

# Modicon M218 Logic Controller

系统功能和变量  
M218 PLCSystem 库指南

26/10/2010

---

本文档中提供的信息包含有关此处所涉及产品的性能的一般说明和 / 或技术特性。本文档并非用于（也不代替）确定这些产品针对特定用户应用的适用性或可靠性。任何此类用户或集成者都有责任就相关特定应用或使用方面对产品执行适当且完整的风险分析、评估和测试。Schneider Electric 或是其任何附属机构或子公司对于误用此处包含的信息而产生的后果概不负责。如果您有关于改进或更正此出版物的任何建议，或者从中发现错误，请通知我们。

未经 Schneider Electric 明确书面许可，不得以任何形式、通过任何电子或机械手段（包括影印）复制本文档的任何部分。

在安装和使用本产品时，必须遵守国家、地区和当地的所有相关的安全法规。出于安全方面的考虑和为了帮助确保符合归档的系统数据，只允许制造商对各个组件进行维修。

当设备用于具有技术安全要求的应用时，必须遵守有关的使用说明。

如果在我们的硬件产品上不正确地使用 Schneider Electric 软件或认可的软件，则可能导致人身伤害、损害或不正确的操作结果。

不遵守此信息可能导致人身伤害或设备损坏。

© 2010 Schneider Electric。保留所有权利。

---

# 目录

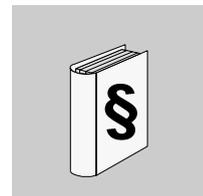


安全信息	5
关于本书	7
<b>章 1 M218 系统变量</b>	<b>9</b>
1.1 系统变量：定义和使用	10
了解系统变量	11
使用系统变量	13
1.2 PLC_R 和 PLC_W 结构	14
PLC_R：控制器只读系统变量	15
PLC_W：控制器读 / 写系统变量	17
1.3 SERIAL_R 和 SERIAL_W 结构	18
SERIAL_R[0..1]：串行线路只读系统变量	19
SERIAL_W[0..1]：串行线路读 / 写系统变量	20
1.4 ETH_R 和 ETH_W 结构	21
ETH_R：以太网端口只读系统变量	22
ETH_W：以太网端口读 / 写系统变量	24
<b>章 2 M218 系统功能</b>	<b>25</b>
2.1 M218 读取功能	26
GetBatteryLevel：返回电池的剩余电量	27
GetBootProjectStatus：返回引导项目状态	28
GetEventsNumber：返回检测到的外部事件数	29
GetFirmwareVersion：返回有关固件、引导和协处理器版本的信息	30
GetHardwareVersion：返回硬件版本	31
GetLastStopCause：返回上次停止的原因	32
GetLastStopTime：返回上次检测到停止的日期和时间	33
GetLocalIOStatus：返回嵌入式 I/O 状态	34
GetLocalAIOStatus：返回嵌入式模拟量 I/O 状态	36
GetPlcFault：返回控制器 I/O 上检测到的错误	37
GetRightBusStatus：返回扩展总线的状态	38
GetSerialNumber：返回控制器的序列号	42

	GetShortCutStatus: 返回嵌入式输出上的短路状态 . . . . .	43
	IsFirstMastColdCycle: 指示循环是否为第一个 MAST 冷启动循环 . . . . .	44
	IsFirstMastCycle: 指示循环是否为第一个 MAST 循环 . . . . .	45
	IsFirstMastWarmCycle: 指示循环是否为第一个 MAST 热启动循环 . . . . .	47
2.2	M218 写入功能 . . . . .	48
	InhibitBatLowLed: 禁用或重新启用电池 LED . . . . .	49
	ResetEventsNumber: 复位事件数 . . . . .	50
	SetRTCDrift: 每周调整实时时钟 . . . . .	51
<b>章 3</b>	<b>M218 PLCSystem 库数据类型 . . . . .</b>	<b>53</b>
3.1	PLC_R/W 系统变量数据类型 . . . . .	54
	PLC_R_STATUS: 控制器状态代码 . . . . .	55
	PLC_R_APPLICATION_ERROR: 检测到的应用程序错误状态代码 . . . . .	56
	PLC_R_BOOT_PROJECT_STATUS: 引导项目状态代码 . . . . .	57
	PLC_R_IO_STATUS: I/O 状态代码 . . . . .	58
	PLC_R_STOP_CAUSE: 从“运行”向其他状态转换的原因代码 . . . . .	59
	PLC_W_COMMAND: 控制命令代码 . . . . .	60
3.2	系统功能数据类型 . . . . .	61
	FIRMWARE_VERSION: GetFirmwareVersion 功能输出类型 . . . . .	62
	BOOT_PROJECT_STATUS: GetBootProjectStatus 功能输出代 码 . . . . .	63
	STOP_WHY: GetLastStopCause 功能输出代码 . . . . .	64
	LOCAL_IO_GET_STATUS: GetLocalIOStatus 功能参数代码 . . . . .	65
	LOCAL_IO_GEN_STATUS: GetLocalIOStatus 功能输出代码 . . . . .	66
	RIGHTBUS_GET_STATUS: GetRightBusStatus 功能参数代码 . . . . .	67
	DAY_OF_WEEK: SetRTCDrift 功能日期参数代码 . . . . .	68
	HOUR: SetRTCDrift 功能小时参数类型 . . . . .	69
	MINUTE: SetRTCDrift 功能分钟参数类型 . . . . .	70
3.3	ETH_R/W 系统变量数据类型 . . . . .	71
	ETH_R_IP_MODE: IP 地址源代码 . . . . .	72
	ETH_R_FRAME_PROTOCOL: 帧传输协议代码 . . . . .	73
	ETH_R_PORT_DUPLEX_STATUS: 传输模式代码 . . . . .	74
	ETH_R_PORT_LINK_STATUS: 通讯链路方向代码 . . . . .	75
	ETH_R_PORT_SPEED: 以太网端口的通讯速度代码 . . . . .	76
<b>附录</b>	<b>. . . . .</b>	<b>77</b>
<b>附录 A</b>	<b>功能和功能块表示形式 . . . . .</b>	<b>79</b>
	功能与功能块之间的差异 . . . . .	80
	如何通过 IL 语言使用功能或功能块 . . . . .	81
	如何通过 ST 语言使用功能或功能块 . . . . .	84
<b>术语表</b>	<b>. . . . .</b>	<b>87</b>
<b>索引</b>	<b>. . . . .</b>	<b>93</b>

---

## 安全信息



---

### 重要信息

#### 声明

在尝试安装、操作或维护设备之前，请仔细阅读下述说明并通过查看来熟悉设备。下述特别信息可能会在本文其他地方或设备上出现，提示用户潜在的危險，或者提醒注意有关阐明或简化某一过程的信息。



在“危險”或“警告”安全标签上添加此符号表示存在触电危險，如果不遵守使用说明，将导致人身伤害。



这是提醒注意安全的符号。提醒用户可能存在人身伤害的危險。请遵守所有带此符号的安全注意事项，以避免可能的人身伤害甚至死亡。

### 危險

“危險”表示极可能存在危險，如果不遵守说明，可导致严重的人身伤害甚至死亡。

### 警告

“警告”表示可能存在危險，如果不遵守说明，可导致严重的人身伤害甚至死亡，或设备损坏。

---

**⚠ 注意**

“注意”表示可能存在危险，如果不遵守说明，可导致严重的人身伤害或设备损坏。

**注意**

注意(无安全警告符号)，表示存在潜在的危险，如果忽视，可能导致设备损坏。

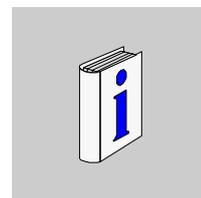
**请注意**

电气设备的安装、操作、维修和维护工作仅限于合格人员执行。对于使用本资料所引发的任何后果，Schneider Electric 概不负责。

专业人员是指掌握与电气设备的制造和操作相关的技能和知识的人员，他们经过安全培训能够发现和避免相关的危险。

---

## 关于本书



---

### 概览

#### 文档范围

本文档将向您介绍 Modicon M218 Logic Controller 中提供的系统功能和变量。M218 PLCSystem 库包含的功能和变量可用于获取信息并向控制器系统发送命令。

本文档描述 M218 PLCSystem 库的数据类型功能和变量。

需要了解以下基本知识：

- M218 的功能、结构和配置的基本信息
- 使用 FBD、LD、ST、IL 或 CFC 语言进行编程
- 系统变量（全局变量）

#### 有效性说明

本文档已随 SoMachine V2.0 的发布进行了更新。

#### 相关的文件

文件名称	参考编号
M218 Logic Controller 编程指南	EIO0000000827 (英语)； EIO0000000828 (简体中文)；

您可以从我们的网站下载这些技术出版物和其它技术信息，网址是：  
[www.schneider-electric.com](http://www.schneider-electric.com)。

## 关于产品的资讯

### 警告

#### 失去控制

- 任何控制方案的设计者都必须考虑到控制路径可能出现故障的情况，并为某些关键控制功能提供一种方法，使其在出现路径故障时，以及出现路径故障后恢复至安全状态。紧急停止和越程停止、断电和重启都属于关键控制功能。
- 对于关键控制功能，必须提供单独或冗余的控制路径。
- 系统控制路径可包括通讯链路。必须对暗含的无法预料的传输延迟或链接失效问题加以考虑。
- 遵守所有事故预防规定和当地的安全指南。<sup>1</sup>
- 为了保证正确运行，在投入使用前，必须对设备的每次执行情况分别进行全面测试。

**如果不遵守这些说明，将会导致死亡、严重伤害或设备损坏。**

<sup>1</sup> 有关详细信息，请参阅 NEMA ICS 1.1（最新版）中的“安全指导原则 - 固态控制器的应用、安装和维护”以及 NEMA ICS 7.1（最新版）中的“结构安全标准及可调速驱动系统的选择、安装与操作指南”或您特定地区的类似规定。

### 警告

#### 意外的设备操作

- 仅使用 Schneider Electric 认可的可与本设备配合使用的软件。
- 每次更改物理硬件配置后，请更新应用程序。

**如果不遵守这些说明，将会导致死亡、严重伤害或设备损坏。**

## 用户意见

欢迎对本书提出意见。您可以给我们发邮件，我们的邮件地址是 [techcomm@schneider-electric.com](mailto:techcomm@schneider-electric.com)。

---

# M218 系统变量



---

## 概述

本章内容:

- 提供系统变量的简介 (参见第 10 页)
- 描述 M218 PLCSystem 库中包含的系统变量 (参见第 15 页)

## 本章包含了哪些内容?

本章包含了以下部分:

节	主题	页
1.1	系统变量: 定义和使用	10
1.2	PLC_R 和 PLC_W 结构	14
1.3	SERIAL_R 和 SERIAL_W 结构	18
1.4	ETH_R 和 ETH_W 结构	21

# 1.1 系统变量：定义和使用

---

## 概述

本节定义系统变量以及如何在 Modicon M218 Logic Controller 中实现这些变量。

## 本节包含了哪些内容？

本节包含了以下主题：

主题	页
了解系统变量	11
使用系统变量	13

## 了解系统变量

### 简介

本节介绍如何在控制器上实现系统变量。这些变量具有以下属性：

- 使用系统变量可以访问一般系统信息、执行系统诊断以及通过命令进行简单操作。
- 系统变量是符合 IEC 61131 定义和命名约定的结构化变量。这些变量可使用 IEC 符号名称 `PLC_GVL` 进行访问。
- 其中一些 `PLC_GVL` 变量是只读变量（例如 `PLC_R`），另有一些变量是读 / 写变量（例如 `PLC_W`）。
- 系统变量会自动声明为全局变量。这些变量会对整个系统造成影响，必须谨慎处理，因为它们可以在任何任务中通过任何程序组织单元 (POU) 进行访问。

### 系统变量命名约定

系统变量的标识方法如下：

- 表示系统变量类别的结构名称（例如，`PLC_R` 表示用于控制器诊断的只读变量的结构名称）。
- 一组标识变量用途的组件名称（例如，`i_wVendorID` 表示控制器的供应商 ID）。

访问变量时，您可以键入变量的结构名称，然后再输入组件名称。

下面是一个实现系统变量的示例：

```
VAR myCtr_Serial :DWORD; myCtr_ID :DWORD; myCtr_FramesRx
:UDINT; END_VAR

myCtr_Serial := PLC_R.i_dwSerialNumber; myCtr_ID :=
PLC_R.i_wVendorID; myCtr_FramesRx :=
SERIAL_R[0].i_udiFramesReceivedOK;
```

**注意：**上面示例中系统变量的合格完整名称为 `PLC_GVL.PLC_R.i_wVendorID`。在使用**输入助手**声明变量时，`PLC_GVL` 是隐式的，但是也可以进行完整输入。优秀编程实践通常会在在声明中使用合格的完整名称。

### 系统变量位置

在对控制器进行编程时，可定义两种要使用的系统变量类型：

- 定位变量
- 非定位变量

定位变量：

- 在静态 `%MW` 区域中的位置不变：
  - `%MW60000` 到 `%MW60199`，用于只读系统变量
  - `%MW62000` 到 `%MW62199`，用于读 / 写系统变量

- 可在“运行”和“已停止”状态下，通过 Modbus 串行请求访问这些变量
- 根据前面介绍的 `structure_name.component_name` 惯例（介于 0 和 59999 之间的 %MW 地址可直接访问；更大的地址将被视为超出了 SoMachine 的范围，只能通过 `structure_name.component_name` 惯例访问）在 SoMachine 程序中使用。

非定位变量：

- 在 %MW 区域中没有实际位置
- 不能通过任何现场总线或网络请求访问这些变量，除非在重新定位表中找到它们，也只有这样才能在“运行”和“已停止”状态中访问它们。重新定位表使用以下动态 %MW 区域：
  - %MW60200 到 %MW61999，用于只读变量
  - %MW62200 到 %MW63999，用于读 / 写变量
- 根据前面介绍的 `structure_name.component_name` 惯例（介于 0 和 59999 之间的 %MW 地址可直接访问；更大的地址将被视为超出了 SoMachine 的范围，只能通过 `structure_name.component_name` 惯例访问）在 SoMachine 程序中使用。

## 使用系统变量

### 简介

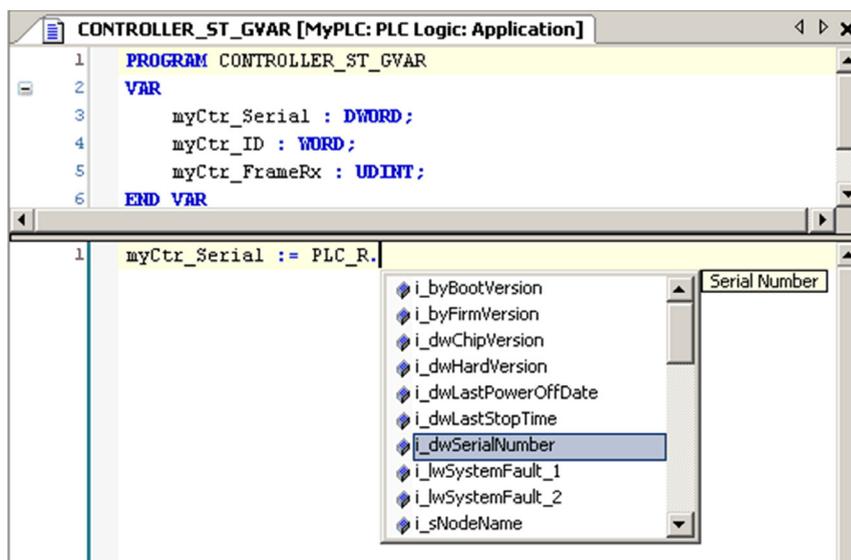
本主题介绍在 SoMachine 中编程和使用系统变量所需的步骤。

系统变量可以用在应用程序的所有程序组织单元 (POU) 中。

系统变量无需在 GVL 中声明。它们会自动从控制器系统库声明。

### 在 POU 中使用系统变量

在 **POU** 中，首先输入系统变量的结构名称（PLC\_R、PLC\_W 等），然后在其后添加一个句点。系统变量会在输入助手<sup>1</sup>中显示。您可以选择所需的变量或手动输入完整名称。



**注意：** SoMachine 具有自动填写功能。在上面的示例中，键入结构名称 PLC\_R. 后，SoMachine 会提供一个包含可能组件名称 / 变量的弹出菜单。

### 示例

下面的示例介绍某些系统变量的使用方法：

```
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;
```

## 1.2 PLC\_R 和 PLC\_W 结构

---

### 概述

本节列出并描述 PLC\_R 和 PLC\_W 结构中包含的不同**系统变量**。

### 本节包含了哪些内容？

本节包含了以下主题：

主题	页
PLC_R: 控制器只读系统变量	15
PLC_W: 控制器读 / 写系统变量	17

## PLC\_R: 控制器只读系统变量

### 变量结构

下表介绍 PLC\_R 系统变量 (PLC\_R\_STRUCT 类型) 的参数:

%MW	变量名称	类型	注释
60000	i_wVendorID	WORD	控制器供应商 ID。 101A (十六进制) = Schneider Electric
60001	i_wProductID	WORD	控制器参考 ID。 <b>注意:</b> 供应商 ID 和参考 ID 是“通讯设置”视图中显示的控制器的目标 ID 的组成部分 (目标 ID = 十六进制的 101A XXXX)。
60002	i_dwSerialNumber	DWORD	控制器序列号
60004	i_byFirmVersion[0..3]	ARRAY [0..3] OF BYTE	控制器固件版本 [aa.bb.cc.dd] : ● i_byFirmVersion[0]= aa ● ... ● i_byFirmVersion[3]= dd
60006	i_byBootVersion[0..3]	ARRAY [0..3] OF BYTE	控制器引导版本 [aa.bb.cc.dd] : ● i_byBootVersion[0]= aa ● ... ● i_byBootVersion[3]= dd
60008	i_dwHardVersion	DWORD	控制器硬件版本。
60010	i_dwChipVersion	DWORD	控制器协处理器版本。
60012	i_wStatus	PLC_R_STATUS (参见第 55 页)	控制器的状态。
60013	i_wBootProjectStatus	PLC_R_BOOT_PROJECT_STATUS (参见第 57 页)	返回有关闪存中存储的引导应用程序的信息。
60014	i_wLastStopCause	PLC_R_STOP_CAUSE (参见第 59 页)	上次从“运行”转换为其他状态的原因。
60015	i_wLastApplicationError	PLC_R_APPLICATION_ERROR (参见第 56 页)	上一次控制器例外的原因。
60016	i_lwSystemFault_1	LWORD	位域 FFFF FFFF FFFF FFFF (十六进制) 表示未检测到错误。 如果检测到 TM2 错误, 则位 1 为 FALSE。
60020	i_lwSystemFault_2	LWORD	未使用。
60024	i_wIOStatus1	PLC_R_IO_STATUS (参见第 58 页)	嵌入式 I/O 状态。

%MW	变量名称	类型	注释
60025	i_wIOStatus2	PLC_R_IO_STAT US (参见 第 58 页)	TM2 I/O 状态。
60026	i_wBatteryStatus	WORD	电池中剩余电量。 该系统变量可采用以下有效值： <ul style="list-style-type: none"> <li>● FFFF (十六进制) = 100% = 3.5 V</li> <li>● 7FFF (十六进制) = 50% = 2.75 V</li> <li>● 0000 (十六进制) = 0% = 2 V</li> </ul> 电池 LED 的闪烁阈值： <ul style="list-style-type: none"> <li>● 100% = 始终为“关”</li> <li>● 50% = 定期闪烁</li> <li>● 0% = 始终为“开”</li> </ul>
60028	i_dwAppliSignature1	DWORD	4 个 DWORD 签名 (总共 16 个字节) 的第 1 个 DWORD。 应用程序签名由软件在编译过程中生成。
60030	i_dwAppliSignature2	DWORD	4 个 DWORD 签名 (总共 16 个字节) 的第 2 个 DWORD。 应用程序签名由软件在编译过程中生成。
60032	i_dwAppliSignature3	DWORD	4 个 DWORD 签名 (总共 16 个字节) 的第 3 个 DWORD。 应用程序签名由软件在编译过程中生成。
60034	i_dwAppliSignature4	DWORD	4 个 DWORD 签名 (总共 16 个字节) 的第 4 个 DWORD。 应用程序签名由软件在编译过程中生成。

## PLC\_W: 控制器读 / 写系统变量

### 变量结构

下表介绍 PLC\_W 系统变量 (PLC\_W\_STRUCT 类型) 的参数:

%MW	变量名称	类型	注释
	q_uiOpenPLCControl	UINT	当值从 0 转到 6699 后, 将执行以前在下面的 PLC_W.q_wPLCControl 中写入的命令。
	q_wPLCControl	PLC_W_COMMAND (参见第 60 页)	当系统变量 PLC_R.q_uiOpenPLCControl 值从 0 转到 6699 后, 将执行控制器运行 / 停止命令。

---

## 1.3 SERIAL\_R 和 SERIAL\_W 结构

---

### 概述

本节列出并描述 SERIAL\_R 和 SERIAL\_W 结构中包含的不同系统变量。

### 本节包含了哪些内容？

本节包含了以下主题：

主题	页
SERIAL_R[0..1]: 串行线路只读系统变量	19
SERIAL_W[0..1]: 串行线路读 / 写系统变量	20

## SERIAL\_R[0..1]: 串行线路只读系统变量

### 简介

SERIAL\_R 是具有 2 个 SERIAL\_R\_STRUCT 类型的数组。该数组的每个元素都返回对应串行线路的诊断系统变量。

### 变量结构

下表描述 SERIAL\_R[0..1] 系统变量的参数：

%MW	变量名称	类型	注释
<b>串行线路</b>			
无	i_udiFramesTransmittedOK	UDINT	已成功传输的帧数。
无	i_udiFramesReceivedOK	UDINT	没有检测到任何错误的已接收帧数。
无	i_udiRX_MessagesError	UDINT	检测到错误（校验和、校验位）的已接收帧数。
<b>特定于 Modbus</b>			
无	i_uiSlaveExceptionCount	UINT	控制器返回的 Modbus 例外响应数。
无	i_udiSlaveMsgCount	UINT	从主站接收并发送到控制器的消息数。
无	i_uiSlaveNoRespCount	UINT	控制器接收的 Modbus 广播请求数。
无	i_uiSlaveNakCount	UINT	未使用
无	i_uiSlaveBusyCount	UINT	未使用
无	i_uiCharOverrunCount	UINT	溢出的字符数。

**注意：** 无表示此系统变量没有预定义的 %MW 映射。

#### 注意：

SERIAL\_R 计数器在以下情况时复位：

- 下载。
- 控制器复位。
- SERIAL\_W[x].q\_wResetCounter 命令。
- 通过 Modbus 请求功能代码 #8 复位命令。

## SERIAL\_W[0..1]: 串行线路读 / 写系统变量

### 简介

SERIAL\_W 是具有 2 个 SERIAL\_W\_STRUCT 类型的数组。该数组中各元素将强制相应串行线路的 SERIAL\_R 系统变量复位。

M218:

- Serial\_W[0] 引用串行线路 1
- Serial\_W[1] 引用串行线路 2

### 变量结构

下表介绍了 SERIAL\_W[0..1] 系统变量的参数:

%MW	变量名称	类型	注释
无	q_wResetCounter	WORD	从 0 转换为 1 后，将复位所有 SERIAL_R[0..1] 计数器。 若要再次复位这些计数器，需要先将此寄存器写为 0，之后才能再次进行从 0 到 1 的转换。

**注意：** 无表示没有用于此系统变量的预定义 %MW 映射。

---

## 1.4 ETH\_R 和 ETH\_W 结构

---

### 概述

本节列出并描述 ETH\_R 和 ETH\_W 结构中包含的不同系统变量。

### 本节包含了哪些内容？

本节包含了以下主题：

主题	页
ETH_R: 以太网端口只读系统变量	22
ETH_W: 以太网端口读 / 写系统变量	24

## ETH\_R: 以太网端口只读系统变量

### 变量结构

下表介绍 ETH\_R 系统变量（ETH\_R\_STRUCT 类型）的参数：

%MW	变量名称	类型	注释
60050	i_byIPAddress[0..3]	ARRAY [0..3] OF BYTE	IP 地址 [aaa.bbb.ccc.ddd] : ● i_byIPAddress[0]= aaa ● ... ● i_byIPAddress[3]= ddd
60052	i_bySubNetMask[0..3]	ARRAY [0..3] OF BYTE	子网掩码 [aaa.bbb.ccc.ddd] : ● i_bySub-netMask[0]= aaa ● ... ● i_bySub-netMask[3]= ddd
60054	i_byGateway[0..3]	ARRAY [0..3] OF BYTE	网关地址 [aaa.bbb.ccc.ddd] : ● i_byGateway[0]= aaa ● ... ● i_byGateway[3]= ddd
60056	i_byMACAddress[0..5]	ARRAY [0..5] OF BYTE	MAC 地址 [aa.bb.cc.dd.ee.ff] : ● i_byMACAddress[0]= aa ● ... ● i_byMACAddress[5]= ff
60059	i_sDeviceName	STRING (16)	用于获取服务器 IP 地址的名称。
无	i_wIpMode	ETH_R_IP_MODE (参见第 72 页)	用于获取 IP 地址的方法
无	i_byFDRServerIPAddress[0..3]	ARRAY [0..3] OF BYTE	DHCP 或 BootP 服务器的 IP 地址 [aaa.bbb.ccc.ddd] : ● i_byFDRServerIPAddress[0]= aaa ● ... ● i_byFDRServerIPAddress[3]= ddd 如果使用存储的 IP 或默认 IP, 则等于 0.0.0.0。
无	i_udiOpenTcpConnections	UDINT	打开的 TCP 连接数。
无	i_wFrameSendingProtocol	ETH_R_FRAME_PROTOCOL (参见第 73 页)	为帧发送配置的以太网协议 (IEEE 802.3 或 Ethernet II)。
无	i_udiFramesTransmittedOK	UDINT	已成功传输的帧数。在电源接通或使用复位命令 ETH_W.q_wResetCounter 后复位。

%MW	变量名称	类型	注释
无	i_udiFramedReceivedOK	UDINT	已成功接收的帧数。在电源接通或使用复位命令 ETH_W.q_wResetCounter 后复位。
无	i_udiTransmitBufferErrors	UDINT	已传输但检测到错误的帧的数量。在电源接通或使用复位命令 ETH_W.q_wResetCounter 后复位。
无	i_udiReceiveBufferErrors	UDINT	已接收但检测到错误的帧的数量。在电源接通或使用复位命令 ETH_W.q_wResetCounter 后复位。
无	i_wPortALinkStatus	ETH_R_PORT_LINK_STATUS (参见第 75 页)	以太网端口的链路方向 (0= 向下, 1= 向上)。
无	i_wPortASpeed	ETH_R_PORT_SPEED (参见第 76 页)	以太网端口网速 (10Mb/s 或 100Mb/s)。
无	i_wPortADuplexStatus	ETH_R_PORT_DUPLEX_STATUS (参见第 74 页)	以太网端口双工状态 (0= 半双工, 1= 全双工)。
无	i_udiPortACollisions	UDINT	遭遇一个或多个冲突但随后成功传输的帧的数量。在电源接通或使用复位命令 ETH_W.q_wResetCounter 后复位。
<b>特定于 Modbus TCP/IP</b>			
无	i_udiModbusMessageTransmitted	UDINT	已传输的 Modbus 消息数。 在电源接通或使用复位命令 ETH_W.q_wResetCounter 后复位。
无	i_udiModbusMessageReceived	UDINT	已接收的 Modbus 消息数。 在电源接通或使用复位命令 ETH_W.q_wResetCounter 后复位。
无	i_udiModbusErrorMessage	UDINT	传输和接收的已检测到的 Modbus 错误消息。 在电源接通或使用复位命令 ETH_W.q_wResetCounter 后复位。

**注意：** 无表示没有用于此系统变量的预定义 %MW 映射。

## ETH\_W: 以太网端口读 / 写系统变量

### 变量结构

下表介绍 ETH\_W 系统变量（ETH\_W\_STRUCT 类型）的参数：

%MW	变量名称	类型	注释
无	q_wResetCounter	WORD	从 0 转换为 1 后，将复位所有 ETH_R 计数器。 若要再次复位，需要先将此寄存器写为 0，之后才能再次进行从 0 到 1 的转换。

**注意：** 无表示没有用于此系统变量的预定义 %MW 映射。

---

# M218 系统功能

# 2

---

## 概述

本章描述 M218 PLCSystem 库中包含的功能。

## 本章包含了哪些内容？

本章包含了以下部分：

节	主题	页
2.1	M218 读取功能	26
2.2	M218 写入功能	48

## 2.1 M218 读取功能

### 概述

本节描述 M218 PLCSystem 库中包含的读取功能。

在许多情况下，也可以通过系统变量 (参见第 11 页) 获取这些功能返回的类似信息。系统变量在控制平台间是通用的，因此建议尽可能地使用这些变量，以提高应用程序的可移植性。系统变量中包含的信息通常是类似的，尽管并不总是与功能返回的信息完全相同。如果您要使用系统变量来替换程序中的这些功能，请确保了解两者所提供的信息的差异。

以下功能描述指示何时可使用系统变量来获得类似信息。

### 本节包含了哪些内容？

本节包含了以下主题：

主题	页
GetBatteryLevel: 返回电池的剩余电量	27
GetBootProjectStatus: 返回引导项目状态	28
GetEventsNumber: 返回检测到的外部事件数	29
GetFirmwareVersion: 返回有关固件、引导和协处理器版本的信息。	30
GetHardwareVersion: 返回硬件版本	31
GetLastStopCause: 返回上次停止的原因	32
GetLastStopTime: 返回上次检测到停止的日期和时间	33
GetLocalIOStatus: 返回嵌入式 I/O 状态	34
GetLocalAIOSStatus: 返回嵌入式模拟量 I/O 状态	36
GetPlcFault: 返回控制器 I/O 上检测到的错误	37
GetRightBusStatus: 返回扩展总线的状态	38
GetSerialNumber: 返回控制器的序列号	42
GetShortCutStatus: 返回嵌入式输出上的短路状态	43
IsFirstMastColdCycle: 指示循环是否为第一个 MAST 冷启动循环	44
IsFirstMastCycle: 指示循环是否为第一个 MAST 循环	45
IsFirstMastWarmCycle: 指示循环是否为第一个 MAST 热启动循环	47

## GetBatteryLevel: 返回电池的剩余电量

### 功能描述

此功能返回外部备用电池的剩余电量（以百分比表示）。

有关内部和外部电池的详细信息，请参阅 M218 硬件指南。

**注意：** 也可通过系统变量 `PLC_R.i_wBatteryStatus`（参见第 15 页）获得类似信息

### 图形表示形式



### IL 和 ST 表示形式

若要查看 IL 或 ST 语言的一般表示形式，请参阅 *功能和功能块表示形式*（参见第 79 页）一章。

### I/O 变量描述

下表对输出变量进行了描述：

输出	类型	注释
GetBatteryLevel	WORD	电池中剩余电量的百分比。 范围：0..100: <ul style="list-style-type: none"> <li>● 100%= 3.5 V</li> <li>● 50% = 2.75 V</li> <li>● 0%= 2 V</li> </ul> 电池 LED 的闪烁阈值: <ul style="list-style-type: none"> <li>● 100%= 常灭</li> <li>● 50%= 定期闪烁</li> <li>● 0%= 常亮</li> </ul>

## GetBootProjectStatus: 返回引导项目状态

### 功能描述

此功能返回引导项目状态。

**注意:** 也可通过系统变量 PLC\_R.i\_wBootProjectStatus (参见第 15 页) 获得类似信息。

### 图形表示形式



### IL 和 ST 表示形式

若要查看 IL 或 ST 语言的一般表示形式，请参阅功能和功能块表示形式 (参见第 79 页) 一章。

### I/O 变量描述

下表对输出变量进行了描述:

输出	类型	注释
GetBootProjectStatus	BOOT_PROJECT_STATUS (参见第 63 页)	引导项目的状态。

## GetEventsNumber: 返回检测到的外部事件数

### 功能描述

该功能返回自上一次冷启动以来发生的事件数（包括输入上检测到的事件和 HSC 阈值比较事件）。

### 图形表示形式



### IL 和 ST 表示形式

若要查看 IL 或 ST 语言的一般表示形式，请参阅 *功能和功能块表示形式*（参见第 79 页）一章。

### I/O 变量描述

下表对输出变量进行了描述：

输出	类型	注释
GetEventsNumber	DWORD	该值表示自上一次冷启动以来发生的事件数。通过调用 ResetEventsNumber (参见第 50 页) 功能复位为 0。

## GetFirmwareVersion: 返回有关固件、引导和协处理器版本的信息。

### 功能描述

此功能返回有关控制器的以下信息：

- 固件版本
- 引导版本（固件更新的内核）
- 协处理器版本（专用于嵌入式 I/O 管理的 ASIC）

**注意：**也可通过系统变量 PLC\_R.i\_byFirmVersion、PLC\_R.i\_byBootVersion 和 PLC\_R.i\_byChipVersion（参见第 15 页）获得类似信息

### 图形表示形式



### IL 和 ST 表示形式

若要查看 IL 或 ST 语言的一般表示形式，请参阅功能和功能块表示形式（参见第 79 页）一章。

### I/O 变量描述

下表对输出变量进行了描述：

输出	类型	注释
GetFirmwareVersion	FIRMWARE_VERSION (参见第 62 页)	固件、引导和协处理器版本。

### 示例

此示例描述如何使用结构化变量获取版本值：

```
VAR MyFirmwareVersion :FIRMWARE_VERSION; MyFwVersion :DWORD;
MyBootVersion :WORD; MyCoProcVersion :WORD; END_VAR
MyFirmwareVersion := GetFirmwareVersion(); MyFwVersion :=
MyFirmwareVersion.FwVersion; MyBootVersion :=
MyFirmwareVersion.BootVersion; MyCoProcVersion :=
MyFirmwareVersion.AsicVersion;
```

## GetHardwareVersion: 返回硬件版本

### 功能描述

此功能返回控制器的硬件版本。

**注意:** 也可通过系统变量 PLC\_R.i\_dwHardVersion (参见第 15 页) 获得类似信息

### 图形表示形式



### IL 和 ST 表示形式

若要查看 IL 或 ST 语言的一般表示形式，请参阅功能和功能块表示形式 (参见第 79 页) 一章。

### I/O 变量描述

下表对输出变量进行了描述:

输出	类型	注释
GetHardwareVersion	WORD	控制器的硬件版本。

## GetLastStopCause: 返回上次停止的原因

### 功能描述

此功能返回上次从“运行”转换为其他状态的原因。

**注意** 也可通过系统变量 PLC\_R.i\_wLastStopCause (参见第 15 页) 获得类似信息值。

### 图形表示形式



### IL 和 ST 表示形式

若要查看 IL 或 ST 语言的一般表示形式，请参阅功能和功能块表示形式 (参见第 79 页) 一章。

### I/O 变量描述

下表对输出变量进行了描述：

输出	类型	注释
GetLastStopCause	STOP_WHY (参见第 64 页)	上次从“运行”转换为其他状态的原因。

## GetLastStopTime: 返回上次检测到停止的日期和时间

### 功能描述

此功能返回上次从“运行”转换为其他状态的日期和时间。

### 图形表示形式



### IL 和 ST 表示形式

若要查看 IL 或 ST 语言的一般表示形式，请参阅[功能和功能块表示形式](#)（参见第 79 页）一章。

### I/O 变量描述

下表对输出变量进行了描述：

输出	类型	注释
GetLastStopTime	DWORD	上次检测到“停止”的时间（以秒为单位，从 1970 年 1 月 1 日 00:00 开始计起）。

## GetLocalIOStatus: 返回嵌入式 I/O 状态

### 功能描述

此功能返回嵌入式 I/O 状态。

**注意:** 也可通过系统变量 `PLC_R.i_wLocalIOStatus` (参见第 15 页) 获得类似信息。

### 图形表示形式



### IL 和 ST 表示形式

若要查看 IL 或 ST 语言的一般表示形式，请参阅 *功能和功能块表示形式* (参见第 79 页) 一章。

### I/O 变量描述

下表对输入参数进行描述:

输入	类型	注释
Mode	LOCAL_IO_GET_STATUS (参见第 65 页)	功能的参数: 当前只能使用 LOCAL_IO_GET_GEN_STATUS (00 十六进制) (参见第 65 页)。

下表对输出变量进行了描述:

输出	类型	注释
GetLocalIOStatus	LOCAL_IO_GEN_STATUS (参见第 66 页)	嵌入式 I/O 的状态。

### 示例 1

此示例演示如何将 LOCAL\_IO\_GET\_GEN\_STATUS 枚举类型的 LOCAL\_IO\_GET\_STATUS 枚举器直接用于 Mode 输入参数:

```
VAR MyLocalIOStatus :LOCAL_IO_GEN_STATUS; END_VAR
```

```
MyLocalIOStatus := GetLocalIOStatus(LOCAL_IO_GET_GEN_STATUS);
```

**示例 2**

此示例演示如何将中间变量用于 Mode 输入参数。

```
VAR MyLocalIOStatus :LOCAL_IO_GEN_STATUS; MyMode  
:LOCAL_IO_GEN_STATUS; END_VAR
```

```
MyMode := LOCAL_IO_GET_GEN_STATUS; MyLocalIOStatus :=  
GetLocalIOStatus(MyMode);
```

## GetLocalAIStatus: 返回嵌入式模拟量 I/O 状态

### 功能描述

此功能返回本地输入和输出的状态。

### 图形表示形式



### IL 和 ST 表示形式

若要查看 IL 或 ST 语言的一般表示形式，请参阅 *功能和功能块表示形式*（参见第 79 页）一章。

### I/O 变量描述

下表对输出变量进行了描述：

输出	类型	注释
位 0	WORD	模拟量初始化失败。
位 1	WORD	模拟量通讯丢失。
位 2	WORD	模拟量 15V 丢失。
位 3	WORD	模拟量 5V 丢失。
位 4	WORD	模拟量通道 1 输出错误。
位 5	WORD	模拟量通道 2 输出错误。
位 6	WORD	模拟量通道 1 输出超出范围。
位 7	WORD	模拟量通道 1 输出超出范围。
位 8... 位 15	WORD	保留。

## GetPlcFault: 返回控制器 I/O 上检测到的错误

### 功能描述

此功能采用位域返回控制器 I/O 的常规诊断（十六进制的 FFFF 表示未检测到错误）：

**注意：**也可通过系统变量（位 0 和位 1）PLC\_R.i\_lwSystemFault\_1（参见第 15 页）获得类似信息。

### 图形表示形式



### IL 和 ST 表示形式

若要查看 IL 或 ST 语言的一般表示形式，请参阅[功能和功能块表示形式](#)（参见第 79 页）一章。

### I/O 变量描述

下表对输出变量进行了描述：

输出	类型	注释
GetPlcFault	WORD	采用位域的控制器 I/O 诊断（十六进制的 FFFF 表示未检测到错误）。

下表详细介绍输出位域中包含的位：

位	含义
0	FALSE = 在 I/O 扩展总线上检测到错误。 使用 GetRightBusStatus（参见第 38 页）功能可获取详细诊断信息。
1	FALSE = 在嵌入式 I/O 上检测到错误。 使用 GetLocalIOStatus（参见第 34 页）功能可获取详细诊断信息。
2..15	保留位，始终为 1

## GetRightBusStatus: 返回扩展总线的状态

### 功能描述

此功能采用位域返回 I/O 扩展总线状态。根据输入参数 (Mask)，该功能返回对 I/O 扩展总线的配置诊断 (I/O 扩展总线配置与插入的模块不匹配) 或所请求扩展模拟量 I/O 模块的详细诊断。

### 图形表示形式



### IL 和 ST 表示形式

若要查看 IL 或 ST 语言的一般表示形式，请参阅 *功能和功能块表示形式* (参见第 79 页) 一章。

### I/O 变量说明

下表对输入参数进行描述:

输入	类型	注释
Mask	RIGHTBUS_GET_STATUS (参见第 67 页)	定义功能返回的 I/O 扩展总线诊断的类型: 总线配置或 7 个可能的扩展模块之一。

下表对输出变量进行了描述:

输出	类型	注释
GetRightBusStatus	字	I/O 扩展总线诊断。 请参见下面的详细信息

### I/O 扩展总线配置诊断

下表描述当扩展总线配置诊断的输入参数 (Mask) 为 RIGHT\_BUS\_GET\_GEN\_STATUS (参见第 67 页) 时, GetRightBusStatus 功能返回的位域 (如果未检测到错误, 则为十六进制的 0000):

位	说明
0	保留位 (始终为 0)
1	如果 TM2 模块 1 配置与插入的模块不匹配, 则为 TRUE。

位	说明
2	如果 TM2 模块 2 配置与插入的模块不匹配, 则为 TRUE。
3	如果 TM2 模块 3 配置与插入的模块不匹配, 则为 TRUE。
4	如果 TM2 模块 4 配置与插入的模块不匹配, 则为 TRUE。
5	如果 TM2 模块 5 配置与插入的模块不匹配, 则为 TRUE。
6	如果 TM2 模块 6 配置与插入的模块不匹配, 则为 TRUE。
7	如果 TM2 模块 7 配置与插入的模块不匹配, 则为 TRUE。
8...15	保留位 (始终为 0)

### I/O 扩展总线模块诊断

下表介绍当扩展模块“x”的详细诊断的输入参数 (Mask) 为 RIGHTBUS\_GET\_STATUS<sub>x</sub> (参见第 67 页) (其中 x=1 至 7) 时, GetRight-BusStatus 功能返回的位域:

**注意:** 详细诊断仅对模拟量 I/O 模块有效, 位域含义取决于相关模拟量模块的类型。

当关联模块为标准模拟量 I/O 模块时 (最多 2 路输入通道):

- TM2AMM3HT
- TM2ALM3LT
- TM2AMI2HT
- TM2AMI2LT
- TM2AVO2HT
- TM2AMO1HT

位	说明
0	所有模拟量通道处于正常状态
1	模块处于初始化状态
2	电源未正常运行
3	配置不正确 - 需要分析
4	输入通道 0 处于转换过程
5	输入通道 1 处于转换过程
6	输入通道 0 的参数无效
7	输入通道 1 的参数无效
8	未使用
9	未使用
10	输入通道 0 的值溢出
11	输入通道 1 的值溢出
12	输入通道 0 的值下溢

位	说明
13	输入通道 1 的值下溢
14	未使用
15	输出通道的参数无效

当关联模块为 4 路或 8 路模拟量输入通道模块之一时：

- TM2ARI8HT
- TM2AMI8HT
- TM2ARI8LT
- TM2ARI8LRJ
- TM2AMI4LT
- TM2AMM6HT

位	说明	含义
0, 1	通道 0 状态	00: 模拟量通道处于正常状态 01: 输入通道的参数无效 10: 输入值不可用（模块处于初始化状态、转换过程） 11: 输入通道的值无效（值溢出或下溢）
2, 3	通道 1 状态	请参见位 0、1
4, 5	通道 2 状态	请参见位 0、1
6, 7	通道 3 状态	请参见位 0、1
8, 9	通道 4 状态	请参见位 0、1（仅用于 8 路输入通道模块）
10, 11	通道 5 状态	请参见位 0、1（仅用于 8 路输入通道模块）
12, 13	通道 6 状态	请参见位 0、1（仅用于 8 路输入通道模块）
14, 15	通道 7 状态	请参见位 0、1（仅用于 8 路输入通道模块）

**注意：**当目标扩展模块是数字量 I/O 模块、高速计数器模块或 AS-i 主模块时，返回的诊断无效（十六进制的 0000）。

要获取对 HSC 和 AS-i 扩展模块的诊断，可以使用：

- 用于 TM200 HSC206D• 模块的 HSCGetDiag（参见 *Modicon M218 Logic Controller, 高速计数, M218 HSC 库指南*）功能块
- 用于 TWDNOI10M3 模块的 ASI\_MasterStatusCheck

## 示例

下面的示例介绍使用 GetRightBusStatus 诊断 I/O 扩展总线和模拟量模块的方法：

```
VAR (* 模块 1 至 7 配置诊断 = MyRightBusStatus 位 1 至 7*)
MyRightBusStatus:WORD; (* 模块 1 至 7 诊断代码 *)
ModuleError:Array [1..7] of WORD; END_VAR

(* 扩展总线上的配置诊断 *)
MyRightBusStatus:=GetRightBusStatus(RIGHTBUS_GET_GEN_STATUS)
;

IF MyRightBusStatus<>0 THEN (* 检测到配置不匹配 => 设置警报, 检查位
值... *)END_IF;

(* 获取模块诊断: 检测到错误, 如果诊断 <> 0*) (* 将列表限制为仅用于已配置
的模拟量模块 *)
ModuleError[1]:=GetRightBusStatus(RIGHTBUS_GET_STATUS1);
ModuleError[2]:=GetRightBusStatus(RIGHTBUS_GET_STATUS2);
ModuleError[3]:=GetRightBusStatus(RIGHTBUS_GET_STATUS3);
ModuleError[4]:=GetRightBusStatus(RIGHTBUS_GET_STATUS4);
ModuleError[5]:=GetRightBusStatus(RIGHTBUS_GET_STATUS5);
ModuleError[6]:=GetRightBusStatus(RIGHTBUS_GET_STATUS6);
ModuleError[7]:=GetRightBusStatus(RIGHTBUS_GET_STATUS7);
```

## GetSerialNumber: 返回控制器的序列号

### 功能描述

此功能返回控制器的序列号。

**注意:** 也可通过系统变量 (位 0 和位 1) PLC\_R.i\_dwSerialNumber (参见第 15 页) 获得类似信息。

### 图形表示形式



### IL 和 ST 表示形式

若要查看 IL 或 ST 语言的一般表示形式, 请参阅功能和功能块表示形式 (参见第 79 页) 一章。

### I/O 变量描述

下表对输出变量进行了描述:

输出	类型	注释
GetSerialNumber	DWORD	控制器序列号。

## GetShortCutStatus: 返回嵌入式输出上的短路状态

### 功能描述

此功能返回嵌入式输出上的短路或过载诊断。

**注意:** 有关嵌入式输出管理的详细信息, 请参阅 M218 硬件指南。

### 图形表示形式



### IL 和 ST 表示形式

若要查看 IL 或 ST 语言的一般表示形式, 请参阅 *功能和功能块表示形式* (参见第 79 页) 一章。

### I/O 变量描述

下表对输出变量进行了描述:

参数	类型	注释
GetShortCutStatus	WORD	请参见下面的位域说明

下表描述 TM218LDA40DR4PHN、TM218LDA40DRPHN 以及 TM218LDAE40DRPHN 的位域:

位	说明
0	TRUE= 输出组 1 (Q0 至 Q3) 上短路。

## IsFirstMastColdCycle: 指示循环是否为第一个 MAST 冷启动循环

### 功能描述

此功能在冷启动之后的第一个 MAST 循环期间（下载或冷复位后的第一个循环）返回 TRUE。

### 图形表示形式



### IL 和 ST 表示形式

若要查看 IL 或 ST 语言的一般表示形式，请参阅 *功能和功能块表示形式*（参见第 79 页）一章。

### I/O 变量描述

下表对输出变量进行描述：

输出	类型	注释
IsFirstMastColdCycle	BOOL	冷启动之后的第一个 MAST 任务循环期间为 TRUE。

### 示例

请参阅功能 IsFirstMastCycle（参见第 45 页）。

## IsFirstMastCycle: 指示循环是否为第一个 MAST 循环

### 功能介绍

此功能在启动后的第一个 MAST 循环期间返回 TRUE。

### 图形表示形式



### IL 和 ST 表示形式

若要查看 IL 或 ST 语言的一般表示形式，请参阅 *功能和功能块表示形式*（参见第 79 页）一章。

### I/O 变量介绍

输出	类型	注释
IsFirstMastCycle	BOOL	启动之后的第一个 MAST 任务循环期间为 TRUE。

### 示例

此示例介绍三个一起使用的功能 IsFirstMastCycle、IsFirstMastColdCycle 和 IsFirstMastWarmCycle。

此示例必须在 MAST 任务中使用，否则可能运行多次或一次也不运行（在第一个 MAST 任务循环期间可能会多次调用或不调用某个附件任务）：

```
VAR MyIsFirstMastCycle :BOOL; MyIsFirstMastWarmCycle :BOOL;
MyIsFirstMastColdCycle :BOOL; END_VAR
```

```
MyIsFirstMastWarmCycle := IsFirstMastWarmCycle();
```

```
MyIsFirstMastColdCycle := IsFirstMastColdCycle();
```

```
MyIsFirstMastCycle := IsFirstMastCycle();
```

```
IF (MyIsFirstMastWarmCycle) THEN
```

(\* 这是热启动后的第一个 Mast 循环: 所有变量都设置为其初始化值, 除了保留变量 \*)

(\*=> 初始化所需变量, 以便使应用程序在这种情况下按预期运行 \*)

```
END_IF;
```

```
IF (MyIsFirstMastColdCycle) THEN
```

*(\* 这是冷启动后的第一个 Mast 循环: 所有变量都设置为其初始化值, 包括保留变量 \*)*

*(\*=> 初始化所需变量, 以便使应用程序在这种情况下按预期运行 \*)*

END\_IF;

IF (MyIsFirstMastCycle) THEN

*(\* 这是启动后的第一个 Mast 循环, 这里的启动是指热启动或冷启动, 以及停止 / 运行命令 \*)*

*(\*=> 初始化所需变量, 以便使应用程序在这种情况下按预期运行 \*)*

END\_IF;

## IsFirstMastWarmCycle: 指示循环是否为第一个 MAST 热启动循环

### 功能描述

此功能在热启动之后的第一个 MAST 循环期间返回 TRUE。

### 图形表示形式



### IL 和 ST 表示形式

若要查看 IL 或 ST 语言的一般表示形式，请参阅 *功能和功能块表示形式*（参见第 79 页）一章。

### I/O 变量描述

下表对输出变量进行描述：

输出	类型	注释
IsFirstMastWarmCycle	BOOL	热启动之后的第一个 MAST 任务循环期间为 TRUE。

### 示例

请参阅功能 IsFirstMastCycle（参见第 45 页）。

---

## 2.2 M218 写入功能

---

### 概述

本节描述 M218 PLCSystem 库中包含的写入功能。

在许多情况下，也可以通过系统变量 (参见第 11 页) 获取这些功能返回的类似信息。系统变量在控制平台间是通用的，因此建议尽可能地使用这些变量，以提高应用程序的可移植性。系统变量中包含的信息通常是类似的，尽管并不总是与功能返回的信息完全相同。如果您要使用系统变量来替换程序中的这些功能，请确保了解两者所提供的信息的差异。

以下功能描述指示何时可使用系统变量来获得类似信息。

### 本节包含了哪些内容？

本节包含了以下主题：

主题	页
InhibitBatLowLed: 禁用或重新启用电池 LED	49
ResetEventsNumber: 复位事件数	50
SetRTCDrift: 每周调整实时时钟	51

## InhibitBatLowLed: 禁用或重新启用电池 LED

### 功能描述

此功能禁用或重新启用低电池电量 LED 指示灯（电池）的显示。

在控制器不使用外部电池的情况下可以使用此功能，以避免出现持久的错误信号。在执行**复位到起点**命令或从编程面板下载到控制器后，会自动恢复低电池电量 LED 指示灯（电池）的显示。

**注意：**有关外部电池的详细信息，请参阅 M218 硬件指南。

**注意：**有关 LED 指示灯的详细信息，请参阅 M218 硬件指南。

### 图形表示形式



### IL 和 ST 表示形式

若要查看 IL 或 ST 语言的一般表示形式，请参阅**功能和功能块表示形式**（参见第 79 页）一章。

### I/O 变量描述

下表对输入参数进行描述：

输入	类型	注释
Inhibit	BOOL	TRUE = 禁用电池 LED 的显示。 FALSE = 重新启用电池 LED 的显示。

下表对输出参数进行描述：

输出	类型	注释
InhibitBatLowLed	WORD	00 十六进制 = 操作成功 否则 = 操作不成功

## ResetEventsNumber: 复位事件数

### 功能描述

该功能将自上一次冷启动以来发生的事件数（包括输入上检测到的事件和 HSC 阈值比较事件）复位。

**注意：**输入或 HSC 阈值检测上检测到的事件数由 GetEventsNumber（参见第 29 页）功能返回。

### 图形表示形式



### IL 和 ST 表示形式

若要查看 IL 或 ST 语言的一般表示形式，请参阅 *功能和功能块表示形式*（参见第 79 页）一章。

### I/O 变量描述

下表对输出变量进行了描述：

输出	类型	注释
ResetEventsNumber	BOOL	如果计数器成功复位为 0，则为 TRUE。

## SetRTCDrift: 每周调整实时时钟

### 功能介绍

此功能用于在每周的指定日的某时某分 30 秒对实时时钟增加或减去指定的秒数。

**注意:** 必须至少调用一次 SetRTCDrift 功能需要对 进行编程, 使其只能在第一个 Mast 循环期间执行。

### 图形表示形式



### IL 和 ST 表示形式

若要查看 IL 或 ST 语言的一般表示形式, 请参阅*功能和功能块表示形式* (参见第 79 页)一章。

### I/O 变量介绍

下表介绍了输入参数:

输入	类型	注释
RTCDrift	SINT (-29..29)	按秒修正 (-29 ... +29)
日	DAY_OF_WEEK (参见第 68 页)	在一周中的哪一天执行。
时	HOUR (参见第 69 页)	在几点钟进行更改。
分	MINUTE (参见第 70 页)	在第几分钟进行更改。 在每周的指定日的几时几分 30 秒, RTC 被设置成几时几分 (30+RTCDrift) 秒。

**注意:** 如果为 RTCDrift、日、时、分输入的值超过了限制值, 则控制器固件会将所有值设置为其最大值。

下表介绍了输出变量:

输出	类型	注释
SetRTCDrift	WORD	如果命令正常运行, 则返回十六进制的 00, 否则返回检测到的错误的 ID 代码。

**示例**

在此示例中，第一个 MAST 任务循环期间仅调用此功能一次，每个星期二的上午 5:45:30 为 RTC 增加 20 秒：

```
VAR MyRTCDrift :SINT (-29..29) := 0; MyDay :DAY_OF_WEEK;  
MyHour :HOUR; MyMinute :MINUTE; END_VAR  
  
IF IsFirstMastCycle() THEN MyRTCDrift := 20; MyDay := TUESDAY;  
MyHour := 5; MyMinute := 45; SetRTCDrift(MyRTCDrift, MyDay,  
MyHour, MyMinute); END_IF
```

---

# M218 PLCSystem 库数据类型

# 3

---

## 概述

本章介绍 M218 PLCSystem 库的**数据类型**。

可使用**数据类型**有两种：

- **系统变量数据类型**，由 M218 PLCSystem 库的**系统变量**（参见第 9 页）（PLC\_R、PLC\_W 等）使用。
- **系统功能数据类型**，由 M218 PLCSystem 库的**读 / 写系统功能**（参见第 25 页）使用。

## 本章包含了哪些内容？

本章包含了以下部分：

节	主题	页
3.1	PLC_R/W 系统变量数据类型	54
3.2	系统功能数据类型	61
3.3	ETH_R/W 系统变量数据类型	71

## 3.1 PLC\_R/W 系统变量数据类型

### 概述

本节列出并描述 PLC\_R 和 PLC\_W 结构中包含的**系统变量数据类型**。

### 本节包含了哪些内容？

本节包含了以下主题：

主题	页
PLC_R_STATUS: 控制器状态代码	55
PLC_R_APPLICATION_ERROR: 检测到的应用程序错误状态代码	56
PLC_R_BOOT_PROJECT_STATUS: 引导项目状态代码	57
PLC_R_IO_STATUS: I/O 状态代码	58
PLC_R_STOP_CAUSE: 从“运行”向其他状态转换的原因代码	59
PLC_W_COMMAND: 控制命令代码	60

## PLC\_R\_STATUS: 控制器状态代码

### 枚举类型介绍

PLC\_R\_STATUS 枚举数据类型包含以下值：

枚举器	值	注释
PLC_R_EMPTY	00 (十六进制)	控制器未进行编程。
PLC_R_STOPPED	01 (十六进制)	控制器已停止。
PLC_R_RUNNING	02 (十六进制)	控制器正在运行。
PLC_R_HALT	04 (十六进制)	控制器处于“暂停”状态。(请参见控制器编程指南中的“控制器状态图”)。
PLC_R_BREAKPOINT	08 (十六进制)	控制器已在断点处暂停。

**PLC\_R\_APPLICATION\_ERROR: 检测到的应用程序错误状态代码****枚举类型介绍**

PLC\_R\_APPLICATION\_ERROR 枚举数据类型包含以下值:

枚举器	值	注释
PLC_R_APP_ERR_UNKNOWN	FFFF (十六进制)	未知错误。
PLC_R_APP_ERR_NOEXCEPTION	0000 (十六进制)	未检测到错误。
PLC_R_APP_ERR_WATCHDOG	0010 (十六进制)	任务的应用程序看门狗已过期。
PLC_R_APP_ERR_HARDWAREWATCHDOG	0011 (十六进制)	硬件看门狗已过期。
PLC_R_APP_ERR_IO_CONFIG_ERROR	0012 (十六进制)	检测到不正确的 I/O 配置参数。
PLC_R_APP_ERR_UNRESOLVED_EXTREFS	0018 (十六进制)	检测到未知功能。
PLC_R_APP_ERR_IEC_TASK_CONFIG_ERROR	0025 (十六进制)	检测到不正确的任务配置参数。
PLC_R_APP_ERR_ILLEGAL_INSTRUCTION	0050 (十六进制)	检测到未知指令。
PLC_R_APP_ERR_ACCESS_VIOLATION	0051 (十六进制)	对保留存储器区域的访问。
PLC_R_APP_ERR_PROCESSORLOAD_WATCHDOG	0105 (十六进制)	处理器由于应用程序任务而过载。

**PLC\_R\_BOOT\_PROJECT\_STATUS: 引导项目状态代码****枚举类型描述**

PLC\_R\_BOOT\_PROJECT\_STATUS 枚举数据类型包含以下值:

枚举器	值	注释
PLC_R_NO_BOOT_PROJECT	0000 (十六进制)	
PLC_R_BOOT_PROJECT_CREATION_IN_PROGRESS	0001 (十六进制)	正在创建引导项目。
PLC_R_DIFFERENT_BOOT_PROJECT	0002 (十六进制)	
PLC_R_VALID_BOOT_PROJECT	FFFF (十六进制)	闪存中的引导项目与 RAM 中加载的项目相同。

## PLC\_R\_IO\_STATUS: I/O 状态代码

### 枚举类型介绍

PLC\_R\_IO\_STATUS 枚举数据类型包含以下值:

枚举器	值	注释
PLC_R_IO_OK	FFFF (十六进制)	输入 / 输出运行正常。
PLC_R_IO_NO_INIT	0001 (十六进制)	输入 / 输出未初始化。
PLC_R_IO_CONF_FAULT	0002 (十六进制)	检测到不正确的 I/O 配置参数。
PLC_R_IO_SHORTCUT_FAULT	0003 (十六进制)	检测到输入 / 输出短路。
	0004 (十六进制)	

## PLC\_R\_STOP\_CAUSE: 从“运行”向其他状态转换的原因代码

### 枚举类型介绍

PLC\_R\_STOP\_CAUSE 枚举数据类型包含以下值:

## PLC\_W\_COMMAND: 控制命令代码

### 枚举类型描述

PLC\_W\_COMMAND 枚举数据类型包含以下值:

枚举器	值	注释
PLC_W_STOP	01 (十六进制)	用于停止控制器的命令。
PLC_W_RUN	02 (十六进制)	用于运行控制器的命令。
PLC_W_RESET_COLD	04 (十六进制)	用于启动控制器冷复位的命令。
PLC_W_RESET_WARM	08 (十六进制)	用于启动控制器热复位的命令。

## 3.2 系统功能数据类型

### 概述

本节描述 M218 PLCSystem 库的不同系统功能数据类型。

### 本节包含了哪些内容？

本节包含了以下主题：

主题	页
FIRMWARE_VERSION: GetFirmwareVersion 功能输出类型	62
BOOT_PROJECT_STATUS: GetBootProjectStatus 功能输出代码	63
STOP_WHY: GetLastStopCause 功能输出代码	64
LOCAL_IO_GET_STATUS: GetLocalIOStatus 功能参数代码	65
LOCAL_IO_GEN_STATUS: GetLocalIOStatus 功能输出代码	66
RIGHTBUS_GET_STATUS: GetRightBusStatus 功能参数代码	67
DAY_OF_WEEK: SetRTCDrift 功能日期参数代码	68
HOUR: SetRTCDrift 功能小时参数类型	69
MINUTE: SetRTCDrift 功能分钟参数类型	70

**FIRMWARE\_VERSION: GetFirmwareVersion 功能输出类型****结构描述**

该数据结构包含下列变量：

变量	类型	说明
FwVersion	DWORD	包含固件版本（例如：2.0.20.4= 十六进制的 02001404）。
BootVersion	WORD	包含引导版本（例如：1.4= 十六进制的 0104）。
AsicVersion	WORD	包含协处理器版本。

**BOOT\_PROJECT\_STATUS: GetBootProjectStatus 功能输出代码****枚举类型描述**

枚举数据类型包含下列值：

枚举器	值	说明
NO_BOOT_PROJECT	0000 (十六进制)	未使用。
BOOT_PROJECT_CREATION_IN_PROGRESS	0001 (十六进制)	正在创建引导项目。
DIFFERENT_BOOT_PROJECT	0002 (十六进制)	闪存中的引导项目与 RAM 中加载的项目不同，或不存在。
VALID_BOOT_PROJECT	FFFF (十六进制)	闪存中的引导项目与 RAM 中加载的项目相同。

**STOP\_WHY: GetLastStopCause 功能输出代码****枚举类型描述**

枚举数据类型包含具有下列值的状态：

枚举器	值	说明
STOP_REASON_UNKNOWN	00 (十六进制)	未定义初始值或停止原因。
STOP_REASON_HW_WATCHDOG	01 (十六进制)	硬件看门狗后停止。
STOP_REASON_RESET	02 (十六进制)	复位后停止。
STOP_REASON_EXCEPTION	03 (十六进制)	例外后停止。
STOP_REASON_USER	04 (十六进制)	用户请求后停止。
STOP_REASON_IECPROGRAM	05 (十六进制)	程序命令请求后 (例如: 带参数的控制命令 PLC_W.q_wPLCControl := PLC_W_COMMAND.PLC_W_STOP;) 停止。
STOP_REASON_DELETE	06 (十六进制)	删除应用程序命令后停止。
STOP_REASON_DEBUGGING	07 (十六进制)	进入调试模式后停止。
STOP_FROM_NETWORK_REQUEST	0A (十六进制)	从网络进行请求后 (USB 盘或 PLC_W 命令) 停止。
STOP_FROM_INPUT	0B (十六进制)	控制器输入要求停止。

**LOCAL\_IO\_GET\_STATUS: GetLocalIOStatus 功能参数代码****枚举类型描述**

LOCAL\_IO\_GET\_STATUS 枚举数据类型包含下列值:

枚举器	值	说明
LOCAL_IO_GET_GEN_STATUS	00 (十六进制)	用于请求嵌入式 I/O 的常规状态的值

**LOCAL\_IO\_GEN\_STATUS: GetLocalIOStatus 功能输出代码****枚举类型描述**

LOCAL\_IO\_GEN\_STATUS 枚举数据类型包含具有下列值的本地 I/O 状态:

枚举器	值	说明
LOCAL_IO_OK	00 (十六进制)	输入 / 输出运行正常。
LOCAL_IO_NO_INIT	01 (十六进制)	检测到错误的 I/O 配置参数。
LOCAL_IO_COM_LOST	02 (十六进制)	协处理器的通讯错误。 未更新嵌入式 I/O。
LOCAL_IO_CONF_FAULT	03 (十六进制)	检测到错误的 I/O 配置参数。

**RIGHTBUS\_GET\_STATUS: GetRightBusStatus 功能参数代码****枚举类型描述**

枚举数据类型包含下列值：

枚举器	值	说明
RIGHTBUS_GET_GEN_STATUS	00 (十六进制)	扩展总线配置诊断的参数
RIGHTBUS_GET_STATUS1	01 (十六进制)	扩展总线模块 1 诊断的参数
RIGHTBUS_GET_STATUS2	02 (十六进制)	扩展总线模块 2 诊断的参数
RIGHTBUS_GET_STATUS3	03 (十六进制)	扩展总线模块 3 诊断的参数
RIGHTBUS_GET_STATUS4	04 (十六进制)	扩展总线模块 4 诊断的参数
RIGHTBUS_GET_STATUS5	05 (十六进制)	扩展总线模块 5 诊断的参数
RIGHTBUS_GET_STATUS6	06 (十六进制)	扩展总线模块 6 诊断的参数
RIGHTBUS_GET_STATUS7	07 (十六进制)	扩展总线模块 7 诊断的参数

**注意：**有关使用 RIGHTBUS\_GET\_STATUS 参数类型的详细信息，请参阅功能 GetRightBusStatus (参见第 38 页)。

**DAY\_OF\_WEEK: SetRTCDrift 功能日期参数代码****枚举类型描述**

枚举数据类型包含下列值：

枚举器	值	注释
MONDAY	01 (十六进制)	将“星期几”设置为星期一
TUESDAY	02 (十六进制)	将“星期几”设置为星期二
WEDNESDAY	03 (十六进制)	将“星期几”设置为星期三
THURSDAY	04 (十六进制)	将“星期几”设置为星期四
FRIDAY	05 (十六进制)	将“星期几”设置为星期五
SATURDAY	06 (十六进制)	将“星期几”设置为星期六
SUNDAY	07 (十六进制)	将“星期几”设置为星期日

## **HOURL: SetRTCDrift 功能小时参数类型**

### **数据类型描述**

该数据类型包含从 0 到 23 的小时值。

## **MINUTE: SetRTCDrift 功能分钟参数类型**

### **数据类型描述**

该数据类型包含从 0 到 59 的分钟值。

## 3.3 ETH\_R/W 系统变量数据类型

### 概述

本节列出并描述 ETH\_R 和 ETH\_W 结构中包含的**系统数据类型**。

### 本节包含了哪些内容？

本节包含了以下主题：

主题	页
ETH_R_IP_MODE: IP 地址源代码	72
ETH_R_FRAME_PROTOCOL: 帧传输协议代码	73
ETH_R_PORT_DUPLEX_STATUS: 传输模式代码	74
ETH_R_PORT_LINK_STATUS: 通讯链路方向代码	75
ETH_R_PORT_SPEED: 以太网端口的通讯速度代码	76

## ETH\_R\_IP\_MODE: IP 地址源代码

### 枚举类型描述

ETH\_R\_IP\_MODE 枚举数据类型包含以下值:

枚举器	值	注释
ETH_R_STORED	00 (十六进制)	使用存储的 IP 地址。
ETH_R_BOOTP	01 (十六进制)	使用引导程序协议获取 IP 地址。
ETH_R_DHCP	02 (十六进制)	使用 DHCP 协议获取 IP 地址。
ETH_DEFAULT_IP	FF (十六进制)	使用缺省 IP 地址。

## ETH\_R\_FRAME\_PROTOCOL: 帧传输协议代码

### 枚举类型描述

ETH\_R\_FRAME\_PROTOCOL 枚举数据类型包含以下值:

枚举器	值	注释
ETH_R_802_3	00 (十六进制)	用于帧传输的协议为 802.3。
ETH_R_ETHERNET_II	01 (十六进制)	用于帧传输的协议为 Ethernet II。

**ETH\_R\_PORT\_DUPLEX\_STATUS: 传输模式代码****枚举类型描述**

ETH\_R\_PORT\_DUPLEX\_STATUS 枚举数据类型包含以下值:

枚举器	值	注释
ETH_R_PORT_HALF_DUPLEX	00 (十六进制)	使用半双工传输模式。
ETH_R_FULL_DUPLEX	01 (十六进制)	使用全双工传输模式。

## ETH\_R\_PORT\_LINK\_STATUS: 通讯链路方向代码

### 枚举类型描述

ETH\_R\_PORT\_LINK\_STATUS 枚举数据类型包含以下值:

枚举器	值	注释
ETH_R_LINK_DOWN	00 (十六进制)	通讯链路从服务器到设备。
ETH_R_LINK_UP	01 (十六进制)	通讯链路从设备到服务器。

## ETH\_R\_PORT\_SPEED: 以太网端口的通讯速度代码

### 枚举类型描述

ETH\_R\_PORT\_SPEED 枚举数据类型包含以下值:

枚举器	值	注释
ETH_R_SPEED_10_MB	10 (十进制)	网络速度为每秒 10 兆位。
ETH_R_100_MB	100 (十进制)	网络速度为每秒 100 兆位。

---

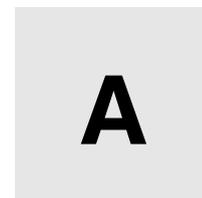
# 附录





---

# 功能和功能块表示形式



---

## 概述

每个功能可以使用以下语言表示：

- IL：指令列表
- ST：结构化文本
- LD：梯形图
- FBD：功能块图
- CFC：连续功能图

本章提供功能和功能块表现形式示例，并解释如何将它们用于 IL 和 ST 语言。

## 本章包含了哪些内容？

本章包含了以下主题：

主题	页
功能与功能块之间的差异	80
如何通过 IL 语言使用功能或功能块	81
如何通过 ST 语言使用功能或功能块	84

## 功能与功能块之间的差异

### 功能

功能:

- 是返回一个直接结果的 **POU**（程序组织单元）
- 通过其名称（而不是通过**实例**）直接调用
- 从一个调用到另一个调用不会保持原有状态
- 可以用作其他表达式中的操作数

**示例:** 布尔操作符 (AND)、计算、转换 (BYTE\_TO\_INT)

### 功能块

功能块:

- 是返回一个或多个输出的 **POU**（程序组织单元）
- 始终通过**实例**（具有专用名称和变量的功能块副本）进行调用
- 每个**实例**在从一个调用到另一个调用会保持原有状态（输出和内部变量）

**示例:** 定时器、计数器

在下面的示例中，Timer\_ON 是功能块 TON 的实例:

```
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);
```

## 如何通过 IL 语言使用功能或功能块

### 一般信息

本部分介绍如何使用 IL 语言实现功能和功能块。

我们以功能 `IsFirstMastCycle`、功能 `SetRTCDrift` 和功能块 `TON` 为例来演示实现的过程。

### 通过 IL 语言使用功能

以下步骤描述如何插入采用 IL 语言的功能：

步骤	操作
1	通过 <b>指令列表</b> 语言打开 POU 或创建新 POU。 <b>注意：</b> 此处未详细介绍创建 POU 的步骤。有关详细信息，请参阅 SoMachine 全局帮助。
2	创建功能所需的变量。
3	如果功能具有 1 个或多个输入，则使用 LD 指令开始加载第一个输入。
4	在下面插入新行，并执行以下操作： <ul style="list-style-type: none"> <li>在操作符列（左侧字段）中键入功能的名称，或</li> <li>使用<b>输入助手</b>选择功能（在上下文菜单中选择<b>插入运算块</b>）。</li> </ul>
5	如果功能具有多个输入，则在使用 <b>输入助手</b> 时，会在右侧字段中使用 ??? 自动创建必需的行数。使用与输入顺序对应的适当值或变量来替换 ???。
6	插入新行，以便将功能的结果存储到适当的变量中：在操作符列（左侧字段）中键入 ST 指令，并在右侧字段中键入变量名。

要阐释该过程，请考虑下面以图形方式表示的功能 `IsFirstMastCycle`（不带输入参数）和功能 `SetRTCDrift`（带输入参数）：

功能	图形表示形式
不带输入参数： <code>IsFirstMastCycle</code>	
带输入参数： <code>SetRTCDrift</code>	

在 IL 语言中，功能名称直接用在操作符列中：

功能	SoMachine POU IL 编辑器中的表示形式
不带输入参数的功能的 IL 示例： IsFirstMastCycle	<pre> 1  PROGRAM MyProgram_IL 2  VAR 3      FirstCycle: BOOL; 4  END_VAR                     </pre> <hr/> <pre> 1  IsFirstMastCycle    ST          FirstCycle                     </pre>
带输入参数的功能的 IL 示例： 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 9                     </pre> <hr/> <pre> 1  LD          myDrift    SetRTCDrift myDay                     myHour                     myMinute    ST          myDiag                     </pre>

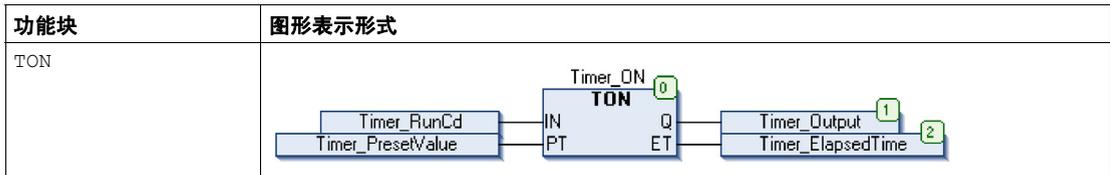
### 通过 IL 语言使用功能块

以下步骤描述如何插入采用 IL 语言的功能块：

步骤	操作
1	通过指令列表语言打开 POU 或创建新 POU。 <b>注意：</b> 此处未详细介绍创建 POU 的步骤。有关详细信息，请参阅 SoMachine 全局帮助。
2	创建功能块所需的变量（包括实例名称）。

步骤	操作
3	使用 CAL 指令调用功能块： <ul style="list-style-type: none"> <li>● 使用<b>输入助手</b>选择 FB（在上下文菜单中右键单击并选择<b>插入运算块</b>）。</li> <li>● 会自动创建 CAL 指令和必要的 I/O。</li> </ul> 每个参数 (I/O) 都是一条指令： <ul style="list-style-type: none"> <li>● 输入的值通过 “:= ” 进行设置。</li> <li>● 输出的值通过 “=&gt; ” 进行设置。</li> </ul>
4	在 CAL 右侧字段中，使用实例名称替换 ??? 。
5	使用适当的变量或立即值替换其他 ??? 。

要阐释该过程，请考虑下面以图形方式表示的 TON 功能块示例：



在 IL 语言中，功能块名称直接用在**操作符**列中：

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

## 如何通过 ST 语言使用功能或功能块

### 一般信息

本部分介绍如何使用 ST 语言实现功能和功能块。

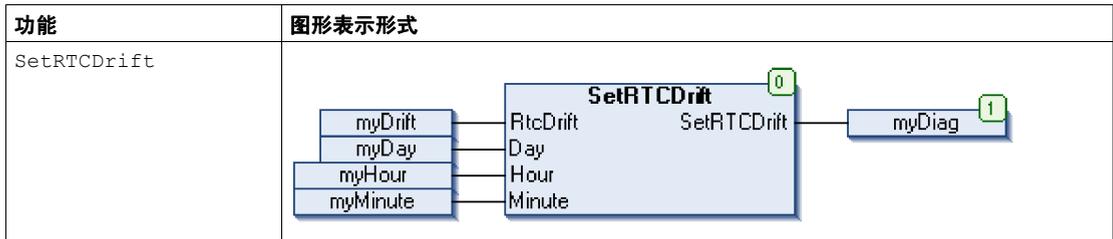
我们以功能 `SetRTCDrift` 和功能块 `TON` 为例演示实现的过程。

### 通过 ST 语言使用功能

以下步骤描述如何插入采用 ST 语言的功能：

步骤	操作
1	通过结构化文本语言打开 POU 或创建新 POU。 <b>注意：</b> 此处未详细介绍创建 POU 的步骤。有关详细信息，请参阅 SoMachine 全局帮助。
2	创建功能所需的变量。
3	在 <b>POU ST 编辑器</b> 中，使用功能 ST 语言的常规语法。常规语法为： <code>FunctionResult:= FunctionName(VarInput1, VarInput2,..VarInputx);</code>

要阐释该过程，请考虑下面以图形方式表示的功能 `SetRTCDrift`：



此功能的 ST 语言如下所示：

```
myRTCAdjust:= SetRTCDrift(myDrift, myDay, myHour, myMinute);
```

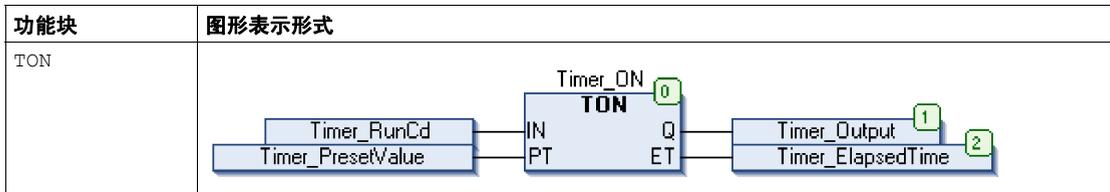
### 通过 ST 语言使用功能块

以下步骤描述如何插入采用 ST 语言的功能块：

步骤	操作
1	通过结构化文本语言打开 POU 或创建新 POU。 <b>注意：</b> 此处未详细介绍创建 POU 的步骤。有关详细信息，请参阅 SoMachine 全局帮助。

步骤	操作
2	创建功能块所需的输入和输出变量以及实例： <ul style="list-style-type: none"> <li>● 输入变量是功能块所需的输入参数</li> <li>● 输出变量接收功能块返回的值</li> </ul>
3	在 <b>POU ST 编辑器</b> 中，使用功能块 ST 语言的常规语法。常规语法为： FunctionBlock_InstanceName (Input1:=VarInput1, Input2:=VarInput2,...Ouput1=>VarOutput1, Ouput2=>VarOutput2,...);

要阐释该过程，请考虑下面以图形方式表示的 TON 功能块示例：



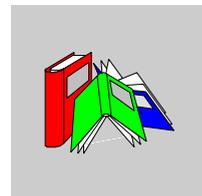
下表显示了采用 ST 语言的功能块调用的示例：

功能块	SoMachine POU ST 编辑器中的表示形式
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=&gt;Timer_Output, 5      ET=&gt;Timer_ElapsedTime); </pre>



---

# 术语



---

## 0-9

### 1 相位计数器

1 *相位计数器*使用一路硬件输入作为计数器输入。该计数器通常在输入中存在脉冲信号时进行加减计数。

### 2 相位计数器

2 *相位计数器*使用两个输入计数器信号之间的相位差进行加减计数。

### 保留数据

*保留数据*值，用于下一次电源接通或热启动。即使在控制器意外关闭或正常情况下关闭控制器后，该值也仍然保留。

### 功能块

(FB) 输入和变量的程序单元，被组织起来以根据定义的功能（例如定时器或计数器）来计算输出值。

### 功能块图语言

(FBD) 一种功能块图，描述输入变量和输出变量之间的功能。一个功能被描述为一组基本块。输入和输出变量通过连接线路与功能块连接。一个功能块的输出可能也与其他功能块的输入相连接。

### 反射输出

在计数模式下，高速计数器的当前值以其配置阈值为基础测得，以此确定这些专用输出的状态。

## 固件

*固件*表示控制器上的操作系统。

## 字节 (BYTE)

8 位组合在一起称为一个字节。可以按二进制或八进制模式输入一个字节。字节类型以八位的格式编码，其范围为 16#00 到 16#FF（以十六进制表示）。

## 定位变量

*定位变量*具有地址。（请参见*非定位变量*。）

## 引导应用程序

一些包含与机器相关的参数的文件：

- 机器名
- 设备名或 IP 地址
- Modbus 串行线路地址
- 路由表

## 扩展 I/O 模块

*扩展输入或输出模块*，是将其他 I/O 添加到本体控制器的数字量或模拟量模块。

## 扩展总线

*扩展总线*是扩展模块和 CPU 之间的电子通讯总线。

## 控制器

*控制器*（或称为“可编程逻辑控制器”，或“可编程控制器”）用于工业流程的自动化。

## 系统变量

系统变量结构提供控制器数据和诊断信息，并可以使用它向控制器发送命令。

## 结构化文本

以*结构化文本* (ST) 语言编写的程序，包括复杂的语句和嵌套指令（例如：迭代循环、条件执行或功能）。ST 符合 IEC 61131-3。

## 非定位变量

*非定位变量*没有地址。（请参见*定位变量*。）

## B

### BOOL

*布尔类型*，用于计算的基本数据类型。BOOL 变量可为以下值之一：0 (FALSE)，1 (TRUE)。从 WORD 中抽取的位为 BOOL 类型，例如：%MW10.4 是编号为 10 的存储器 WORD 的五分之一位。

### BOOTP

*引导程序协议*，是一种 UDP 网络协议，可由网络客户端用于从服务器自动获取 IP 地址（可能还包括其他数据）。客户端使用客户端的 MAC 地址向服务器标识自己。服务器会维护预先配置的客户端设备 MAC 地址及关联 IP 地址表，从而向客户端发送其预先配置的 IP 地址。BOOTP 最初用于使无盘主机能够通过网络远程启动。BOOTP 进程分配一个无限租期的 IP 地址。BOOTP 服务使用 UDP 端口 67 和 68。

## C

### CAN

*控制器局域网协议 (ISO 11898)*，用于串行总线网络，旨在实现智能系统中智能设备（来自多家制造商）之间的互连，以处理实时的工业应用。CAN 多主站系统可通过实施广播消息传递和先进的诊断机制，确保高度的数据完整性。CAN 最初为汽车行业而开发，现在已应用于多种工业自动控制环境中。

### CANopen

CANopen 是一种开放工业标准通讯协议和设备配置文件规范。

### CFC

*连续功能图*（IEC61131-3 标准的扩展），是一种图形化编程语言，工作方式与流程图类似。通过添加简单的逻辑块（AND、OR 等等），即可使用此图形格式来表示程序中的每个功能或功能块。每个功能块的输入位于左侧，输出位于右侧。功能块输出可链接到其他功能块的输入，从而创建复合表达式。

## F

### FBD

*功能块图*，是面向图形的编程语言，与 IEC 61131-3 兼容。可用于一系列网络，其中每个网络包含一个框和连接线路的图形结构，该图形结构表示逻辑或算术表达式、功能块的调用、跳转或返回指令。

## G

### GVL

*全局变量列表*，用于管理每个应用程序 POU 中可用的全局变量。

## H

### HMI

*人机界面*，是工业设备采用的一种操作员界面（通常为图形界面）。

### HSC

*高速计数器*

## I

### IEC 61131-3

IEC 61131-3 是工业自动化设备（如控制器）采用的一种 *国际电工委员会* 标准。IEC 61131-3 针对控制器编程语言，并定义了两个图形编程语言和两个文本编程语言标准：

- **图形：**梯形图、功能块图
- **文本：**结构化文本、指令列表

### IL

以 *指令列表* 语言编写的程序，包括由控制器按顺序执行的一系列指令。每个指令包括一个行号、一个指令代码和一个操作数。（IL 符合 IEC 61131-3。）

---

## L

### LD

以*梯形图*语言编写的程序，它包括一个控制器程序指令图形表示，其中包含控制器按顺序执行的一系列梯级中的触点、线圈和块符号。符合 IEC 61131-3。

## M

### Modbus

Modbus 通信协议允许在连接到同一网络的多个设备之间进行通讯。

## N

### NEMA

*美国国家电气制造商协会*，负责发布各种类型的电气机箱的性能标准。NEMA 标准涉及防腐蚀、防雨淋和防淹没等性能。对于 IEC 成员国家，IEC 60529 标准还对机箱的入口防护等级进行了分类。

## P

### PLC

*可编程逻辑控制器*，是工业制造过程的“大脑”。它可以让过程自动化，而不是使用继电器控制系统。PLC 是适合在条件苛刻的工业环境中使用的计算机。

### PLI

*脉冲锁存输入*

### POU

*程序组织单元*，包括源代码变量声明和相应的指令集。POU 有助于简化软件程序、功能和功能块的模块化重用。经过声明后，POU 便可相互使用。SoMachine 编程需要使用 POU。

### PTO

*脉冲串输出*，用于控制，例如开放回路中的步进器电机。

**PWM**

*脉冲宽度调制*，用于调制脉冲信号长度的调节过程（例如用于温度控制的执行器）。对于此类信号，要使用晶体管输出。

**R**

**RTC**

*实时时钟*选件，在控制器断电后，该选件可以确保一定时间长度内的计时。

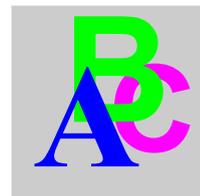
**S**

**SFC**

以*顺序功能图*语言编写的程序，可用于能被拆分为数个步骤的过程。SFC 包括具有关联操作的步骤、具有相关联逻辑条件的转换，以及步骤和转换之间的定向链接。（SFC 标准在 IEC 848 中定义。符合 IEC 61131-3。）

---

## 索引



- 
- BOOT\_PROJECT\_STATUS  
数据类型, 63
- DAY\_OF\_WEEK  
数据类型, 68
- ETH\_R  
系统变量, 22
- ETH\_R\_FRAME\_PROTOCOL  
数据类型, 73
- ETH\_R\_IP\_MODE  
数据类型, 72
- ETH\_R\_PORT\_DUPLEX\_STATUS  
数据类型, 74
- ETH\_R\_PORT\_LINK\_STATUS  
数据类型, 75
- ETH\_R\_PORT\_SPEED  
数据类型, 76
- ETH\_W  
系统变量, 24
- FIRMWARE\_VERSION  
数据类型, 62
- GetBatteryLevel  
功能, 27
- GetBootProjectStatus  
功能, 28
- GetEventsNumber  
功能, 29
- GetFirmwareVersion  
功能, 30
- GetHardwareVersion  
功能, 31
- GetLastStopCause  
功能, 32
- GetLastStopTime  
功能, 33
- GetLocalIOStatus  
功能, 34
- GetPlcFault  
功能, 37
- GetRightBusStatus  
功能, 38
- GetRunStopSwitch  
功能, 36
- GetSerialNumber  
功能, 42
- GetShortCutStatus  
功能, 43
- HOUR  
数据类型, 69
- InhibitBatLowLed  
功能, 49
- IsFirstMastColdCycle  
功能, 44
- IsFirstMastCycle  
功能, 45
- IsFirstMastWarmCycle  
功能, 47
- LOCAL\_IO\_GEN\_STATUS  
数据类型, 66
- LOCAL\_IO\_GET\_STATUS  
数据类型, 65
- MINUTE  
数据类型, 70
- PLC\_R  
系统变量, 15

## PLC\_R\_APPLICATION\_ERROR

数据类型, 56

## PLC\_R\_BOOT\_PROJECT\_STATUS

数据类型, 57

## PLC\_R\_IO\_STATUS

数据类型, 58

## PLC\_R\_STATUS

数据类型, 55

## PLC\_R\_STOP\_CAUSE

数据类型, 59

## PLC\_W

系统变量, 17

## PLC\_W\_COMMAND

数据类型, 60

## ResetEventsNumber

功能, 50

## RIGHTBUS\_GET\_STATUS

数据类型, 67

## SERIAL\_R

系统变量, 19

## SERIAL\_W

系统变量, 20

## SetRTCDrift

功能, 51

## STOP\_WHY

数据类型, 64

## 功能

GetBatteryLevel, 27

GetBootProjectStatus, 28

GetEventsNumber, 29

GetFirmwareVersion, 30

GetHardwareVersion, 31

GetLastStopCause, 32

GetLastStopTime, 33

GetLocalIOStatus, 34

GetPlcFault, 37

GetRightBusStatus, 38

GetRunStopSwitch, 36

GetSerialNumber, 42

GetShortCutStatus, 43

InhibitBatLowLed, 49

IsFirstMastColdCycle, 44

IsFirstMastCycle, 45

IsFirstMastWarmCycle, 47

ResetEventsNumber, 50

SetRTCDrift, 51

功能与功能块之间的差异, 80

如何通过 IL 语言使用功能或功能块, 81

如何通过 ST 语言使用功能或功能块, 84

## 数据类型

BOOT\_PROJECT\_STATUS, 63

DAY\_OF\_WEEK, 68

ETH\_R\_FRAME\_PROTOCOL, 73

ETH\_R\_IP\_MODE, 72

ETH\_R\_PORT\_DUPLEX\_STATUS, 74

ETH\_R\_PORT\_LINK\_STATUS, 75

ETH\_R\_PORT\_SPEED, 76

FIRMWARE\_VERSION, 62

HOUR, 69

LOCAL\_IO\_GEN\_STATUS, 66

LOCAL\_IO\_GET\_STATUS, 65

MINUTE, 70

PLC\_R\_APPLICATION\_ERROR, 56

PLC\_R\_BOOT\_PROJECT\_STATUS, 57

PLC\_R\_IO\_STATUS, 58

PLC\_R\_STATUS, 55

PLC\_R\_STOP\_CAUSE, 59

PLC\_W\_COMMAND, 60

RIGHTBUS\_GET\_STATUS, 67

STOP\_WHY, 64

## 系统变量

ETH\_R, 22

ETH\_W, 24

PLC\_R, 15

PLC\_W, 17

SERIAL\_R, 19

SERIAL\_W, 20

使用, 13

定义, 11

