

# Modicon M258 Logic Controller

系统功能和变量

Modicon M258 PLCSystem 库指南

04/2012

---

本文档中提供的信息包含有关此处所涉及产品之性能的一般说明和 / 或技术特性。本文档并非用于（也不代替）确定这些产品对于特定用户应用场合的适用性或可靠性。任何此类用户或集成者都有责任就相关特定应用场合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场合时，必须遵守有关的使用说明。

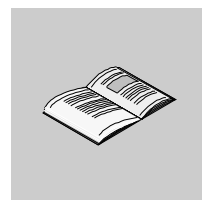
如果在我们的硬件产品上不正确地使用 Schneider Electric 软件或认可的软件，则可能导致人身伤害、损害或不正确的操作结果。

不遵守此信息可能导致人身伤害或设备损坏。

© 2012 Schneider Electric。保留所有权利。

---

# 目录



|  |           |
|--|-----------|
| 安全信息                                       | 7         |
| 关于本书                                       | 9         |
| <b>章 1 M258 系统变量</b>                       | <b>11</b> |
| 1.1 系统变量：定义和使用                             | 12        |
| 了解系统变量                                     | 13        |
| 使用系统变量                                     | 15        |
| 1.2 PLC_R 和 PLC_W 结构                       | 16        |
| PLC_R：控制器只读系统变量                            | 17        |
| PLC_W：控制器读 / 写系统变量                         | 21        |
| 1.3 SERIAL_R 和 SERIAL_W 结构                 | 22        |
| SERIAL_R[0..2]：串行线路只读系统变量                  | 23        |
| SERIAL_W[0..2]：串行线路读 / 写系统变量               | 24        |
| 1.4 ETH_R 和 ETH_W 结构                       | 25        |
| ETH_R：以太网端口只读系统变量                          | 26        |
| ETH_W：以太网端口读 / 写系统变量                       | 30        |
| 1.5 TM5_MODULE_R 结构                        | 31        |
| TM5_MODULE_R[1..254]：TM5 模块只读系统变量          | 31        |
| <b>章 2 M258 系统功能</b>                       | <b>33</b> |
| 2.1 M258 读取功能                              | 34        |
| DM72FGetImmediateInput：读取嵌入式专用 I/O 的输入     | 35        |
| getTM5Delay：无有效交换的 TM5 总线循环数               | 36        |
| IsFirstMastColdCycle：指示循环是否为第一个 MAST 冷启动循环 | 38        |
| IsFirstMastCycle：指示循环是否为第一个 MAST 循环        | 39        |
| IsFirstMastWarmCycle：指示循环是否为第一个 MAST 热启动循环 | 41        |
| 2.2 M258 写入功能                              | 42        |
| DM72F•SetImmediateOutput：写入嵌入式专用 I/O 的输出   | 43        |
| SetLEDBehaviour：决定 LED 的行为                 | 45        |
| SetRTCDrift：每周调整实时时钟                       | 47        |
| 2.3 M258 用户功能                              | 49        |
| DataFileCopy：复制文件命令                        | 50        |
| ExecuteScript：脚本命令                         | 52        |

|            |   |           |
|------------|---|-----------|
| <b>章 3</b> | <b>M258 PLCSystem 库数据类型</b>                 | <b>55</b> |
| 3.1        | PLC_R/W 系统变量数据类型                            | 56        |
|            | PLC_R_APPLICATION_ERROR: 检测到的应用程序错误状态代码     | 57        |
|            | PLC_R_BOOT_PROJECT_STATUS: 引导项目状态代码         | 58        |
|            | PLC_R_IO_STATUS: I/O 状态代码                   | 59        |
|            | PLC_R_STATUS: 控制器状态代码                       | 60        |
|            | PLC_R_STOP_CAUSE: 从“运行”状态向其他状态转换的原因代码       | 61        |
|            | PLC_R_TERMINAL_PORT_STATUS: 编程端口连接状态代码      | 62        |
|            | PLC_R_USB_HOST_STATUS: USB 主机端口连接状态代码       | 63        |
|            | PLC_W_COMMAND: 控制命令代码                       | 64        |
| 3.2        | DataFileCopy 系统变量数据类型                       | 65        |
|            | DataFileCopyError: 错误代码                     | 65        |
| 3.3        | ExecScript 系统变量数据类型                         | 66        |
|            | ExecuteScriptError: 错误代码                    | 66        |
| 3.4        | ETH_R/W 系统变量数据类型                            | 67        |
|            | ETH_R_IP_MODE: IP 地址源代码                     | 68        |
|            | ETH_R_FRAME_PROTOCOL: 帧传输协议代码               | 69        |
|            | ETH_R_PORT_DUPLEX_STATUS: 传输模式代码            | 70        |
|            | ETH_R_PORT_LINK_STATUS: 通讯链路方向代码            | 71        |
|            | ETH_R_PORT_SPEED: 以太网端口的通讯速度代码              | 72        |
|            | ETH_R_PORT_IP_STATUS: 以太网 TCP/IP 端口状态代码     | 73        |
|            | ETH_R_RUN_IDLE: 以太网 /IP 运行和闲置状态代码           | 74        |
| 3.5        | TM5_MODULE_R/W 系统变量数据类型                     | 75        |
|            | TM5_MODULE_STATE: TM5 扩展模块状态代码              | 75        |
| 3.6        | PROFIBUS_R 系统变量数据类型                         | 76        |
|            | PROFIBUS_R: Profibus 诊断系统变量                 | 76        |
| 3.7        | 系统功能数据类型                                    | 77        |
|            | LED_ID: SetLEDBehaviour 功能 LedId 参数代码       | 78        |
|            | LED_BHV: SetLEDBehaviour 功能 LedBhv 参数代码     | 79        |
|            | LED_BHV_ERROR: 检测到的 SetLEDBehaviour 功能错误代码  | 80        |
|            | LED_COLOR: SetLEDBehaviour 功能 LedColor 参数代码 | 81        |
|            | RTCSETDRIFT_ERROR: 检测到的 SetRTCDrift 功能错误代码  | 82        |
|            | DAY_OF_WEEK: SetRTCDrift 功能日期参数代码           | 83        |
|            | HOUR: SetRTCDrift 功能小时参数类型                  | 84        |
|            | MINUTE: SetRTCDrift 功能分钟参数类型                | 85        |
| <b>附录</b>  |   | <b>87</b> |

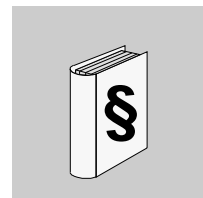
---

|                        |            |
|------------------------|------------|
| <b>附录 A 功能和功能块表示形式</b> | <b>89</b>  |
| 功能与功能块之间的差异            | 90         |
| 如何通过 IL 语言使用功能或功能块     | 91         |
| 如何通过 ST 语言使用功能或功能块     | 94         |
| <b>术语表</b>             | <b>97</b>  |
| <b>索引</b>              | <b>103</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”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，会导致人身伤害。



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”用于表示与人身伤害无关的危害。

**请注意**

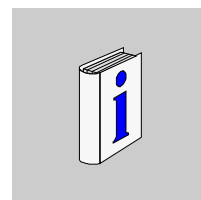
电气设备的安装、操作、维修和维护工作仅限于合格人员执行。对于使用本资料所引发的任何后果，Schneider Electric 概不负责。

专业人员是指掌握与电气设备的制造和操作及其安装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 M258 Logic Controller 中提供的系统功能和变量。M258 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 M258 PLCSystem 库的数据类型功能和变量。

需要了解以下基本知识：

- M258 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 V3.1 的发布进行了更新。

#### 相关的文件

| 文件名称                               | 参考编号  |
|------------------------------------|---|
| Modicon M258 Logic Controller 编程指南 | EIO0000000402（英语）、<br>EIO0000000403（法语）、<br>EIO0000000404（德语）、<br>EIO0000000405（西班牙语）、<br>EIO0000000406（意大利语）、<br>EIO0000000407（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)。

 **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)。

---

# M258 系统变量



---

## 概述

本章内容:

- 提供系统变量的简介 (参见第 12 页)
- 描述 M258 PLCSystem 库中包含的系统变量 (参见第 17 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            | 12 |
| 1.2 | PLC_R 和 PLC_W 结构       | 16 |
| 1.3 | SERIAL_R 和 SERIAL_W 结构 | 22 |
| 1.4 | ETH_R 和 ETH_W 结构       | 25 |
| 1.5 | TM5_MODULE_R 结构        | 31 |

---

## 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 M258 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 | 13 |
| 使用系统变量 | 15 |

## 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 TCP、Modbus 串行和 EtherNet/IP 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

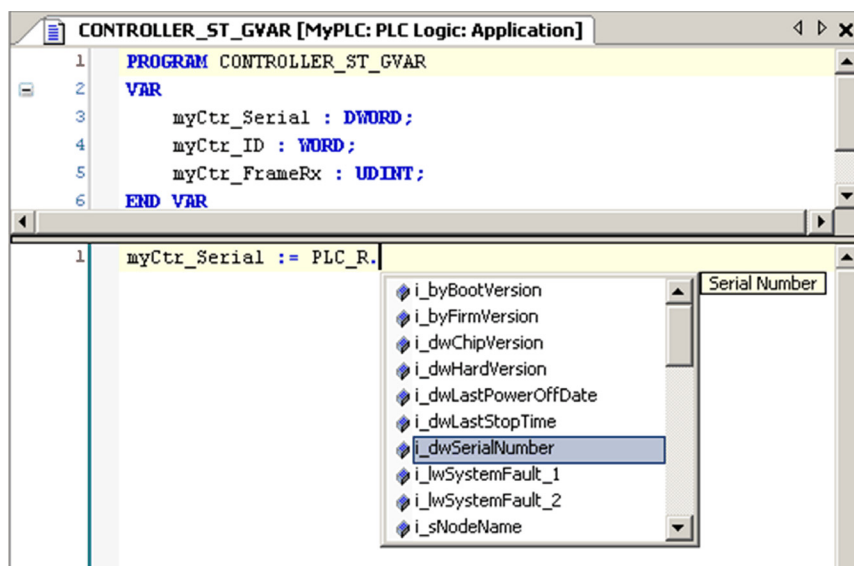
本主题介绍在 SoMachine 中编程和使用系统变量所需的步骤。

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    | 17 |
| PLC_W: 控制器读 / 写系统变量 | 21 |



## 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。<br>101A（十六进制）= Schneider Electric  |
| 60001 | i_wProductID            | WORD                                 | 控制器参考 ID。<br><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<br>[aa.bb.cc.dd]:<br>● i_byFirmVersion[0]= aa<br>● ...<br>● i_byFirmVersion[3]= dd  |
| 60006 | i_byBootVersion[0..3]   | ARRAY [0..3] OF BYTE                 | 控制器引导版本<br>[aa.bb.cc.dd]:<br>● i_byBootVersion[0]= aa<br>● ...<br>● 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 60 页)              | 控制器的状态。   |
| 60013 | i_wBootProjectStatus    | PLC_R_BOOT_PROJECT_STATUS (参见第 58 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 61 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 57 页)   | 上一次控制器例外的原因。  |

| %MW   | 变量名称                  | 类型                         | 注释   |
|-------|-----------------------|----------------------------|--|
| 60016 | i_lwSystemFault_1     | LWORD                      | 位域 FFFF FFFF FFFF FFFF (十六进制) 表示未检测到错误。某个位处于低电平表示检测到错误：<br><ul style="list-style-type: none"> <li>● 位 0 = 检测到嵌入式专用错误。有关诊断，请参见 i_wIOStatus1</li> <li>● 位 1 = 检测到 TM5 I/O 错误。有关诊断，请参见 i_wIOStatus2</li> <li>● 位 2 = 检测到以太网 0 错误</li> <li>● 位 3 = 检测到串行 0 错误</li> <li>● 位 4 = 检测到 CAN 0 错误</li> <li>● 位 5 = 检测到 CAN 1 错误</li> <li>● 位 6 = 检测到接口总线模块 0 错误</li> <li>● 位 7 = 检测到接口总线模块 1 错误</li> </ul> |
| 60020 | i_lwSystemFault_2     | LWORD                      | 未使用。   |
| 60024 | i_wIOStatus1          | PLC_R_IO_STATUS (参见第 59 页) | 嵌入式专用 I/O 状态。  |
| 60025 | i_wIOStatus2          | PLC_R_IO_STATUS (参见第 59 页) | TM5 I/O 状态。  |
| 60026 | i_wClockBatteryStatus | WORD                       | 实时时钟电池电量的状态：<br><ul style="list-style-type: none"> <li>● 0000 (十六进制) = 电池电量低</li> <li>● FFFF (十六进制) = 电池电量正常</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。   |

| %MW   | 变量名称                | 类型    | 注释   |
|-------|---------------------|-------|--|
| 60034 | i_dwAppliSignature4 | DWORD | 4 个 DWORD 签名（总共 16 个字节）的第 4 个 DWORD。<br>应用程序签名由软件在编译过程中生成。 |

|   |                       |  |   |
|---|-----------------------|--|---|
| 无 | i_sVendorName         | STRING (31)                              | 供应商的名称：“Schneider Electric”。  |
| 无 | i_sProductRef         | STRING (31)                              | 控制器的参考。   |
| 无 | i_sNodeName           | STRING (31)                              | SoMachine 网路上的节点名称  |
| 无 | i_dwiLastStopTime     | 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）。  |
| 无 | i_dwLastPowerOffDate  | DWORD                                    | 上次检测到电源关闭的日期和时间（以秒为单位，从 1970 年 1 月 1 日 00:00 开始计起）。<br><b>注意：</b> 可以使用功能 <code>SysTimeRtcConvertUtcToDate</code> 将此值转换为日期和时间。有关时间和日期转换的详细信息，请参阅系统库指南（参见 <i>SoMachine, 获取和设置实时时钟, SysTime 库指南</i> ）。 |
| 无 | i_uiEventsCounter     | UINT                                     | 自上次冷启动起，在为外部事件检测配置的输入上检测到的外部事件数。<br>可通过冷启动或 <code>PLC_W.q_wResetCounterEvent</code> 命令进行复位。   |
| 无 | i_wTerminalPortStatus | PLC_R_TERMINAL_PORT_STATUS<br>(参见第 62 页) | USB 编程端口 (USB Mini-B) 的状态。  |
| 无 | i_wUSBHostStatus      | PLC_R_USB_HOST_STATUS (参见第 63 页)         | USB 主机端口 (USB A) 的状态。   |
| 无 | i_wUsrFreeFileHdl     | WORD                                     | 可自由打开的文件句柄数。<br>文件句柄是系统在您打开文件时分配的资源。  |
| 无 | i_udiUsrFsTotalBytes  | UDINT                                    | 用户文件系统总存储器大小（以字节为单位）。<br>这是用于目录“/usr/”的闪存大小。  |
| 无 | i_udiUsrFsFreeBytes   | UDINT                                    | 用户文件系统可用存储器大小（以字节为单位）。  |

|   |                       |      |   |
|---|-----------------------|------|---|
| 无 | i_uiTM5BusState       | UINT | <p>TM5 总线状态位字段：</p> <ul style="list-style-type: none"> <li>● 位 0..3 = 未使用</li> <li>● 位 4 = 可用 TM5 总线驱动程序</li> <li>● 位 5 = 已找到 TM5 总线硬件</li> <li>● 位 6 = TM5 总线配置已成功完成</li> <li>● 位 7 = TM5 总线运行正常</li> <li>● 位 8 = 未使用</li> <li>● 位 9 = 在 TM5 总线配置过程中检测到错误</li> <li>● 位 10..15 = 未使用</li> </ul> |
| 无 | i_uiTM5SyncErrCnt     | UINT | 在 TM5 总线上检测到的无效同步帧数。通过 PLC_W.q_wResetTM5Counters 命令和关闭电源复位。   |
| 无 | i_uiTM5AsynErrCnt     | UINT | 在 TM5 总线上检测到的无效异步帧数。通过 PLC_W.q_wResetTM5Counters 命令和关闭电源复位。   |
| 无 | i_uiTM5BreakCnt       | UINT | 检测到的 TM5 总线复位次数。通过 PLC_W.q_wResetTM5Counters 命令和关闭电源复位。   |
| 无 | i_uiTM5TopoChangedCnt | UINT | TM5 总线拓扑的更改次数。通过 PLC_W.q_wResetTM5Counters 命令和关闭电源复位。   |
| 无 | i_uiTM5BusCycleCnt    | UINT | 自冷启动起的 TM5 总线循环数。通过 PLC_W.q_wResetTM5Counters 命令和关闭电源复位。  |
| 无 | i_wTM5BrokendownSlot  | WORD | <p>00..FE（十六进制）= 故障 TM5 模块的插槽号。<br/> FF（十六进制）= 所有正常工作的 TM5 模块报告。</p>  |

**注意：**无表示没有用于此系统变量的预定义 %MW 映射。

## 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   |
|-----|----------------------|--------------------------|--|
| 无   | q_wResetCounterEvent | WORD                     | 从 0 转换为 1 后, 将复位事件计数器 (PLC_R.i_uiEventsCounter)。若要再次复位该计数器, 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。   |
| 无   | q_wResetTM5counters  | WORD                     | 从 0 转换为 1 后, 将复位 PLC_R 结构化系统变量的所有 TM5 计数器 (从 PLC_R.i_uiTM5SyncErrCnt 到 PLC_R.i_uiTM5BusCycleCnt) 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。 |
| 无   | 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。   |
| 无   | q_wPLCControl        | PLC_W_COMMAND (参见第 64 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。  |

**注意:** 无表示没有用于此系统变量的预定义 %MW 映射。

---

## 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..2]: 串行线路只读系统变量    | 23 |
| SERIAL_W[0..2]: 串行线路读 / 写系统变量 | 24 |

## SERIAL\_R[0..2]: 串行线路只读系统变量

### 简介

SERIAL\_R 是包含 3 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。

对于 M258:

- Serial\_R[0] 引用嵌入式串行线路
- Serial\_R[1] 引用可选的串行线路 PCI 模块（如果已安装）
- Serial\_R[2] 引用可选的串行线路 PCI 模块（如果已安装）

### 变量结构

下表介绍 SERIAL\_R[0..2] 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..2]: 串行线路读 / 写系统变量

### 简介

SERIAL\_W 是包含 3 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。

M258:

- Serial\_W[0] 引用嵌入式串行线路
- Serial\_W[1] 引用 PCI 串行线路
- Serial\_W[2] 引用 PCI 串行线路

### 变量结构

下表介绍 SERIAL\_W[0..2] 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。<br>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    | 26 |
| ETH_W: 以太网端口读 / 写系统变量 | 30 |

## 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<br>[aaa.bbb.ccc.ddd]:<br>● i_byIPAddress[0]= aaa<br>● ...<br>● i_byIPAddress[3]= ddd   |
| 60052 | i_bySubNetMask[0..3]         | ARRAY [0..3] OF BYTE            | 子网掩码<br>[aaa.bbb.ccc.ddd]:<br>● i_bySub-netMask[0]= aaa<br>● ...<br>● i_bySub-netMask[3]= ddd  |
| 60054 | i_byGateway[0..3]            | ARRAY [0..3] OF BYTE            | 网关地址<br>[aaa.bbb.ccc.ddd]:<br>● i_byGateway[0]= aaa<br>● ...<br>● i_byGateway[3]= ddd  |
| 60056 | i_byMACAddress[0..5]         | ARRAY [0..5] OF BYTE            | MAC 地址<br>[aa.bb.cc.dd.ee.ff]:<br>● i_byMACAddress[0]= aa<br>● ...<br>● 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 68 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]:<br>● i_byFDRServerIPAddress[0]= aaa<br>● ...<br>● i_byFDRServerIPAddress[3]= ddd<br><br>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 69 页) | 为帧发送配置的以太网协议 (IEEE 802.3 或 Ethernet II)。   |

| %MW                      | 变量名称                          | 类型                                  | 注释  |
|--------------------------|-------------------------------|-------------------------------------|---|
| 无                        | i_udiFramesTransmittedOK      | UDINT                               | 已成功传输的帧数。在电源接通或使用复位命令<br>ETH_W.q_wResetCounter<br>后复位。              |
| 无                        | i_udiFramedReceivedOK         | UDINT                               | 已成功接收的帧数。在电源接通或使用复位命令<br>ETH_W.q_wResetCounter<br>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<br>ETH_W.q_wResetCounter<br>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<br>ETH_W.q_wResetCounter<br>后复位。        |
| 无                        | i_wPortALinkStatus            | ETH_R_PORT_LINK_STATUS (参见第 71 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 72 页)         | 以太网端口网速 (10Mb/s 或 100Mb/s)。   |
| 无                        | i_wPortADuplexStatus          | ETH_R_PORT_DUPLEX_STATUS (参见第 70 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<br>ETH_W.q_wResetCounter<br>后复位。 |
| 无                        | i_wPortAIPStatus              | ETH_R_PORT_IP_STATUS (参见第 73 页)     | 以太网 TCP/IP 端口栈状态。   |
| <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<br>ETH_W.q_wResetCounter<br>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<br>ETH_W.q_wResetCounter<br>后复位。       |

| %MW | 变量名称                    | 类型        | 注释   |
|-----|-------------------------|-----------|--|
| 无   | i_udiModbusErrorMessage | UDINT     | 传输和接收的已检测到的 Modbus 错误消息。<br>在电源接通或使用复位命令 ETH_W.q_wResetCounter 后复位。  |
| 无   | i_byMasterIpTimeouts    | 字节 (BYTE) | 以太网 Modbus TCP 主站超时事件计数器。<br>在电源接通或使用复位命令 ETH_W.q_wResetCounter 后复位。 |
| 无   | i_byMasterIpLost        | 字节 (BYTE) | 以太网 Modbus TCP 主站链路状态: 0 = 链路良好, 1 = 链路丢失。                           |

| 特定于 EtherNet/IP |                                   |       |  |
|-----------------|-----------------------------------|-------|--|
| 无               | i_udiETHIP_IOMessagingTransmitted | UDINT | 已传输的 EtherNet/IP 1 类帧的数量。<br>在电源接通或使用复位命令 ETH_W.q_wResetCounter 后复位。     |
| 无               | i_udiETHIP_IOMessagingReceived    | UDINT | 已接收的 EtherNet/IP 1 类帧的数量。<br>在电源接通或使用复位命令 ETH_W.q_wResetCounter 后复位。     |
| 无               | i_udiUCMM_Request                 | UDINT | 已接收的 EtherNet/IP 未连接消息的数量。<br>在电源接通或使用复位命令 ETH_W.q_wResetCounter 后复位。    |
| 无               | i_udiUCMM_Error                   | UDINT | 已接收的 EtherNet/IP 无效未连接消息的数量。<br>在电源接通或使用复位命令 ETH_W.q_wResetCounter 后复位。  |
| 无               | i_udiClass3_Request               | UDINT | 已接收的 EtherNet/IP 3 类请求的数量。<br>在电源接通或使用复位命令 ETH_W.q_wResetCounter 后复位。    |
| 无               | i_udiClass3_Error                 | UDINT | 已接收的 EtherNet/IP 无效 3 类请求的数量。<br>在电源接通或使用复位命令 ETH_W.q_wResetCounter 后复位。 |
| 无               | i_uiAssemblyInstanceInput         | UINT  | 输入组件实例编号。有关详细信息, 请参阅相应的控制器编程指南。  |

|   |                                |                              |  |
|---|--------------------------------|------------------------------|--|
| 无 | i_uiAssemblyInstanceInputSize  | UINT                         | 输入组件实例大小。有关详细信息，请参阅相应的控制器编程指南。                 |
| 无 | i_uiAssemblyInstanceOutput     | UINT                         | 输出组件实例编号。有关详细信息，请参阅相应的控制器编程指南。                 |
| 无 | i_uiAssemblyInstanceOutputSize | UINT                         | 输出组件实例大小。有关详细信息，请参阅相应的控制器编程指南。                 |
| 无 | i_uiETHIP_ConnectionTimeouts   | UINT                         | 连接超时次数。在电源接通或使用复位命令 ETH_W.q_wResetCounter 后复位。 |
| 无 | i_ucEipRunIdle                 | ETH_R_RUN_IDLE<br>(参见第 74 页) | EtherNet/IP 1 类连接的运行 (值 =1) / 空闲 (值 =0) 标志。    |

**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。<br>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。

## 1.5 TM5\_MODULE\_R 结构

### TM5\_MODULE\_R[1..254]: TM5 模块只读系统变量

#### 简介

TM5\_MODULE\_R 是包含 254 种 TM5\_MODULE\_R\_STRUCT 类型的数组。该数组的每个元素都返回对应 TM 模块的诊断系统变量。

对于 M258:

- TM5\_MODULE\_R[1] 引用 TM5 模块 1
- ...
- TM5\_MODULE\_R[254] 引用 TM5 模块 254

#### 变量结构

下表描述 TM5\_MODULE\_R[1..254] 系统变量的参数:

| %MW | 变量名称             | 类型                             | 注释                                     |
|-----|------------------|--------------------------------|--|
| 无   | i_wVendorID      | WORD                           | 目标的 TM5 模块供应商 ID。                      |
| 无   | i_wProductID     | WORD                           | 目标的 TM5 模块类型 ID。                       |
| 无   | i_dwSerialNumber | DWORD                          | TM5 模块序列号。                             |
| 无   | i_wFirmVersion   | WORD                           | TM5 模块固件版本。                            |
| 无   | i_wBootVersion   | WORD                           | TM5 模块引导版本。                            |
| 无   | i_wModuleState   | TM5_MODULE_STATE<br>(参见第 75 页) | 描述 TM5 模块的状态。当返回 TM5_ACTIVE 时, 模块正常运行。 |

**注意:** 无表示此系统变量没有预定义的 %MW 映射。





---

## M258 系统功能

# 2

---

### 概述

本章描述 M258 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 | M258 读取功能 | 34 |
| 2.2 | M258 写入功能 | 42 |
| 2.3 | M258 用户功能 | 49 |

---

## 2.1 M258 读取功能

---

### 概述

本节描述 M258 PLCSystem 库中包含的读取功能。

### 本节包含了哪些内容？

本节包含了以下主题：

| 主题  | 页  |
|---|----|
| DM72FGetImmediateInput: 读取嵌入式专用 I/O 的输入     | 35 |
| getTM5Delay: 无有效交换的 TM5 总线循环数               | 36 |
| IsFirstMastColdCycle: 指示循环是否为第一个 MAST 冷启动循环 | 38 |
| IsFirstMastCycle: 指示循环是否为第一个 MAST 循环        | 39 |
| IsFirstMastWarmCycle: 指示循环是否为第一个 MAST 热启动循环 | 41 |

## DM72FGetImmediateInput: 读取嵌入式专用 I/O 的输入

### 功能介绍

该功能适用于**嵌入式专用 I/O 功能块** DM72F0 和 DM72F1。它将返回输入的当前物理值，当前物理值可能与该输入的当前逻辑值不同。该输入的变量值在下一个总线循环前不会改变。

### 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 89 页）一章。

### I/O 变量介绍

下表介绍输入变量：

| 输入  | 类型  | 注释                                   |
|-----|-----|--------------------------------------|
| 功能块 | INT | 目标功能块：<br>● 0= DM72F0<br>● 1= DM72F1 |
| 输入  | INT | 功能块的目标输入。<br>0..6= DI0..DI6          |

下表介绍了输出变量：

| 输出                     | 类型   | 注释                                      |
|------------------------|------|---|
| DM72FGetImmediateInput | BOOL | 功能块 <block> 的输入 <Input> 值 = FALSE/TRUE. |

下表介绍了输入 / 输出变量：

| 输入 / 输出 | 类型                      | 注释                                     |
|---------|-------------------------|--|
| Error   | BOOL                    | FALSE= 运行正常。<br>TRUE= 检测到运行错误，功能返回无效值。 |
| ErrID   | IMMEDIATE_FUNC_ERR_TYPE | Error 为 TRUE 时检测到的运行错误代码。              |

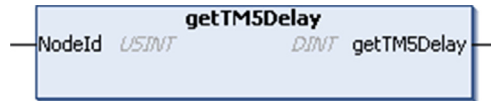
## getTM5Delay: 无有效交换的 TM5 总线循环数

### 功能介绍

此功能会返回没有与目标 TM5 模块进行有效交换的 TM5 总线循环数。

**注意:** 有关 TM5 模块诊断, 请参见系统变量 TM5\_MODULE\_R (参见第 31 页)。

### 图形表示形式

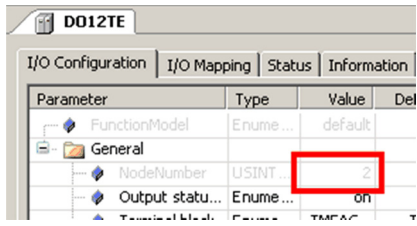


### IL 和 ST 表示形式

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 89 页) 一章。

### I/O 变量介绍

下表介绍了输入变量:

| 输入     | 类型   | 注释   |
|--------|------|--|
| NodeId | DINT | 目标 TM5 模块 <b>NodeNumber</b> (要获取 <b>NodeNumber</b> , 请在 TM5 模块编辑器中 <b>IO 配置屏幕</b> 下查看该值)。<br> |

下表介绍了输出变量:

| 输出          | 类型    | 注释  |
|-------------|-------|---|
| getTM5Delay | USINT | 该变量可以采用以下值: <ul style="list-style-type: none"> <li>● 0= 正常</li> <li>● [1..3]= 1 到 3 个循环没有有效交换]</li> <li>● -1= 没有有效交换的循环多于 3 个, 或参数无效</li> </ul> |

## 示例

下面的示例介绍如何获取第一个 TM5 模块的延迟:

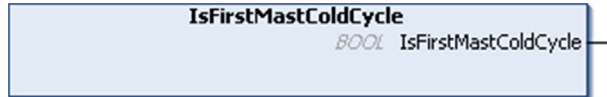
```
VAR delay :DINT; //Slot ID is 1 for the first TM5 module
slot_ID :USINT := 1; END_VAR
delay = getTM5Delay(slot_ID);
```

## 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 89 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 39 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 89 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;

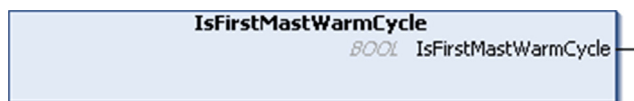


## IsFirstMastWarmCycle: 指示循环是否为第一个 MAST 热启动循环

### 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 89 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 39 页)。

---

## 2.2 M258 写入功能

---

### 概述

本节描述 M258 PLCSystem 库中包含的写入功能。

### 本节包含了哪些内容？

本节包含了以下主题：

| 主题  | 页  |
|---|----|
| DM72F•SetImmediateOutput: 写入嵌入式专用 I/O 的输出 | 43 |
| SetLEDBehaviour: 决定 LED 的行为               | 45 |
| SetRTCDrift: 每周调整实时时钟                     | 47 |

## DM72F•SetImmediateOutput: 写入嵌入式专用 I/O 的输出

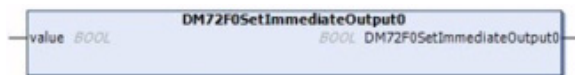
### 功能介绍

此功能用于设置嵌入式专用 I/O (DM72F) 的快速输出的当前物理值。

每个快速输出都有一个功能:

- DM72F0SetImmediateOutput0
- DM72F0SetImmediateOutput1
- DM72F1SetImmediateOutput0
- DM72F1SetImmediateOutput1

### 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 89 页) 一章。

### I/O 变量介绍

下表介绍了输入变量:

| 输入 | 类型   | 注释      |
|----|------|---------|
| 值  | BOOL | 要求的输出值。 |

下表介绍了输出变量:

| 输出   | 类型   | 注释              |
|--|------|-----------------|
| DM72Fb <sup>1</sup> SetImmediateOutputn <sup>2</sup>   | BOOL | TRUE = 设置物理输出值。 |
| (1) <b>b</b> 是目标功能块: <ul style="list-style-type: none"> <li>● 0= DM72F0</li> <li>● 1= DM72F1</li> </ul> (2) <b>n</b> 是功能块的目标输出: <ul style="list-style-type: none"> <li>● 0= DO0</li> <li>● 1= DO1</li> </ul> |      |                 |

**实现 DM72F•SetImmediateOutput•**

SoMachine 对于下列情况返回编译错误:

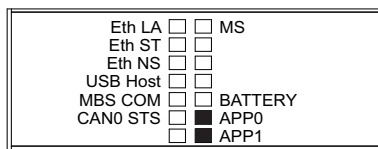
- 在多个任务中使用 DM72F•SetImmediateOutput• 功能。
- 在应用程序中使用与 DM72F•SetImmediateOutput• 关联的 %Q。
- 该输出已经专用于某一**嵌入式专用 I/O 功能块**功能（例如：PWM、频率发生器、编码器的反射输出、警报）。

## SetLEDBehaviour: 决定 LED 的行为

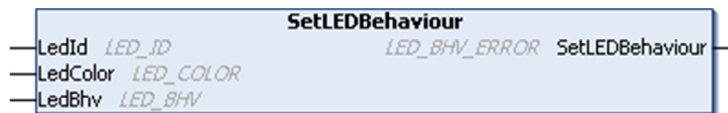
### 功能介绍

此功能控制诊断 LED APP0 和 APP1。

下图显示了位于前面板显示区的 LED:



### 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 89 页) 一章。

### I/O 变量介绍

下表介绍了输入参数:

| 输入       | 类型                   | 注释             |
|----------|----------------------|----------------|
| LedId    | LED_ID (参见第 78 页)    | 应用程序 LED 的 ID。 |
| LedColor | LED_COLOR (参见第 81 页) | 应用程序 LED 的颜色。  |
| LedBhv   | LED_BHV (参见第 79 页)   | 应用程序 LED 的模式。  |

下表介绍了输出变量:

| 输出              | 类型                       | 注释  |
|-----------------|--------------------------|---|
| SetLEDBehaviour | LED_BHV_ERROR (参见第 80 页) | 如果命令正常执行，则返回 NO_ERROR (十六进制的 00)，否则返回所检测到错误的 ID 代码。 |

## 示例

此示例介绍如何命令 LED APP0 亮起绿色指示灯：

```

VAR
    myLEDStatus : LED_BHV_ERROR;
    myLED : LED_ID := LED_0;
    myLEDColor : LED_COLOR := LED_GREEN;
    myLEDMode : LED_BHV := LED_ON;
END_VAR

myLEDStatus := SetLedBehaviour(myLED, myLEDColor, myLEDMode);

```

**注意：**LED 颜色是单独控制的并且可以混合，因此点亮新的前要关闭当前的颜色。  
下表显示 SetLedBehaviour 命令序列示例以及相关的 LED 行为：

| 步骤 | LedId | LedColor  | LedBhv  | 绿色闪烁模式 | 红色闪烁模式 |
|----|-------|-----------|---------|--------|--------|
| 1  | LED_0 | -         | -       | 关闭     | 关闭     |
| 2  | LED_0 | LED_GREEN | LED_ON  | 开启     | 关闭     |
| 3  | LED_0 | LED_GREEN | LED_OFF | 关闭     | 关闭     |
| 4  | LED_0 | LED_RED   | LED_ON  | 关闭     | 开启     |

## 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和时间对实时时钟增加或减去指定的秒数（小时：分钟）。

**注意：**必须至少调用一次 SetRTCDrift 功能需要对 进行编程，使其只在第一个 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 89 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 83 页) | 在一周中的哪一天执行。        |
| 时        | HOUR (参见第 84 页)        | 在几点钟进行更改。          |
| 分        | MINUTE (参见第 85 页)      | 在第几分钟进行更改。         |

**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 | RTCSETDRIFT_ERROR (参见第 82 页) | 如果命令正常运行，则返回 RTC_OK（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，每个星期二清晨 5:45 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
```



---

## 2.3 M258 用户功能

---

### 概述

本节介绍 M258 **PLCSystem** 库中的 **DataFileCopy** 功能和 **ExecuteScript** 功能。

### 本节包含了哪些内容？

本节包含了以下主题：

| 主题                           | 页  |
|------------------------------|----|
| <b>DataFileCopy</b> : 复制文件命令 | 50 |
| <b>ExecuteScript</b> : 脚本命令  | 52 |

## DataFileCopy: 复制文件命令

### 功能介绍

此功能可将存储器数据复制到文件，或将文件复制到存储器。文件位于内部文件系统或外部文件系统（USB 存储盘）。

DataFileCopy 功能块可以：

- 从具有一定格式的文件中读取数据，或
- 将存储器缓冲区中的数据复制到一定格式的文件中。有关详细信息，请参阅闪存结构（参见 *Modicon M258 Logic Controller, 编程指南*）。

### 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 89 页）一章。

### I/O 变量介绍

下表介绍了输入变量：

| 输入                | 类型     | 注释  |
|-------------------|--------|---|
| xExecute          | BOOL   | 在 rising edge, 开始功能块的执行。<br>在 falling edge, 在功能块的执行终止时复位功能块的输出。 |
| sFileName         | STRING | 不含扩展名的文件名称（将自动添加扩展名 .DTA）。<br>只使用 a...z、A...Z、0...9 字母数字字符。     |
| (1) 最大为 65534 字节。 |        |   |

| 输入                | 类型    | 注释  |
|-------------------|-------|---|
| xRead             | BOOL  | TRUE: 从文件复制到存储器。文件位置是内部文件系统中的 /usr/Dta。<br>FALSE: 从存储器复制到文件。文件位置是外部文件系统 (USB 存储盘) 中的 /usr/Dta。<br>只使用字母数字字符。        |
| xSecure           | BOOL  | TRUE: MAC 地址始终存储在文件中。只有具有相同 MAC 地址的控制器才能从该文件中读取。<br>FALSE: 采用相同类型存储器的其他 PLC 可以从该文件中读取。                              |
| iLocation         | INT   | 0: 文件位置是内部文件系统中的 /usr/Dta。<br>1: 文件位置是外部文件系统 (USB 存储盘) 中的 /usr/Dta。   |
| uiSize            | UINT  | 表示以字节为单位的大小 <sup>1</sup> 。<br>您只能使用符合 IEC 61131 (变量、数组、结构) 规范的变量地址, 例如:<br>变量: int;<br>uiSize := SIZEOF (Variable); |
| dwAdd             | DWORD | 表示存储器中的地址。<br>您只能使用符合 IEC 61131 (变量、数组、结构) 规范的变量地址, 例如:<br>变量: int;<br>dwAdd := ADR (Variable);                     |
| (1) 最大为 65534 字节。 |       |   |

## 警告

### 意外的设备操作

在将文件复制到存储器之前, 确保存储器位置的大小正确, 文件的类型正确。  
**如果不遵守这些说明, 将会导致死亡、严重伤害或设备损坏。**

下表介绍了输出变量:

| 输出     | 类型                           | 注释                       |
|--------|------------------------------|--------------------------|
| xDone  | BOOL                         | TRUE = 表示操作已成功完成。        |
| xBusy  | BOOL                         | TRUE = 功能块正在运行。          |
| xError | BOOL                         | TRUE = 表示检测到错误, 功能块中止操作。 |
| eError | DataFileCopyError (参见第 65 页) | 表示检测到的数据文件复制错误的类型。       |

**注意:** 如果您在文件写入的区域内写入存储器变量, 将会产生 CRC 错误。

## ExecuteScript: 脚本命令

### 功能介绍

此功能可以运行以下 USB 脚本命令：

- 下载
- 上载
- SetNodeName
- 删除
- 重新启动

使用与 USB 脚本相同的语法运行这些命令（区分大小写）。请参阅使用 USB 大容量存储生成脚本和文件（参见 *Modicon M258 Logic Controller, 编程指南*）。

### 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 89 页）一章。

### I/O 变量介绍

下表介绍了输入变量：

| 输入                | 类型     | 注释  |
|-------------------|--------|---|
| xExecute          | BOOL   | 在 rising edge, 开始功能块的执行。<br>在 falling edge, 在功能块的执行终止时复位功能块的输出。 |
| sCmd <sup>1</sup> | STRING | USB 脚本命令语法。   |

(1) 不允许并行命令执行：如果正在从其他功能块或从 USB 脚本执行某一命令，则该功能块不执行请求的命令。  
**注意：**从 USB 存储盘执行的 USB 脚本被视为在移除 USB 存储盘之前一直执行。

下表介绍了输出变量：

| 输出     | 类型                               | 注释                      |
|--------|----------------------------------|-------------------------|
| xDone  | BOOL                             | TRUE = 表示操作已成功完成。       |
| xBusy  | BOOL                             | TRUE = 功能块正在运行。         |
| xError | BOOL                             | TRUE = 表示检测到错误，功能块中止操作。 |
| eError | ExecuteScriptError<br>(参见第 66 页) | 表示检测到的执行脚本错误的类型。        |



---

## 概述

本章介绍 M258 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 M258 PLCSystem 库的**系统变量**（参见第 11 页）（PLC\_R、PLC\_W 等）使用。
- **系统功能数据类型**，由 M258 PLCSystem 库的**读 / 写系统功能**（参见第 33 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        | 56 |
| 3.2 | DataFileCopy 系统变量数据类型   | 65 |
| 3.3 | ExecScript 系统变量数据类型     | 66 |
| 3.4 | ETH_R/W 系统变量数据类型        | 67 |
| 3.5 | TM5_MODULE_R/W 系统变量数据类型 | 75 |
| 3.6 | PROFIBUS_R 系统变量数据类型     | 76 |
| 3.7 | 系统功能数据类型                | 77 |

## 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_APPLICATION_ERROR: 检测到的应用程序错误状态代码 | 57 |
| PLC_R_BOOT_PROJECT_STATUS: 引导项目状态代码     | 58 |
| PLC_R_IO_STATUS: I/O 状态代码               | 59 |
| PLC_R_STATUS: 控制器状态代码                   | 60 |
| PLC_R_STOP_CAUSE: 从“运行”状态向其他状态转换的原因代码   | 61 |
| PLC_R_TERMINAL_PORT_STATUS: 编程端口连接状态代码  | 62 |
| PLC_R_USB_HOST_STATUS: USB 主机端口连接状态代码   | 63 |
| PLC_W_COMMAND: 控制命令代码                   | 64 |



**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_DIVIDE_BY_ZERO         | 0102 (十六进制) | 检测到整数除 0。                       |
| 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_APP_ERR_DIVIDE_REAL_BY_ZERO    | 0152 (十六进制) | 检测到实数除 0。                       |
| PLC_R_APP_ERR_TOO_MANY_EVENT         | 4E20 (十六进制) | 外部 I/O 事件率过高。降低链接到事件任务的外部事件的频率。 |

**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) | 闪存中不存在引导项目。             |
| 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) | 闪存中的引导项目与 RAM 中加载的项目不同。 |
| 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。<br>如果重置模式为手动, 则当电源恢复时, PLC_R_IO_STATUS 被设置为 PLC_R_IO_SHORTCUT_FAULT。 |
| PLC_R_IO_POWER_SUPPLY_FAULT | 0004 (十六进制) | 检测到输入 / 输出电源错误。  |

## 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\_STOP\_CAUSE: 从“运行”状态向其他状态转换的原因代码

### 枚举类型介绍

PLC\_R\_STOP\_CAUSE 枚举数据类型包含下列值:

| 枚举器                                   | 值         | 注释  |
|---------------------------------------|-----------|---|
| PLC_R_STOP_REASON_UNKNOWN             | 00 (十六进制) | 未定义初始值或停止原因。  |
| PLC_R_STOP_REASON_HW_WATCHDOG         | 01 (十六进制) | 硬件看门狗后停止。   |
| PLC_R_STOP_REASON_RESET               | 02 (十六进制) | 复位后停止。  |
| PLC_R_STOP_REASON_EXCEPTION           | 03 (十六进制) | 例外后停止。  |
| PLC_R_STOP_REASON_USER                | 04 (十六进制) | 用户请求后停止。  |
| PLC_R_STOP_REASON_IECPROGRAM          | 05 (十六进制) | 程序命令请求后停止 (例如: 带有参数<br>PLC_W.q_wPLCControl:=PLC_W.COMMAND.PLC_W_STOP; 的控制命令)。 |
| PLC_R_STOP_REASON_DELETE              | 06 (十六进制) | 删除应用程序命令后停止。  |
| PLC_R_STOP_REASON_DEBUGGING           | 07 (十六进制) | 进入调试模式后停止。  |
| PLC_R_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。  |
| PLC_R_STOP_FROM_INPUT                 | 0B (十六进制) | 控制器输入要求停止。  |
| PLC_R_STOP_REASON_RETAIN_MISMATCH     | 0C (十六进制) | 重新启动过程中检查环境测试不成功后停止。  |
| PLC_R_STOP_REASON_BOOT_APPLI_MISMATCH | 0D (十六进制) | 重新启动之前比较引导应用程序和已在存储器中的应用程序不成功后停止。   |
| PLC_R_STOP_REASON_POWERFAIL           | 0E (十六进制) | 电源中断后停止。  |

有关控制器停止原因的详细信息, 请参阅控制器状态描述 (参见 *Modicon M258 Logic Controller, 编程指南*)。

**PLC\_R\_TERMINAL\_PORT\_STATUS: 编程端口连接状态代码****枚举类型描述**

PLC\_R\_TERMINAL\_PORT\_STATUS 枚举数据类型包含以下值:

| 枚举器                             | 值         | 注释            |
|---------------------------------|-----------|---------------|
| TERMINAL_NOT_CONNECTED          | 00 (十六进制) | 无 PC 连接到编程端口。 |
| TERMINAL_CONNECTION_IN_PROGRESS | 01 (十六进制) | 连接正在进行。       |
| TERMINAL_CONNECTED              | 02 (十六进制) | 编程端口已连接到 PC。  |
| TERMINAL_ERROR                  | 0F (十六进制) | 在连接过程中检测到错误。  |

## PLC\_R\_USB\_HOST\_STATUS: USB 主机端口连接状态代码

### 枚举类型描述

PLC\_R\_USB\_HOST\_STATUS 枚举数据类型包含以下值:

| 枚举器  | 值         | 注释                      |
|--|-----------|-------------------------|
| USB_NOT_CONNECTED  | 00 (十六进制) | 无设备 (存储盘) 连接到 USB 主机端口。 |
| USB_CONNECTION_IN_PROGRESS   | 01 (十六进制) | 正在连接 <sup>(1)</sup> 。   |
| USB_CONNECTED  | 02 (十六进制) | USB 主机端口已连接到设备 (存储盘)。   |
| USB_ERROR  | 0F (十六进制) | 在连接过程中检测到错误。            |
| <p><b>(1)</b> 满足下列要求的受支持 USB 存储盘:</p> <ul style="list-style-type: none"> <li>● 1 GB 最小容量</li> <li>● USB 2.0 规格或更低版本</li> <li>● FAT16 或 FAT32 文件系统</li> <li>● 必须设置卷标</li> </ul> |           |                         |

## 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 DataFileCopy 系统变量数据类型

### DataFileCopyError: 错误代码

#### 枚举类型描述

DataFileCopyError 枚举数据类型包含下列值:

| 枚举器                     | 值         | 说明                                   |
|-------------------------|-----------|--------------------------------------|
| ERR_NO_ERR              | 00 (十六进制) | 未检测到错误。                              |
| ERR_FILE_NOT_FOUND      | 01 (十六进制) | 文件不存在。                               |
| ERR_FILE_ACCESS_REFUSED | 02 (十六进制) | 无法打开文件。                              |
| ERR_INCORRECT_SIZE      | 03 (十六进制) | 请求大小与从文件中读取的大小不一致。                   |
| ERR_CRC_ERR             | 04 (十六进制) | CRC 不正确, 文件大概已损坏。                    |
| ERR_INCORRECT_MAC       | 05 (十六进制) | 试图读取文件的控制器的 MAC 地址与文件中包含的 MAC 地址不一致。 |

### 3.3 ExecScript 系统变量数据类型

#### ExecuteScriptError: 错误代码

##### 枚举类型描述

ExecuteScriptError 枚举数据类型包含下列值:

| 枚举器                          | 值         | 说明  |
|------------------------------|-----------|---|
| CMD_OK                       | 00 (十六进制) | 未检测到错误。   |
| ERR_CMD_UNKNOWN              | 01 (十六进制) | 不识别该命令。   |
| ERR_USB_KEY_MISSING          | 02 (十六进制) | USB 存储盘不存在。   |
| ERR_SEE_FWLOG                | 03 (十六进制) | 在命令执行过程中检测到错误, 参见 FwLog.txt。有关详细信息, 请参阅文件类型 (参见 <i>Modicon M258 Logic Controller, 编程指南</i> )。 |
| ERR_ONLY_ONE_COMMAND_ALLOWED | 04 (十六进制) | 试图同时执行多个脚本。   |
| CMD_BEING_EXECUTED           | 05 (十六进制) | 表示某一脚本正在执行。   |

## 3.4 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                 | 68 |
| ETH_R_FRAME_PROTOCOL: 帧传输协议代码           | 69 |
| ETH_R_PORT_DUPLEX_STATUS: 传输模式代码        | 70 |
| ETH_R_PORT_LINK_STATUS: 通讯链路方向代码        | 71 |
| ETH_R_PORT_SPEED: 以太网端口的通讯速度代码          | 72 |
| ETH_R_PORT_IP_STATUS: 以太网 TCP/IP 端口状态代码 | 73 |
| ETH_R_RUN_IDLE: 以太网 /IP 运行和闲置状态代码       | 74 |

## 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。 |



## ETH\_R\_PORT\_IP\_STATUS: 以太网 TCP/IP 端口状态代码

### 枚举类型描述

ETH\_R\_PORT\_IP\_STATUS 枚举数据类型包含以下值:

| 枚举器             | 值         | 注释                                    |
|-----------------|-----------|---------------------------------------|
| WAIT_FOR_PARAMS | 00 (十六进制) | 等待参数。                                 |
| WAIT_FOR_CONF   | 01 (十六进制) | 等待配置。                                 |
| DATA_EXCHANGE   | 02 (十六进制) | 数据交换准备就绪。                             |
| ETH_ERROR       | 03 (十六进制) | 检测到以太网 TCP/IP 端口错误 (电缆断开连接、无效配置... )。 |
| DUPLICATE_IP    | 04 (十六进制) | IP 地址已被其他设备使用。                        |

## ETH\_R\_RUN\_IDLE: 以太网 /IP 运行和闲置状态代码

### 枚举类型描述

ETH\_R\_RUN\_IDLE 枚举数据类型包含以下值:

| 枚举器  | 值         | 注释              |
|------|-----------|-----------------|
| IDLE | 00 (十六进制) | 以太网 /IP 连接闲置。   |
| 运行   | 01 (十六进制) | 以太网 /IP 连接正在运行。 |

## 3.5 TM5\_MODULE\_R/W 系统变量数据类型

### TM5\_MODULE\_STATE: TM5 扩展模块状态代码

#### 枚举类型描述

TM5\_MODULE\_STATE 枚举数据类型包含以下值:

| 枚举器          | 值         | 注释               |
|--------------|-----------|------------------|
| TM5_INACTIVE | 00 (十六进制) | 状态机处于非活动状态。      |
| TM5_BOOT     | 0A (十六进制) | 正在进行引导。          |
| TM5_FWDNLD   | 0B (十六进制) | 正在进行固件下载。        |
| TM5_PREOP    | 14 (十六进制) | 基本初始化。           |
| TM5_OPERATE  | 1E (十六进制) | 寄存器初始化。          |
| TM5_ACTIVE   | 64 (十六进制) | Modbus 通讯处于活动状态。 |
| TM5_ERROR    | C8 (十六进制) | 模块处于检测到错误状态。     |
| TM5_UNSUP    | C9 (十六进制) | 不支持 TM5 模块。      |
| TM5_NOCFG    | CA (十六进制) | 无可用配置。           |

## 3.6 PROFIBUS\_R 系统变量数据类型

### PROFIBUS\_R: Profibus 诊断系统变量

#### 说明

该表列出了 PROFIBUS\_R 系统变量 (PROFIBUS\_R\_STRUCT) 类型的参数:

| 变量名              | 类型    | 值   | 注释   |
|------------------|-------|---|--|
| i_wPNOIdentifier | WORD  | 0x0D73  | Profibus 从站标识号   |
| i_wBusAdr        | WORD  | 1...126   | Profibus 从站地址编号<br>保留地址 126。   |
| i_CommState      | UDINT | 0 - 未知<br>1 - 未配置<br>2 - 停止<br>3 - 空闲<br>4 - 运行 | Profibus 状态<br>运行模式: 循环 I/O 正在运行                                     |
| i_CommError      | UDINT | 0 - 无错误<br><>0 错误代码                             | Profibus 模块检测到通讯错误, 显示为错误代码 (参见 <i>Modicon TM5, PCI 模块配置, 编程指南</i> ) |
| i_ErrorCount     | UDINT | ≥0  | 检测到的通讯错误数量   |

## 3.7 系统功能数据类型

### 概述

本节描述 M258 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  |
|---|----|
| LED_ID: SetLEDBehaviour 功能 LedId 参数代码       | 78 |
| LED_BHV: SetLEDBehaviour 功能 LedBhv 参数代码     | 79 |
| LED_BHV_ERROR: 检测到的 SetLEDBehaviour 功能错误代码  | 80 |
| LED_COLOR: SetLEDBehaviour 功能 LedColor 参数代码 | 81 |
| RTCSETDRIFT_ERROR: 检测到的 SetRTCDrift 功能错误代码  | 82 |
| DAY_OF_WEEK: SetRTCDrift 功能日期参数代码           | 83 |
| HOUR: SetRTCDrift 功能小时参数类型                  | 84 |
| MINUTE: SetRTCDrift 功能分钟参数类型                | 85 |

**LED\_ID: SetLEDBehaviour 功能 LedId 参数代码****枚举类型介绍**

LED\_ID 枚举数据类型包含以下值：

| 枚举器   | 值         | 注释                  |
|-------|-----------|---------------------|
| LED_0 | 00 (十六进制) | 应用程序 LED APP0 的标识符。 |
| LED_1 | 01 (十六进制) | 应用程序 LED APP1 的标识符。 |

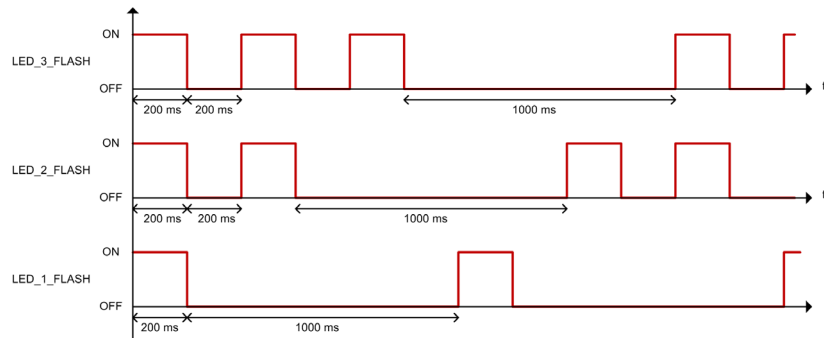
## LED\_BHV: SetLEDBehaviour 功能 LedBhv 参数代码

### 枚举类型描述

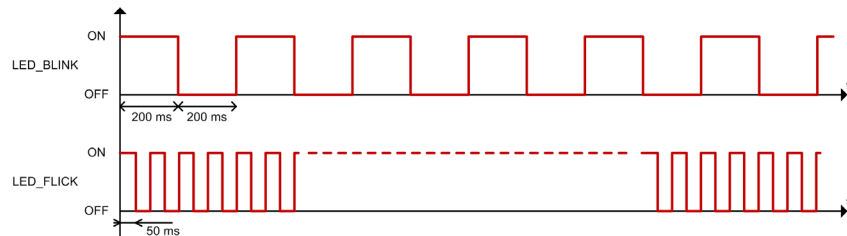
LED\_BHV 枚举数据类型包含以下值:

| 枚举器         | 值        | 注释                       |
|-------------|----------|--------------------------|
| LED_3_FLASH | -3 (十进制) | LED 在模式 3 下闪烁 (请参见下图)。   |
| LED_2_FLASH | -2 (十进制) | LED 在模式 2 下闪烁 (请参见下图)。   |
| LED_1_FLASH | -1 (十进制) | LED 在模式 1 下闪烁 (请参见下图)。   |
| LED_OFF     | 0 (十进制)  | LED 常灭。                  |
| LED_ON      | 1 (十进制)  | LED 常亮。                  |
| LED_BLINK   | 2 (十进制)  | LED 以 2.5 Hz 闪烁 (请参见下图)。 |
| LED_FLICK   | 3 (十进制)  | LED 以 10 Hz 闪烁 (请参见下图)。  |

下面的时钟图描述了应用程序 LED 闪烁模式 LED\_x\_FLASH:



下面的时钟图描述了应用程序 LED 闪烁模式 LED\_BLINK 和 LED\_FLICK:



## LED\_BHV\_ERROR: 检测到的 SetLEDBehaviour 功能错误代码

### 枚举类型介绍

LED\_BHV\_ERROR 枚举数据类型包含以下值:

| 枚举器            | 值         | 注释                          |
|----------------|-----------|-----------------------------|
| NO_ERROR       | 00 (十六进制) | 在没有检测到错误的情况下执行了 LED 行为设置功能。 |
| UNKNOWN_LED    | 01 (十六进制) | LED_ID 参数未知。                |
| UNKNOWN_COLOR  | 02 (十六进制) | LED_COLOR 参数未知。             |
| UNKNOWN_STATE  | 03 (十六进制) | 包含在 LED_BHV 参数中的 LED 状态未知。  |
| FIRMWARE_ERROR | 04 (十六进制) | FW 由于检测到错误而拒绝命令。            |



**LED\_COLOR: SetLEDBehaviour 功能 LedColor 参数代码****枚举类型介绍**

LED\_COLOR 枚举数据类型包含以下值：

| 枚举器       | 值         | 注释         |
|-----------|-----------|------------|
| LED_RED   | 00 (十六进制) | LED 颜色为红色。 |
| LED_GREEN | 01 (十六进制) | LED 颜色为绿色。 |

**RTCSETDRIFT\_ERROR: 检测到的 SetRTCDrift 功能错误代码****枚举类型描述**

RTCSETDRIFT\_ERROR 枚举数据类型包含以下值:

| 枚举器                | 值         | 注释                     |
|--------------------|-----------|------------------------|
| RTC_OK             | 00 (十六进制) | RTC 漂移配置正确。            |
| RTC_BAD_DAY        | 01 (十六进制) | 日期参数未知。                |
| RTC_BAD_HOUR       | 02 (十六进制) | 小时参数未知。                |
| RTC_BAD_MINUTE     | 03 (十六进制) | 分钟参数未知。                |
| RTC_BAD_DRIFT      | 04 (十六进制) | RTC 漂移参数超出范围。          |
| RTC_INTERNAL_ERROR | 05 (十六进制) | 由于内部检测到错误而拒绝 RTC 漂移设置。 |

**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 |

## **HOUR: 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**

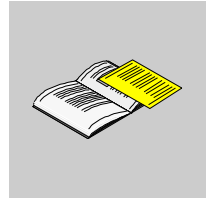
### **数据类型描述**

该数据类型包含从 0 到 59 的分钟值。



---

# 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        | 90 |
| 如何通过 IL 语言使用功能或功能块 | 91 |
| 如何通过 ST 语言使用功能或功能块 | 94 |

## 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。<br><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：<br><code>IsFirstMastCycle</code> |        |
| 带输入参数：<br><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：<br>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：<br>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> <hr/> <pre> 1  LD          myDrift    SetRTCDrift myDay                     myHour                     myMinute    ST          myDiag                     </pre> |

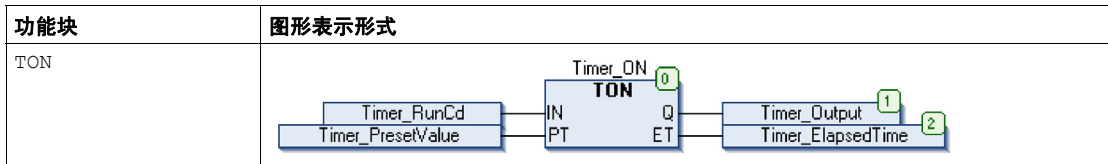
### 通过 IL 语言使用功能块

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 <b>指令列表</b> 语言打开 POU 或创建新 POU。<br><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。

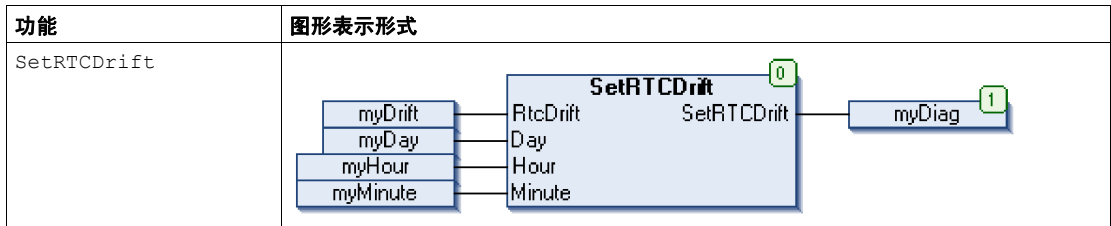
我们以功能 SetRTCDrift 和功能块 TON 为例演示实现的过程。

### 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。<br><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：<br>FunctionResult:= FunctionName(VarInput1, VarInput2,..VarInputx); |

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：

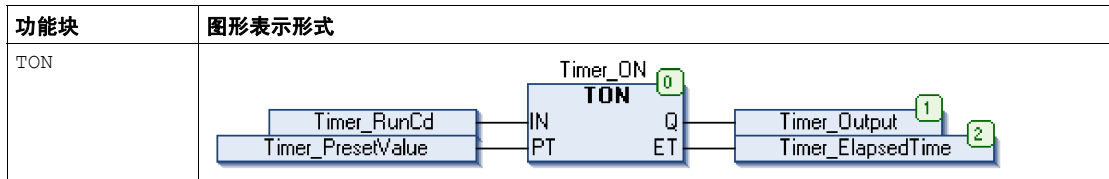
| 功能          | SoMachine POU ST 编辑器中的表示形式   |
|-------------|--|
| SetRTCDrift | <pre>PROGRAM MyProgram_ST VAR myDrift:SINT(-29..29) := 5; myDay:DAY_OF_WEEK := SUNDAY; myHour:HOUR := 12; myMinute:MINUTE; myRTCAdjust:RTCDRIFT_ERROR; END_VAR  myRTCAdjust:= SetRTCDrift(myDrift, myDay, myHour, myMinute);</pre> |

## 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 <b>结构化文本</b> 语言打开 POU 或创建新 POU。<br><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  | 创建功能块所需的输入和输出变量以及实例：<br><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：<br>FunctionBlock_InstanceName (Input1:=VarInput1,<br>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：

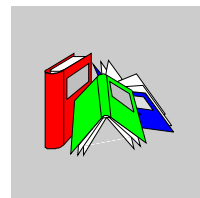
| 功能块 | SoMachine POU ST 编辑器中的表示形式  |
|-----|---|
| TON | <pre> 1  PROGRAM MyProgram_ST 2  VAR 3      Timer_ON: TON; // Function Block Instance 4      Timer_RunCd: BOOL; 5      Timer_PresetValue: TIME := T#5S; 6      Timer_Output: BOOL; 7      Timer_ElapsedTime: TIME; 8  END_VAR  1  Timer_ON( 2      IN:=Timer_RunCd, 3      PT:=Timer_PresetValue, 4      Q=&gt;Timer_Output, 5      ET=&gt;Timer_ElapsedTime); </pre> |





---

# 术语



---

## 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*。）

## 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。

## 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。

## 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。

## 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。

## 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*

## 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。）

## 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。

## 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。

## 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。

## 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。

**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。

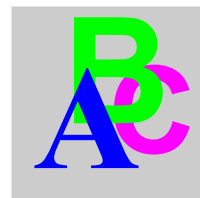
**SFC**

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



- 
- DataFileCopy  
功能, 50
- DataFileCopyError  
数据类型, 65
- DAY\_OF\_WEEK  
数据类型, 83
- DM72F•SetImmediateOutput•  
功能, 43
- DM72FGetImmediateInput  
功能, 35
- ETH\_R  
系统变量, 26
- ETH\_R\_FRAME\_PROTOCOL  
数据类型, 69
- ETH\_R\_IP\_MODE  
数据类型, 68
- ETH\_R\_PORT\_DUPLEX\_STATUS  
数据类型, 70
- ETH\_R\_PORT\_LINK\_STATUS  
数据类型, 71
- ETH\_R\_PORT\_SPEED  
数据类型, 72
- ETH\_W  
系统变量, 30
- ExecuteScript  
功能, 52
- ExecuteScriptError  
数据类型, 66
- getTM5Delay  
功能, 36
- HOUR  
数据类型, 84
- IsFirstMastColdCycle  
功能, 38
- IsFirstMastCycle  
功能, 39
- IsFirstMastWarmCycle  
功能, 41
- LED\_BHV  
数据类型, 79
- LED\_BHV\_ERROR  
数据类型, 80
- LED\_COLOR  
数据类型, 81
- LED\_ID  
数据类型, 78
- MINUTE  
数据类型, 85
- PLC\_R  
系统变量, 17
- PLC\_R\_APPLICATION\_ERROR  
数据类型, 57
- PLC\_R\_BOOT\_PROJECT\_STATUS  
数据类型, 58
- PLC\_R\_IO\_STATUS  
数据类型, 59
- PLC\_R\_STATUS  
数据类型, 60
- PLC\_R\_STOP\_CAUSE  
数据类型, 61
- PLC\_R\_TERMINAL\_PORT\_STATUS  
数据类型, 62
- PLC\_R\_USB\_HOST\_STATUS  
数据类型, 63

## PLC\_W

系统变量, 21

## PLC\_W\_COMMAND

数据类型, 64

## PROFIBUS\_R, 76

数据类型, 76

## RTCSETDRIFT\_ERROR

数据类型, 82

## SERIAL\_R

系统变量, 23

## SERIAL\_W

系统变量, 24

## SetLEDBehaviour

功能, 45

## SetRTCdrift

功能, 47

## TM5\_MODULE\_R

系统变量, 31

## TM5\_MODULE\_STATE

数据类型, 75

## 功能

DataFileCopy, 50

DM72F•SetImmediateOutput•, 43

DM72FGetImmediateInput, 35

ExecuteScript, 52

getTM5Delay, 36

IsFirstMastColdCycle, 38

IsFirstMastCycle, 39

IsFirstMastWarmCycle, 41

SetLEDBehaviour, 45

SetRTCdrift, 47

功能与功能块之间的差异, 90

如何通过 IL 语言使用功能或功能块, 91

如何通过 ST 语言使用功能或功能块, 94

## 数据类型

DataFileCopyError, 65

DAY\_OF\_WEEK, 83

ETH\_R\_FRAME\_PROTOCOL, 69

ETH\_R\_IP\_MODE, 68

ETH\_R\_PORT\_DUPLEX\_STATUS, 70

ETH\_R\_PORT\_IP\_STATUS, 73

ETH\_R\_PORT\_LINK\_STATUS, 71

ETH\_R\_PORT\_SPEED, 72

ETH\_R\_RUN\_IDLE, 74

ExecuteScriptError, 66

HOUR, 84

LED\_BHV, 79

LED\_BHV\_ERROR, 80

LED\_COLOR, 81

LED\_ID, 78

MINUTE, 85

PLC\_R\_APPLICATION\_ERROR, 57

PLC\_R\_BOOT\_PROJECT\_STATUS, 58

PLC\_R\_IO\_STATUS, 59

PLC\_R\_STATUS, 60

PLC\_R\_STOP\_CAUSE, 61

PLC\_R\_TERMINAL\_PORT\_STATUS,  
62

PLC\_R\_USB\_HOST\_STATUS, 63

PLC\_W\_COMMAND, 64

PROFIBUS\_R, 76

RTCSETDRIFT\_ERROR, 82

TM5\_MODULE\_STATE, 75

## 系统变量

ETH\_R, 26

ETH\_W, 30

PLC\_R, 17

PLC\_W, 21

SERIAL\_R, 23

SERIAL\_W, 24

TM5\_MODULE\_R, 31

使用, 15

定义, 13