo 3

Tipos de datos de la biblioteca PLCSystem de M251

Descripción general

En este capítulo se describen los tipos de datos de la biblioteca PLCSystem de M251.

Hay dos tipos de datos disponibles:

- Los tipos de datos de variables de sistema son utilizados por las variables de sistema (*véase página 13*) de la biblioteca PLCSystem de M251 (PLC_R, PLC_W,...).
- Los tipos de datos de funciones de sistema son utilizados por las funciones de sistema (*véase página 39*) de lectura/escritura de la biblioteca PLCSystem de M251.

Contenido de este capítulo

Este capítulo contiene las siguientes secciones:

Sección	Apartado	Página
3.1	Tipos de datos de variables de sistema de PLC_RW	66
3.2	Tipos de datos de variables del sistema de DataFileCopy	78
3.3	Tipos de datos de variables del sistema de ExecScript	81
3.4	Tipos de datos de variables de sistema de ETH_RW	82
3.5	Tipos de datos de variables de sistema de TM3_MODULE_RW	90
3.6	Tipos de datos de funciones de sistema	95

Sección 3.1

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`.

Contenido de esta sección

Esta sección contiene los siguientes apartados:

Apartado	Página
PLC_R_APPLICATION_ERROR: Códigos de estado del error detectado de la aplicación	67
PLC_R_BOOT_PROJECT_STATUS: Códigos de estado del proyecto de inicio	69
PLC_R_IO_STATUS: Códigos de estado de E/S	70
PLC_R_SDCARD_STATUS: Códigos de estado de slots para tarjeta SD	71
PLC_R_STATUS: Códigos de estado del controlador	72
PLC_R_STOP_CAUSE: Códigos de causa de transición de estado RUN a otro estado	73
PLC_R_TERMINAL_PORT_STATUS: Códigos de estado de la conexión del puerto de programación	75
PLC_R_TM3_BUS_STATE: Códigos de estado del bus TM3	76
PLC_W_COMMAND: Códigos de comando de control	77

PLC_R_APPLICATION_ERROR: Códigos de estado del error detectado de la aplicación

Descripción del tipo enumerado

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

Enumerador	Valor	Comentario	Qué hacer
PLC_R_APP_ERR_UNKNOWN	FFFF hex	Error no definido detectado.	Póngase en contacto con su representante de servicio de Schneider Electric.
PLC_R_APP_ERR_NOEXCEPTION	0000 hex	No se ha detectado ningún error.	–
PLC_R_APP_ERR_WATCHDOG	0010 hex	Ha caducado el watchdog de la tarea.	Compruebe la aplicación. Se necesita reiniciar para entrar en modalidad de ejecución.
PLC_R_APP_ERR_HARDWAREWATCHDOG	0011 hex	Ha caducado el watchdog de sistema.	Si el problema se reproduce, verifique que no haya puertos de comunicación configurados y desconectados. Si el problema persiste, actualice el firmware. Si aún no se ha resuelto, póngase en contacto con su representante de servicio de Schneider Electric.
PLC_R_APP_ERR_IO_CONFIG_ERROR	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 EcoStruxure Machine Experta la última versión.
PLC_R_APP_ERR_UNRESOLVED_EXTREFS	0018 hex	Funciones no definidas detectadas.	Elimine las funciones no resueltas de la aplicación.

Enumerador	Valor	Comentario	Qué hacer
PLC_R_APP_ERR_IEC_TASK_CONFIG_ERROR	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: 1. Compilar → Limpiar todo 2. Exporte/Importe la aplicación. 3. Actualice EcoStruxure Machine Experta la última versión.
PLC_R_APP_ERR_ILLEGAL_INSTRUCTION	0050 hex	Instrucción no definida detectada.	Para resolver el problema, depure la aplicación.
PLC_R_APP_ERR_ACCESS_VIOLATION	0051 hex	Intento de acceso al área de memoria reservada.	Para resolver el problema, depure la aplicación.
PLC_R_APP_ERR_DIVIDE_BY_ZERO	0102 hex	Detectada división de entero por cero.	Para resolver el problema, depure la aplicación.
PLC_R_APP_ERR_PROCESSORLOAD_WATCHDOG	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.
PLC_R_APP_ERR_DIVIDE_REAL_BY_ZERO	0152 hex	Detectada división real por cero.	Para resolver el problema, depure la aplicación.
PLC_R_APP_ERR_EXPIO_EVENTS_COUNT_EXCEEDED	4E20 hex	Se han detectado demasiados eventos en E/S expertas.	Reduzca el número de tareas de evento.
PLC_R_APP_ERR_APPLICATION_VERSION_MISMATCH	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 EcoStruxure Machine Expert. Consulte Aplicaciones (<i>véase EcoStruxure Machine Expert, Guía de programación</i>).

PLC_R_BOOT_PROJECT_STATUS: Códigos de estado del proyecto de inicio

Descripción de los tipos enumerados

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

Enumerador	Valor	Comentario
PLC_R_NO_BOOT_PROJECT	0000 hex	El proyecto de inicio no existe en la memoria Flash.
PLC_R_BOOT_PROJECT_CREATION_IN_PROGRESS	0001 hex	Se va a crear el proyecto de inicio.
PLC_R_DIFFERENT_BOOT_PROJECT	0002 hex	El proyecto de inicio en la memoria Flash es distinto del cargado en la RAM.
PLC_R_VALID_BOOT_PROJECT	FFFF hex	El proyecto de inicio en la memoria Flash es el mismo que el cargado en la RAM.

PLC_R_IO_STATUS: Códigos de estado de E/S

Descripción de los tipos enumerados

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

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

PLC_R_SDCARD_STATUS: Códigos de estado de slots para tarjeta SD

Descripción del tipo enumerado

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

Enumerador	Valor	Comentario
NO_SDCARD	0000 hex	No se ha detectado ninguna tarjeta SD en el slot o el slot no está conectado.
SDCARD_READONLY	0001 hex	La tarjeta SD se encuentra en la modalidad de solo lectura.
SDCARD_READWRITE	0002 hex	La tarjeta SD se encuentra en la modalidad de lectura/escritura.
SDCARD_ERROR	0003 hex	Error detectado en la tarjeta SD. Se escriben más detalles sobre el error que se ha producido en el archivo FwLog.txt.

PLC_R_STATUS: Códigos de estado del controlador

Descripción del tipo enumerado

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

Enumerador	Valor	Comentario
PLC_R_EMPTY	0000 hex	El controlador no contiene ninguna aplicación.
PLC_R_STOPPED	0001 hex	El controlador se ha detenido.
PLC_R_RUNNING	0002 hex	El controlador está en ejecución.
PLC_R_HALT	0004 hex	El controlador está en estado HALT (consulte el diagrama de estado del controlador en la guía de programación (<i>véase Modicon M251 Logic Controller, Guía de programación</i>) del controlador).
PLC_R_BREAKPOINT	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 valores siguientes:

Enumerador	Valor	Comentario	Qué hacer
PLC_R_STOP_REASON_UNKNOWN	00 hex	El valor inicial o la causa de la detención no se puede determinar.	Póngase en contacto con su representante local de Schneider Electric.
PLC_R_STOP_REASON_HW_WATCHDOG	01 hex	Detenido tras un timeout de watchdog.	Póngase en contacto con su representante local de Schneider Electric.
PLC_R_STOP_REASON_RESET	02 hex	Detenido tras resetear.	Consulte las posibilidades de reseteo en Diagrama de estado del controlador.
PLC_R_STOP_REASON_EXCEPTION	03 hex	Detenido tras la excepción.	Compruebe su aplicación y corríjala si es necesario. Consulte Watchdogs del sistema y de tareas. Se necesita reiniciar para entrar en modalidad de ejecución.
PLC_R_STOP_REASON_USER	04 hex	Detenido tras una petición de usuario.	Consulte el comando Parada en Comandos de transiciones de estado (<i>véase Modicon M251 Logic Controller, Guía de programación</i>).
PLC_R_STOP_REASON_IECPROGRAM	05 hex	Detenido después de una petición de comando de programa (por ejemplo: comando de control con parámetro <code>PLC_W.q_wPLCControl:=PLC_W_COMMAND.PLC_W_STOP;</code>).	–
PLC_R_STOP_REASON_DELETE	06 hex	Detenido tras un comando de eliminación de aplicación.	Consulte la ficha Aplicaciones del Editor de dispositivos de controlador (<i>véase Modicon M251 Logic Controller, Guía de programación</i>).
PLC_R_STOP_REASON_DEBUGGING	07 hex	Detenido tras entrar en la modalidad de depuración.	–
PLC_R_STOP_FROM_NETWORK_REQUEST	0A hex	Detenido tras una petición de la red, el servidor web del controlador o el comando PLC_W.	–

Enumerador	Valor	Comentario	Qué hacer
PLC_R_STOP_FROM_INPUT	0B hex	Detención requerida por una entrada de controlador.	–
PLC_R_STOP_FROM_RUN_STOP_SWITCH	0C hex	Detención requerida por el interruptor del controlador.	–
PLC_R_STOP_REASON_RETAIN_MISMATCH	0D hex	Detenido tras una prueba de comprobación de contexto incorrecta durante el reinicio.	Algunas variables 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.
PLC_R_STOP_REASON_BOOT_APPLI_MISMATCH	0E hex	Detenido tras una comparación incorrecta entre la aplicación de inicio y la aplicación que se encontraba en la memoria antes de reiniciar.	Cree una aplicación de arranque válida.
PLC_R_STOP_REASON_POWERFAIL	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 (*véase Modicon M251 Logic Controller, Guía de programación*).

PLC_R_TERMINAL_PORT_STATUS: Códigos de estado de la conexión del puerto de programación

Descripción de los tipos enumerados

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

Enumerador	Valor	Comentario
TERMINAL_NOT_CONNECTED	00 hex	No hay ningún PC conectado al puerto de programación.
TERMINAL_CONNECTION_IN_PROGRESS	01 hex	La conexión está en curso.
TERMINAL_CONNECTED	02 hex	El PC está conectado al puerto de programación.
TERMINAL_ERROR	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 valores siguientes:

Enumerador	Valor	Comentario
TM3_CONF_ERROR	01 hex	Error detectado debido a una discrepancia en la configuración física y la configuración en EcoStruxure Machine Expert.
TM3_OK	03 hex	La configuración física y la configuración de EcoStruxure Machine Expert coinciden.
TM3_POWER_SUPPLY_ERROR	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 valores siguientes:

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

Sección 3.2

Tipos de datos de variables del sistema de DataFileCopy

Descripción general

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

Contenido de esta sección

Esta sección contiene los siguientes apartados:

Apartado	Página
DataFileCopyError: Códigos de errores detectados	79
DataFileCopyLocation: Códigos de ubicación	80

DataFileCopyError: Códigos de errores detectados

Descripción de los tipos enumerados

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

Enumerador	Valor	Descripción
<code>ERR_NO_ERR</code>	00 hex	No se ha detectado ningún error.
<code>ERR_FILE_NOT_FOUND</code>	01 hex	El archivo no existe.
<code>ERR_FILE_ACCESS_REFUSED</code>	02 hex	No se puede abrir el archivo.
<code>ERR_INCORRECT_SIZE</code>	03 hex	El tamaño de la petición es diferente al tamaño que se ha leído en el archivo.
<code>ERR_CRC_ERR</code>	04 hex	El CRC es incorrecto y se supone que el archivo está dañado.
<code>ERR_INCORRECT_MAC</code>	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 valores siguientes:

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

Sección 3.3

Tipos de datos de variables del sistema de ExecScript

ExecuteScriptError: Códigos de errores detectados

Descripción de los tipos enumerados

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

Enumerador	Valor	Descripción
<code>CMD_OK</code>	00 hex	No se ha detectado ningún error.
<code>ERR_CMD_UNKNOWN</code>	01 hex	No se reconoce el comando.
<code>ERR_SD_CARD_MISSING</code>	02 hex	La tarjeta SD no está presente.
<code>ERR_SEE_FWLOG</code>	03 hex	Se ha detectado un error durante la ejecución del comando, consulte <code>FwLog.txt</code> . Para obtener más información, consulte Tipo de archivo (véase <i>Modicon M251 Logic Controller, Guía de programación</i>).
<code>ERR_ONLY_ONE_COMMAND_ALLOWED</code>	04 hex	Se ha intentado ejecutar diversas secuencias de comandos simultáneamente.
<code>CMD_BEING_EXECUTED</code>	05 hex	Ya hay una secuencia de comandos en curso.

Sección 3.4

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`.

Contenido de esta sección

Esta sección contiene los siguientes apartados:

Apartado	Página
ETH_R_FRAME_PROTOCOL: Códigos de protocolo de transmisión de tramas	83
ETH_R_IP_MODE: Códigos de origen de la dirección IP	84
ETH_R_PORT_DUPLEX_STATUS: Códigos de modalidad de transmisión	85
ETH_R_PORT_IP_STATUS: Códigos de estado del puerto TCP/IP Ethernet	86
ETH_R_PORT_LINK_STATUS: Códigos de estado de enlaces de comunicación	87
ETH_R_PORT_SPEEDETH_R_PORT_SPEED: Códigos de velocidad de comunicación del puerto Ethernet	88
ETH_R_RUN_IDLE: Códigos de estado de ejecución e inactividad de Ethernet/IP	89

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 valores siguientes:

Enumerador	Valor	Comentario
<code>ETH_R_802_3</code>	00 hex	El protocolo que se usa en la transmisión de tramas es IEEE 802.3.
<code>ETH_R_ETHERNET_II</code>	01 hex	El protocolo usado 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 valores siguientes:

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

ETH_R_PORT_DUPLEX_STATUS: Códigos de modalidad de transmisión

Descripción de los tipos enumerados

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

Enumerador	Valor	Comentario
ETH_R_PORT_HALF_DUPLEX	00 hex	Se utiliza la modalidad de transmisión de semidúplex.
ETH_R_FULL_DUPLEX	01 hex	Se utiliza la modalidad de transmisión de dúplex completo.
ETH_R_PORT_NA_DUPLEX	03 hex	Se utiliza la modalidad de transmisión de dúplex.

ETH_R_PORT_IP_STATUS: Códigos de estado del puerto TCP/IP Ethernet

Descripción de los tipos enumerados

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

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

ETH_R_PORT_LINK_STATUS: Códigos de estado de enlaces de comunicación

Descripción de los tipos enumerados

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

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

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 valores siguientes:

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

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

Descripción de los tipos enumerados

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

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

Sección 3.5

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`.

Contenido de esta sección

Esta sección contiene los siguientes apartados:

Apartado	Página
TM3_ERR_CODE: Códigos de error detectados en el módulo de ampliación TM3	91
TM3_MODULE_R_ARRAY_TYPE: Tipo de matriz de lectura del módulo de ampliación TM3	92
TM3_MODULE_STATE: Códigos de estado del módulo de ampliación TM3	93
TM3_BUS_W_IOBUSERRMOD: Modalidad de error de bus TM3	94

TM3_ERR_CODE: Códigos de error detectados en el módulo de ampliación TM3

Descripción del tipo enumerado

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

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

TM3_MODULE_R_ARRAY_TYPE: Tipo de matriz de lectura del módulo de ampliación 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 ampliación TM3

Descripción del tipo enumerado

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

Enumerador	Valor	Comentario
TM3_EMPTY	00 hex	No hay módulo.
TM3_CONF_ERROR	01 hex	El módulo de ampliación física no coincide con el configurado en EcoStruxure Machine Expert.
TM3_BUS_ERROR	02 hex	Error de bus detectado en el último intercambio con el módulo.
TM3_OK	03 hex	El último intercambio del bus con este módulo se ha realizado correctamente.
TM3_MISSING_OPT_MOD	05 hex	El módulo opcional no está físicamente presente.

TM3_BUS_W_IOBUSERRMOD: Modalidad de error de bus TM3

Descripción del tipo enumerado

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

Enumerador	Valor	Comentario
IOBUS_ERR_ACTIVE	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 (<i>véase Modicon M251 Logic Controller, Guía de programación</i>).
IOBUS_ERR_PASSIVE	01 hex	Modalidad pasiva. Los intercambios de E/S continúan en el bus TM3 aunque se detecte un error.

Sección 3.6

Tipos de datos de funciones de sistema

RTCSETDRIFT_ERROR: Códigos de error de función `SetRTCDrift` detectados

Descripción de los tipos enumerados

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

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

Apéndices



Apéndice A

Representación de funciones y de bloques de funciones

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.

Contenido de este capítulo

Este capítulo contiene los siguientes apartados:

Apartado	Página
Diferencias entre una función y un bloque de funciones	100
Cómo utilizar una función o un bloque de funciones en lenguaje IL	101
Cómo utilizar una función o un bloque de funciones en lenguaje ST	105

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 un estado persistente desde una llamada hasta la otra.
- Se puede utilizar como un operando en otras expresiones.

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

Bloque de funciones

Bloque de funciones

- Es una POU (Unidad de organización de programa) que devuelve una o más salidas.
- Debe llamarse a través de una instancia (copia del bloque de funciones con nombre y variables dedicados).
- 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);

```

Cómo utilizar 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 utilizan como ejemplos para mostrar implementaciones.

Uso de una función en lenguaje IL

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

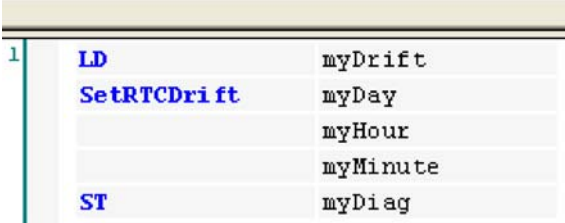
Paso	Acción
1	Abra o cree una nueva POU en el lenguaje de Lista de instrucciones (IL). NOTA: Aquí no se detalla el procedimiento para crear una POU. Para obtener más información, consulte <i>Adding and Calling POUs</i> (véase <i>EcoStruxure Machine Expert, Guía de programación</i>).
2	Cree las variables que necesite la función.
3	Si la función tiene 1 o más entradas, empiece a cargar la primera entrada utilizando la instrucción LD.
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 llamada de módulo en el menú contextual).
5	Si la función tiene más de una entrada y se utiliza Accesibilidad , se crea automáticamente el número necesario de líneas con ??? en los campos de la derecha. Sustituya los ??? 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 ST 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:

Función	Representación gráfica
sin parámetros de entrada: <code>IsFirstMastCycle</code>	
con parámetros de entrada: <code>SetRTCDrift</code>	

En lenguaje IL, el nombre de la función se utiliza 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 data-bbox="375 776 989 1096"> 1 PROGRAM MyProgram_IL 2 VAR 3 FirstCycle: BOOL; 4 END_VAR 5 ----- 1 IsFirstMastCycle ST FirstCycle </pre>

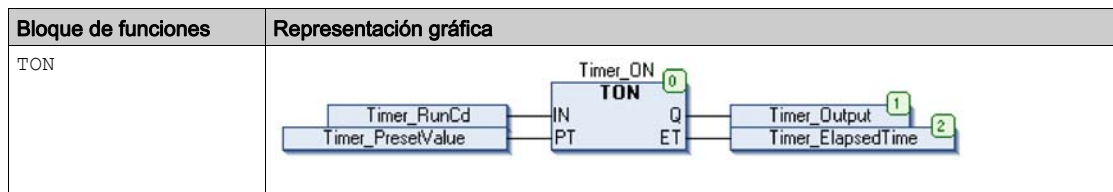
Función	Representación en el Editor IL de POU
Ejemplo en IL de una función con parámetros de entrada: SetRTCDrift	<pre> 1 PROGRAM MyProgram_IL 2 VAR 3 myDrift: SINT (-29..29) := 5; 4 myDay: DAY_OF_WEEK := SUNDAY; 5 myHour: HOUR := 12; 6 myMinute: MINUTE; 7 myDiag: RTCSETDRIFT_ERROR; 8 END_VAR </pre> 

Uso de un bloque de funciones en lenguaje IL

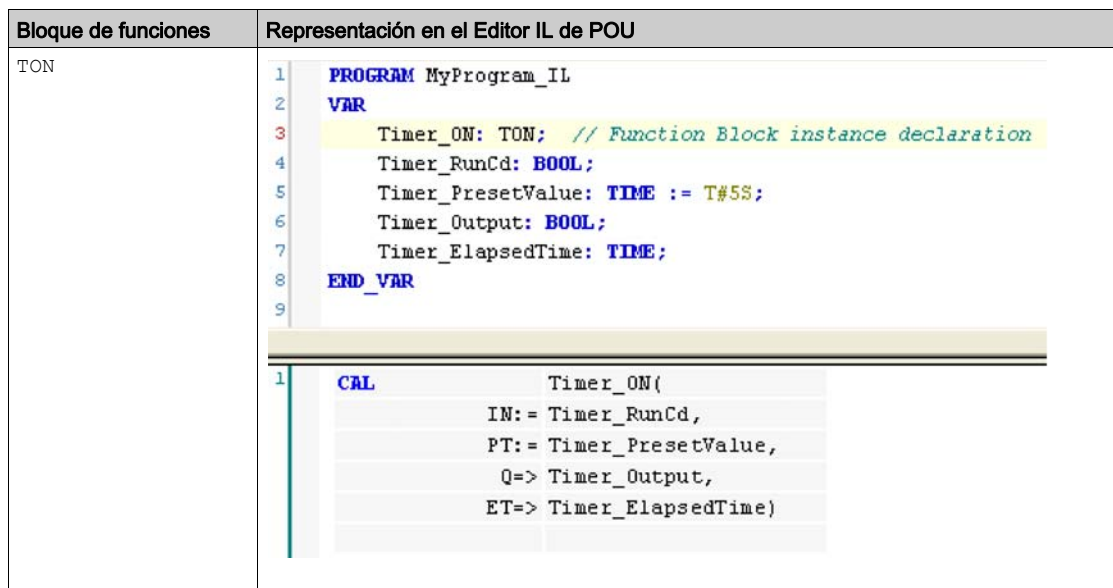
En este procedimiento se describe cómo insertar un bloque de funciones en lenguaje IL:

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 <i>Adding and Calling POU</i> (véase <i>EcoStruxure Machine Expert, Guía de programación</i>).
2	Cree las variables que necesita el bloque de funciones, incluido el nombre de instancia.
3	Se llama a los bloques de funciones utilizando una instrucción CAL : <ul style="list-style-type: none"> ● Utilice la opción Accesibilidad para seleccionar el bloque de funciones (botón derecho del ratón y seleccionar Insertar llamada de 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 utiliza directamente en la columna de operadores:



Cómo utilizar 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 utilizan como ejemplos para mostrar implementaciones.

Uso de 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 Adding and Calling POUs (<i>véase EcoStruxure Machine Expert, Guía de programación</i>).
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:

Función	Representación gráfica
SetRTCDrift	

El lenguaje ST de esta función es este:

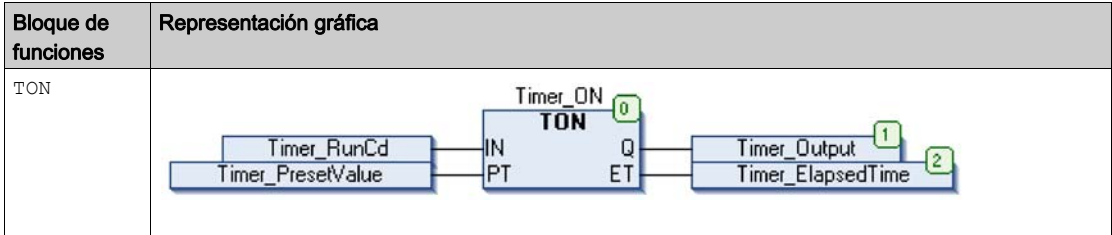
Función	Representación en el editor POU ST de
SetRTCDrift	<pre> PROGRAM MyProgram_ST VAR myDrift: SINT(-29..29) := 5; myDay: DAY_OF_WEEK := SUNDAY; myHour: HOUR := 12; myMinute: MINUTE; myRTCAdjust: RTCDRIFT_ERROR; END_VAR myRTCAdjust:= SetRTCDrift(myDrift, myDay, myHour, myMinute); </pre>

Uso de un bloque de funciones en lenguaje ST

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

Paso	Acción
1	<p>Abra o cree una POU nueva en el lenguaje de Texto estructurado (ST).</p> <p>NOTA: Aquí no se detalla el procedimiento para crear una POU. Para obtener más información sobre la adición, declaración y llamadas de POU, consulte la documentación (<i>véase EcoStruxure Machine Expert, Guía de programación</i>) relacionada.</p>
2	<p>Cree las variables de entrada y salida y la instancia requeridas para el bloque de funciones:</p> <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	<p>Utilice la sintaxis general en el Editor POU ST para el lenguaje ST de un bloque de funciones. La sintaxis general es:</p> <pre> FunctionBlock_InstanceName (Input1:=VarInput1, Input2:=VarInput2, ... Ouput1=>VarOutput1, Ouput2=>VarOutput2, ...); </pre>

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 de
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>



!

%MW

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

A

aplicación

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

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.

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.

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.

equipo

Una parte de la máquina que incluye subconjuntos tales como cintas transportadoras, plataformas giratorias, etc.

Ethernet

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

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.

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

(hexadecima)

I

ID

(identificador/identificación)

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.

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.

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).

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.

memoria Flash

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

P**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's facilitan la reutilización modular de programas de software, funciones y bloques de funciones. Una vez declaradas, cada una de las POU's 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

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

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.

S

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.

STOP

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

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**

Una unidad de memoria direccionada y modificada por un programa.

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*).

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.



Specials

B

bloques de funciones
 FB_ControlClone, 49

C

ciclo
 IsFirstMastColdCycle, 42
 IsFirstMastCycle, 43
 IsFirstMastWarmCycle, 45
comandos de script
 ExecuteScript, 53
comandos para copiar archivos
 DataFileCopy, 50

D

DataFileCopy
 copiar datos a o desde un archivo, 50
DataFileCopyError
 Tipos de datos, 79
DataFileCopyLocation
 tipos de datos, 80

E

estado de bus de módulo TM3
 TM3_GetModuleBusStatus, 60
estado interno de módulo TM3
 TM3_GetModuleInternalStatus, 62
ETH_R
 variable de sistema, 29
ETH_R_FRAME_PROTOCOL
 tipos de datos, 83
ETH_R_IP_MODE
 tipos de datos, 84

ETH_R_PORT_DUPLEX_STATUS
 Tipos de datos, 85
ETH_R_PORT_LINK_STATUS
 Tipos de datos, 87
ETH_R_PORT_SPEED
 tipos de datos, 88
ETH_W
 variable de sistema, 35
ExecuteScript
 ejecutar comandos de script, 53
ExecuteScriptError
 Tipos de datos, 81

F

FB_ControlClone
 bloque de funciones, 49
FC_GetFreeDiskSpace, 56
FC_GetLabel, 57
FC_GetTotalDiskSpace, 58
funciones
 cómo utilizar una función o un bloque de funciones en lenguaje IL, 101
 cómo utilizar una función o un bloque de funciones en lenguaje ST, 105
 diferencias entre una función y un bloque de funciones, 100

G

GetRtc
 obtener valor del reloj de tiempo real (RTC), 41

I

IsFirstMastColdCycle
 primer ciclo del arranque en frío, 42
IsFirstMastCycle
 primer ciclo MAST, 43

IsFirstMastWarmCycle
 primer ciclo del arranque en caliente, 45

M

M241 PLCSystem
 DataFileCopy, 50
 ExecuteScript, 53
 GetRtc, 41
 IsFirstMastColdCycle, 42
 IsFirstMastCycle, 43
 IsFirstMastWarmCycle, 45
 TM3_GetModuleBusStatus, 60, 62
 TM3_GetModuleFWVersion, 61

P

PLC_R
 Variable de sistema, 20
 PLC_R_APPLICATION_ERROR
 Tipos de datos, 67
 PLC_R_BOOT_PROJECT_STATUS
 Tipos de datos, 69
 PLC_R_IO_STATUS
 Tipos de datos, 70
 PLC_R_SDCARD_STATUS
 tipos de datos, 71
 PLC_R_STATUS
 tipos de datos, 72
 PLC_R_STOP_CAUSE
 tipos de datos, 73
 PLC_R_TERMINAL_PORT_STATUS
 Tipos de datos, 75
 PLC_R_TM3_BUS_STATE
 tipos de datos, 76
 PLC_W
 Variable de sistema, 24
 PLC_W_COMMAND
 tipos de datos, 77
 PROFIBUS_R
 variable de sistema, 38

R

reloj de tiempo real
 GetRtc, 41
 reloj de tiempo real
 SetRTCDrift, 46
 RTC
 GetRtc, 41
 SetRTCDrift, 46
 RTCSETDRIFT_ERROR
 Tipos de datos, 95

S

SERIAL_R
 Variable del sistema, 26
 SERIAL_W
 variable de sistema, 27
 SetRTCDrift
 aumentar o reducir la frecuencia del RTC,
 46

T

tipos de datos, 88
 Tipos de datos
 DataFileCopyError, 79
 tipos de datos
 DataFileCopyLocation, 80
 ETH_R_FRAME_PROTOCOL, 83
 ETH_R_IP_MODE, 84
 Tipos de datos
 ETH_R_PORT_DUPLEX_STATUS, 85
 ETH_R_PORT_IP_STATUS, 86
 ETH_R_PORT_LINK_STATUS, 87
 ETH_R_RUN_IDLE, 89
 ExecuteScriptError, 81
 PLC_R_APPLICATION_ERROR, 67
 PLC_R_BOOT_PROJECT_STATUS, 69
 PLC_R_IO_STATUS, 70
 tipos de datos
 PLC_R_SDCARD_STATUS, 71
 PLC_R_STATUS, 72
 PLC_R_STOP_CAUSE, 73
 Tipos de datos
 PLC_R_TERMINAL_PORT_STATUS, 75

- tipos de datos
 - PLC_R_TM3_BUS_STATE, 76
 - PLC_W_COMMAND, 77
- Tipos de datos
 - RTCSETDRIFT_ERROR, 95
 - TM3_BUS_W_IOBUSERRMOD, 94
- tipos de datos
 - TM3_ERR_CODE, 91
 - TM3_MODULE_R_ARRAY_TYPE, 92
 - TM3_MODULE_STATE, 93
- TM3_BUS_W
 - variable de sistema, 37
- TM3_BUS_W_IOBUSERRMOD
 - Tipos de datos, 94
- TM3_ERR_CODE
 - tipos de datos, 91
- TM3_GetModuleBusStatus
 - obtener estado del bus de un módulo TM3, 60
- TM3_GetModuleFWVersion
 - obtener versión del firmware de un módulo TM3, 61
- TM3_GetModuleInternalStatus
 - obtener estado interno de un módulo TM3, 62
- TM3_MODULE_R
 - variable de sistema, 36
- TM3_MODULE_R_ARRAY_TYPE
 - tipos de datos, 92
- TM3_MODULE_STATE
 - tipos de datos, 93
- Variable del sistema
 - SERIAL_R, 26
- variables de sistema
 - definición, 15
 - utilización, 17
- versión del firmware del módulo TM3
 - TM3_GetModuleFWVersion, 61

V

- variable de sistema
 - ETH_R, 29
 - ETH_W, 35
- Variable de sistema
 - PLC_R, 20
 - PLC_W, 24
- variable de sistema
 - PROFIBUS_R, 38
 - SERIAL_W, 27
 - TM3_BUS_W, 37
 - TM3_MODULE_R, 36

