


# Modicon LMC058 Motion Controller 系统功能和变量 LMC058 PLCSystem 库指南

04/2017

EIO0000000571.08

[www.schneider-electric.com](http://www.schneider-electric.com)

**Schneider**  
 Electric

---

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，只有制造商才能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。

未能使用 Schneider Electric 软件或认可的软件配合我们的硬件，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。

© 2017 Schneider Electric。保留所有权利。



<b>安全信息</b> .....	7
<b>关于本书</b> .....	9
<b>第1章 LMC058 系统变量</b> .....	13
1.1 系统变量：定义和使用 .....	14
了解系统变量 .....	15
使用系统变量 .....	17
1.2 PLC_R 和 PLC_W 结构 .....	18
PLC_R：控制器只读系统变量 .....	19
PLC_W：控制器读/写系统变量 .....	23
1.3 SERIAL_R 和 SERIAL_W 结构 .....	24
SERIAL_R[0..2]：串行线路只读系统变量 .....	25
SERIAL_W[0..2]：串行线路读/写系统变量 .....	26
1.4 ETH_R 和 ETH_W 结构 .....	27
ETH_R：以太网端口只读系统变量 .....	28
ETH_W：以太网端口读/写系统变量 .....	31
1.5 TM5_MODULE_R 结构 .....	32
TM5_MODULE_R[1..254]：TM5 模块只读系统变量 .....	32
1.6 PROFIBUS_R 系统变量数据类型 .....	33
PROFIBUS_R：Profibus 诊断系统变量 .....	33
<b>第2章 LMC058 系统功能</b> .....	35
2.1 LMC058 读取功能 .....	36
DM72FGetImmediateInput：读取嵌入式专用 I/O 的输入 .....	37
getTM5Delay：无有效交换的 TM5 总线循环数 .....	38
IsFirstMastColdCycle：指示此循环是否为第一个 MAST 冷启动循环 .....	40
IsFirstMastCycle：指示此循环是否为第一个 MAST 循环 .....	41
IsFirstMastWarmCycle：指示此循环是否为第一个 MAST 热启动循环 .....	43
2.2 LMC058 写入功能 .....	44
DM72F*SetImmediateOutput：写入嵌入式专用 I/O 的输出 .....	45
SetLEDBehaviour：确定 LED 的行为 .....	47
SetRTCDrift：每周调整实时时钟 .....	49
2.3 LMC058 用户功能 .....	51
DataFileCopy：复制文件命令 .....	52
ExecuteScript：运行脚本命令 .....	55

<b>第3章 LMC058 PLCSystem 库数据类型</b> .....	<b>57</b>
3.1 PLC_RW 系统变量数据类型.....	<b>58</b>
PLC_R_APPLICATION_ERROR : 检测到的应用程序错误状态代码 ..	<b>59</b>
PLC_R_BOOT_PROJECT_STATUS : 引导项目状态代码.....	<b>61</b>
PLC_R_IO_STATUS : I/O 状态代码 .....	<b>62</b>
PLC_R_STATUS : 控制器状态代码 .....	<b>63</b>
PLC_R_STOP_CAUSE : 从 RUN 状态向其他状态过度的原因代码.....	<b>64</b>
PLC_R_TERMINAL_PORT_STATUS : 编程端口连接状态代码 .....	<b>65</b>
PLC_R_USB_HOST_STATUS : USB 主机端口连接状态代码.....	<b>66</b>
PLC_W_COMMAND : 控制命令代码 .....	<b>67</b>
3.2 DataFileCopy 系统变量数据类型 .....	<b>68</b>
DataFileCopyError : 检测到的错误代码.....	<b>68</b>
3.3 ExecScript 系统变量数据类型 .....	<b>69</b>
ExecuteScriptError : 检测到的错误代码.....	<b>69</b>
3.4 ETH_RW 系统变量数据类型 .....	<b>70</b>
ETH_R_IP_MODE : IP 地址源代码 .....	<b>71</b>
ETH_R_FRAME_PROTOCOL : 帧传输协议代码 .....	<b>72</b>
ETH_R_PORT_DUPLEX_STATUS : 传输模式代码 .....	<b>73</b>
ETH_R_PORT_LINK_STATUS : 通讯链路状态代码 .....	<b>74</b>
ETH_R_PORT_SPEED : 以太网端口的通讯速度代码.....	<b>75</b>
ETH_R_PORT_IP_STATUS : 以太网 TCP/IP 端口状态代码.....	<b>76</b>
ETH_R_RUN_IDLE : Ethernet/IP 运行和空闲状态代码.....	<b>77</b>
3.5 TM5_MODULE_R/W 系统变量数据类型 .....	<b>78</b>
TM5_MODULE_STATE : TM5 扩展模块状态代码.....	<b>78</b>
3.6 系统功能数据类型 .....	<b>79</b>
LED_ID : SetLEDBehaviour 功能 LedId 参数代码.....	<b>80</b>
LED_BHV : SetLEDBehaviour 功能 LedBhv 参数代码 .....	<b>81</b>
LED_BHV_ERROR : 检测到的 SetLEDBehaviour 功能错误代码 .....	<b>82</b>
LED_COLOR : SetLEDBehaviour 功能 LedColor 参数代码 .....	<b>83</b>
RTCSETDRIFT_ERROR : SetRTCDrift 功能检测到的错误代码 .....	<b>84</b>
DAY_OF_WEEK : SetRTCDrift 功能日期参数代码 .....	<b>85</b>
HOUR : SetRTCDrift 功能小时参数类型 .....	<b>86</b>
MINUTE : SetRTCDrift 功能分钟参数类型 .....	<b>87</b>
<b>附录</b> .....	<b>89</b>

---

<b>附录 A</b>	<b>功能和功能块表示形式</b> .....	<b>91</b>
	功能与功能块的区别 .....	<b>92</b>
	如何通过 IL 语言使用功能或功能块 .....	<b>93</b>
	如何通过 ST 语言使用功能或功能块 .....	<b>96</b>
<b>术语表</b>	.....	<b>99</b>
<b>索引</b>	.....	<b>105</b>





## 重要信息

### 声明

在尝试安装、操作、维修或维护设备之前，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”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

危險表示若不加以避免，将会导致严重人身伤害甚至死亡的危險情况。

## 警告

警告表示若不加以避免，可能会导致严重人身伤害甚至死亡的危險情况。

## 小心

小心表示若不加以避免，可能会导致轻微或中度人身伤害的危險情况。

## 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。Schneider Electric 不承担由于使用本资料所引起的任何后果。

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 LMC058 Motion Controller 中提供的系统功能和变量。LMC058 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 LMC058 PLCSystem 库的数据类型功能和变量。

需要了解以下基本知识：

- LMC058、
- FBD、LD、ST、IL 或 CFC 语言的编程
- 系统变量 ( 全局变量 )

### 有效性说明

本文档已随 SoMachine V4.3 的发布进行了更新。

### 相关的文件

文件名称	参考编号
Modicon LMC058 Logic Controller - 编程指南	<a href="#">EIO0000000408 (Eng)</a> <a href="#">EIO0000000409 (Fre)</a> <a href="#">EIO0000000410 (Ger)</a> <a href="#">EIO0000000411 (Spa)</a> <a href="#">EIO0000000412 (Ita)</a> <a href="#">EIO0000000413 (Chs)</a>

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：<http://www.schneider-electric.com/en/download>

## 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。这些关键控制功能包括紧急停止、越程停止、断电重启以及类似的安全措施。
- 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路失效问题加以考虑。
- 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。

**不遵循上述说明可能导致人员伤亡或设备损坏。**

<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。

**不遵循上述说明可能导致人员伤亡或设备损坏。**

## 摘自标准的术语

本手册中的或者出现在产品自身中/上的技术术语、术语、符号和相应描述基本上均源自国际标准的条款或定义。

在功能安全系统、驱动器和一般自动化领域，这可能包括但不限于 *安全、安全功能、安全状态、故障、故障复位、失灵、失效、错误、错误消息、危险* 等词语。

这些标准包括：

标准	描述
EN 61131-2:2007	编程控制器，第 2 部分：设备要求和测试。
ISO 13849-1:2008	机器安全：控制系统的安全相关部分。设计通则。
EN 61496-1:2013	机械安全：电子感应式防护设备。第 1 部分：一般要求和测试。
ISO 12100:2010	机械安全 - 设计的一般原则 - 风险评估和风险抑制
EN 60204-1:2006	机械安全 - 电气机械设备 - 第 1 部分：一般要求

标准	描述
EN 1088:2008 ISO 14119:2013	机械安全 - 与防护设备关联的联锁设备 - 设计和选择原则
ISO 13850:2006	机械安全 - 紧急停止 - 设计原则
EN/IEC 62061:2005	机械安全 - 安全相关的电气、电子和可编程电子控制系统的功能性安全
IEC 61508-1:2010	电气/电子/可编程电子安全相关系统的功能性安全：一般要求。
IEC 61508-2:2010	电气/电子/可编程电子安全相关系统的功能性安全：电气/电子/可编程电子安全相关系统的要求。
IEC 61508-3:2010	电气/电子/可编程电子安全相关系统的功能性安全：软件要求。
IEC 61784-3:2008	用于测量和控制的数字数据通讯：功能性安全现场总线。
2006/42/EC	机械指令
2014/30/EU	电磁兼容性规程
2014/35/EU	低电压规程

此外，本文中所用的名词可能是被无意中使用，因为它们是从其他标准中衍生出来的，如：

标准	描述
IEC 60034 系列	旋转电机
IEC 61800 系列	可调速电力驱动系统
IEC 61158 系列	用于测量和控制的数字数据通讯：用于工业控制系统的现场总线

最后，*操作区*一词可结合特定危险的描述一起使用，其定义相当于 *机器指令()* 和 *:2010* 中的 *2006/42/EC* 风险区 *ISO 12100* 或 *危险区*。

**注意：**对于当前文档中引用的特定产品，上述标准可能适用，也可能不适用。若要了解与适用于此处所述产品的各项标准有关的更多信息，请参阅这些产品参考的特性表。



---

# 第1章

## LMC058 系统变量

---

### 概述

本章内容：

- 提供系统变量的简介 (参见第 14 页)
- 描述 (参见第 19 页) LMC058 PLC 系统库中包含的系统变量

### 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	14
1.2	PLC_R 和 PLC_W 结构	18
1.3	SERIAL_R 和 SERIAL_W 结构	24
1.4	ETH_R 和 ETH_W 结构	27
1.5	TM5_MODULE_R 结构	32
1.6	PROFIBUS_R 系统变量数据类型	33

# 第1.1节

## 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 LMC058 Motion 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	15
使用系统变量	17

## 了解系统变量

### 简介

本节介绍实现系统变量的方式。系统变量：

- 使您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。
- 是符合 IEC 61131-3 定义和命名约定的结构化变量。您可以通过 IEC 符号名称 PLC\_GVL 访问系统变量。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）。
- 会自动声明为全局变量。这些变量的作用范围是整个系统，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。

### 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。

访问系统变量时，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完全限定名称为 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用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：

- 定位变量
- 非定位变量

定位变量：

- 在静态 %MW 区域中具有固定位置：
  - %MW60000 到 %MW60199，用于只读系统变量
  - %MW62000 到 %MW62199，用于读/写系统变量
- 可在 RUNNING 和 STOPPED 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 约定在 SoMachine 程序中使用。可以直接访问从 0 到 59999 的 %MW 地址。大于此地址的地址将被 SoMachine 视为超出范围，只能通过 structure\_name.component\_name 约定进行访问。

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 RUNNING 和 STOPPED 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 约定在 SoMachine 程序中使用。可以直接访问从 0 到 59999 的 %MW 地址。大于此地址的地址将被 SoMachine 视为超出范围，只能通过 structure\_name.component\_name 约定进行访问。



## 使用系统变量

### 简介

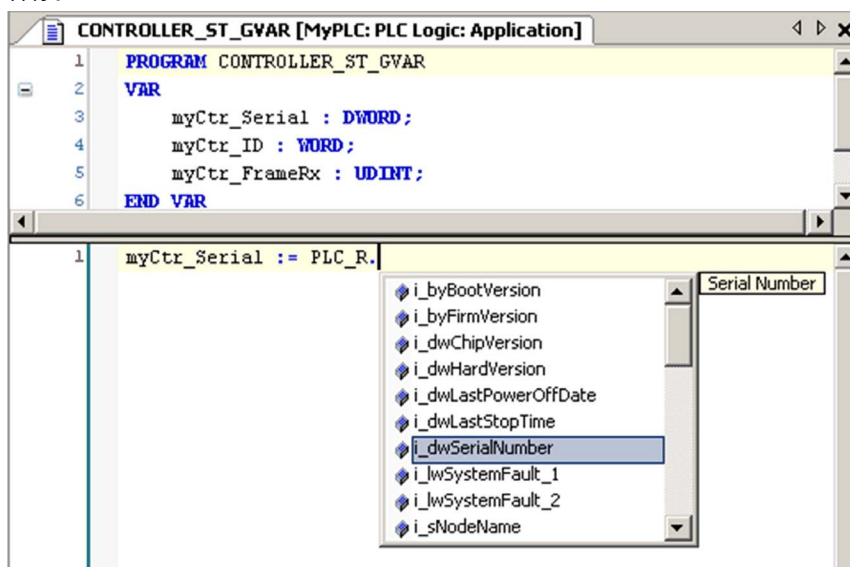
本节介绍在 SoMachine 中编程和使用系统变量所需的步骤。

系统变量是全局范围的，您可以在应用程序的所有程序组织单元 (POU) 中使用它们。

系统变量无需在全局变量列表 (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

SoMachine 具有自动完成功能。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。系统变量显示在**输入助手**中。您可以选择所需的变量或手动输入完整名称。



**注意：** 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;

```

## 第1.2节

### PLC\_R 和 PLC\_W 结构

#### 概述

本节列出 PLC\_R 和 PLC\_W 结构中包括的不同系统变量并加以说明。

#### 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	19
PLC_W：控制器读/写系统变量	23

## PLC\_R : 控制器只读系统变量

### 变量结构

下表对 PLC\_R 系统变量 ( PLC\_R\_STRUCT 类型 ) 的参数进行了描述 :

Modbus 地址 <sup>(1)</sup>	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] : <ul style="list-style-type: none"> <li>● i_byFirmVersion[0]= aa</li> <li>● ...</li> <li>● i_byFirmVersion[3]= dd</li> </ul>
60006	i_byBootVersion[0..3]	ARRAY[0..3] OF BYTE	控制器引导版本 [aa. bb. cc. dd] : <ul style="list-style-type: none"> <li>● i_byBootVersion[0]= aa</li> <li>● ...</li> <li>● i_byBootVersion[3]= dd</li> </ul>
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 63 页)	控制器的状态。
60013	i_wBootProjectStatus	PLC_R_BOOT_PROJECT_STATUS (参见第 61 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 64 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 59 页)	上一次控制器异常的原因。

Modbus 地址 <sup>(1)</sup>	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： <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_wl0Status1</li> <li>● 位 1 = 检测到 TM5 I/O 错误有关诊断，请参见 i_wl0Status2</li> <li>● 位 2 = 处于没有主站的 BOOTP 或 DHCP 中时未检测到以太网 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> <b>注意：</b> 电源重置后，SystemFault 可能会在多个循环中处于错误状态。
60020	i_lwSystemFault_2	LWORD	未使用。
60024	i_wl0Status1	PLC_R_IO_STATUS (参见第 62 页)	嵌入式专用 I/O 状态。
60025	i_wl0Status2	PLC_R_IO_STATUS (参见第 62 页)	TM5 I/O 状态。
60026	i_wClockBatterystatus	WORD	实时时钟电池电量的状态： <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。
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。
(1) 无法通过应用程序进行访问。			

无	i_sVendorName	STRING (31)	供应商名称：“Schneider Electric”。
无	i_sProductRef	STRING (31)	控制器的参考号。
无	i_sNodeName	STRING (99)	SoMachine 网络上的节点名称。
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 UTC 00:00 开始计起）。 <b>注意：</b> 使用功能 SysTimeRtcConvertUtcToDate 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 SoMachine, 获取和设置实时时钟, SysTimeRtc 和 SysTimeCore 库指南）。
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。 可通过冷启动或 PLC_W.q_wResetCounterEvent 命令进行复位。
无	i_wTerminalPortStatus	PLC_R_TERMINAL_PORT_STATUS (参见第 65 页)	USB 编程端口 (USB Mini-B) 的状态。
无	i_wUSBHostStatus	PLC_R_USB_HOST_STATUS (参见第 66 页)	USB 主机端口 (USB A) 的状态。
无	i_wUsrFreeFileHdl	WORD	可用的文件句柄数。 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）。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	TM5 总线状态位字段： <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_wTM5BroktdownSlot	WORD	00..FE (十六进制) = 故障 TM5 模块的插槽号。 FF (十六进制) = 所有 TM5 模块报告它们正常工作。

**注意：** 无表示没有用于此系统变量的预定义 Modbus 地址映射。

## PLC\_W : 控制器读/写系统变量

### 变量结构

下表对 PLC\_W 系统变量 ( PLC\_W\_STRUCT 类型 ) 的参数进行了描述 :

%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)。要再次复位计数器, 需要先向此变量写入 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 67 页)	当系统变量 PLC_R.q_uiOpenPLCControl 值从 0 转到 6699 后, 将执行控制器 RUN / STOP 命令。
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) 要再次复位这些计数器, 需要先向此变量写入 0, 之后才能再次进行从 0 到 1 的转换。

**注意 :** n/a 表示没有用于此系统变量的预定义 %MW 映射。

## 第1.3节

### SERIAL\_R 和 SERIAL\_W 结构

---

#### 概述

本节列出 SERIAL\_R 和 SERIAL\_W 结构中包括的不同系统变量并加以说明。

#### 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	25
SERIAL_W[0..2]：串行线路读/写系统变量	26



## SERIAL\_R[0...2] : 串行线路只读系统变量

### 简介

对于 LMC058 Motion Controller:

- Serial\_R[0] 指嵌入式串行线路
- Serial\_R[1] 指可选串行线路 PCI 模块 ( 如果已安装 )
- Serial\_R[2] 指可选串行线路 PCI 模块 ( 如果已安装 )

### 变量结构

下表对 SERIAL\_R[0...2] 系统变量的参数进行了描述 :

%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时的已接收帧数。
无	i_udiRX_MessagesError	UDINT	检测到错误 ( 校验和、校验位 ) 时的已接收帧数。
<b>特定于 Modbus</b>			
无	i_uiSlaveExceptionCount	UINT	逻辑控制器返回的 Modbus 异常响应数。
无	i_udiSlaveMsgCount	UINT	从主站接收并发送到逻辑控制器的消息数。
无	i_uiSlaveNoRespCount	UINT	逻辑控制器接收的 Modbus 广播请求数。
无	i_uiSlaveNakCount	UINT	未使用
无	i_uiSlaveBusyCount	UINT	未使用
无	i_uiCharOvrrunCount	UINT	字符溢出数。
无表示没有用于此系统变量的预定义 %MW 映射。 未使用表示系统未维护该变量，并且如果该变量的值非零，应被视为额外变量。			

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 请求功能代码数字 8 发出复位命令。

## SERIAL\_W[0...2] : 串行线路读/写系统变量

### 简介

对于 LMC058 Motion Controller:

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

### 变量结构

下表对 SERIAL\_W[0...2] 系统变量的参数进行了描述：

%MW	变量名称	类型	注释
无	q_wResetCounter	WORD	从 0 转换为 1 后，将复位所有计数器。 要再次复位这些计数器，需要先向此变量写入 0，之后才能再次进行从 0 到 1 的转换。

**注意：** n/a 表示没有用于此系统变量的预定义 %MW 映射。

## 第1.4节

### ETH\_R 和 ETH\_W 结构

#### 概述

本节列出 ETH\_R 和 ETH\_W 结构中包括的不同系统变量并加以说明。

#### 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	28
ETH_W：以太网端口读/写系统变量	31

## ETH\_R : 以太网端口只读系统变量

### 变量结构

下表对 ETH\_R 系统变量 ( ETH\_R\_STRUCT 类型 ) 的参数进行了描述 :

%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] : <ul style="list-style-type: none"> <li>● i_byIPAddress[0]= aaa</li> <li>● ...</li> <li>● i_byIPAddress[3]= ddd</li> </ul>
60052	i_bySubNetMask[0..3]	ARRAY [0..3] OF BYTE	子网掩码 [aaa.bbb.ccc.ddd] : <ul style="list-style-type: none"> <li>● i_bySub-netMask[0]= aaa</li> <li>● ...</li> <li>● i_bySub-netMask[3]= ddd</li> </ul>
60054	i_byGateway[0..3]	ARRAY [0..3] OF BYTE	网关地址 [aaa.bbb.ccc.ddd] : <ul style="list-style-type: none"> <li>● i_byGateway[0]= aaa</li> <li>● ...</li> <li>● i_byGateway[3]= ddd</li> </ul>
60056	i_byMACAddress[0..5]	ARRAY [0..5] OF BYTE	MAC 地址 [aa.bb.cc.dd.ee.ff] : <ul style="list-style-type: none"> <li>● i_byMACAddress[0]= aa</li> <li>● ...</li> <li>● i_byMACAddress[5]= ff</li> </ul>
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 71 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] : <ul style="list-style-type: none"> <li>● i_byFDRServerIPAddress[0]= aaa</li> <li>● ...</li> <li>● i_byFDRServerIPAddress[3]= ddd</li> </ul> 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 72 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。
<b>无</b> 表示没有用于此系统变量的预定义 %MW 映射。			

%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 74 页)	以太网端口链路 ( 0 = 无链路, 1 = 链路连接到其他以太网设备 )。
无	i_wPortASpeed	ETH_R_PORT_SPEED (参见第 75 页)	以太网端口网速 ( 10Mb/s 或 100Mb/s )。
无	i_wPortADuplexStatus	ETH_R_PORT_DUPLEX_STATUS (参见第 73 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。
无	i_wPortAlpStatus	ETH_R_PORT_IP_STATUS (参见第 76 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 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。
无	i_byMasterIpTimeouts	BYTE	以太网 Modbus TCP 主站超时事件计数器。在电源接通或使用复位命令 ETH_W.q_wResetCounter 后复位。
无	i_byMasterIpLost	BYTE	以太网 Modbus TCP 主站链路状态 : 0 = 链路正常, 1 = 链路丢失。
<b>无表示没有用于此系统变量的预定义 %MW 映射。</b>			

%MW	变量名称	类型	注释
<b>特定于 EtherNet/IP</b>			
无	i_udiETHIP_IOMessagingTransmitted	UDINT	已传输的 EtherNet/IP 1 类帧的数量。 在电源接通或使用复位命令 ETH_W.q_wResetCounter 后复位。
无	i_udiETHIP_IOMessagingReceived	UDINT	已接收的 EtherNet/IP 1 类帧的数量。 在电源接通或使用复位命令 ETH_W.q_wResetCounter 后复位。
无	i_udiUCMM_Request	UDINT	已接收的 EtherNet/IP 未连接消息的数量。 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。 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。 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。 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 (参见第 77 页)	EtherNet/IP 1 类连接的运行 (值=1) /空闲 (值=0) 标志。
<p><b>无</b>表示没有用于此系统变量的预定义 %MW 映射。  <b>未使用</b>表示系统未维护该变量，并且如果该变量的值非零，应被视为额外变量。</p>			

**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

下表对 ETH\_W 系统变量 ( ETH\_W\_STRUCT 类型 ) 的参数进行了描述 :

%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。 要再次复位, 需要先向此变量写入 0, 之后才能再次进行从 0 到 1 的转换。

**注意 :** n/a 表示没有用于此系统变量的预定义 %MW 映射。

## 第1.5节

### 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 LMC058 :

- 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 (参见第 78 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。



## 第1.6节

### 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_wPNIdentifier	WORD	0x0D73	Profibus 从站标识号
i_wBusAdr	WORD	1...126	Profibus 从站地址编号 保留地址 126。
i_CommState	UDINT	0 - 未知 1 - 未配置 2 - 停止 3 - 空闲 4 - 运行	Profibus 状态 运行模式：循环 I/O 正在运行
i_CommError	UDINT	0 - 无错误 <>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



---

# 第2章

## LMC058 系统功能

---

### 概述

本章介绍 LMC058 PLCSystem 库中包含的系统功能。

### 本章包含了哪些内容？

本章包含了以下部分：

节	主题	页
2.1	LMC058 读取功能	36
2.2	LMC058 写入功能	44
2.3	LMC058 用户功能	51

## 第2.1节

### LMC058 读取功能

---

#### 概述

本节描述 LMC058 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：

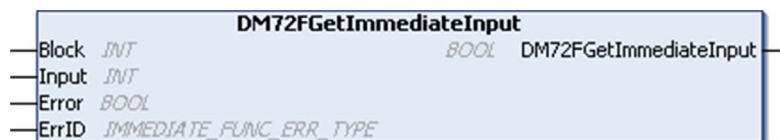
主题	页
DM72FGetImmediateInput：读取嵌入式专用 I/O 的输入	37
getTM5Delay：无有效交换的 TM5 总线循环数	38
IsFirstMastColdCycle：指示此循环是否为第一个 MAST 冷启动循环	40
IsFirstMastCycle：指示此循环是否为第一个 MAST 循环	41
IsFirstMastWarmCycle：指示此循环是否为第一个 MAST 热启动循环	43

## 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 91 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： ● 0= DM72F0 ● 1= DM72F1
输入	INT	功能块的目标输入。 0..6= DI0..DI6DI6

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> 的值 = FALSE/TRUE。

下表介绍了输入/输出变量：

输入/输出	类型	注释
错误	BOOL	FALSE = 正常运行。 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。

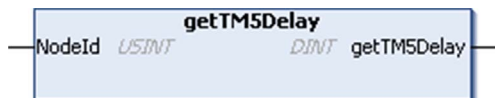
## getTM5Delay : 无有效交换的 TM5 总线循环数

### 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 32 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 91 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 模块的 <b>模块地址</b> ( 要获取 <b>模块地址</b> ，请验证 <b>I/O 配置</b> 选项卡中的值。 ) 

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：

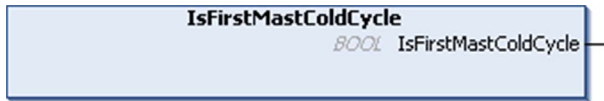
```
VAR delay :DINT; //Slot ID is 1 for the first TM5 module slot_ID :USINT := 1; END_VAR  
delay = getTM5Delay(slot_ID);
```

## IsFirstMastColdCycle : 指示此循环是否为第一个 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 91 页) 一章。

### I/O 变量描述

下表描述了输出变量：

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 41 页)。



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

### 功能描述

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 91 页) 一章。

### I/O 变量描述

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。

请在 MAST 任务中使用此示例。否则，可能会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加任务）：

```

VAR
MylsFirstMastCycle : BOOL;
MylsFirstMastWarmCycle : BOOL;
MylsFirstMastColdCycle : BOOL;
END_VAR

MylsFirstMastWarmCycle := IsFirstMastWarmCycle(); MylsFirstMastColdCycle :=
IsFirstMastColdCycle(); MylsFirstMastCycle := IsFirstMastCycle();

IF (MylsFirstMastWarmCycle) THEN
(*这是热启动后的第一个 Mast 循环：除了保留变量，所有变量都设置为其初始化值*)
(*=> 初始化所需的变量，以便应用程序在这种情况下按预期运行*)
END_IF;

IF (MylsFirstMastColdCycle) 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*)
(*=> 初始化所需的变量，以便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*)
```

```
(*=> 初始化所需的变量，以便应用程序在这种情况下按预期运行*)
```

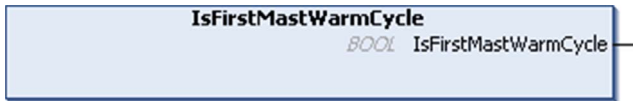
```
END_IF;
```

## IsFirstMastWarmCycle：指示此循环是否为第一个 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 91 页) 一章。

### I/O 变量描述

下表描述了输出变量：

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 IsFirstMastCycle (参见第 41 页) 功能。

## 第2.2节

### LMC058 写入功能

---

#### 概述

本节描述 LMC058 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	45
SetLEDBehaviour：确定 LED 的行为	47
SetRTCDrift：每周调整实时时钟	49

## 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

**注意：**输出强制 (参见 *Modicon LMC058 Motion Controller, 编程指南*) 将覆盖对输出执行的所有其他命令 (无论正在执行何种任务编程)。

### 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 91 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= D00</li> <li>● 1= D01</li> </ul>		

### 实现 DM72F?SetImmediateOutput?

SoMachine 在下列情况下返回检测到的编译错误：

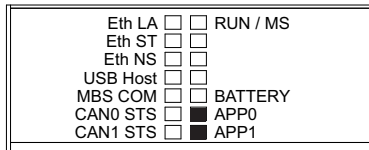
- 在多个任务中使用 DM72F?SetImmediateOutput? 功能。
- 在应用程序中使用与 DM72F?SetImmediateOutput? 关联的 %Q。
- 该输出已经专用于某一**嵌入式专用 I/O 功能块**功能（例如：PWM、频率发生器、编码器的反射输出、警报）。

## SetLEDBehaviour : 确定 LED 的行为

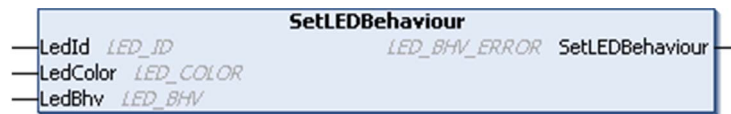
### 功能描述

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 91 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 80 页)	应用程序 LED 的 ID。
LedColor	LED_COLOR (参见第 83 页)	应用程序 LED 的颜色。
LedBhv	LED_BHV (参见第 81 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 82 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，因此在点亮新颜色之前关闭当前的颜色。下表显示 SetLedBehaviour 命令序列以及相关 LED 行为的示例：

步	LedId	LedColor	LedBhv	绿色闪烁模式	红色闪烁模式
1	LED_0	-	-	不亮	不亮
2	LED_0	LED_GREEN	LED_ON	亮起	不亮
3	LED_0	LED_GREEN	LED_OFF	不亮	不亮
4	LED_0	LED_RED	LED_ON	不亮	亮起



## SetRTCDrift : 每周调整实时时钟

### 功能描述

此功能用于在每周的指定日和时间 ( 小时:分钟 ) 对实时时钟 (RTC) 增加或减去指定的秒数。

**注意：** SetRTCDrift 功能需要进行编程，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 91 页) 一章。

### I/O 变量描述

下表介绍输入参数：

输入	类型	注释
RtcDrift	SINT (-29...+29)	按秒修正 (-29 ... +29)
日	DAY_OF_WEEK (参见第 85 页)	星期几进行更改。
时	HOUR (参见第 86 页)	在几点钟进行更改。
分	MINUTE (参见第 87 页)	在第几分钟进行更改。

**注意：** 如果为 RtcDrift、Day、Hour、Minute 输入的值超过了限制值，则逻辑控制器固件会将所有值设置为其最大值。

下表描述了输出变量：

输出	类型	注释
SetRTCDrift	RTCSETDRIFT_ERROR (参见第 84 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，仅在第一个 MAST 任务循环期间调用此功能一次，每个星期二清晨 5:45 向 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
```

---

## 第2.3节

### LMC058 用户功能

---

#### 概述

本节介绍 DataFileCopy PLCSystem 库中包含的 ExecuteScript 和 LMC058 功能。

#### 本节包含了哪些内容？

本节包含了以下主题：

主题	页
DataFileCopy : 复制文件命令	52
ExecuteScript : 运行脚本命令	55

## DataFileCopy : 复制文件命令

### 功能块描述

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。该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：

- 从格式化文件中读取数据，或者
- 将存储器缓冲区中的数据复制到一定格式的文件中。有关更多信息，请参阅闪存组织（参见 *Modicon LMC058 Motion 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 91 页）一章。

### I/O 变量描述

下表介绍输入变量：

输入	类型	注释
xExecute	BOOL	在上升沿启动功能块的执行。 在下降沿上，则在其执行终结时，复位功能块的输出。
sFileName	STRING	不含扩展名的文件名（自动添加扩展名 <i>.DTA</i> ）。只使用 a...z、A...Z、0...9 字母数字字符。
xRead	BOOL	TRUE：从文件复制到存储器。 FALSE：从存储器复制到文件。
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。 FALSE：采用相同类型存储器的其他控制器可以从该文件中读取数据。
iLocation	INT	0：文件位置是内部文件系统中的 <i>/usr/Dta</i> 。 1：文件位置是外部文件系统（USB 存储盘）中的 <i>/usr/Dta</i> 。

输入	类型	注释
uiSize	UINT	表示以字节为单位的大小。最大为 65534 字节。 仅使用符合 IEC 6113-1 ( 变量、数组、结构 ) 规范的变量地址，例如： Variable : int; uiSize := SIZEOF (Variable);
dwAdd	DWORD	表示存储器中的地址。 仅使用符合 IEC 6113-1 ( 变量、数组、结构 ) 规范的变量地址，例如： Variable : int; dwAdd := ADR (Variable);

## 警告

### 意外的设备操作

在将文件复制到存储器之前，确认存储器位置的大小正确，文件的类型正确。

**不遵循上述说明可能导致人员伤亡或设备损坏。**

下表介绍输出变量：

输出	类型	注释
xDone	BOOL	TRUE = 表示操作已成功完成。
xBusy	BOOL	TRUE = 表示功能块正在运行。
xError	BOOL	TRUE = 表示检测到错误，功能块中止操作。
eError	DataFileCopyError (参见第 68 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到存储器变量，将生成检测到的 CRC 错误。

## 示例

此示例介绍如何复制文件命令：

```

VAR
LocalArray : ARRAY [0..29] OF BYTE;
myFileName: STRING := 'exportfile';
EXEC_FLAG: BOOL;
DataFileCopy: DataFileCopy;
END_VAR
DataFileCopy(
xExecute:= EXEC_FLAG,
sFileName:= myFileName,

```

```
xRead:= FALSE,  
xSecure:= FALSE,  
iLocation:= DFCL_INTERNAL,  
uiSize:= SIZEOF(LocalArray),  
dwAdd:= ADR(LocalArray),  
xDone=> ,  
xBusy=> ,  
xError=> ,  
eError=> );
```

## ExecuteScript : 运行脚本命令

### 功能块描述

此功能块可以运行以下 USB 脚本命令：

- Download
- Upload
- SetNodeName
- Delete
- Reboot

请参阅使用 USB 大容量存储生成脚本和文件 (参见 *Modicon LMC058 Motion 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 91 页) 一章。

### I/O 变量描述

下表介绍输入变量：

输入	类型	注释
xExecute	BOOL	在检测到上升沿时，启动功能块的执行。 在检测到下降沿时，则在其执行终结时，复位功能块的输出。
sCmd	STRING	USB 脚本命令语法。 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，则功能块会将命令排入队列，不会立即执行该命令。 <b>注意：</b> 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 表示功能块正在运行。
xError	BOOL	TRUE 表示检测到错误；功能块中止操作。

输出	类型	注释
eError	ExecuteScriptError (参见第 69 页)	表示检测到的执行脚本错误的类型。

### 示例

此示例介绍了如何执行 Upload 脚本命令：

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



---

# 第3章

## LMC058 PLCSystem 库数据类型

---

### 概述

本章介绍 LMC058 PLCSystem 库的数据类型。

可用的数据类型有 2 种：

- 系统变量数据类型，由 LMC058 PLCSystem 库的系统变量 (参见第 13 页) ( PLC\_R、PLC\_W 等 ) 使用。
- 系统功能数据类型，由 LMC058 PLCSystem 库的读/写 系统功能 (参见第 35 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_RW 系统变量数据类型	58
3.2	DataFileCopy 系统变量数据类型	68
3.3	ExecScript 系统变量数据类型	69
3.4	ETH_RW 系统变量数据类型	70
3.5	TM5_MODULE_R/W 系统变量数据类型	78
3.6	系统功能数据类型	79

## 第3.1节

### PLC\_RW 系统变量数据类型

#### 概述

本节列出 PLC\_R 和 PLC\_W 结构中包括的系统变量数据类型并加以说明。

#### 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	59
PLC_R_BOOT_PROJECT_STATUS : 引导项目状态代码	61
PLC_R_IO_STATUS : I/O 状态代码	62
PLC_R_STATUS : 控制器状态代码	63
PLC_R_STOP_CAUSE : 从 RUN 状态向其他状态过度的原因代码	64
PLC_R_TERMINAL_PORT_STATUS : 编程端口连接状态代码	65
PLC_R_USB_HOST_STATUS : USB 主机端口连接状态代码	66
PLC_W_COMMAND : 控制命令代码	67

## 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	操作方法
PLC_R_APP_ERR_UNKNOWN	FFFF (十六进制)	检测到未定义的错误。	联系当地的支持部门。
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)	任务看门狗已到期。	检查您的应用程序。请参阅一章。需要复位才能进入 Run 模式。
PLC_R_APP_ERR_HARDWAREWATCHDOG	0011 (十六进制)	系统看门狗已到期。	如果此问题反复发生, 请检查通讯端口是否已断开。如果此问题继续存在, 请更新固件。如果此问题仍旧存在, 请联系您当地的支持部门。
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。	您的应用程序可能已经损坏。若要解决此问题, 请使用以下方法之一: <b>1. 编译 → 清除全部</b> <b>2. 导出/导入您的应用程序。</b> <b>3. 将 SoMachine 升级到最新版本。</b>
PLC_R_APP_ERR_UNRESOLVED_EXTREFS	0018 (十六进制)	检测到未定义的功能。	从应用程序中删除未解决的功能。
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。	您的应用程序可能已经损坏。若要解决此问题, 请使用以下方法之一: <b>1. 编译 → 清除全部</b> <b>2. 导出/导入您的应用程序。</b> <b>3. 将 SoMachine 升级到最新版本。</b>
PLC_R_APP_ERR_ILLEGAL_INSTRUCTION	0050 (十六进制)	检测到未定义的指令。	调试您的应用程序以解决问题。
PLC_R_APP_ERR_ACCESS_VIOLATION	0051 (十六进制)	试图访问保留存储器区域。	调试您的应用程序以解决问题。
PLC_R_APP_ERR_DIVIDE_BY_ZERO	0102 (十六进制)	检测到整数除 0。	调试您的应用程序以解决问题。
PLC_R_APP_ERR_PROCESSORLOAD_WATCHDOG	0105 (十六进制)	处理器由于应用程序任务而超载。	通过改善应用程序架构以减少应用程序工作负荷。增加任务循环持续时间。减少事件的发生频率。

---

枚举器	值	注释	操作方法
PLC_R_APP_ERR_DIVIDE_REAL_BY_ZERO	0152 (十六进制)	检测到实数除 0。	调试您的应用程序以解决问题。
PLC_R_APP_ERR_TOO_MANY_EVENT	4E20 (十六进制)	外部 I/O 事件率过高。	降低专用 I/O 事件频率。

## PLC\_R\_BOOT\_PROJECT\_STATUS : 引导项目状态代码

### 枚举类型介绍

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。 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	0000 ( 十六进制 )	控制器不包含应用程序。
PLC_R_STOPPED	0001 ( 十六进制 )	控制器已停止。
PLC_R_RUNNING	0002 ( 十六进制 )	控制器正在运行。
PLC_R_HALT	0004 ( 十六进制 )	控制器处于“暂停”状态。( 请参见控制器 <i>编程指南</i> 中的“控制器状态图” )。
PLC_R_BREAKPOINT	0008 ( 十六进制 )	控制器已在断点处暂停。

## PLC\_R\_STOP\_CAUSE : 从 RUN 状态向其他状态过度的原因代码

### 枚举类型介绍

PLC\_R\_STOP\_CAUSE 枚举数据类型包含以下值：

枚举器	值	注释	操作方法
PLC_R_STOP_REASON_UNKNOWN	00 (十六进制)	未定义初始值或停止原因。	出现未定义停止原因时，请联系您当地的支持部门。
PLC_R_STOP_REASON_HW_WATCHDOG	01 (十六进制)	在硬件看门狗超时后停止。	联系当地的支持部门。
PLC_R_STOP_REASON_RESET	02 (十六进制)	复位后停止。	有关复位可能性请参阅第 4 章。
PLC_R_STOP_REASON_EXCEPTION	03 (十六进制)	例外后停止。	检查您的应用程序。请参阅 1.1 章。需要复位才能进入 Run 模式。
PLC_R_STOP_REASON_USER	04 (十六进制)	用户请求后停止。	请参阅 1.1 章。
PLC_R_STOP_REASON_IECPROGRAM	05 (十六进制)	在发出程序命令请求 (例如：带参数的控制命令 PLC_W.q_wPLCControl:=PLC_W.COMMAND.PLC_W.STOP;) 后停止。	-
PLC_R_STOP_REASON_DELETE	06 (十六进制)	删除应用程序命令后停止。	请参阅 1.1 章。
PLC_R_STOP_REASON_DEBUGGING	07 (十六进制)	进入调试模式后停止。	-
PLC_R_STOP_REASON_NETWORK_REQUEST	0A hex	从网络、USB 存储盘或 PLC_W command. 发出请求后停止	-
PLC_R_STOP_REASON_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时停止。	保持变量因未在应用程序中引用而被删除。如果应用程序将保持变量设为其初始值，则可用。
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时停止。	创建有效的引导应用程序。
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 LMC058 Motion Controller, 编程指南*)。



## PLC\_R\_TERMINAL\_PORT\_STATUS : 编程端口连接状态代码

### 枚举类型介绍

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)	PC 已连接到编程端口。
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。
<b>(1)</b> 满足下列要求的受支持 USB 存储盘： <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

### 枚举类型介绍

PLC\_W\_COMMAND 枚举数据类型包含以下值：

枚举器	值	注释
PLC_W_STOP	0001 (十六进制)	用于停止控制器的命令。
PLC_W_RUN	0002 (十六进制)	用于运行控制器的命令。
PLC_W_RESET_COLD	0004 (十六进制)	用于启动控制器冷复位的命令。
PLC_W_RESET_WARM	0008 (十六进制)	用于启动控制器热复位的命令。

## 第3.2节

### DataFileCopy 系统变量数据类型

#### DataFileCopyError : 检测到的错误代码

##### 枚举类型介绍

DataFileCopyError 枚举数据类型包含以下值：

枚举器	值	描述
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。

## 第3.3节

### ExecScript 系统变量数据类型

#### ExecuteScriptError : 检测到的错误代码

##### 枚举类型介绍

ExecuteScriptError 枚举数据类型包含以下值：

枚举器	值	描述
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 LMC058 Motion 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)	某一脚本已经在执行。

## 第3.4节

### ETH\_RW 系统变量数据类型

#### 概述

本节列出 ETH\_R 和 ETH\_W 结构中包括的系统变量数据类型并加以说明。

#### 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	71
ETH_R_FRAME_PROTOCOL : 帧传输协议代码	72
ETH_R_PORT_DUPLEX_STATUS : 传输模式代码	73
ETH_R_PORT_LINK_STATUS : 通讯链路状态代码	74
ETH_R_PORT_SPEED : 以太网端口的通讯速度代码	75
ETH_R_PORT_IP_STATUS : 以太网 TCP/IP 端口状态代码	76
ETH_R_RUN_IDLE : Ethernet/IP 运行和空闲状态代码	77

## ETH\_R\_IP\_MODE : IP 地址源代码

### 枚举类型介绍

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

### 枚举类型介绍

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 IEEE 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

### 枚举类型介绍

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 : 通讯链路状态代码

### 枚举类型介绍

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

### 枚举类型介绍

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

### 枚举类型介绍

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 )	正在等待配置。
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、无效配置等 )。
DUPLICATE_IP	04 ( 十六进制 )	IP 地址已被其他设备使用。

## ETH\_R\_RUN\_IDLE : Ethernet/IP 运行和空闲状态代码

### 枚举类型介绍

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)	EtherNet/IP 连接空闲。
RUN	01 (十六进制)	EtherNet/IP 连接正在运行。

## 第3.5节

### 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。

## 第3.6节

### 系统功能数据类型

#### 概述

本节介绍 LMC058 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	80
LED_BHV : SetLEDBehaviour 功能 LedBhv 参数代码	81
LED_BHV_ERROR : 检测到的 SetLEDBehaviour 功能错误代码	82
LED_COLOR : SetLEDBehaviour 功能 LedColor 参数代码	83
RTCSETDRIFT_ERROR : SetRTCDrift功能检测到的错误代码	84
DAY_OF_WEEK : SetRTCDrift 功能日期参数代码	85
HOUR : SetRTCDrift 功能小时参数类型	86
MINUTE : SetRTCDrift 功能分钟参数类型	87

## 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。



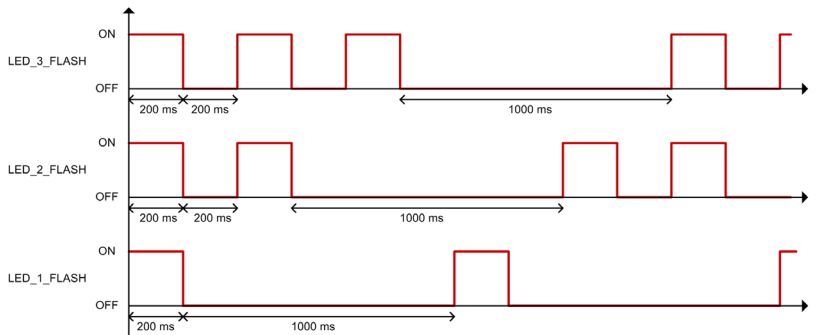
## LED\_BHV : SetLEDBehaviour 功能 LedBhv 参数代码

### 枚举类型描述

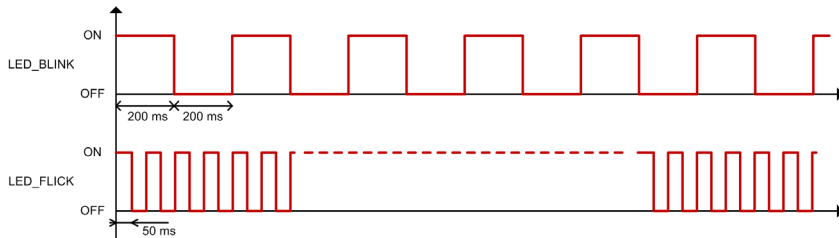
LED\_BHV 枚举数据类型包含以下值：

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 : SetRTCDrift功能检测到的错误代码

### 枚举类型介绍

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

### 枚举类型介绍

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





---

# 附录 A

## 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
功能与功能块的区别	92
如何通过 IL 语言使用功能或功能块	93
如何通过 ST 语言使用功能或功能块	96

## 功能与功能块的区别

### 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。
- 从一次调用到另一次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 )。
- 需要通过实例 ( 具有专用名称和变量的功能块副本 ) 进行调用。
- 从功能块或程序的一次调用到另一次调用，每个实例都具有持续状态 ( 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 的一个实例：

```
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。

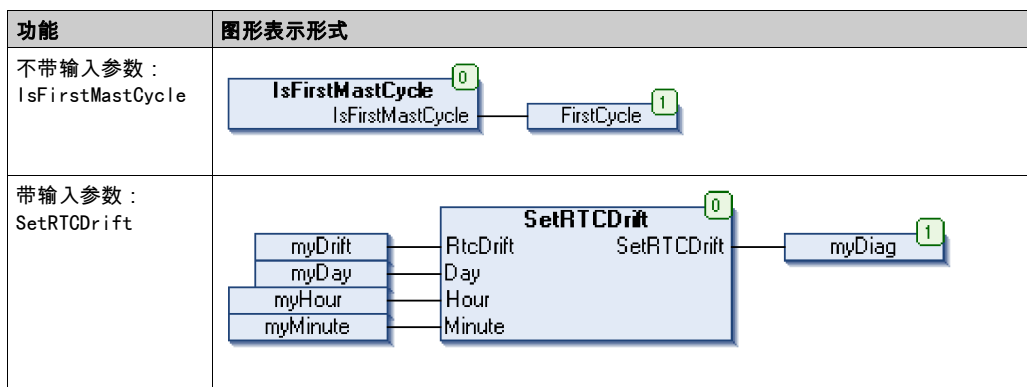
我们以功能 IsFirstMastCycle、功能 SetRTCDrift 和功能块 TON 为例来演示实现的过程。

### 通过 IL 语言使用功能

以下过程描述如何用 IL 语言插入一个功能：

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，请参阅添加和调用 POU (参见 <i>SoMachine, 编程指南</i> )。
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，则在使用输入助手时，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	插入新的行，将功能的结果存储到相应的变量中：在操作符列（左侧字段）中输入 ST 指令，并在右侧的字段中输入变量名称。

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）：



在 IL 语言中，功能名称直接用在操作符列中：

功能	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 5 </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>

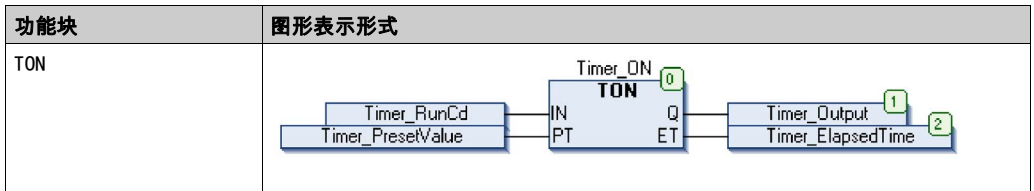
### 通过 IL 语言使用功能块

以下过程描述如何用 IL 语言插入一个功能块：

步骤	操作
1	<p>通过指令列表语言打开 POU 或创建新 POU。</p> <p><b>注意：</b> 此处未详细介绍创建 POU 的步骤。有关详细信息，请参阅添加和调用 POU (参见 <i>SoMachine, 编程指南</i>)。</p>

步骤	操作
2	创建功能块所需的变量（包括实例名称）。
3	<p>使用 CAL 指令调用功能块：</p> <ul style="list-style-type: none"> <li>● 使用<b>输入助手</b>选择 FB（右键单击并在上下文菜单中选择<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> <p>每个参数 (I/O) 都是一条指令：</p> <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	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。

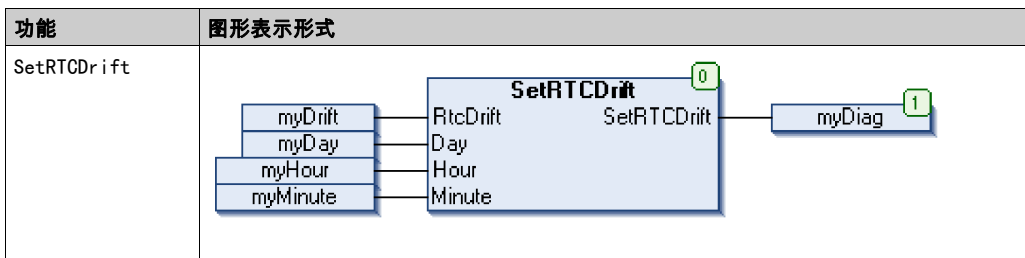
我们以功能 SetRTCDrift 和功能块 TON 为例演示实现的过程。

### 通过 ST 语言使用功能

以下过程描述如何用 ST 语言插入一个功能：

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，请参阅添加和调用 POU (参见 <i>SoMachine, 编程指南</i> )。
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： FunctionResult:= FunctionName (Var Input1, Var Input2,.. Var Inputx);

要阐释该过程，请考虑下面以图形方式表示的 SetRTCDrift 功能：



此功能的 ST 语言如下所示：

功能	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; myRTCAadjust: RTCDRIFT_ERROR; END_VAR myRTCAadjust:= SetRTCDrift(myDrift, myDay, myHour, myMinute);</pre>

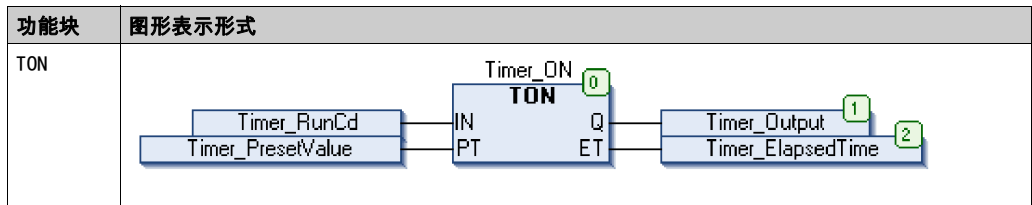


## 通过 ST 语言使用功能块

以下过程描述如何用 ST 语言插入一个功能块：

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。有关添加、声明和调用 POU 的更多信息，请参阅相关文档 (参见 <i>SoMachine, 编程指南</i> )。
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:=Var Input1, Input2:=Var Input2, ... Output1:=VarOutput1, Output2:=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	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>



## %

根据 IEC 标准，% 是标识逻辑控制器中用于存储程序变量、常量和 I/O 等值的内部存储器地址的前缀。

## %MW

根据 IEC 标准，%MW 表示存储器字寄存器（例如，存储器字类型的语言对象）。

## 任务

一组段和子程序，MAST 任务为循环或周期性执行，FAST 任务为周期性执行。

任务具有优先级，并且链接到控制器的输入和输出。可以根据任务来刷新这些 I/O。

一个控制器可以有多个任务。

## 功能

拥有一个输入和返回一个直接结果的编程单元。但是，与 FBs 不同，它通过其名称（而不是通过实例）直接调用、不具备从一个调用到下一个调用的持久状态且可以用作其他编程表达式中的操作数。

示例：布尔 (AND) 操作符、计算、转换 (BYTE\_TO\_INT)

## 功能块

拥有一个或多个输入并返回一个或多个输出的编程单元。FBs 通过实例（具有专用名称和变量的功能块副本）进行调用，且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

## 功能块图

控制系统的标准 IEC 61131-3 所支持的五种逻辑或控制语言中的其中一种语言。功能块图是面向图形的编程语言。它可以与一系列网络搭配使用，其中每个网络均包含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。

## 十六进制

(十六进制)

## 协议

一种用于控制和启用两个计算端点和设备之间的连接、通讯和数据传输的惯例或标准。

## 反射输出

HSC 的输出是与计数器值进行比较的阈值关联的反射输出，具体取决于 HSC 的配置。反射输出可以在开启或关闭之间切换，具体取决于使用阈值配置的关系。

## 变量

由程序寻址和修改的存储器单元。

## 固件

表示构成控制器上操作系统的 BIOS、数据参数和编程指令。固件存储在控制器内的非易失性存储器上。

## 字符串

一系列 ASCII 字符的变量。

## 字节

采用 8 位格式编辑的类型，范围从十六进制 00 到十六进制 FF。

## 应用程序

包括配置数据、符号和文档的程序。

## 引导应用程序

( *引导应用程序* ) 包含应用程序的二进制文件。通常它存储在控制器中，使控制器启动用户生成的应用程序。

## 扩展总线

扩展 I/O 模块和控制器之间的电子通讯总线。

## 控制器

自动化工业流程 ( 也称为可编程逻辑控制器或可编程控制器 ) 。

## 控制网络

此网络中包含逻辑控制器、SCADA 系统、PC、HMI、交换机.....

支持以下两种拓扑：

- 扁平：此网络中的所有模块和设备都属于同一个子网。
- 2 层：网络分为操作网络 and 控制器间网络。

这两个网络可以在物理上独立，但通常通过路由设备链接。

## 看门狗

看门狗是一种特殊的定时器，用于确保程序不会超过为它们所分配的扫描时间。通常将看门狗定时器设置为比扫描时间较高的值，并在每个扫描周期结束后重置为零。如果警戒时钟定时器达到预设值 ( 例如，因为程序陷入了死循环 ) ，则表明出现了错误且程序将会停止。

## 程序

应用程序的组成部分，其中包括可以在逻辑控制器的存储器中安装的经过编译的源代码。

## 系统变量

用于提供控制器数据和诊断信息，并用来向控制器发送命令的变量。

## 网络

共享一个公用数据路径和通讯协议的各种互联设备系统。

## 设备

包括子组件 ( 如传送带和转盘等 ) 的机器的一部分。

## 配置

一个系统内硬件组件的布局 and 互连以及硬件和软件的参数，可决定系统的运行特性。

**闪存**

可覆盖的非易失性存储器它存储在一个特殊的可擦除和可重编程的 EEPROM 上。

**非定位变量**

没有地址的变量（请参阅**定位变量**）。

**ARRAY**

在逻辑控制器存储器中以表格形式定义的单一类型数据对象的系统排列。语法如下：ARRAY [ <dimension> ] OF <Type>

示例 1：ARRAY [1..2] OF BOOL 是由两个 BOOL 类型的元素组成的一维表。

示例 2：ARRAY [1..10, 1..20] OF INT 是由 10 x 20 个 INT 类型的元素组成的二维表。

**BOOL**

（**布尔**）用于计算的基本数据类型。BOOL 变量可以是以下两个值之一：0 (FALSE) 或 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**）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）的 UDP 网络协议。客户端使用客户端 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 服务利用 UDP 端口 67 和 68。

**CAN**

（**控制器局域网**）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）之间互连，以及用于处理实时工业应用的协议 (ISO 11898)。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。

**CFC**

（**连续功能图**）一种基于功能块图语言的图形编程语言（IEC 61131-3 标准的扩展），工作原理与流程图类似。但是，不可以使用网络并对图形元素进行任意定位，允许反馈回路。每个功能块的输入位于左侧，输出位于右侧。可以将功能块输出链接到其他功能块的输入以创建复合表达式。

**CRC**

（**循环冗余校验**）用来确定通讯传输的有效性的方法。传输包含构成校验和的位域。发射器根据消息的内容来计算所使用消息的校验和。接收节点后，按照相同的方式再次计算该字段。如果两次 CRC 计算的值存在任何差异，则说明传输的消息与收到的消息不同。

**DHCP**

（**动态主机配置协议**）BOOTP 的高级扩展。DHCP 虽然较为高级，但是 DHCP 和 BOOTP 可以通用。（DHCP 可以处理 BOOTP 客户端请求。）

**DWORD**

（**双字**）以 32 位格式进行编码的类型。

**Ethernet**

用于 LANs 的物理和数据链路层技术，也称为 IEEE 802.3。

**EtherNet/IP**

( *Ethernet工业协议* ) 用于工业系统中自动化解决方案制造的开放式通讯协议。EtherNet/IP 是在其上层执行公共工业协议的网络家庭成员。支持组织 (ODVA) 规定 EtherNet/IP 是为了实现全球适应性和介质独立性。

**FB**

( *功能块* ) 用于整合一组编程指令以执行特定和规范化操作 ( 如速度控制、间隔控制或计数 ) 的实用编程机制。功能块可以包含配置数据和一组内部或外部操作参数，通常是一个或多个数据输入和输出。

**GVL**

( *全局变量列表* ) 管理 项目中的全局变量。

**I/O**

( *输入/输出* )

**ID**

( *标识符/标识* )

**IEC**

( *国际电工委员会* ) 负责为所有电器、电子和相关技术制定和发布国际标准的非盈利性和非政府性的国际标准组织。

**IEC 61131-3**

工业自动化设备的 3 部分标准的第 3 部分。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。图形编程语言既是梯形图语言又是功能块图语言。文本编程语言包括结构化文本和指令列表。

**IEEE 802.3**

定义了有线IEEE的物理层以及数据链路层的介质访问控制子层的 Ethernet 标准集合。

**IL**

( *指令列表* ) 以某种语言编写的程序，包括由控制器按顺序执行的一系列基于文本的指令。每个指令均包括一个行号、一个指令代码和一个操作数 ( 请参阅 IEC 61131-3 )。

**INT**

( *整数* ) 以 16 位格式进行编码的整数。

**IP**

( *因特网协议* ) TCP/IP 协议系列的一部分，用于跟踪设备的因特网地址、对传出消息进行路由并识别传入消息。

**LD**

( *梯形图* ) 控制器程序指令的图形表示，其中包括控制器按顺序执行的一系列梯级中的触点、线圈和块符号 ( 请参阅 IEC 61131-3 )。

**LED**

( *发光二极管* ) 在低电平电荷时亮起的指示灯。

**LWORD**

( *长字* ) 以 64 位格式进行编码的数据类型。

**MAC 地址**

(*介质访问控制地址*) 与特定硬件设备关联的唯一 48 位编号。在生产网卡或设备过程中, 需要为每个网卡或设备编入一个 MAC 地址。

**MAST**

通过其编程软件运行的处理器任务。MAST 任务有两个段:

- **IN**: 在 MAST 任务执行之前, 将输入复制到 IN 段。
- **OUT**: 在 MAST 任务执行完后, 将输出复制到 OUT 段。

**PLC**

(*可编程逻辑控制器*) 用于自动化制造、工业和其他机电进程的工业计算机。此外, PLCs 与普通计算机不同, 因为这些计算机拥有多个输入和输出数组, 并且符合冲击、振动、温度和电气干扰的更强大的规范。

**POU**

(*程序组织单元*) 源代码的变量声明和相应的指令集。POUs 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。

**PWM**

(*脉冲宽度调制*) 以可调占空比在关闭和开启之间振荡以产生矩形波形式的快速输出 ( 尽管可以调整它来产生方形波 ) 。

**RTC**

(*实时时钟*) 由电池供电可连续运转以显示当天时间和日历的时钟, 即使在为延长电池使用寿命而未对控制器通电时也一样。

**run**

使控制器根据程序的逻辑解决方案扫描应用程序、读取物理输入并写入物理输出的命令。

**ST**

(*结构化文本*) 一种包括复杂的语句和嵌套指令 ( 如迭代循环、条件执行或功能 ) 的语言。ST 符合 IEC 61131-3

**STOP**

使控制器停止运行应用程序的命令。

**TCP**

(*传输控制协议*) 基于连接的传输层协议, 可提供同步双向数据传输。TCP 是 TCP/IP 协议套件的一部分。

**UDINT**

(*无符号双精度整数*) 以 32 位格式进行编码的整数。

**UINT**

(*无符号整数*) 以 16 位格式进行编码的整数。

**WORD**

一种以 16 位格式进行编码的类型。







- DataFileCopy
  - 将数据复制到文件或从文件复制数据, 52
- DataFileCopyError
  - 数据类型, 68
- DAY\_OF\_WEEK
  - 数据类型, 85
- DM72F•SetImmediateOutput
  - 功能, 45
- DM72FGetImmediateInput
  - 功能, 37
- ETH\_R
  - 系统变量, 28
- ETH\_R\_FRAME\_PROTOCOL
  - 数据类型, 72
- ETH\_R\_IP\_MODE
  - 数据类型, 71
- ETH\_R\_PORT\_DUPLEX\_STATUS
  - 数据类型, 73
- ETH\_R\_PORT\_LINK\_STATUS
  - 数据类型, 74
- ETH\_R\_PORT\_SPEED
  - 数据类型, 75
- ETH\_W
  - 系统变量, 31
- ExecuteScript
  - 运行脚本命令, 55
- ExecuteScriptError
  - 数据类型, 69
- getTM5Delay
  - 功能, 38
- HOUR
  - 数据类型, 86
- IsFirstMastColdCycle
  - 第一个冷启动循环, 40
- IsFirstMastCycle
  - 第一个 MAST 循环, 41
- IsFirstMastWarmCycle
  - 第一个热启动循环, 43
- LED\_BHV
  - 数据类型, 81
- LED\_BHV\_ERROR
  - 数据类型, 82
- LED\_COLOR
  - 数据类型, 83
- LED\_ID
  - 数据类型, 80
- M241 PLCSystem
  - DataFileCopy, 52
  - ExecuteScript, 55
  - IsFirstMastColdCycle, 40
  - IsFirstMastCycle, 41
  - IsFirstMastWarmCycle, 43
  - SetRTCDrift, 49
- MINUTE
  - 数据类型, 87
- PLC\_R
  - 系统变量, 19
- PLC\_R\_APPLICATION\_ERROR
  - 数据类型, 59
- PLC\_R\_BOOT\_PROJECT\_STATUS
  - 数据类型, 61
- PLC\_R\_IO\_STATUS
  - 数据类型, 62
- PLC\_R\_STATUS
  - 数据类型, 63
- PLC\_R\_STOP\_CAUSE
  - 数据类型, 64
- PLC\_R\_TERMINAL\_PORT\_STATUS
  - 数据类型, 65
- PLC\_R\_USB\_HOST\_STATUS
  - 数据类型, 66
- PLC\_W
  - 系统变量, 23
- PLC\_W\_COMMAND
  - 数据类型, 67
- PROFIBUS\_R, 33
  - 数据类型, 33
- RTC
  - SetRTCDrift, 49
- RTCSETDRIFT\_ERROR
  - 数据类型, 84

## SERIAL\_R

系统变量, 25

## SERIAL\_W

系统变量, 26

## SetLEDBehaviour

功能, 47

## SetRTCDrift

加快或减慢 RTC 频率, 49

## TM5\_MODULE\_R

系统变量, 32

## TM5\_MODULE\_STATE

数据类型, 78

## 功能

DM72F•SetImmediateOutput, 45

DM72FGetImmediateInput, 37

getTM5Delay, 38

SetLEDBehaviour, 47

功能与功能块的区别, 92

如何通过 IL 语言使用功能或功能块, 93

如何通过 ST 语言使用功能或功能块, 96

## 实时时钟

SetRTCDrift, 49

## 循环

IsFirstMastColdCycle, 40

IsFirstMastCycle, 41

IsFirstMastWarmCycle, 43

## 数据类型

DataFileCopyError, 68

DAY\_OF\_WEEK, 85

ETH\_R\_FRAME\_PROTOCOL, 72

ETH\_R\_IP\_MODE, 71

ETH\_R\_PORT\_DUPLEX\_STATUS, 73

ETH\_R\_PORT\_IP\_STATUS, 76

ETH\_R\_PORT\_LINK\_STATUS, 74

ETH\_R\_PORT\_SPEED, 75

ETH\_R\_RUN\_IDLE, 77

ExecuteScriptError, 69

HOUR, 86

LED\_BHV, 81

LED\_BHV\_ERROR, 82

LED\_COLOR, 83

LED\_ID, 80

MINUTE, 87

PLC\_R\_APPLICATION\_ERROR, 59

PLC\_R\_BOOT\_PROJECT\_STATUS, 61

PLC\_R\_IO\_STATUS, 62

PLC\_R\_STATUS, 63

PLC\_R\_STOP\_CAUSE, 64

PLC\_R\_TERMINAL\_PORT\_STATUS, 65

PLC\_R\_USB\_HOST\_STATUS, 66

PLC\_W\_COMMAND, 67

PROFIBUS\_R, 33

RTCSETDRIFT\_ERROR, 84

TM5\_MODULE\_STATE, 78

## 文件复制命令

DataFileCopy, 52

## 系统变量, 23, 31

ETH\_R, 28

PLC\_R, 19

SERIAL\_R, 25

SERIAL\_W, 26

TM5\_MODULE\_R, 32

使用, 17

定义, 15

## 脚本命令

ExecuteScript, 55