

工业PC平台

NY系列

IPC 机器控制器

工业平板电脑 / 工业BOX PC

用户手册  
软件篇

NY532-1500

NY532-1400

NY532-1300

NY512-1500

NY512-1400

NY512-1300

工业平板电脑  
工业BOX PC



#### 声明

- 禁止擅自誊写、复制、转载本手册内容的一部分或全部。
- 本手册的内容如因改良需要而有所变更，恕不另行通知。敬请谅解。
- 本手册的内容经仔细确认以期万全，若发现不明之处或错误，麻烦请与本公司分店或营业所联系。此时，请同时告知卷末记载的Man.No.（手册No.）。

#### 商标

- Sysmac是欧姆龙株式会社FA设备产品在日本及其他国家的商标或注册商标。
- Microsoft、Windows、Windows Vista、Excel、Visual Basic是美国Microsoft Corporation在美国及其它国家的注册商标或商标。
- EtherCAT® 是德国Beckhoff Automation GmbH的注册商标和专利技术，由该公司授权使用。
- ODVA、CIP、CompoNet、DeviceNet、EtherNet/IP是ODVA的商标。
- SD、SDHC标志是SD-3C, LLC的商标。 

其他正文中记载的公司名称及产品名称为各公司的商标或注册商标。

#### 著作权

屏幕截图的使用已获得微软的许可。

# 前言

感谢您购买 NY 系列 IPC 机器控制器 工业平板电脑 / 工业 BOX PC。

本手册中，将对象产品，即工业平板电脑和工业 BOX PC 统称为“NY 系列工业 PC”。此外，由 NY 系列工业 PC 的实时 OS 中装载的控制器功能直接控制的范围记载为“控制器”。

本手册记载了使用 NY 系列控制器所必需的信息。请在使用前仔细阅读本手册，在充分理解功能和性能的基础上，灵活用于系统构建。

此外，请妥善保管本手册，阅读之后放于随时可取阅之处。

## 阅读对象

本手册提供给下列阅读对象。

具有电工专业知识的人员（合格的电气工程师或具有同等知识的人员）：

- 引进 FA 设备的人员；
- 设计 FA 系统的人员；
- 安装和连接 FA 设备的人员；
- FA 现场管理人员。

此外，编程语言的对象为掌握国际标准规格 IEC 61131-3 或日本国内标准规格格 JIS B 3503 之规定内容的人员。

## 对象产品

本手册的对象为以下产品：

- NY 系列 IPC 机器控制器 工业平板电脑
  - NY532-15□□
  - NY532-14□□
  - NY532-13□□
- NY 系列 IPC 机器控制器 工业 BOX PC
  - NY512-15□□
  - NY512-14□□
  - NY512-13□□

各产品的部分规格或限制事项可能在其他手册中有记载。请确认☐「分册构成 (P.2)」及☐「相关手册 (P.35)」。

# 分册构成

本产品的手册如下表所示分册。请根据不同目的选读，灵活运用本产品。

本产品的操作主要使用自动化软件 Sysmac Studio。关于 Sysmac Studio，请参考 [□](#) 《Sysmac Studio Version 1 操作手册（SBCA-CN5-362）》。

使用目的	手册							
	基本信息							
	NY 系列 故障排除手册	NY 系列 IPC 机器控制器 工业平板电脑 / 工业 BOX PC 用户手册 内置 EtherNet/IP 端口篇	NY 系列 IPC 机器控制器 工业平板电脑 / 工业 BOX PC 用户手册 内置 EtherCAT 端口篇	NY 系列 指令基准手册 运动篇	NY 系列 IPC 机器控制器 工业平板电脑 / 工业 BOX PC 用户手册 运动控制篇	NY 系列 指令基准手册 基本篇	NY 系列 IPC 机器控制器 工业平板电脑 / 工业 BOX PC 用户手册 软件篇	NY 系列 IPC 机器控制器 工业平板电脑 / 工业 BOX PC 用户手册 设置篇
了解 NY 系列平板电脑的概要	○							
了解 NY 系列 BOX PC 的概要		○						
进行安装、设置和硬件设定								
进行运动控制时	○	○						
使用 EtherCAT 时								○
使用 EtherNet/IP 时								○
进行设置 <sup>1</sup>								
进行初始设置时			○					
进行控制器的使用准备时								
进行软件设定								
进行运动控制时								
使用 EtherCAT 时				○				○
使用 EtherNet/IP 时								○
创建用户程序								
进行运动控制时								
使用 EtherCAT 时				○	○			○
使用 EtherNet/IP 时								○
组合异常处理时								○
进行动作确认、调试时								
进行运动控制时								
使用 EtherCAT 时				○				○
使用 EtherNet/IP 时								○
了解异常管理功能或故障发生时的处理方法 <sup>2</sup>								○
了解维护作业								
进行运动控制时	○	○						
使用 EtherCAT 时								○
使用 EtherNet/IP 时								○

\*1. 关于 Windows 上的设置方法和实用工具等的使用方法请参考 [□](#) 《NY 系列 工业平板电脑 / 工业 BOX PC 用户手册 安装篇（SBCA-CN5-441）》。

\*2. 关于异常管理的说明和异常项目，请参考 [□](#) 《NY 系列 故障排除手册（SBCA-CN5-438）》。

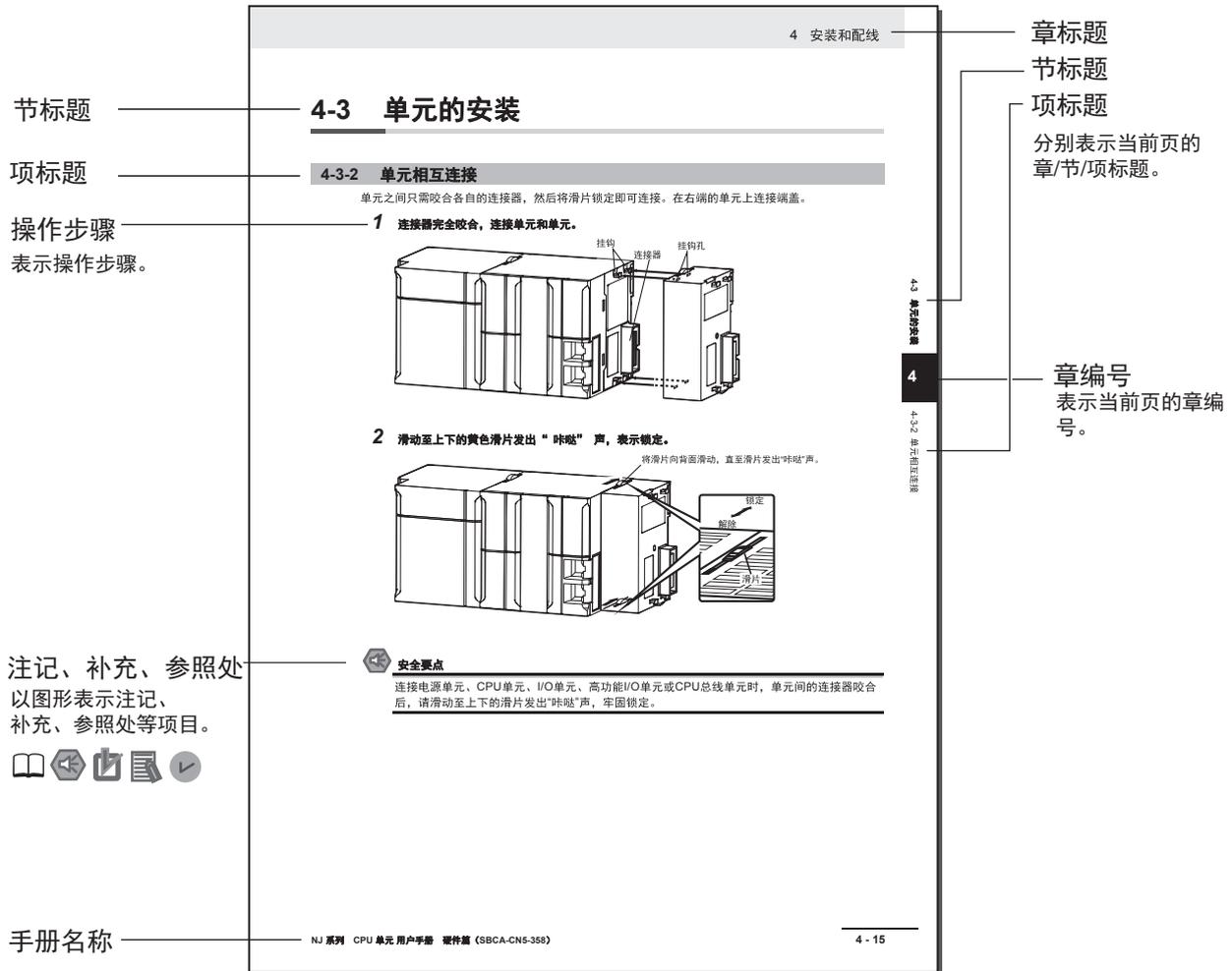
# 手册说明

本手册中记载的功能说明可能有部分内容与 NJ/NX 系列共通。因此，请注意以下事项。

- NJ/NX/NY 系列的共通功能中使用相同的功能名称。“CPU 单元名称功能”、“CPU 单元的写入保护功能”等，功能名称中包含“CPU 单元”时，NY 系列中是指“控制器”。
- 本手册的功能规格一览中记载为“CPU 单元”时，在 NY 系列中也是指“控制器”。

## 页面构成

本手册各页的构成如下。



本页为说明用样本。与实际内容不同。

## 图标

本手册中使用的图标含义如下。



### 安全要点

表示为了安全使用，应该实施或避免的行为。



### 使用注意事项

表示为了防止产品出现动作不良、误动作或严重影响其性能、功能，应该实施或避免的行为。



### 参考

需要时阅读的项目。

了解后有助于使用的信息以及使用时可参考的内容。



表示记载有详细信息、相关信息的位置。



### 版本相关信息

介绍不同版本的控制器或 Sysmac Studio 的性能和功能区别。

## 表述注意事项

本手册中，从 Sysmac Studio 向控制器实体传送数据记载为“下载”，从控制器实体向 Sysmac Studio 传送数据记载为“上传”。

在 Sysmac Studio 中，无论是“下载”还是“上传”，都使用“同步”功能。Sysmac Studio 的“同步”功能可以自动核对电脑上的数据和控制器实体上的数据，由用户选择传送方向。

# 目录结构

1	NY系列控制器的概要	10	通信设定	1	10
2	NY系列控制器的动作	11	实际使用步骤示例	2	11
3	I/O单元和从站/单元构成	A	附录	3	A
4	控制器设定	I	索引	4	I
5	设计任务			5	
6	编程			6	
7	动作确认和运行			7	
8	控制器内置功能			8	
9	备份功能			9	

# 目录

前言 .....	1
阅读对象 .....	1
对象产品 .....	1
分册构成 .....	2
手册说明 .....	3
页面构成 .....	3
图标 .....	4
表述注意事项 .....	4
目录结构 .....	5
承诺事项 .....	14
<b>NY 系列 工业平板电脑 / 工业 BOX PC 用户手册 软件篇 (SBCA-CN5-436)</b>	
安全注意事项 .....	16
安全信息的标识及其含义 .....	16
图标说明 .....	16
警告 .....	17
注意 .....	21
安全要点 .....	22
使用注意事项 .....	28
法规和标准 .....	31
日本国外的使用 .....	31
符合 EU 指令 .....	31
软件许可和著作权 .....	31
版本 .....	32
版本的确认方法 .....	32
相关手册 .....	35
术语说明 .....	37
手册修订记录 .....	40

## 第 1 章 NY 系列控制器的概要

1-1 NY 系列控制器定义 .....	1-2
1-1-1 特点 .....	1-3
1-1-2 系统构成概要 .....	1-5
1-2 规格概要 .....	1-7
1-3 产品型号 .....	1-9
1-4 NY 系列的整体使用步骤 .....	1-10
1-4-1 步骤概要 .....	1-10
1-4-2 步骤详情 .....	1-11

## 第 2 章 NY 系列控制器的动作

<b>2-1 NY 系列控制器的动作概要</b> .....	<b>2-2</b>
2-1-1 NY 系列控制器的原理概要 .....	2-2
2-1-2 NY 系列控制器在各状态下的动作概要 .....	2-3
<b>2-2 软件的原理</b> .....	<b>2-4</b>
2-2-1 软件构成 .....	2-4
2-2-2 软件的动作 .....	2-5
<b>2-3 与 I/O 的访问原理</b> .....	<b>2-10</b>
2-3-1 变量的种类 .....	2-10
2-3-2 变量和 I/O 的访问 .....	2-13
<b>2-4 顺序控制和运动控制的原理</b> .....	<b>2-16</b>
2-4-1 控制的概要 .....	2-16
2-4-2 顺序控制的原理 .....	2-18
2-4-3 运动控制的原理 .....	2-19
2-4-4 顺序控制和运动控制的同步 .....	2-20
<b>2-5 控制器内的数据概要</b> .....	<b>2-21</b>
<b>2-6 控制器在各种状态下的动作</b> .....	<b>2-22</b>
2-6-1 控制器的状态 .....	2-22
2-6-2 各种状态下的动作 .....	2-24
2-6-3 动作模式 .....	2-25
<b>2-7 Windows 状态的监视及变更</b> .....	<b>2-27</b>
2-7-1 Windows 的状态 .....	2-27
2-7-2 Windows 状态的监视 .....	2-27
2-7-3 Windows 状态的变更 .....	2-28
2-7-4 Windows 状态的变化 .....	2-28
<b>2-8 关机处理</b> .....	<b>2-29</b>
2-8-1 断电及按下电源按钮时的处理 .....	2-30
2-8-2 关机处理的设定 .....	2-31
2-8-3 关机处理的内容 .....	2-32
<b>2-9 控制器重置</b> .....	<b>2-35</b>

## 第 3 章 I/O 单元和从站 / 单元构成

<b>3-1 从站 / 单元构成的步骤</b> .....	<b>3-2</b>
<b>3-2 EtherCAT 从站构成的创建</b> .....	<b>3-4</b>
<b>3-3 I/O 端口和设备变量</b> .....	<b>3-5</b>
3-3-1 I/O 端口 .....	3-5
3-3-2 I/O 端口名称 .....	3-6
3-3-3 设备变量 .....	3-6
<b>3-4 向单元分配变量</b> .....	<b>3-8</b>
3-4-1 向单元分配变量的方法 .....	3-8
3-4-2 单元中分配的变量的使用方法 .....	3-9
<b>3-5 “轴”的创建和伺服驱动器 / 编码器输入从站的分配</b> .....	<b>3-11</b>
3-5-1 概要 .....	3-11
3-5-2 轴变量 / 轴组变量 .....	3-11
3-5-3 “轴”及轴变量的创建和使用方法 .....	3-13

## 第 4 章 控制器设定

<b>4-1 控制器设定的概要</b> .....	<b>4-2</b>
<b>4-2 PLC 功能模块的初始设定</b> .....	<b>4-4</b>
4-2-1 概要 .....	4-4

4-2-2	控制器设定 .....	4-4
4-2-3	任务设置 .....	4-7
<b>4-3</b>	<b>运动控制功能模块的初始设定 .....</b>	<b>4-12</b>
4-3-1	概要 .....	4-12
4-3-2	设定方法 .....	4-13
<b>4-4</b>	<b>EtherCAT 主机功能模块的初始设定 .....</b>	<b>4-14</b>
<b>4-5</b>	<b>EtherNet/IP 功能模块的初始设定 .....</b>	<b>4-15</b>

## 第 5 章 设计任务

<b>5-1</b>	<b>任务设计步骤的概要 .....</b>	<b>5-2</b>
<b>5-2</b>	<b>任务的概要 .....</b>	<b>5-4</b>
5-2-1	任务定义 .....	5-4
5-2-2	任务相关的指令 .....	5-6
5-2-3	任务相关的系统定义变量 .....	5-6
<b>5-3</b>	<b>NY 系列的任务规格和基本动作 .....</b>	<b>5-8</b>
5-3-1	NY 系列的任务规格 .....	5-8
5-3-2	NY 系列的任务分配说明 .....	5-8
5-3-3	NY 系列的任务基本动作 .....	5-9
5-3-4	NY 系列事件任务的执行条件 .....	5-15
5-3-5	NY 系列事件任务的执行时序 .....	5-20
5-3-6	事件任务完成前，执行条件再次成立时的动作 .....	5-24
<b>5-4</b>	<b>标签数据链接服务和系统服务 .....</b>	<b>5-25</b>
5-4-1	标签数据链接服务和系统服务的执行优先度和执行顺序 .....	5-26
5-4-2	标签数据链接服务的处理内容和执行时序 .....	5-28
5-4-3	系统服务的处理内容和执行时序 .....	5-30
<b>5-5</b>	<b>任务相关的分配和设定 .....</b>	<b>5-31</b>
5-5-1	分配到 I/O 刷新的任务中 .....	5-31
5-5-2	将程序分配到任务中 .....	5-37
5-5-3	任务的设定项目 .....	5-38
<b>5-6</b>	<b>确保变量的同时性 .....</b>	<b>5-40</b>
5-6-1	确保任务间变量值同时性的方法 .....	5-40
5-6-2	从控制器外部进行的变量访问 .....	5-46
<b>5-7</b>	<b>任务相关的异常 .....</b>	<b>5-50</b>
<b>5-8</b>	<b>任务执行状态、任务执行时间的监视 .....</b>	<b>5-52</b>
<b>5-9</b>	<b>任务设计方法及输入输出响应时间 .....</b>	<b>5-56</b>
5-9-1	任务执行时间的确认 .....	5-56
5-9-2	任务设计示例 .....	5-57
5-9-3	系统输入输出响应时间 .....	5-58

## 第 6 章 编程

<b>6-1</b>	<b>编程步骤概要 .....</b>	<b>6-3</b>
<b>6-2</b>	<b>POU (程序构成单位) .....</b>	<b>6-5</b>
6-2-1	POU 定义 .....	6-5
6-2-2	三个类型 POU 的概要 .....	6-6
6-2-3	程序、功能块、函数的区别 .....	6-7
6-2-4	程序的详情 .....	6-8
6-2-5	功能块 (FB) 的详情 .....	6-9
6-2-6	函数 (FUN) 的详情 .....	6-17
6-2-7	功能块 (FB)、函数 (FUN) 共通动作 .....	6-23
6-2-8	POU 的限制事项 .....	6-25
<b>6-3</b>	<b>变量 .....</b>	<b>6-27</b>
6-3-1	变量定义 .....	6-27

6-3-2	变量的种类	6-27
6-3-3	POU 对应的用户定义变量的种类	6-28
6-3-4	变量的属性	6-29
6-3-5	数据类型	6-30
6-3-6	派生数据类型	6-40
6-3-7	数据类型的排列 / 范围指定	6-50
6-3-8	变量的各属性说明	6-56
6-3-9	与状态变化对应的变量值的变化	6-62
6-3-10	功能块 (FB) 的实例	6-72
6-3-11	变量值的监视	6-72
6-3-12	变量名称等程序相关名称的限制	6-73
<b>6-4</b>	<b>常数 (文本)</b>	<b>6-75</b>
6-4-1	常数定义	6-75
6-4-2	各数据类型的记述形式	6-75
<b>6-5</b>	<b>编程语言</b>	<b>6-80</b>
6-5-1	编程语言定义	6-80
6-5-2	梯形图语言 (LD)	6-80
6-5-3	ST 语言	6-86
<b>6-6</b>	<b>指令</b>	<b>6-124</b>
6-6-1	指令定义	6-124
6-6-2	指令的基本知识	6-124
6-6-3	指令错误	6-127
<b>6-7</b>	<b>名称空间</b>	<b>6-133</b>
6-7-1	名称空间定义	6-133
6-7-2	名称空间的规格	6-134
6-7-3	名称空间的使用步骤	6-137
<b>6-8</b>	<b>库</b>	<b>6-138</b>
6-8-1	库的概要	6-138
6-8-2	库的规格	6-139
6-8-3	库部件的规格	6-140
6-8-4	库的使用步骤	6-141
<b>6-9</b>	<b>创建用户程序时的注意事项</b>	<b>6-142</b>
6-9-1	输入变量 / 输出变量 / 输入输出变量为排列指定的变量时	6-142
6-9-2	输入变量 / 输出变量 / 输入输出变量为结构体型变量时	6-142
6-9-3	主控	6-143

## 第 7 章 动作确认和运行

<b>7-1</b>	<b>动作确认和运行的步骤概要</b>	<b>7-2</b>
<b>7-2</b>	<b>离线调试</b>	<b>7-3</b>
7-2-1	模拟功能的特点	7-3
7-2-2	模拟功能的执行	7-3
7-2-3	模拟功能的设定	7-6
<b>7-3</b>	<b>上机动作确认和运行</b>	<b>7-8</b>
7-3-1	从上机动作确认到运行开始之间的步骤	7-8
7-3-2	项目的传送	7-9
7-3-3	I/O 的配线确认	7-9
7-3-4	MC 调试	7-9
7-3-5	用户程序的动作确认	7-10
7-3-6	正式运行的开始	7-10

## 第 8 章 控制器内置功能

<b>8-1</b>	<b>数据管理、时钟、运行相关功能</b>	<b>8-2</b>
8-1-1	存储器全部清除	8-2
8-1-2	时钟功能	8-2

<b>8-2</b>	<b>SD 存储卡功能</b>	<b>8-5</b>
8-2-1	SD 存储卡功能一览	8-5
8-2-2	共享文件夹和虚拟 SD 存储卡设定	8-6
8-2-3	虚拟 SD 存储卡的识别和识别解除	8-6
8-2-4	Windows 的共享文件夹及文件规格	8-8
8-2-5	SD 存储卡指令	8-9
8-2-6	FTP 客户端通信指令	8-10
8-2-7	FTP 服务器功能	8-10
8-2-8	通过 Sysmac Studio 进行的文件操作	8-10
8-2-9	SD 存储卡相关的系统定义变量一览	8-11
8-2-10	虚拟 SD 存储卡中文件访问的排他控制	8-12
<b>8-3</b>	<b>安全功能</b>	<b>8-13</b>
8-3-1	用户程序执行用 ID 认证功能	8-14
8-3-2	无用户程序复原信息传送功能	8-17
8-3-3	项目文件整体的保护功能	8-18
8-3-4	数据保护功能	8-19
8-3-5	操作权限的认证功能	8-21
8-3-6	CPU 单元的写入保护功能	8-23
8-3-7	CPU 单元名称功能及序列 ID 功能	8-24
<b>8-4</b>	<b>调试功能</b>	<b>8-27</b>
8-4-1	强制值刷新	8-27
8-4-2	当前值变更	8-31
8-4-3	在线编辑	8-33
8-4-4	数据追踪	8-35
8-4-5	微分监控	8-40
<b>8-5</b>	<b>事件日志功能</b>	<b>8-45</b>
8-5-1	概要	8-45
8-5-2	事件日志功能的详情	8-46
8-5-3	控制器事件 (控制器异常 / 信息)	8-51
8-5-4	用户事件 (用户异常 / 信息)	8-52
<b>8-6</b>	<b>事件重要程度变更功能</b>	<b>8-58</b>
8-6-1	事件重要程度变更的用途	8-58
8-6-2	可变更事件重要程度的事件	8-58
8-6-3	事件重要程度变更的操作方法	8-59

## 第 9 章 备份功能

<b>9-1</b>	<b>备份功能的概要</b>	<b>9-3</b>
9-1-1	备份功能的用途	9-3
9-1-2	备份功能的对象数据	9-4
9-1-3	备份功能的种类	9-5
9-1-4	备份功能的种类和数据表的关系	9-7
9-1-5	备份功能的对象范围	9-8
<b>9-2</b>	<b>SD 存储卡的备份功能</b>	<b>9-10</b>
9-2-1	备份 (从控制器到虚拟 SD 存储卡)	9-11
9-2-2	核对 (控制器和虚拟 SD 存储卡之间)	9-15
<b>9-3</b>	<b>SD 存储卡备份禁止功能</b>	<b>9-19</b>
<b>9-4</b>	<b>Sysmac Studio 控制器备份功能</b>	<b>9-20</b>
9-4-1	备份 (从控制器到计算机)	9-21
9-4-2	恢复 (从计算机到控制器)	9-22
9-4-3	核对 (控制器和计算机之间)	9-23
<b>9-5</b>	<b>Sysmac Studio 备份文件的导入 / 导出功能</b>	<b>9-24</b>
<b>9-6</b>	<b>Sysmac Studio 变量 / 存储器的备份功能</b>	<b>9-25</b>
9-6-1	Sysmac Studio 变量 / 存储器备份功能的对象数据	9-25
9-6-2	Sysmac Studio 变量 / 存储器备份功能的操作方法	9-25
9-6-3	NY 系列控制器的型号兼容性	9-25
<b>9-7</b>	<b>连接 EtherCAT 从站时的备份功能</b>	<b>9-26</b>

9-7-1	EtherCAT 从站的备份对象数据	9-26
9-7-2	控制器在各种状态下的备份功能执行可否	9-27
9-7-3	EtherCAT 从站的恢复执行条件	9-28
9-7-4	备份功能的对象 EtherCAT 从站	9-29
<b>9-8</b>	<b>连接 EtherCAT 从站终端时的备份功能</b>	<b>9-31</b>
9-8-1	EtherCAT 从站终端的备份对象数据	9-31
9-8-2	EtherCAT 从站终端在各状态下的备份功能执行可否	9-32
9-8-3	EtherCAT 从站终端的恢复执行条件	9-32
<b>9-9</b>	<b>备份相关文件</b>	<b>9-33</b>
9-9-1	备份相关文件的种类	9-33
9-9-2	备份文件的规格	9-34
9-9-3	恢复指令文件的规格	9-35
9-9-4	控制器核对结果文件的规格	9-37
9-9-5	EtherCAT 核对结果文件的规格	9-38
9-9-6	EtherCAT 从站终端核对结果文件的规格	9-39
<b>9-10</b>	<b>备份功能的文件间兼容性</b>	<b>9-40</b>
9-10-1	备份功能间的兼容性	9-40
9-10-2	NY 系列控制器的型号兼容性	9-41
9-10-3	NY 系列控制器的单元版本兼容性	9-42
<b>9-11</b>	<b>无法与备份功能同时执行的功能</b>	<b>9-43</b>

## 第 10 章 通信设定

<b>10-1</b>	<b>通信系统的概要</b>	<b>10-2</b>
10-1-1	概要	10-3
<b>10-2</b>	<b>Sysmac Studio 的连接</b>	<b>10-4</b>
10-2-1	可在线连接的构成	10-4
10-2-2	无法在线连接的构成	10-5
<b>10-3</b>	<b>与其他控制器或从站连接</b>	<b>10-6</b>
10-3-1	控制器与控制器之间的连接构成	10-6
10-3-2	控制器与从站的连接构成	10-9
<b>10-4</b>	<b>与触摸屏的连接</b>	<b>10-10</b>

## 第 11 章 实际使用步骤示例

<b>11-1</b>	<b>实际使用示例</b>	<b>11-2</b>
11-1-1	系统构成	11-2
11-1-2	动作	11-2
<b>11-2</b>	<b>操作顺序示例的概要</b>	<b>11-3</b>
11-2-1	配线和设定	11-3
11-2-2	软件设计	11-3
11-2-3	Sysmac Studio 上的软件设定	11-4
11-2-4	Sysmac Studio 上的编程	11-8
11-2-5	上机确认动作和运行	11-9

## 第 A 章 附录

<b>A-1</b>	<b>规格</b>	<b>A-3</b>
A-1-1	性能规格	A-3
A-1-2	功能规格	A-6
<b>A-2</b>	<b>NY 系列中任务实际处理时间的计算标准</b>	<b>A-12</b>
A-2-1	任务实际处理时间平均值的计算方法	A-13
A-2-2	任务实际处理时间平均值的计算和任务周期的设定示例	A-19
<b>A-3</b>	<b>系统定义变量一览</b>	<b>A-22</b>

A-3-1	NY 系列控制器的系统整体（类别名称：无）	A-22
A-3-2	PLC 功能模块（类别名称：_PLC）	A-30
A-3-3	运动控制功能模块（类别名称：_MC）	A-31
A-3-4	EtherCAT 主机功能模块（类别名称：_EC）	A-32
A-3-5	EtherNet/IP 功能模块（类别名称：_EIP）	A-36
A-3-6	异常状态中各个位的含义	A-44
<b>A-4</b>	<b>系统定义变量的个别规格</b>	<b>A-45</b>
A-4-1	NY 系列控制器的系统整体（类别名称：无）	A-46
A-4-2	PLC 功能模块（类别名称：_PLC）	A-57
A-4-3	运动控制功能模块（类别名称：_MC）	A-59
A-4-4	EtherCAT 主机功能模块（类别名称：_EC）	A-61
A-4-5	EtherNet/IP 功能模块（类别名称：_EIP）	A-69
<b>A-5</b>	<b>控制器内数据的属性</b>	<b>A-79</b>
<b>A-6</b>	<b>变量的存储器确保方法</b>	<b>A-82</b>
A-6-1	变量的存储器确保规则	A-82
A-6-2	应注意的案例	A-91
<b>A-7</b>	<b>CX-Designer 的变量表登录方法</b>	<b>A-94</b>
<b>A-8</b>	<b>EtherCAT 从站或轴的有效 / 无效切换</b>	<b>A-97</b>
A-8-1	使用 EtherCAT 从站和轴时的项目设定	A-97
A-8-2	执行 EtherCAT 从站或轴的有效 / 无效切换的指令	A-97
A-8-3	表示 EtherCAT 从站或轴状态的系统定义变量	A-98
A-8-4	程序的执行 / 停止指令	A-99
A-8-5	确认程序的执行 / 停止	A-99
A-8-6	Sysmac Studio 上的设定	A-100
A-8-7	EtherCAT 从站或轴的有效 / 无效切换应用示例	A-101
<b>A-9</b>	<b>用户程序的大小相关的限制</b>	<b>A-104</b>
A-9-1	用户程序的各部件限制	A-104
A-9-2	用户程序的各部件数量	A-106
<b>A-10</b>	<b>NY 系列的版本相关信息</b>	<b>A-108</b>

## 索引



# 承诺事项

关于“本产品”，若无特殊协议，无论客户从何处购买，均适用本承诺事项中的条件。

## • 定义

本承诺事项中术语的定义如下所示。

- “本公司产品”：“本公司”的 FA 系统设备、通用控制设备、传感设备、电子和机械零件
- “产品样本等”：与“本公司产品”相关的欧姆龙工控设备、电子和机械零件综合样本、其他产品样本、规格书、使用说明书、手册等，还包括通过电磁介质提供的资料。
- “使用条件等”：“产品样本等”中的“本公司产品”的使用条件、额定值、性能、运行环境、使用方法、使用注意事项、禁止事项等
- “用户用途”：用户使用“本公司产品”的方法，包括直接使用或将“本公司产品”装入用户制造的零件、印刷电路板、机械、设备或系统等。
- “适用性等”：“用户用途”中“本公司产品”的（a）适用性、（b）动作、（c）不侵犯第三方知识产权、（d）遵守法律以及（e）遵守各种标准

## • 记载内容的注意事项

关于“产品样本等”中的内容，请注意以下几点。

- 额定值和性能值是在各条件下进行单独试验后获取的值，并不保证在复合条件下可获取各额定值和性能值。
- 参考数据仅供参考，并不保证在该范围内始终正常运行。
- 使用实例仅供参考，“本公司”不保证“适用性等”。
- “本公司”可能会因产品改良、本公司的原因而中止“本产品”的生产或变更“本产品”的规格。

## • 使用注意事项

使用时，请注意以下几点。

- 使用时请符合额定值、性能以及“使用条件等”。
- 请用户自行确认“适用性等”，判断是否可使用“本公司产品”。“本公司”对“适用性等”不作任何保证。
- 用户将“本公司产品”用于整个系统时，请务必事先自行确认配电、设置是否恰当。
- 使用“本公司产品”时，请注意以下各事项。（i）使用“本公司产品”时，应在额定值和性能方面留有余量，采用冗余设计等安全设计，（ii）采用安全设计，即使“本公司产品”发生故障，也可将“用户用途”造成的危险降至较低程度，（iii）对整个系统采取安全措施，以便向使用者告知危险，（iv）定期维护“本公司产品”及“用户用途”。
- 本公司设计并制造面向一般工业产品的通用产品。但是，不可用于以下用途。如果用户将“本公司产品”用于以下用途，则“本公司”不对“本公司产品”作任何保证。但如果属于本公司许可的特别产品用途或与本公司签订特殊协议的场合除外。
  - (a) 需高安全性的用途（例：原子能控制设备、燃烧设备、航空航天设备、铁路设备、起重设备、娱乐设备、医疗设备、安全装置以及其他危及生命、健康的用途）
  - (b) 需高可靠性的用途（例：煤气、自来水、电力等供应系统、24 小时连续运行的系统、支付系统等涉及权利、财产的用途等）
  - (c) 用于严格条件或环境下（例：需设置在室外的设备、会受化学污染的设备、会受电磁波干扰的设备、会受振动、冲击影响的设备等）
  - (d) 在“产品样本等”中未记载的条件或环境下使用
- 上述（a）～（d）以及“本产品样本等中记载的产品”不可用于汽车（含两轮车。下同）。请勿装入汽车进行使用。关于可装入汽车的产品，请咨询本公司销售负责人。

- **保修条件**

“本产品”的保修条件如下所述。

- 保修期为购买本产品后的 1 年内。  
(“产品样本等”中另有记载的情况除外。)
- 保修内容 对发生故障的“本产品”，经“本公司”判断后提供以下任一服务。
  - (a) 发生故障的“本产品”可在本公司维修服务网点免费维修  
(不提供电子和机械零件的维修服务。)
  - (b) 免费提供与发生故障的“本产品”数量相同的替代品
- 非保修范围 如果因以下任一原因造成故障，则不在保修范围内。
  - (a) 用于非“本产品”原本用途的用途时
  - (b) 未按“使用条件等”进行使用
  - (c) 违反本承诺事项中的“使用注意事项”进行使用
  - (d) 改造或维修未经“本公司”
  - (e) 使用的软件程序非由“本公司”人员编制
  - (f) 因以出厂时的科学技术水平无法预见的原因
  - (g) 除上述以外，因“本公司”或“本产品”以外的原因（包括自然灾害等不可抗力）

- **责任免除**

本承诺事项中的保修即与“本产品”相关的保修的所有内容。

对因“本产品”造成的损害，“本公司”及“本产品”的销售店概不负责。

- **出口管理**

出口“本产品”或技术资料或向非居民的人员提供时，应遵守日本及各国安全保障贸易管理相关的法律法规。如果用户违反上述法律法规，则可能无法向其提供“本产品”或技术资料。

# NY 系列 工业平板电脑 / 工业 BOX PC 用户手册 软件篇 (SBCA-CN5-436) 安全注意事项

## 安全信息的标识及其含义

为了安全使用 NY 系列控制器，本手册使用下列标识及图标说明注意事项。这里所记载的注意事项均为与安全有重大相关的内容。请务必遵守。

标识及含义如下所示。

 <b>警告</b>	操作不当时可能导致操作人员轻度、中度受伤，严重时可能导致重伤或死亡。此外还有可能引发重大财产损失。
---	---

 <b>注意</b>	如果不正确操作处理，本危险有时可能会导致轻伤、中等程度伤害，也可能造成物质损失。
---	--

## 图标说明



⊙符号表示禁止。  
具体内容参见⊙内和文字叙述。  
左图表示“禁止拆解”。



该三角形符号表示注意事项 (包括警告)。  
具体内容参见三角形内和文字叙述。  
左图表示“小心触电”。



该三角形符号表示注意事项 (包括警告)。  
具体内容参见三角形内和文字叙述。  
左图表示“一般注意事项”。



●符号表示强制。  
具体内容参见●内和文字叙述。  
左图表示“一般强制事项”。



该三角形符号表示注意事项 (包括警告)。  
具体内容参见三角形内和文字叙述。  
左图表示“注意高温”。

## 警告



### 分解和掉落

请勿分解、修理或改造产品。否则可能导致故障或起火。



### 安装时

安装时请务必进行第 3 种接地。



产品的安装和安装后的确认请由熟悉所安装机械的负责人进行。



## 故障安全对策

因产品故障或外部原因而导致异常时，请在外部采取安全措施，以确保系统整体的安全性。否则可能会因异常动作而导致严重事故。



请务必通过外部的控制回路构成紧急停止回路、联锁回路、限制回路等安全保护相关回路。



以下情况时，产品将关闭所有输出单元的输出，从站将按照从站侧的动作运行。

- 发生电源部异常时
- 发生 CPU 异常 (WDT 异常) 或 CPU 复位时
- 发生全部停止故障等级的控制器异常时
- 从接通电源到变为运行模式的启动过程中
- 发生系统初始化异常时

以上情况下，必须在外部采取措施以确保系统安全。



若从站等外部电源变为过载状态或短路状态，电压将下降，可能导致输出变为 OFF、输入无法导入。此时，必须在外部采取措施，以确保系统安全，如在必要时监视电源电压，并导入到控制中。



产品内部的数据异常时，可能发生意外动作。此时，必须在外部采取措施以确保系统安全。



请在通信系统或用户程序中采取措施，确保在数据链接通信或远程 I/O 通信中发生通信错误或误动作时的系统安全。



远程 I/O 中发生通信故障或发生全部停止故障等级的异常时，输出状态取决于所用产品的规格。

请在确认发生上述通信故障或全部停止故障等级的异常时，其动作规格如何，再采取安全对策。

关于 EtherCAT 从站，请分别正确地进行各“从站设定”。



如果使用 UPS，在发生瞬时停电时，也可正常运行，因此可能收到受瞬时停电影响的外部设备传来的错误信号。

请在外部采取故障安全对策，必要时，对外部设备的电源电压进行监视，并作为联锁条件导入。



请勿将触摸屏的输入功能用于与人命或重大损害相关的地方，或者作为紧急停止开关功能使用。



## 传送

---

通过 Sysmac Studio 传送用户程序、“构成 / 设定”数据、设备变量时，请在确认传送目标安全后再执行。  
无论产品处于什么动作模式，装置、机器都可能发生意外动作。



## 正式运行时

---

请对编写完成的用户程序及各种数据、设定值进行充分的动作确认后，再转移到正式运行。



安全设定请由对此熟悉的技术人员进行。  
非推荐的 OS 安全设定可能进一步提高系统的安全风险等级。



BIOS 设定的变更请由对此熟悉的技术人员进行。产品的动作可能会发生变化。



接触到湿的触摸屏时，触摸屏的输入精度会降低，可能发生误输入。  
请事先擦干触摸屏上的水分后再操作。



## 安全对策

### 防病毒保护

请在连接控制系统的电脑上安装最新版本的企业级杀毒软件并及时维护。



### 防止非法访问

请对本公司产品采取下列防范非法访问的措施。

- 导入物理控制，确保只有授权人员才能访问控制系统及设备
- 通过将控制系统及设备的网络连接限制在最低程度，防止未获信任的设备访问
- 通过部署防火墙，将控制系统及设备的网络与IT网络隔离（断开未使用的通信端口、限制通信主机）
- 如需远程访问控制系统及设备，应使用虚拟专用网络（VPN）
- 在控制系统及设备的远程访问中导入多重要素认证
- 采用复杂密码并频繁更换
- 如需在控制系统或设备上使用USB存储器等外部存储设备，应事先进行病毒扫描



### 数据输入输出保护

请确认备份、范围检查等妥当性，以防对控制系统和设备的输入输出数据受到意外修改。

- 检查数据范围
- 利用备份确认妥当性，完善还原准备，以防数据遭到篡改或发生异常
- 进行安全设计如紧急停机、应急运行等，以应对数据遭到篡改及异常情况



### 丢失数据的复原

请定期进行设定数据的备份和维护，以防数据丢失。



经由全局地址使用内部网络时，一旦连接至SCADA、HMI等未经授权的终端或未经授权的服务器，可能会面临恶意伪装、数据篡改等网络安全问题。请客户自行采取充分有效的安全防护措施以防范网络攻击，例如限制终端访问，使用配备安全功能的终端，对面板设置区域实施上锁管理等。



构建内部网络时，可能会因电缆断线、未经授权的网络设备的影响，导致通信故障的发生。请采取充分有效的安全防护措施，例如通过对面板设置区域实施上锁管理等方法，限制无权限人员对网络设备的物理访问。



使用配备SD存储卡功能的设备时，可能存在第三方通过拔出或非法卸载移动存储介质等方式非法获取、篡改、替换移动存储介质内的文件及数据的安全风险。

请客户自行采取充分有效的安全防护措施，包括但不限于对面板设置区域实施上锁管理、门禁管理等方式，以限制无权限人员对控制器的物理访问，对移动存储介质采取妥善的管理措施等等。

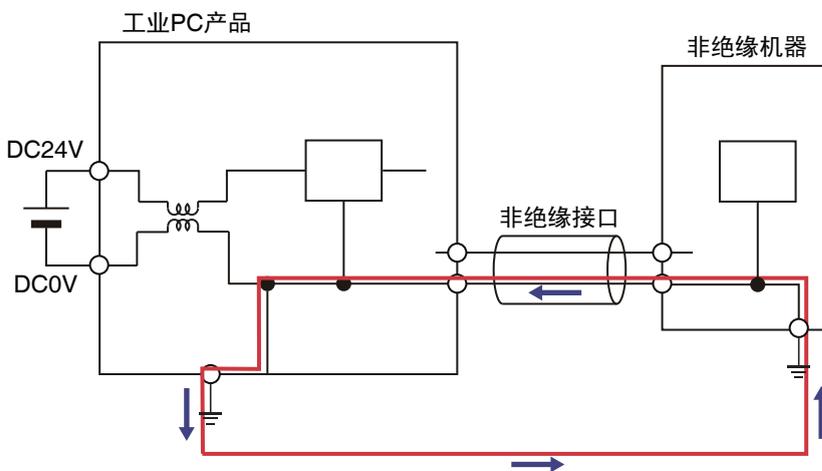


注意

**⚠ 注意**

**配线时**

产品的电源为非绝缘 DC 电源。内部的 0V 和 FG 已直接连接。连接非绝缘的外部设备时，请在外部采取安全措施。否则可能导致通信错误或产品故障。



请勿将外部电源的 24V 侧接地。24V-0V 之间会短路。



**在线编辑**

请确认输入输出时间打乱后也不会受到影响，再执行在线编辑。进行在线编辑后，任务执行时间可能超过任务周期，与外部之间的输入输出无法更新，导致输入信号无法读取、输出时间紊乱。



# 安全要点

## 分解、掉落、安装时、设置时、保管时

- 请勿使产品掉落，或对其施加异常振动和冲击。否则可能导致产品故障、烧毁。
- 拆开时请进行外观检查，确认没有损伤。此外，请轻晃产品，确认没有异响。
- 请务必使用相关手册中指定的设备。
- 请勿使用产品规格范围外的设备。可能导致系统故障、误动作。
- 安装时，请确认安装方向和温度范围在规格范围内。如果在超出规格范围的条件下使用，可能导致产品无法正常动作。
- 将周边设备与产品连接时，请采取充分的干扰和静电屏蔽措施后，再进行安装。
- 产品必须安装在控制柜内。
- 可安装的面板厚度为1.6~6.0mm。为保持防水及防尘性能，安装支架请用0.6N·m的转矩均匀拧紧。如果以其他转矩拧紧或未均匀拧紧，可能导致面板变形。  
请使用没有脏污、扭曲、有足够安装强度的面板。
- 面板加工时，注意防止金属屑进入产品内部。请勿使配线废料或碎屑等异物进入产品内部。否则会导致烧毁、故障、误动作。特别是施工时，请用物品覆盖。

## 配线时

- 连接器的配线及连接请按照本手册正确操作。对配线顺序、连接器的方向充分确认后，再进行通电。
- 如连接器、电缆、PCIe卡、存储设备等带有锁定机构，则请务必确认锁紧后再使用。否则可能在运行过程中脱落。
- 连接计算机和产品时，请先从AC插座上拔掉计算机的电源插头。此外，若计算机上有FG端子，连接时请确保其与产品的功能接地端子为同电位。  
否则计算机和产品之间会出现电位差，可能导致故障或误动作。
- 请勿强行扭曲或拉拽电缆。  
请勿在电缆的电线上放置重物。否则可能导致断线。
- 为防止电压下降或烧毁，请使用足够大的电源电线。电源配线时请注意电线的电流容量。否则可能导致发热或起火。若端子和端子采用交叉配线，各电线中流过的电流为各条线相加的值。采用交叉配线时，请注意所有电线的电流容量。
- 请务必按照本手册或参考手册中指定的扭矩紧固安装支架螺钉和电缆螺钉。螺钉松动可能导致起火及误动作。
- 请使用压接端子进行配线。
- 电源电缆请使用 $0.2\text{mm}^2 \sim 2.5\text{mm}^2$ 大小的电缆。请将电缆的包皮剥去7mm，再将其连接到连接器上。

- 请遵守以下事项，注意防止断线。
  - 剥离包皮时，请勿损伤芯线。
  - 芯线请直接连接。
  - 芯线无需焊接，请直接连接。否则可能因振动而断裂。

## 设计电源时、接通电源时、关闭电源时

- 系统中使用的电源请勿超出本手册中记载的电源额定范围。
- 使用的 DC 电源应具备以下条件：电压变动小，即使输入发生 10ms 的瞬停亦可稳定地输出，且采取强化绝缘或双重绝缘。额定电源电压为 DC24V（容许范围 DC19.2 ~ 28.8V）。
- 请勿进行耐电压测试。
- 为防止产品断电时发生数据损坏或系统文件不匹配，请使用 UPS。同时，为防止误操作等引起数据损坏或系统文件不匹配，请有计划地备份文件。
- 为正常关闭产品，请使用修订号合适的欧姆龙产 S8BA UPS。
- 从接通电源到变为运行模式约需 10 ~ 20 秒。在此期间的输出遵照从站或单元的规格。请设计故障安全回路，避免外部设备发生误动作。
- 请在连接监视和产品的 DVI 电缆后，再打开电源。
- 接通电源前，请确认电源或电源的配线。错误的电源或配线可能导致故障或起火。
- 人体或有导电性的物品接触触摸屏时，请勿接通产品的电源。触摸屏功能将无法使用。要恢复触摸屏的功能，请先清除有导电性的物品，再重新接通电源。
- 进行下列操作时，请关闭电源。
  - 安装或取出 PCIe 卡时
  - 连接电缆或进行配线时
  - 进行系统配线时
  - 连接或断开连接器时
  - 安装或取出 HDD/SSD 时
  - 电池的更换
  - 风扇单元的更换

## 正式运行时

- 请设定操作系统密码，防止非法访问。
- 请牢记操作系统的用户名和密码。操作产品需要密码。
- 进行正式运行前，请确认是否已安装合适的软件且已正确设定。否则可能导致误动作。
- 请确认已安装最新的浏览器。
- 请确认已安装最新的防火墙。
- 请确认装置的操作系统环境受到保护，可防止外部的病毒和非法访问。
- 请确认已安装最新的病毒定义文件。
- 电源处于打开状态时，请勿拆下风扇罩。如果接触运行中的风扇，可能导致受伤。
- 虚拟内存的设定可能会影响系统的性能。安装或升级软件后，请将页面文件设定为无效。
- 请正确地进行配线和设定，确认可通过 UPS 执行关机。
- 请使用 RescueDiskTool，创建产品的恢复数据。可以用它恢复 HDD/SSD 中的设定。

## 操作时

---

- 在进行以下任何一项操作前，请确认其不会对设备造成不良影响。
  - 变更产品的动作模式（包括接通电源时的动作模式设定）
  - 用户程序、设定的变更
  - 设定值 / 当前值的变更
  - 强制值刷新
- 产品正在访问 **USB** 设备或 **SD** 卡时，请勿进行以下操作。
  - 关闭产品的电源
  - 按下电源按钮
  - 拔出 **USB** 设备或 **SD** 卡
- 联合 **PCIe** 连接器上连接的主板和 **EtherCAT** 从站等不同的功能模块对装置进行控制时，请在用户程序或外部采取对策，确保在一个功能模块停止时也不会造成影响。各功能模块发生部分停止故障等级的异常时，相应的输出遵照从站或输出单元的规格。
- 通电过程中或刚关闭电源后，请勿更换风扇单元。如果更换，可能被高温部位烫伤或因旋转单元而受伤。
- 长按电源按钮数秒钟，可强制关闭产品。但这样可能导致数据损坏。请有计划地备份文件。
- 通电过程中或刚关闭电源后，请勿接触产品的机框。否则可能导致烫伤。
- 请在确认系统安全后，再操作触摸屏。
- 如果连续高速地接触摸屏，可能无法输入。请在确认一次输入后，再进行下一次输入操作。
- 背光熄灭时、画面上没有显示时或屏幕太暗无法识别时，请勿随意接触摸屏。
- 请勿用硬物或尖锐物接触或刮擦显示部。否则可能会损坏屏幕表面。

## 一般通信

---

- 数据链接表设定不当时，可能导致设备动作异常。此外，即使设定了适当的数据链接表，也请在确认不会对设备产生不良影响后，再执行传送。传送后将自动开始数据链接。
- 装置的网络和办公室网络请分开。否则可能导致误动作。

## EtherNet/IP 通信

- EtherNet/IP 请以规格范围内的通信距离及连接台数 / 连接方法使用。此外，请勿连接 EtherCAT 等其他网络。否则可能因过载导致网络崩溃或误动作。
- 若从编程设备传送内置 EtherNet/IP 的设定（IP 地址或标签数据链接的设定），设定对象的所有 EtherNet/IP（节点）将重置。这是因为读入设定内容后生效。请在重置后确认不会对设备造成影响，再执行传送。
- 若使用中继集线器进行 EtherNet/IP 的标签数据链接通信（周期链接通信），网络的通信负载会升高，容易发生冲撞（冲突），将无法稳定地通信。在使用标签数据链接的网络中，请务必使用交换式集线器。

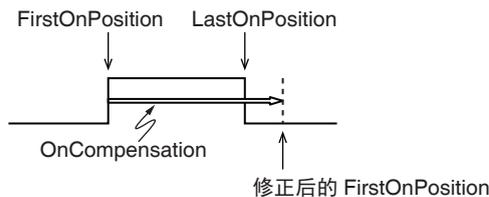
## EtherCAT 通信

- 因 EtherCAT 主机和从站的修订版本组合不同，可能导致误动作或意外动作。将网络设定的修订版本检查功能设为无效后使用时，请通过 Sysmac Studio 确认主机上设定的从站修订版本和实际从站的修订版本，并查阅从站点的手册等，确认功能的兼容性。此外，实际从站的修订版本可通过 Sysmac Studio 或铭牌确认。
- “低效运行设定”设定为“停止”时，若任一从站发生 EtherCAT 通信异常，所有从站和过程数据通信将停止。此时，伺服驱动器的动作遵照伺服驱动器的规格。请充分检验“低效运行设定”，确保装置发生异常时能安全运行。
- 发生于干扰时或从 EtherCAT 从站上拔下网络电缆时，正在通信的帧可能会丢失。帧丢失时，将无法传送从站的输入输出数据，可能发生意外动作。从站的输出遵照从站的规格。详情请参考各从站的手册。需要采取干扰对策或更换从站时，请执行以下处理。
  - 请将系统定义变量的输入数据无效作为联锁条件来创建用户程序。
  - 请务必将 EtherCAT 主机设定项目中的“PDO 通信连续超时检测此时”设定为 2 次以上。详情请参考 [□《NY 系列 工业平板电脑 / 工业 BOX PC 用户手册 内置 EtherCAT® 端口 用户手册 \(SBCD-CN5-368\)》](#)。
- 刚接通电源时，EtherCAT 的通信可能尚未建立。请利用系统定义变量确认通信建立后，再创建用于控制的程序。
- EtherCAT 从站设定为“脱离”或“无效”时，该从站之后的从站都将停止通信，无法控制输出。请在确认不会对设备造成影响后，再发出“脱离”、“无效”的指示。
- 请勿使用通用 Ethernet 集线器、中继集线器。若使用，可能发生全部停止故障等级的异常。
- 请务必使用指定的 EtherCAT 从站电缆。若使用非指定的电缆，EtherCAT 主机或 EtherCAT 从站可能检测出异常，出现以下现象。
  - 过程数据通信的刷新无法持续执行。
  - 过程数据通信的刷新无法在设定周期内完成。

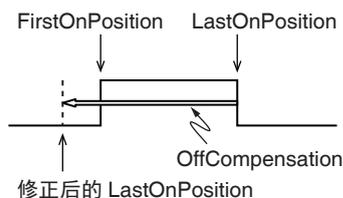
## 运动控制

- 若 MC 调试过程中工具 - 产品之间的通信中断，将强制停止电机。请注意连接电缆未松脱，确认不会对设备造成影响后，再执行 MC 调试。
- 将伺服驱动器的外部输入中分配的正转侧驱动禁止输入（POT）、反转侧驱动禁止输入（NOT）、原点附近输入（DEC）分别作为正转侧极限输入、反转侧极限输入、原点附近输入使用。这些输入信号的信号宽度请务必确保大于控制周期。若输入信号的宽度小于控制周期，将无法检测到输入信号，导致无法正常动作。
- 通过CPU单元的用户程序或Sysmac Studio变更了凸轮数据的值时，请务必执行凸轮表保存指令。若不执行，下次接通电源时，凸轮数据的值将恢复为变更前的状态，装置可能发生意外动作。
- 进行 MC 调试时，请仔细确认轴编号。
- 请在确认 MC\_DigitalCamSwitch（数字凸轮开关有效）指令的 InOperation（操作中）变为 TRUE 后，使用 NX\_AryDOOutTimeStamp（时间戳数字输出排列写入）指令。
- 请务必在很定速度的轴上使用 MC\_DigitalCamSwitch（数字凸轮开关有效）指令。  
若将计数模式变更为旋转模式，使用 OnCompensation（ON 时间补偿）或 OffCompensation（OFF 时间补偿），轴速度发生急剧变化，将发生以下动作。

- OnCompensation（ON 时间补偿）或 OffCompensation（OFF 时间补偿）的值超出轴旋转半圈的时间时，InOperation（操作中）将变为 FALSE。
- OnCompensation（ON 时间补偿）的值超出 LastOnPosition（ON 结束位置）时，输出时间将变得不确定。



- OffCompensation（OFF 时间补偿）的值超出 FirstOnPosition（ON 开始位置）时，输出时间将变得不确定。



## 恢复时

- 产品利用非易失性存储器来保持绝对编码器的信息，即“绝对编码器原点位置偏置”。满足以下任一条件时，请将绝对编码器原点位置偏置从恢复对象中排除，再执行恢复。然后，请重新设定绝对编码器原点。若未设定绝对编码器原点，控制对象可能发生意外动作。
  - 伺服电机或驱动器的个体变为创建备份时的个体。
  - 创建备份后，执行了绝对编码器的设置。
  - 绝对编码器的绝对值数据将丢失。
- 从站或单元中，所有或部分设定数据可能不在备份、恢复、核对的对象范围内。此外，设定为无效的从站或单元也不在备份、恢复、核对的对象范围内。使用恢复功能时，请进行充分的动作确认后再转移到正式运行。

## 更换电池时

- 若电池掉落到地板上或受到过度冲击，请丢弃该电池。受过冲击的电池在使用过程中会发生漏液。
- UL 规格规定，电池务必由熟练的技术人员进行更换。更换作业请由熟练的技术人员负责。
- 切勿短接电池的正负极端子或对电池进行充电、拆解、加热或焚烧。请勿使电池受到强烈的冲击。否则会导致电池漏液、破裂、发热或起火。

## 更换风扇单元时

- 风扇单元在未通电状态保管超过 6 个月时，请在确认能正常动作后再使用。

## 更换产品时

- 更换产品时，请先将开始运行所需的用户程序“构成/设定”的设定数据、变量传送到更换后的产品及外部连接的设备后，再开始运行。标签数据链接的设定、路由表也保存在产品中，也请传送。

## 清洁、检查和废弃时

- 清洁产品时，请勿使用有腐蚀性的化学药品。否则会导致故障、误动作。
- 在有油或水的环境中使用时，请定期确认安装状态。
- 橡胶包装会因使用环境发生老化、收缩或硬化，因此建议定期检查。
- 对产品和电池的废弃处理有时受到当地法规限制。请遵照当地适用法规进行废置处理。



废电池请回收

- 装有锂一次电池（高氯酸含量 6ppb 以上）的产品经过或出口至美国加利福尼亚州时，有标示以下信息的义务。

Perchlorate Material - special handling may apply.

See [www.dtsc.ca.gov/hazardouswaste/perchlorate](http://www.dtsc.ca.gov/hazardouswaste/perchlorate)

- 产品中配备有锂一次电池（高氯酸含量 6ppb 以上）。向美国加利福尼亚州出口装有产品的成品时，请在个装箱、运输箱等包装上标示上述内容。

# 使用注意事项

## 保管、设置和安装时

- 请勿安装或保管在下列场所。否则可能导致运行停止、误动作。
  - 日光直射的场所；
  - 环境温度或相对湿度超出规格中规定范围的场所；
  - 温度变化剧烈容易引起结露的场所；
  - 有腐蚀性气体、可燃性气体的场所；
  - 尘土、粉尘、盐分、铁屑较多的场所；
  - 有水、油、化学品等飞沫喷溅的场所；
  - 直接致使产品产生振动或冲击的场所；
  - 室外受风雨直接影响的场所；
  - 紫外线较强的场所。
- 请勿安装在发热体附近，并确保通风。
- 在下列场所使用时，请充分采取遮蔽措施。
  - 产生强高频干扰的设备附近；
  - 可能因静电等产生干扰的场所；
  - 产生强电场或磁场的场所；
  - 可能受到辐射的场所；
  - 附近有电源线通过的场所；
- 接触产品前请先采取触摸接地金属等除静电措施。
- 请正确安装周边设备。否则会导致烧毁、故障、误动作。
- 与 PCIe 连接器连接的主板可能因主板本身或关联软件而发生启动不良或动作不良。请获取连接主板前的备份并仔细确认动作。
- 背光有使用寿命。如果超过使用寿命，亮度会降低。请定期确认，并在必要时更换。

## 配线时

- 产品请在额定电源电压下使用。
- 请勿使配线废料或碎屑等进入产品内部。否则会导致烧毁、故障、误动作。特别是施工时，请用物品覆盖。
- 可使用软件的确认请由熟悉可用软件的负责人进行。
- 请勿使用长度超出规格范围的电缆。否则可能导致通信不良。
- 请勿在 DC 电源连接器上连接 AC 电源。

## 设计电源时、关闭电源时

- 正在通过工具传送各种数据时，请勿拔掉电缆或关闭产品的电源。否则可能导致数据传送无法完成，发生误动作。

## 正式运行时、操作时

- 升级产品上搭载的操作系统或更新周边设备的驱动程序后，产品可能出现与更新前不同的动作。请仔细确认动作。
- 产品运行过程中，请确认风扇处于运行状态。
- 存储设备、SD 存储卡、电源按钮、风扇单元、电池均有使用寿命。如果超过使用寿命，可能导致产品故障或误动作。
- 请对风扇的状态进行监视。如果使用时间超过风扇的寿命，将显示风扇转速降低的警告信息，产品可能发生过热并停止。
- 请对电池的电压水平进行监视。如果没有电池的状态持续太久，将无法保持时钟的运行。
- 部分手套可在穿戴的状态下操作触摸屏。请在正式运行前，佩戴该手套操作触摸屏，确认可正常操作。
- 更改密码时，请勿在写入完成前进行重置或切断电源。如果密码保存失败，项目可能无法运行。
- 要对 EtherCAT 主机功能模块解除部分停止故障等级以上的控制器异常时，请充分确认连接目标的设备安全。  
因部分停止故障等级以上的控制器异常而变为可操作状态以外（输出无效）的从站将变为可操作状态，输出变为有效。  
解除所有异常或重启从站时，请确认 EtherCAT 主机功能模块是否发生部分停止故障等级的控制器异常。
- 产品的单元版本不同，所配备的功能也不同。使用升级后添加的功能时，需要使用对应版本的 Sysmac Studio。产品的单元版本种类和 Sysmac Studio 版本的关系以及单元版本对应的支持功能一览请参考“NY 系列的版本相关信息”。
- 触摸屏最多可同时输入 5 点。同时输入超过 5 点时，将无法检测到。
- 静电容量方式的触摸屏如果其表面接触到导电物，就会进行输入。请注意避免不小心接触。否则可能导致误输入。

## 异常处理

- 在使用异常状态获取指令检测结果的应用程序中，请在编程时考虑检测时对设备产生的影响。例如，在动作过程中更换了电池等，即使检测到轻微异常，因程序的关系，也可能对动作造成影响。
- 变更了控制器异常的重要程度时，发生异常时的输出状态可能因变更而发生变化。请在确认安全后使用。

## 恢复时

- 编辑“恢复指令文件”时，请只记述可选择的数据组的“yes/no”。若编辑其他记述后执行恢复，控制器可能发生意外动作。

## 任务设置

---

- 发生任务周期超限时，请在创建程序时确保在任务周期内，或设定任务周期。

## 运动控制

---

- 开始 MC 调试时，请在确认动作参数正确后，再开始 MC 调试动作。
- MC 调试过程中，请勿下载运动设定。

## EtherCAT 通信

---

- 请设定伺服驱动器，使其在控制器和伺服驱动器之间的 EtherCAT 通信发生异常时停止动作
- 若要在运行过程中拔出 EtherCAT 从站电缆，请先脱离 EtherCAT 从站，或将后续 EtherCAT 从站在内的从站设为无效后再拔出。
- 要解除 EtherCAT 主机功能模块的“网络配置核对异常”、“过程数据通信异常”、“链接断开异常”时，请事先确认所有恢复对象的从站都处于加入状态。若有未加入的从站，EtherCAT 主机功能模块可能会访问指定节点地址以外的从站，或无法正确解除异常。
- EtherCAT 请以规格范围内的通信距离及连接台数 / 连接方法使用。  
此外，请勿连接 EtherNet/IP 或普通局域网等其他网络。否则可能因过载导致网络崩溃或误动作。
- 若传送用户程序，产品会重启，与 EtherCAT 从站的通信将中断。在此期间的从站输出遵照从站的规格。因 EtherCAT 网络配置不同，通信中断的时间也不同。传送用户程序时，请确认不会对装置造成影响后再执行。

## 更换电池时

---

- 长期闲置的产品应在更换电池后再接通电源。更换电池后，若一次都不打开电源继续放置，可能导致电池寿命缩短。
- 更换电池时，请使用指定的电池组。
- 更换电池前，请通电 5 分钟以上，并在电源关闭后 5 分钟内更换新电池。若无电池的状态持续 5 分钟以上，时钟信息可能会丢失。更换电池后，请确认时钟信息。

## SD 存储卡

---

- 将 SD 存储卡插到底。

## 清洁和检查

---

- 请勿使用汽油、稀释剂等挥发性溶剂和经过化学处理的抹布等。
- 用水清洁时，请关闭产品或将触摸屏的输入设为无效。

# 法规和标准

## 日本国外的使用

出口（或向非居住者提供）本产品中符合外汇及外国贸易法规定的出口许可、批准对象货物（或技术）要求的产品时，须依照该法获得出口许可、批准（或劳务交易许可）。

## 符合 EU 指令

### 适用指令

- EMC 指令

### 概念

#### ● EMC 指令

由于 OMRON 公司的产品为安装在各种机械、制造装置中使用的电气设备，为了使安装 PLC 的机械、装置更容易符合 EMC 标准，必须先使产品本身达到相关 EMC 标准（\*）。

但是，由于客户使用的机械、装置各不相同，EMC 性能因符合 EU 指令的产品安装的设备和控制柜的构成、配线及配置等条件而异，无法在客户使用状态下确认其适合性。因此，机械、装置整体最终的 EMC 适用性确认请客户自行实施。

\* EMC (Electro-Magnetic Compatibility: 电磁兼容性) 相关标准中，  
EMS (Electro-Magnetic Susceptibility: 电磁敏感度) 应符合 EN 61131-2  
EMI (Electro-Magnetic Interference: 电磁干扰) 应符合 EN 61131-2  
此外，Radiated emission 基于 10m 法。

#### ● 符合 EU 指令

NY 系列符合 EU 指令。为确保客户使用的机械、装置符合 EU 指令，必须注意下列几点。

- NY 系列必须安装在控制柜内。
- 连接到 DC 电源单元和 I/O 单元的 DC 电源，请使用 SELV 规格的电源。
- NY 系列同时符合 EU 指令和 EMI 相关的通用发射标准。特别是 Radiated emission(10m 法)，因控制柜的配置、连至控制柜的其它设备、接线和其它条件而异。  
因此，即使使用符合 EU 指令的 NY 系列时，客户也必须确认机械、装置整体是否符合 EU 指令。

## 软件许可和著作权

本产品中组装有第三方开发的软件。关于软件的许可和著作权，请参考 <http://www.fa.omron.com.cn/>。

# 版本

NY 系列及各 EtherCAT 从站的硬件或软件按照硬件修订版本或单元版本等编号管理。每次硬件或软件发生规格变更后，硬件修订版本或单元版本都会更新。因此，即使是同一型号的单元或 EtherCAT 从站，若硬件修订版本或单元版本不同，配备的功能或性能可能不同。

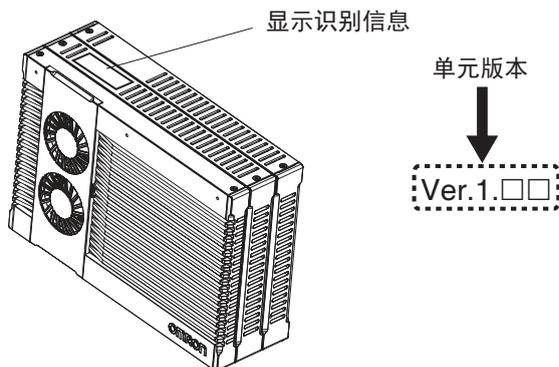
## 版本的确认方法

版本可通过标示的识别信息或 Sysmac Studio 确认。

### 通过标示的识别信息确认

可通过产品背面标示的识别信息确认版本。

NY 系列控制器 NY5□2-1□□□ 的识别信息标示如下图所示。



## 通过 Sysmac Studio 确认的方法

可通过 Sysmac Studio 确认版本。单元和 EtherCAT 从站的确认方法不同。

### ● NY 系列控制器的版本确认方法

单元版本可通过在线状态的 [生产信息] 确认。但是，只能确认控制器单元的版本。

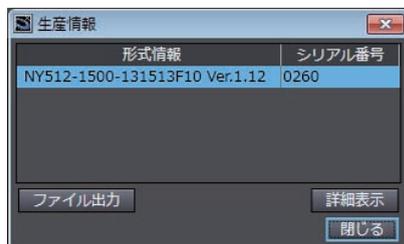
- 1 在多视图浏览器中右击 [配置和设定] - [CPU・扩展装置] - [CPU 装置]，选择 [显示生产信息]。

显示 [生产信息] 对话框。

### ● 切换生产信息的显示内容

- 1 选择 [生产信息] 对话框右下方的 [简易显示] 或 [详细显示]。

切换 [生产信息] 的简易显示和详细显示。



简易显示



详细显示

显示内容因简易显示还是详细显示而异。详细显示中，显示单元版本、硬件修订版本及各种版本。简易显示中，仅显示单元版本。

● EtherCAT 从站的版本确认方法

EtherCAT 从站的版本可通过在线状态的 [生产信息] 确认。确认方法如下所示。

- 1 在多视图浏览器中双击 [配置和设定] - [EtherCAT]。或者右击 [配置和设定] - [EtherCAT]，然后选择 [编辑]。

显示 EtherCAT 配置编辑画面。

- 2 右击 EtherCAT 配置编辑画面中的主机，选择 [显示生产信息]。

显示生产信息对话框。

单元版本以带“Rev”的形式显示。

● 切换生产信息的显示内容

- 1 选择 [生产信息] 对话框右下方的 [简易显示] 或 [详细显示]。

切换 [生产信息] 的简易显示和详细显示。



简易显示



详细显示

# 相关手册

本手册相关的手册如下表所示。请同时参考。

手册名称	Man.No.	型号	用途	内容
NY 系列 IPC 机器控制器 工业平板电脑 用户手册 硬件篇	SBCA-CN5 -435	NY532-1□□□	了解 NY 系列工业平板电脑的概要 / 设计 / 安装 / 保养等基本规格时。 主要是硬件相关的信息。	对 NY 系列的系统整体概要及工业平板电脑相关的如下内容进行说明。 <ul style="list-style-type: none"> <li>• 特长及系统构成</li> <li>• 概要</li> <li>• 各部分的名称和功能</li> <li>• 一般规格</li> <li>• 安装和配线</li> <li>• 维护检查</li> </ul>
NY 系列 IPC 机器控制器 工业 BOX PC 用户手册 硬件篇	SBCA-CN5 -434	NY512-1□□□	了解 NY 系列工业 BOX PC 的概要 / 设计 / 安装 / 保养等基本规格时。 主要是硬件相关的信息。	对 NY 系列的系统整体概要及工业 BOX PC 相关的如下内容进行说明。 <ul style="list-style-type: none"> <li>• 特长及系统构成</li> <li>• 概要</li> <li>• 各部分的名称和功能</li> <li>• 一般规格</li> <li>• 安装和配线</li> <li>• 维护检查</li> </ul>
NY 系列 IPC 机器控制器 工业平板电脑 / 工业 BOX PC 用户手册 设置篇	SBCA-CN5 -441	NY532-1□□□ NY512-1□□□	了解 NY 系列工业电脑的初始设定或控制器的使用准备时。	对 NY 系列的系统整体概要相关的如下内容进行说明。 <ul style="list-style-type: none"> <li>• 两个操作系统的原理</li> <li>• 初始设定</li> <li>• Industrial PC Support Utility</li> <li>• NYCompolet</li> <li>• Industrial PC API</li> <li>• 备份和恢复</li> </ul>
NY 系列 IPC 机器控制器 工业平板电脑 / 工业 BOX PC 用户手册 软件篇	SBCA-CN5 -436	NY532-1□□□ NY512-1□□□	了解 NY 系列工业电脑的控制功能的编程 / 系统启动时。	对 NY 系列的控制器功能相关的如下内容进行说明。 <ul style="list-style-type: none"> <li>• 控制器的动作</li> <li>• 控制器的功能</li> <li>• 控制器的设定</li> <li>• 基于 IEC 61131-3 的语言规格和编程</li> </ul>
NY 系列 指令基准手册 基本篇	SBCA-CN5 -437	NY532-1□□□ NY512-1□□□	了解 NY 系列工业电脑的基本指令规格详情时。	对各指令（IEC 61131-3 规格）的详情进行说明。
NY 系列 IPC 机器控制器 工业平板电脑 / 工业 BOX PC 用户手册 运动控制篇	SBCE-CN5 -379	NY532-1□□□ NY512-1□□□	了解 NY 系列工业电脑的运动控制的设定及编程思路时。	对用于运动控制的控制器的设定、动作及编程思路进行说明。
NY 系列 指令基准手册 运动篇	SBCE-CN5 -380	NY532-1□□□ NY512-1□□□	了解 NY 系列工业电脑的运动指令规格详情时。	对各运动控制指令的详情进行说明。
NY 系列 IPC 机器控制器 工业平板电脑 / 工业 BOX PC 用户手册 内置 EtherCAT® 端口篇	SBCD-CN5 -368	NY532-1□□□ NY512-1□□□	使用 NY 系列工业电脑的内置 EtherCAT 端口时。	对内置 EtherCAT 端口进行说明。 对概要、构成、功能、安装进行说明。
NY 系列 IPC 机器控制器 工业平板电脑 / 工业 BOX PC 用户手册 内置 EtherNet/IP™ 端口篇	SBCD-CN5 -369	NY532-1□□□ NY512-1□□□	使用 NY 系列工业电脑的内置 EtherCAT/IP 端口时。	对内置 EtherNet/IP 端口进行说明。 对基本设定、标签数据链接、其它功能进行说明。
NY 系列 故障排除手册	SBCA-CN5 -438	NY532-1□□□ NY512-1□□□	了解通过 NY 系列工业电脑检测的异常详情时。	对 NY 系列系统检出的异常的管理思路和各异常项目进行说明。
Sysmac Studio Version 1 操作手册	SBCA-CN5 -362	SYSMAC -SE2□□□	了解 Sysmac Studio 的操作方法、功能。	对 Sysmac Studio 的操作方法进行说明。

手册名称	Man.No.	型号	用途	内容
NX 系列 EtherCAT® 耦合器单元 用户手册	SBCD-CN5-361	NX-ECC□□□	希望了解 NX 系列 EtherCAT 耦合器单元及 EtherCAT 从站终端的使用方法时。	对 NX 系列 EtherCAT 耦合器单元和 NX 单元构成的 EtherCAT 从站终端的系统概要、配置方法，以及通过 EtherCAT 对 NX 单元进行设定、控制、监视所需的 EtherCAT 耦合器单元的硬件、设定方法和功能进行说明。
NX 系列 安全控制单元 用户手册	SGFM-CN5-710	NX-SL□□□□ NX-SI□□□□ NX-SO□□□□	了解 NX 系列安全控制单元的使用方法时。	对 NX 系列安全控制单元的硬件、设定方法和功能进行说明。
图像处理系统 FH/FZ5 系列 用户手册	SDNB-CN5-712	FH-1□□□ FH-3□□□ FZ5-L35□ FZ5-6□□ FZ5-11□□	希望了解图像处理系统 FH/FZ5 系列的使用方法时。	对使用 FH/FZ5 系列所需的软件功能、设定和操作进行说明。
视觉传感器 FQ-M 系列 定位专用视觉传感器 用户手册	SDNB-CN5-706	FQ-MS12□	希望了解定位专用视觉传感器的使用方法时。	对定位专用视觉传感器的硬件、设定方法和功能进行说明。
视觉传感器 FZ3 系列 用户手册	SCHB-CN5-762	FZ3-□□□□	希望了解视觉传感器 FZ3 系列的使用方法时。	对视觉传感器 FZ3 系列的软件功能、设定和操作进行说明。
位移传感器 ZW 系列 光纤同轴位移传感器 用户手册	SDNE-CN5-703	ZW-CE1□	希望了解位移传感器 ZW 系列的使用方法时。	对位移传感器 ZW 系列的硬件、设定方法和功能进行说明。
可编程终端 NA 系列 用户手册 软件篇	SBSA-CN5-546	NA5-□W□□□□	了解可编程终端 NA 系列的页面和各对象的功能。	对可编程终端 NA 系列的页面和各对象的功能进行说明。
可编程终端 NS 系列 编程手册	SBSA-CN5-512	NS15-□□□□□ NS12-□□□□□ NS10-□□□□□ NS8-□□□□□ NS5-□□□□□	了解可编程终端 NS 系列的使用方法。	对可编程终端 NS 系列的设定方法和功能进行说明。
CX-Designer 用户手册	SBSA-CN5-532	—	希望创建可编程终端 NS 系列的画面数据时	对 CX-Designer 的操作方法进行说明。

# 术语说明

对控制器功能相关的术语进行说明。

术语	说明
AT（分配目标）指定	变量的属性之一。 该属性表示由用户指定变量的分配目标。可指定 I/O 端口。
EtherCAT 主机功能模块	功能模块之一。作为 EtherCAT 主机控制 EtherCAT 从站。
EtherNet/IP 功能模块	功能模块之一。控制 EtherNet/IP 端口。
FB	功能块的简称。
FUN	函数的简称。
I/O 端口	控制器与外部设备（从站 / 单元）进行数据交换所需的逻辑接口。
I/O 映射设置	对 I/O 分配变量的设定。I/O 端口和变量之间的分配信息。
I/O 刷新	事先确定的存储器区域和外部进行周期性数据交换。
MC 调试	通过 Sysmac Studio 确认电机动作及配线的功能。
NX 单元	在 NX 系列的单元中，执行外部连接设备的 I/O 处理的单元。通信耦合器单元不包含在 NX 单元中。
PDO 通信	过程数据通信的简称。按过程数据通信周期（=主固定周期任务的任务周期），主机—从站之间周期性地进行数据交换。
PLC 功能模块	功能模块之一。进行用户程序的执行、向运动控制功能模块发出指示等。
POU	Program Organization Unit。IEC 61131-3 中程序执行模式的单位。 作为构成整个用户程序的基本单位，对算法和本地变量表进行记述。 包括程序、功能块、函数三种。
SDO 通信	EtherCAT 通信的一种，使用可在任意时间传输信息的服务数据对象（Service Data Objects: SDO）。
Sysmac Studio	计算机软件，进行 NY 系列的设定、编程、调试及故障排除。包括运动控制用的功能及模拟功能。
上传	利用 Sysmac Studio 的“同步”功能，从控制器向 Sysmac Studio 传送数据。
一般信息	表示控制器信息及用户信息的事件重要程度的等级之一。并非异常，但仍记录到事件日志中，并向用户通知的信息。
事件日志功能	识别事件（异常或现象），并记录下来。
事件设定	定义用户异常或用户信息。
事件任务	任务的执行条件成立时，执行一次用户程序。
联机 ST	将 ST 程序作为梯形图程序的一部分写入其中。或者是所写入的 ST 程序。
边缘	变量的属性之一。 BOOL 型变量的值从 FALSE 上升为 TRUE 时，或从 TRUE 下降为 FALSE 时，向功能块传递 TRUE 值。
虚拟 SD 存储卡	控制器功能对各种文件进行处理所需的共享文件夹。 Sysmac 的控制器 NJ/NX 系列 CPU 单元中使用 SD 存储卡，但在 NY 系列控制器中，将 Windows 的共享文件夹作为虚拟 SD 存储卡使用。 因此，有时会省略“虚拟”，称之为“SD 存储卡”。
凸轮数据变量	将凸轮数据转换为结构体排列后的变量。 结构体排列指定的变量，由相位和变位数据组成。
监视信息	表示控制器信息及用户信息的事件重要程度的等级之一。 对控制没有影响的轻微异常，但仍记录到事件日志中，并向用户通知的信息。
功能模块	按功能分割控制器软件构成的单位。
基本数据类型	IEC 61131-3 定义的数据类型。 包括布尔型、位列型、整数型、实数型、持续时间型、日期型、时刻型、日期时刻型、字符串型。 与用户定义的数据类型——“派生数据类型”相对的术语。
强制值刷新	用户进行调试等时，以强制值刷新来自外部的输入或向外部的输出。 即使是强制值刷新对象的区域，也可通过用户程序的指令覆盖。

术语	说明
联合体型	派生数据型的一种，使同一数据可以作为不同的数据类型处理。
全局变量	可从所有 POU（程序、功能块、函数）读写的变量。
轻度故障等级的控制器异常	NY 系列控制器的某些功能模块的部分功能无法控制的异常。 发生轻度故障后，NY 系列控制器将继续运行。
结构体型	派生数据型的一种，将多个不同的数据类型整合为一个。
常量	变量的属性之一。 变量指定为常量后，将无法通过指令、ST 语言的运算符或 CIP 信息通信进行值的写入。
控制器	是指由工业平板电脑 / 工业 BOX PC 的实时 OS 中装载的控制器功能直接控制的范围。 NY 系列系统中，包括 EtherCAT 从站（包括通用 / 伺服驱动器）在内，有时会称之为“控制器”。
控制器功能	是指执行工业平板电脑 / 工业 BOX PC 的实时 OS 中装载的机械控制所需的功能。
控制器异常	NY 系列系统中定义的异常。 “全部停止故障等级”、“部分停止故障等级”、“轻度故障等级”、“监视信息”的控制器事件的总称。
控制器事件	NY 系列系统的一种事件。系统定义为需要通知用户的异常、信息。系统检测到定义的发生原因后，发生控制器事件。
控制器信息	NY 系列系统中定义的信息，但并非异常。属于“一般信息”的控制器异常。
伺服 / 编码器输入从站	EtherCAT 从站中，可向“轴”分配的从站。NY 系列中，指伺服驱动器和编码器输入从站。
轴	运动控制功能模块中的功能单位。分配外部伺服驱动的驱动机构、编码器输入从站的检测机构等。
轴组	运动控制功能模块中，汇集了多个“轴”的功能单位。
轴组变量	结构体中定义的系统定义变量，如各轴组的各种状态信息、部分轴组参数设定信息等。 用于运动控制指令的轴组指定及轴组的指令插补速度、异常信息等的监视。
轴变量	结构体中定义的系统定义变量，如各轴的各种状态信息、部分轴参数设定信息等。 用于运动控制指令的轴指定及轴的指令当前位置、异常信息等的监视。
系统共通处理	与 I/O 刷新或用户程序执行一起，控制器在任务内部执行的系统处理。执行变量的任务间排他处理、数据追踪处理等。
系统服务	控制器所执行的任务以外的处理。执行通信处理、虚拟 SD 存储卡访问、自诊断处理等。
系统定义变量	所有属性由系统定义，用户无法变更的变量。
初始值	变量的属性之一。以下情况下设定的变量值。 <ul style="list-style-type: none"> <li>• 电源接通时</li> <li>• 切换到运行模式时</li> <li>• 指定为传送用户程序时初始化时</li> <li>• 发生全部停止故障等级的控制器异常时</li> </ul>
从站	执行远程 I/O 的主站（主机）相对的子站。
从站终端	在通信耦合器单元上安装 NX 单元群的构件块型远程 I/O 单元的总称。从站终端是从站的一种。
从站 / 单元构成	EtherCAT 构成及单元构成的总称。
绝对编码器原点位置偏置	控制器中保存的原点确定后的指令位置和从绝对编码器读取的绝对值数据的差，用于恢复带绝对编码器的伺服驱动器的当前位置。
全部停止故障等级的控制器异常	NY 系列控制器整体无法控制的异常。控制器将立即停止执行用户程序，断开包括远程 I/O 在内所有从站 / 单元的负载。
下载	利用 Sysmac Studio 的“同步”功能，从 Sysmac Studio 向控制器传送数据。
任务	“程序执行时的属性”。
任务周期	主固定周期任务及固定周期任务的执行间隔。
通信耦合器单元	NX 单元群和上位网络主机在网络上进行远程 I/O 通信所需的接口总称。例如，EtherCAT 耦合器单元为 EtherCAT 通信网络的通信耦合器单元。

术语	说明
固定周期任务	按一定周期执行 I/O 刷新和用户程序的任务。
设备变量	通过 I/O 端口访问特定设备所需的变量。
同步	自动比较 NY 系列控制器和 Sysmac Studio 的信息，分层显示是否有不同以及不同的位置和内容，并将信息统一
名称空间	分割和分层管理函数、功能块定义、数据类型名称的机制。
网络公开	变量的属性之一。 利用 CIP 信息通信或标签数据链接功能，可在其他控制器或上位计算机上读写变量。
排列指定	变量指定的一种。整合相同数据类型的多个要素后得到的一个变量。每个要素从前往后以编号（下标）指定。
派生数据类型	由用户定义规格的数据类型。包括结构体型、联合体型、列举型。
范围指定	变量指定的一种。明确表示只能提取预先设定范围内的变量值。
通用从站	EtherCAT 从站中，不可向“轴”分配的从站。
功能	运算处理等，编写输出与输入一对一的元件时使用的 POU。
功能块	定时器、计数器等，编写输出与输入因状态不同而不同的元件时使用的 POU。
部分停止故障等级的控制器异常	NY 系列控制器的某些功能模块整体无法控制的异常。 发生部分停止故障后，NY 系列控制器将继续运行。
主固定周期任务	优先度最高的任务。
程序	与函数、功能块并列，三种 POU 之一。 分配到任务后动作。
过程数据通信	EtherCAT 通信的一种，使用以固定周期定期进行实时信息交换的过程数据对象（Process Data Objects: PDO）。也称“PDO 通信”。
变量	用户程序中使用的、表示任意数值或字符串的概念。 代入任意值，可变更变量的值。与之相对的是值始终不变的“常数”。
变量存储器	无 AT（分配目标）指定的变量之当前值。只能通过“无 AT（分配目标）指定”属性的变量访问。
保持	变量的属性之一。在以下情况下设定了本属性时，将保持变量的值，若未设定，则为初始值。 <ul style="list-style-type: none"> <li>• 断电后接通电源时</li> <li>• 切换到运行模式时</li> <li>• 指定为传送用户程序时不初始化时</li> </ul>
指令	POU 的算法中记载的、欧姆龙指定的最小单位的处理要素。 指令中包含梯形图（接点、线圈）、FB 型、FUN 型、ST 语言的语句。
主存储器	控制器在使用 OS、执行用户程序时使用的控制器内部的存储器。
运动控制功能模块	功能模块之一。按照用户程序中的运动指令所赋予的指令，执行运动控制、
运动控制指令	执行运动控制功能所需的运动控制定义指令。 运动控制指令中，包括以 PLCopen <sup>®</sup> 的运动控制用功能块为标准的指令和运动控制功能模块独立的指令。
用户事件	NY 系列系统的一种事件。由用户定义的事件。用户异常、用户信息的总称。
用户定义变量	所有变量属性由用户定义，可由用户变更的变量。
用户程序	多个程序的集合体。
单元	可安装到通信耦合器单元上的设备。
单元构成	Sysmac Studio 中设定的单元构成信息，包括控制器上连接的单元种类和连接位置等信息。
文本	用户程序中使用的常数表现。
列举型	派生数据类型的一种。将事先准备的名称列表（列举值）中的一个作为值的数据类型。
列举值	将列举型变量可得到的多个值以文字的形式表示。 列举型变量的值取列举值中的任意一个。
本地变量	只能从定义的 POU 内部访问的变量。与之相对的是可从多个 POU 访问的“全局变量”。 本地变量中包括内部变量、输入变量、输出变量、输入输出变量、外部变量。

# 手册修订记录

手册修订符号在封面和封底中记载的 Man.No. 末尾标注。

Man.No. SBCA-CN5-436C

↑ 修订符号

修订符号	修订年月	修订理由和修订页码
A	2016 年 9 月	第一版
B	2021 年 1 月	修订错误
C	2023 年 4 月	增加安全对策的说明

# 1

## NY 系列控制器的概要

本章介绍 NY 系列控制器的特点、系统构成的概要、规格、使用步骤。

---

<b>1-1 NY 系列控制器定义</b> .....	<b>1-2</b>
1-1-1 特点 .....	1-3
1-1-2 系统构成概要 .....	1-5
<b>1-2 规格概要</b> .....	<b>1-7</b>
<b>1-3 产品型号</b> .....	<b>1-9</b>
<b>1-4 NY 系列的整体使用步骤</b> .....	<b>1-10</b>
1-4-1 步骤概要 .....	1-10
1-4-2 步骤详情 .....	1-11

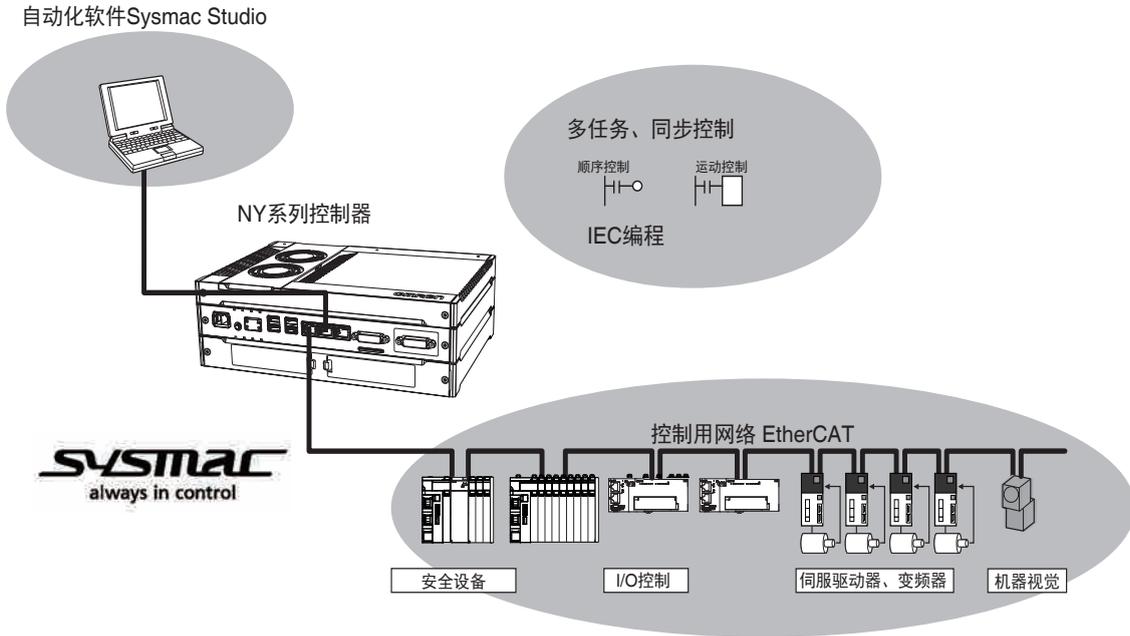
# 1-1 NY 系列控制器定义

工业电脑平台 NY 系列控制器是一款兼顾机械控制必需的功能、高速性能和工业控制器所需的安全性、可靠性、维护性的新型控制器。

包含传统可编程控制器的功能，还附带运动控制所需的各种功能，是一款综合型控制器。可在高速 EtherCAT 上同步控制安全设备、视觉装置、运动机器、I/O 等输入输出设备。

欧姆龙提供以统一的通信规格和用户接口规格设计的欧姆龙控制设备——Sysmac 设备。工业电脑平台 NY 系列可与 EtherCAT 从站等 Sysmac 设备以及自动化软件 Sysmac Studio 组合，实现较佳的性能和操作性。

在由 Sysmac 设备构成的系统中，机器与机器的连接非常方便，能够以统一的思路和实用工具操作。



## 1-1-1 特点

对 NY 系列 IPC 机器控制器 工业平板电脑 / 工业 BOX PC 的控制器功能的特点进行说明。

### 硬件的特点

#### ● 标配控制用网络 EtherCAT

NY 系列控制器标配 EtherCAT 通信主机功能端口。

EtherCAT 是一种以 Ethernet 系统为基础的高性能商用网络系统，可实现更高速、更高效的通信。各节点高速传输 Ethernet 帧，可实现恒定的短周期通信周期时间。

采用控制网络 EtherCAT，可通过一个网络连接机器控制所需的 I/O 系统、伺服驱动器、变频器、机器视觉等设备。

#### ● 支持 EtherCAT 从站终端

使用 EtherCAT 从站终端，节省安装空间。此外，利用种类丰富的 NX 单元，可灵活地构建系统。

#### ● 在 EtherCAT 上实现了安全子系统

通过使用 NX 系列安全控制单元，可在顺序控制、运动控制系统上，将安全控制作为 EtherCAT 上的子系统整合起来。

#### ● 标配 EtherNet/IP 通信功能端口

NY 系列控制器标配 EtherNet/IP 通信功能端口。

EtherNet/IP 为使用 Ethernet 的多供应商网络。可作为控制器之间的网络或现场网络使用。使用标准 Ethernet 技术，可与各种通用 Ethernet 设备一起使用。

#### ● 高硬件可靠性

备有与 PLC 同等的硬件可靠性和 RAS 功能。

#### ● 利用多核处理器并列执行

NY532-1□□□ 及 NY512-1□□□ 中搭载有多核处理器，可并列进行任务的执行和标签数据链接服务或系统服务的执行。因此，大规模设备也可实现高速控制。

### 软件的特点

#### ● 整合顺序控制和运动控制

一台控制器同时实现顺序控制和运动控制。因此，可同时实现顺序控制和多轴同步控制。在同一控制周期中，执行顺序控制、运动控制和 I/O 刷新。

此外，控制周期与 EtherCAT 的过程数据通信周期一致。因此，可实现周期恒定、波动小、精度高的顺序控制及运动控制。

#### ● 支持多任务

将 I/O 刷新或用户程序执行等一系列处理分配到多个任务，并分别指定执行条件和执行顺序，再将它们组合起来，可与应用程序一起灵活地构建控制系统。

#### ● 符合国际标准规格 IEC 61131-3 的编程语言规格

配备符合 IEC 61131-3 的语言规格。部分经欧姆龙改良。备有 PLCopen<sup>®</sup> 标准的运动控制指令以及符合 IEC 规则的丰富的指令群（POU）。

#### ● 无需在意存储器映射，通过变量编程

与在电脑上使用高级语言一样，所有数据都通过变量访问。创建的变量可自动分配到控制器的存储器上，用于无需在意。

#### ● 丰富的安全功能

备有丰富的安全功能，可通过操作权限的设定、ID 来限制用户程序的执行。

#### ● 控制器整体的监视功能

控制器可对包括安装的 EtherCAT 从站在内的控制器整体现象（事件）进行监视。

异常时的处理方法通过 Sysmac Studio 或 NS/NA 系列触摸屏引导。发生的现象（事件）将保存为记录。

#### ● 自动化软件 Sysmac Studio

Sysmac Studio 是一个综合开发环境，能用一个软件覆盖控制器、外围设备或 EtherCAT 上的设备。对不同机器，可提供统一的操作性。从设计到调试、模拟、启动、开始使用后的变更等，可对所有工序提供支援。

#### ● 丰富的模拟功能

备有丰富的模拟功能，包括虚拟控制器的执行功能、调试功能等。

## 1-1-2 系统构成概要

NY 系列的系统构成如下所示。

### ● 基本构成

NY 系列的基本构成为 EtherCAT 网络配置和支持软件。

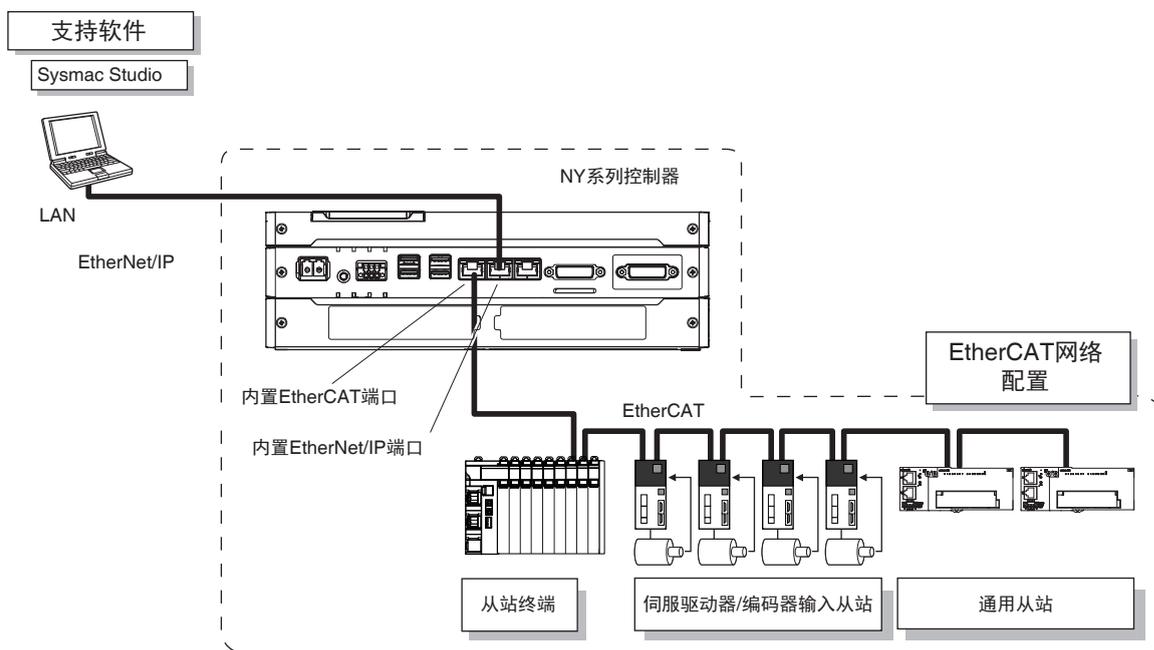
#### ● EtherCAT 网络配置

使用内置 EtherCAT 端口，可连接 EtherCAT 从站终端、数字 I/O、模拟 I/O 等通用从站以及伺服 / 编码器输入从站。使用这样的构成，可实现周期恒定、波动小、精度高的顺序控制及运动控制。

#### ● 支持软件

将 Ethernet 电缆连接到内置 EtherNet/IP 端口上。

支持软件的详细连接构成请参考 □ 「10-2 Sysmac Studio 的连接」 (P.10-4)。

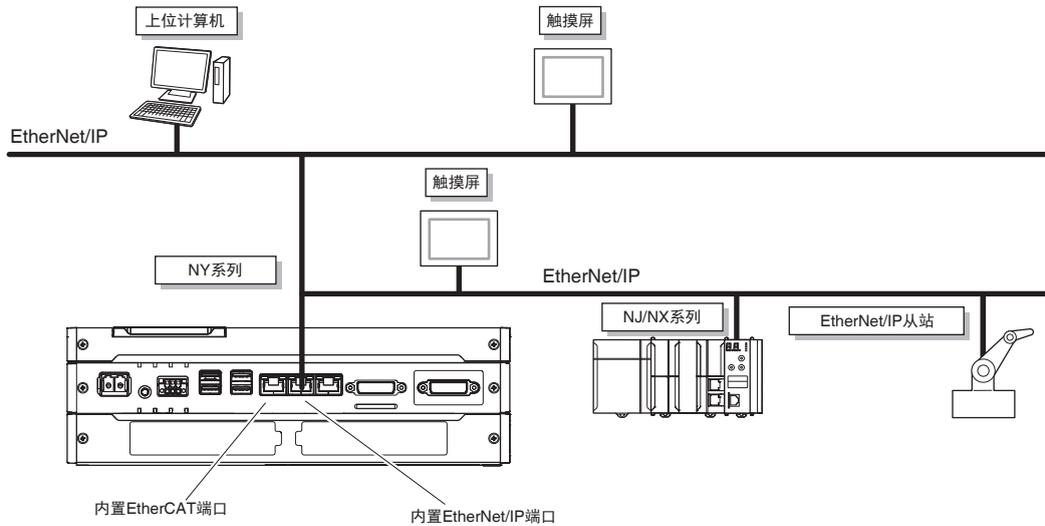


### 参考

可在通信耦合器单元上直接连接 Sysmac Studio，进行从站终端的设定。详情请参考 □ 《NX 系列 EtherCAT 耦合器单元 用户手册 (SBCD-CN5-361)》。

## ● 网络配置

- 在内置 EtherNet/IP 端口上连接上位计算机、触摸屏、NJ/NX/NY 系列。



网络配置的详情请参考□「第 10 章 通信设定」(P.10-1)。

## ● 支持软件

进行 NY 系列的控制器功能设定 / 监视 / 调试的编程设备如下所示。

- Sysmac Studio**  
NY 系列中，基本使用 Sysmac Studio 作为控制器配置和设定、程序创建、调试、模拟的支持软件。
- 其他支持软件**  
除了 Sysmac Studio 以外，软件包 Sysmac Studio 标准版中还包含以下软件。

构成软件	用途
<b>Sysmac Studio</b>	包括顺序控制及运动控制在内，使用除以下外的所有功能时使用。
<b>Network Configurator</b>	通过 EtherNet/IP 端口或单元进行标签数据链接时使用。 <sup>*1</sup>
<b>CX-Designer</b>	创建 NS 系列触摸屏画面时使用。

\*1. CS/CJ 系列 PLC 为始发设备时，在 Network Configurator 上设定。

关于 NY 系列的控制器功能以外的编程设备，请参考□《NY 系列 工业平板电脑 / 工业 BOX PC 用户手册 安装篇 (SBCA-CN5-441)》。

## 1-2 规格概要

下面介绍 NY 系列控制器功能的主要性能规格。一般规格、性能规格、功能规格请参考□□「A-1 规格」(P.A-3)。

项目				NY5□2-15□□	NY5□2-14□□	NY5□2-13□□
编程	程序容量 <sup>*1</sup>	大小		40MB		
		数量	POU 定义数	3,000		
	POU 实例数		24,000			
	变量容量	有保存属性	大小	4MB		
			变量数量	40,000		
		无保存属性	大小	64MB		
变量数量			180,000			
数据类型	数据类型的数量		4,000			
运动控制	控制轴数	最大控制轴数 <sup>*2</sup>		64 轴	32 轴	16 轴
		最大使用实轴数 <sup>*3</sup>		64 轴	32 轴	16 轴
		最大单轴控制数		64 轴	32 轴	16 轴
		直线插补控制最大数		每 1 轴组 4 轴		
		圆弧插补控制轴数		每 1 轴组 2 轴		
	轴组最大数		32 组			
	运动控制周期		与 EtherCAT 通信的过程数据通信周期相同			
	凸轮	凸轮数据点数	每 1 凸轮表的最大点数	65,535 点		
			所有凸轮表的最大点数	1,048,560 点		
		凸轮表最大表数		640 表		
内置 EtherNet/IP 端口	端口数量		1			
	物理层		10BASE-T/100BASE-TX/1000BASE-T			
	帧长度		最大 1,514 字节			
	介质存取方式		CSMA/CD			
	调制方式		基带			
	拓扑结构		星形			
	传送速度		1Gb/s (1000BASE-T)			
	传送介质		双绞线电缆 (带屏蔽: STP) : 类别 5、5e 以上			
	传输距离 (集线器和节点之间的距离) 最大值		100m			
	串联连接最大数量		使用交换式集线器时无限制			
	CIP 服务: 标签数据链接 (周期链接通信)	最大连接数		128		
		Packet 间隔 <sup>*4</sup>		各连接可设定 1 ~ 1000ms (以 1ms 为单位)		
		单元允许通信带宽		20,000 pps <sup>*5</sup> (含 Heartbeat)		
		最大标签集数		128		
		标签种类		网络变量		
		1 连接 (=1 标签集) 的标签数		8 (标签集中包含控制器状态时为 7)		
		标签最大数量		256		
		每 1 节点的最大链接数据容量 (所有标签的总大小)		184,832 字节		
		每 1 连接的最大数据容量		1,444 字节		
		最大可登录的标签集数		128 (1 连接 =1 标签集)		
1 标签集的最大容量		1,444 字节 (标签集中包含控制器状态时使用 2 字节)				
多点传送 Packet 过滤功能 <sup>*6</sup>		支持				

# 1 NY 系列控制器的概要

项目		NY5□2-15□□	NY5□2-14□□	NY5□2-13□□
内置 EtherCAT 端口	通信标准	IEC 61158 Type12		
	EtherCAT 主站规格	符合 Class B (符合 Feature Pack Motion Control)		
	物理层	100BASE-TX		
	调制方式	基带		
	传送速度	100Mb/s (100BASE-TX)		
	Duplex 模式	Auto		
	拓扑结构	线、菊花链、分支布线		
	传送介质	类别 5 以上 双绞线电缆 (推荐铝带编织双重隔离屏蔽电缆、直线型)		
	节点间距离最大值	100m		
	最大从站数	128		
单元构成	可安装的单元数	系统整体的 NX 单元最大数		
		4,096 (EtherCAT 从站终端上)		

- \*1. 执行对象、变量表 (变量名等) 的容量。
- \*2. 所有轴种类的合计。
- \*3. 轴种类为 [伺服轴] 或 [编码器轴], 且轴使用设定为 [使用轴] 的轴合计。
- \*4. 与节点数无关, 以设定周期更新线路中的数据。
- \*5. pps 代表 Packet Per Second, 表示 1 秒内可处理的接收发送 Packet 数。
- \*6. EtherNet/IP 端口安装了 IGMP 客户端, 使用了支持 IGMP Snooping 的交换式集线器, 所以会进行不必要的多点传送数据包的过滤。

## 1-3 产品型号

通过 NY 系列控制器的型号，可识别控制轴数和硬件构成。

**NY 532 - 1500 - 112213C22**  
 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17

编号	项目	选项
1	系列名称	NY : NY 系列工业电脑平台
2	控制器规格	5 : 用于最多 64 轴的大规模且高速高精度控制
3	机型	1 : 工业 BOX PC 3 : 工业平板电脑
4	追加编号	2 : 最新版
5	功能模块	1 : 标准
6	运动控制轴数	3 : 16 轴 4 : 32 轴 5 : 64 轴
7	追加功能软件模块	0 : ---
8	保留	0 : ---
9	扩展插槽	1 : PCIe 插槽 x1
10	框架类型	1 : 铝框架、黑色、透射型电容式触摸型 X : 无触摸屏 (工业 BOX PC)
11	触摸屏尺寸	1 : 12.1 英寸 2 : 15.4 英寸 X : 无触摸屏 (工业 BOX PC)
12	OS	1 : Windows Embedded Standard 7 32bit 2 : Windows Embedded Standard 7 64bit
13	处理器	1 : Intel® Core™ i7-4700EQ Processor
14	主存储器	3 : 8GB Non-ECC
15	存储器	8 : 32GB SSD SLC 9 : 64GB SSD SLC C : 320GB HDD K : 128GB SSD iMLC
16	接口选项	1 : RS-232C 2 : DVI-D
17	标志	0 : OMRON 2 : 自定义 X : 无触摸屏 (工业 BOX PC)



### 参考

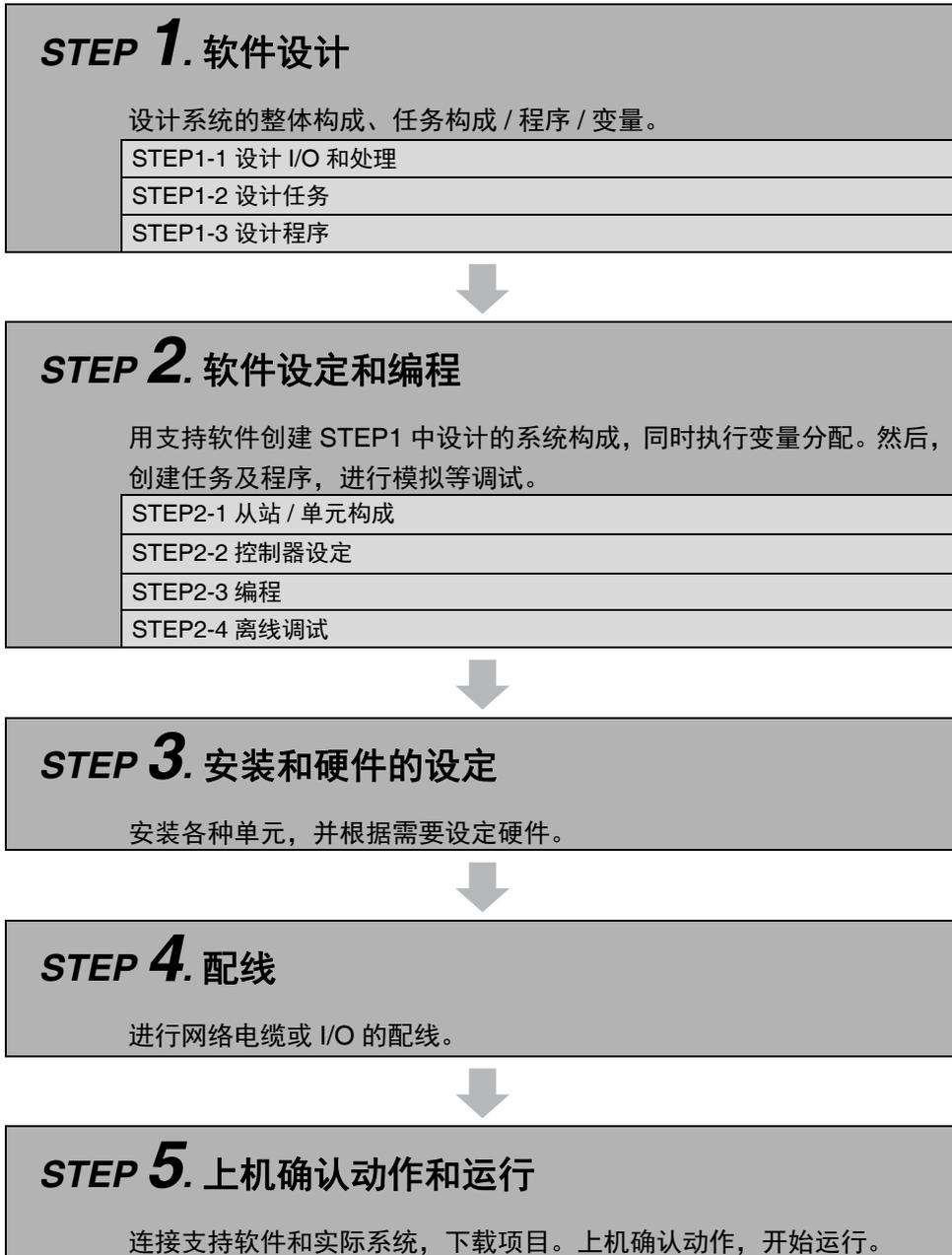
本手册主要介绍控制器功能，因此可能会省略对控制器功能没有影响的 9 号之后的型号。

## 1-4 NY 系列的整体使用步骤

下面介绍 NY 系列控制器功能整体步骤的概要和详细信息。

### 1-4-1 步骤概要

NY 系列的控制器功能按以下流程使用。



## 1-4-2 步骤详情

## STEP1 . 软件设计

步骤	内容	参考
STEP1-1 设计 I/O 和处理	<ul style="list-style-type: none"> <li>外部 I/O 和单元的构成</li> <li>外部设备和刷新周期</li> <li>程序的处理内容</li> </ul>	<input type="checkbox"/> 《NY 系列 工业平板电脑 用户手册 硬件篇 (SBCA-CN5-435)》 <input type="checkbox"/> 《NY 系列 工业 BOX PC 用户手册 硬件篇 (SBCA-CN5-434)》



STEP1-2 设计任务	<ul style="list-style-type: none"> <li>任务构成</li> <li>任务 — 程序的关系</li> <li>各任务的周期</li> <li>从站 / 单元的刷新周期</li> <li>任务间变量的排他方法</li> </ul>	<input type="checkbox"/> 「4-2-3 任务设置」(P.4-7)
-----------------	--	--



STEP1-3 设计程序		
设计 POU (处理的分割单位)	<ul style="list-style-type: none"> <li>程序</li> <li>函数 / 功能块</li> <li>选择各算法的语言</li> </ul>	<input type="checkbox"/> 「第 6 章 编程」(P.6-1)
设计变量	<ul style="list-style-type: none"> <li>区分 POU 共同使用的变量和各 POU 单独使用的变量</li> <li>设计用于访问从站 / 单元的设备变量之变量名</li> <li>设计变量的名称、保持等属性</li> <li>设计变量的数据类型</li> </ul>	<input type="checkbox"/> 「6-3 变量」(P.6-27)



## STEP2 . 软件设定和编程

步骤	内容	Sysmac Studio 上的操作	参考
新建项目	<ol style="list-style-type: none"> <li>用 Sysmac Studio 新建项目。</li> <li>插入控制器。</li> </ol>	[ 新建项目 ] 按钮 [ 插入 ]   [ 控制器 ]	<input type="checkbox"/> 《Sysmac Studio Version 1 操作手册 (SBCA-CN5-362)》



以下“控制器配置和设定”及“编程 / 任务设置”的顺序中，可任选。

STEP2-1 从站 / 单元构成			
1) 从站 / 单元构成的创建	<ol style="list-style-type: none"> <li>离线或在线创建从站 / 单元构成。(在线时, 先执行 STEP 5 的在线连接)</li> <li>使用从站终端时, 应设定从站终端。</li> </ol>	EtherCAT 从站设定编辑器 单元编辑器	<input type="checkbox"/> 「3-2 EtherCAT 从站构成的创建」(P.3-4) <input type="checkbox"/> 《NX 系列 EtherCAT 耦合器单元 用户手册 (SBCD-CN5-361)》



2) 向 I/O 端口分配设备变量	将设备变量登录到变量表中 (变量名由用户定义或自动生成)	I/O 映射	<input type="checkbox"/> 「3-3 I/O 端口和设备变量」(P.3-5)
-------------------	------------------------------	--------	---

# 1 NY 系列控制器的概要

(进行运动控制时)

3) “轴”的创建和伺服 / 编码器输入从站的分配	创建“轴”，分配到实轴或虚拟轴。 插补轴控制时，创建“轴组”。	配置和设定   在运动控制设定中设定	☐「3-5 “轴”的创建和伺服驱动器 / 编码器输入从站的分配」(P.3-11)
---------------------------	------------------------------------	--------------------	--

STEP2-2 控制器设定	在 Sysmac Studio 中设定以下参数。		☐「第 4 章 控制器设定」(P.4-1)
	PLC 功能模块相关的初始设定	配置和设定   在动作设定中设定	☐「4-2 PLC 功能模块的初始设定」(P.4-4)
	(进行运动控制时) 运动控制功能模块相关的初始设定	配置和设定   在运动控制设定中设定	☐「4-3 运动控制功能模块的初始设定」(P.4-12)
	EtherCAT 主机功能模块相关的初始设定	配置和设定   在 EtherCAT 中设定	☐「4-4 EtherCAT 主机功能模块的初始设定」(P.4-14)
	EtherNet/IP 功能模块相关的初始设定	配置和设定   在内置 EtherNet/IP 端口设定中设定	☐「4-5 EtherNet/IP 功能模块的初始设定」(P.4-15)

STEP2-3 编程			
1) 变量的登录	<ul style="list-style-type: none"> <li>在 Sysmac Studio 中，向全局变量表登录 POU 共通的变量</li> <li>登录各程序的本地变量表</li> <li>登录各功能块、函数的本地变量表</li> </ul>	全局变量表编辑器 本地变量表编辑器	☐《Sysmac Studio Version 1 操作手册 (SBCA-CN5-362)》☐「6-3 变量」(P.6-27)
2) 创建 POU 的算法	用各语言创建 POU (程序 / 功能块 / 函数) 的算法	程序编辑器	☐「第 6 章 编程」(P.6-1) ☐《NY 系列 指令基准手册 基本篇 (SBCA-CN5-437)》、 ☐《NY 系列 指令基准手册 运动篇 (SBCE-CN5-380)》
3) 任务设置	进行任务相关的设置。	在任务设置中设置	☐「4-2-3 任务设置」(P.4-7)

STEP2-4 离线调试	在模拟器 (虚拟控制器) 上确认算法		☐「第 7 章 动作确认和运行」(P.7-1)
-----------------	--------------------	--	-------------------------

**STEP3. 安装和硬件的设定**

步骤	内容	参考
1) 安装	<ul style="list-style-type: none"> <li>• 各部位的安装</li> <li>• 安装到控制柜上</li> </ul>	<input type="checkbox"/> 《NY 系列 工业平板电脑 用户手册 硬件篇 (SBCA-CN5-435)》 <input type="checkbox"/> 《NY 系列 工业 BOX PC 用户手册 硬件篇 (SBCA-CN5-434)》
2) 共享文件夹的设定	<ul style="list-style-type: none"> <li>• 作为虚拟SD存储卡使用的共享文件夹的Windows端设定</li> </ul>	<input type="checkbox"/> 《NY 系列 工业平板电脑 / 工业 BOX PC 用户手册 安装篇 (SBCA-CN5-441)》
3) 硬件的设定	<ul style="list-style-type: none"> <li>• EtherCAT 从站的节点地址设定</li> </ul>	<input type="checkbox"/> 各 EtherCAT 从站的用户手册、各高性能单元的用户手册

**STEP4. 配线**

步骤	内容	参考
1) Ethernet 电缆的配线	<ul style="list-style-type: none"> <li>• 内置 EtherCAT 端口的配线</li> <li>• 内置 EtherNet/IP 端口的配线</li> </ul>	<input type="checkbox"/> 《NY 系列 工业平板电脑 用户手册 硬件篇 (SBCA-CN5-435)》 <input type="checkbox"/> 《NY 系列 工业 BOX PC 用户手册 硬件篇 (SBCA-CN5-434)》
2) I/O 配线	<ul style="list-style-type: none"> <li>• EtherCAT 从站的 I/O 配线</li> </ul>	<input type="checkbox"/> 各 EtherCAT 从站的用户手册 <input type="checkbox"/> 《NY 系列 工业平板电脑 用户手册 硬件篇 (SBCA-CN5-435)》 <input type="checkbox"/> 《NY 系列 工业 BOX PC 用户手册 硬件篇 (SBCA-CN5-434)》
	<ul style="list-style-type: none"> <li>• 配线的检查</li> </ul>	<input type="checkbox"/> 《Sysmac Studio Version 1 操作手册 (SBCA-CN5-362)》
3) 电脑 (Sysmac Studio) 的配线	<ul style="list-style-type: none"> <li>• 内置 EtherNet/IP 端口的配线</li> </ul>	<input type="checkbox"/> 《Sysmac Studio Version 1 操作手册 (SBCA-CN5-362)》

**STEP5. 上机确认动作和运行**

步骤	内容	Sysmac Studio 上的操作	参考

# 1 NY 系列控制器的概要

<b>1) 与 Sysmac Studio 的在线连接、项目的下载</b>	打开控制器的电源，与 Sysmac Studio 在线连接。然后，下载项目。 <sup>*1</sup> 在安装单元上进行 STEP2-1 的“从站/单元构成的创建”时，请在 STEP2-1 之前执行。	[ 控制器 ]- [ 通信设定 ] [ 控制器 ]- [ 同步 ]	□□「第 7 章 动作确认和运行」 (P.7-1)
			
<b>2) 上机确认动作</b>	1. 通过 I/O 映射或监视窗口对实际 I/O 进行强制值刷新，确认配线。 2. 进行运动控制时，在程序模式下通过 MC 调试画面确认配线。然后，通过点动进给确认电机的方向，通过相对定位确认移动量（电子齿轮的设定），确认原点复位动作。 3. 设为运行模式，确认用户程序的动作。	—	□□「第 7 章 动作确认和运行」 (P.7-1)
			
<b>3) 控制器的正式运行</b>	开始正式运行。	—	—

\*1. 下载时使用 Sysmac Studio 的同步功能。

# 2

## NY 系列控制器的动作

下面介绍使用 NY 系列控制器需要的知识，包括控制器的原理及各种状态下的动作。

<b>2-1 NY 系列控制器的动作概要</b> .....	<b>2-2</b>
2-1-1 NY 系列控制器的原理概要 .....	2-2
2-1-2 NY 系列控制器在各状态下的动作概要 .....	2-3
<b>2-2 软件的原理</b> .....	<b>2-4</b>
2-2-1 软件构成 .....	2-4
2-2-2 软件的动作 .....	2-5
<b>2-3 与 I/O 的访问原理</b> .....	<b>2-10</b>
2-3-1 变量的种类 .....	2-10
2-3-2 变量和 I/O 的访问 .....	2-13
<b>2-4 顺序控制和运动控制的原理</b> .....	<b>2-16</b>
2-4-1 控制的概要 .....	2-16
2-4-2 顺序控制的原理 .....	2-18
2-4-3 运动控制的原理 .....	2-19
2-4-4 顺序控制和运动控制的同步 .....	2-20
<b>2-5 控制器内的数据概要</b> .....	<b>2-21</b>
<b>2-6 控制器在各种状态下的动作</b> .....	<b>2-22</b>
2-6-1 控制器的状态 .....	2-22
2-6-2 各种状态下的动作 .....	2-24
2-6-3 动作模式 .....	2-25
<b>2-7 Windows 状态的监视及变更</b> .....	<b>2-27</b>
2-7-1 Windows 的状态 .....	2-27
2-7-2 Windows 状态的监视 .....	2-27
2-7-3 Windows 状态的变更 .....	2-28
2-7-4 Windows 状态的变化 .....	2-28
<b>2-8 关机处理</b> .....	<b>2-29</b>
2-8-1 断电及按下电源按钮时的处理 .....	2-30
2-8-2 关机处理的设定 .....	2-31
2-8-3 关机处理的内容 .....	2-32
<b>2-9 控制器重置</b> .....	<b>2-35</b>

## 2-1 NY 系列控制器的动作概要

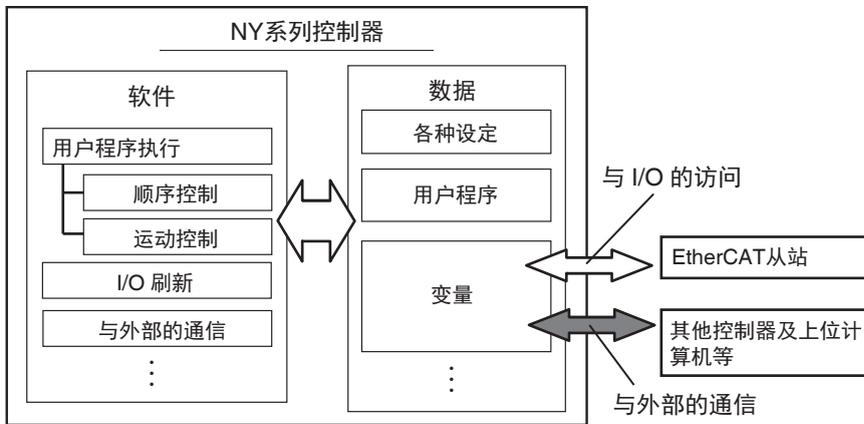
下面介绍 NY 系列控制器的动作概要，包括控制器的原理及控制器在各种状态下的动作。

### 2-1-1 NY 系列控制器的原理概要

为进行顺序控制和运动控制，NY 系列控制器需要执行用户程序、进行 I/O 刷新、与外部进行通信等。这些处理由控制器的软件执行。

此外，控制器中还包含各种设定、用户程序、变量等数据。控制器使用这些数据进行处理。控制器和 I/O 的访问以及与外部的通信使用这些数据中的变量。

控制器通过访问软件及使用变量的 I/O，可以通知执行顺序控制和运动控制。



本章介绍将控制器用于顺序控制、运动控制或两种控制时所需的基本知识，内容如下。

内容	参照页
软件的原理	P.2-4
与 I/O 的访问原理	P.2-10
顺序控制和运动控制的原理	P.2-16
控制器的数据概要	P.2-21
控制器在各种状态下的动作	P.2-22



#### 参考

- 关于使用变量与外部进行通信的原理，请参考 □《NY 系列 工业平板电脑 / 工业 BOX PC 用户手册 内置 EtherNet/IP 端口篇 用户手册 (SBCD-CN5-369)》。
- 可从 Windows 端访问 NY 系列控制器上的变量。关于访问方法，请参考 □《NY 系列 工业平板电脑 / 工业 BOX PC 用户手册 安装篇 (SBCA-CN5-441)》。

## 2-1-2 NY 系列控制器在各状态下的动作概要

NY 系列控制器会因发生异常、动作模式的切换而变为各种状态。因 NY 系列控制器的状态变化，用户程序的执行、I/O 刷新、与外部的通信处理动作也会变化。

□「2-6 控制器在各种状态下的动作」(P.2-22)中，介绍控制器在各种状态下的处理动作。

## 2-2 软件的原理

下面介绍 NY 系列控制器功能的软件构成及软件动作。

### 2-2-1 软件构成

控制器功能的软件由 4 种功能构成。构成功能的单位为“功能模块”。

各功能模块及其处理如下所示。

功能模块名称	处理
PLC 功能模块	进行任务的日程安排、向各功能模块的指示、事件日志、用户程序的执行、虚拟 SD 存储卡服务、数据追踪处理等。
运动控制功能模块	进行运动控制处理。 <sup>*1</sup>
EtherCAT 主机功能模块	EtherCAT 从站的 I/O 刷新 <sup>*2</sup> 、EtherCAT 信息通信 <sup>*3</sup> 等，作为 EtherCAT 主机与 EtherCAT 从站进行通信。
EtherNet/IP 功能模块	标签数据链接处理、内置 EtherNet/IP 端口服务等，进行 EtherNet/IP 通信的处理。

\*1. 根据运动控制指令发出的目标值（位置、速度）等指示，进行运动计算，通过 EtherCAT 主机执行指令值输出、状态管理及信息获取。

\*2. 通过过程数据通信（以下简称 PDO 通信）进行。PDO 通信是指在主机和从站之间以恒定的周期进行周期性数据交换的通信方式。

\*3. 作为 EtherCAT 主机与 EtherCAT 从站进行通信。

## 2-2-2 软件的动作

软件执行的控制器处理，因执行处理的控制器的状态、处理的执行条件不同，分为以下 4 种。

控制器的处理种类	执行处理的控制器状态	执行条件	处理示例
启动处理	启动中	仅接通电源时执行。	电源接通时的自诊断
任务中执行的处理	正常运行中及异常状态	在分配的任务中执行。任务分为按周期反复执行的任务和在指定的条件成立时仅执行 1 次的任务。	用户程序执行、I/O 刷新
标签数据链接服务 系统服务		仅在硬件或外部发出请求时执行。	标签数据链接 虚拟 SD 存储卡服务

控制器的状态请参考 □ 「2-6-1 控制器的状态」(P.2-22)。

下面介绍各处理的动作。

### 启动处理

启动处理仅接通电源时执行。

在启动处理中，执行以下处理。

处理	内容
电源接通时的自诊断	检测“电源部异常”、“CPU 复位”、“CPU 异常 (WDT 异常)”。 <sup>*1</sup>
数据检查	若带电源断开保持属性的变量值在电源断开时无法保持，则系统定义变量“_RetainFail” (电源断开保持失败标志) 为 TRUE。
电源接通及电源断开的事件记录	记录电源接通及电源断开的事件。

\*1. 关于“电源部异常”、“CPU 复位”、“CPU 异常 (WDT 异常)”，请参考 □ 《NY 系列 故障排除手册 (SBCA-CN5-438)》。

## 任务中执行的处理

### ● 任务中执行的处理种类

任务中，将执行以下处理

处理	内容
I/O 刷新	与 EtherCAT 从站进行数据输入输出。
用户程序执行	执行顺序控制的用户程序。或向运动控制发行指令。
运动控制	收到来自用户程序的指令，执行运动控制。
系统共通处理	执行数据追踪处理、标签数据链接处理等。

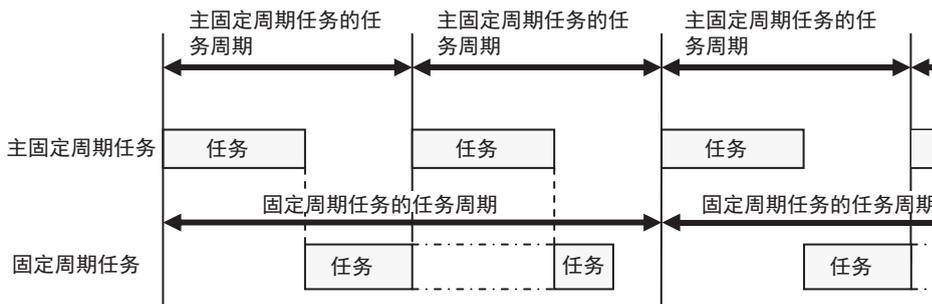
任务中执行的处理详情请参考及 □□「5-3-3 NY 系列的任務基本動作」(P.5-9)。

### ● 任务的动作

各处理将分配到任务中。因执行优先度和执行条件不同，任务分为以下 3 种。

固定周期任务的种类	执行优先度 (值越小, 优先度越高)	执行条件	主要处理内容
主固定周期任务	4 执行优先度最高的任务。	以一定周期反复执行。	I/O 刷新、用户程序执行、运动控制
固定周期任务	16,17,18	以一定周期反复执行。 任务周期为主固定周期任务的任务周期整数倍。	处理内容因执行优先度不同而异。 执行优先度 16、17、18: 用户程序执行
事件任务	8,48	在事先指定的条件成立时, 仅执行 1 次。	用户程序执行

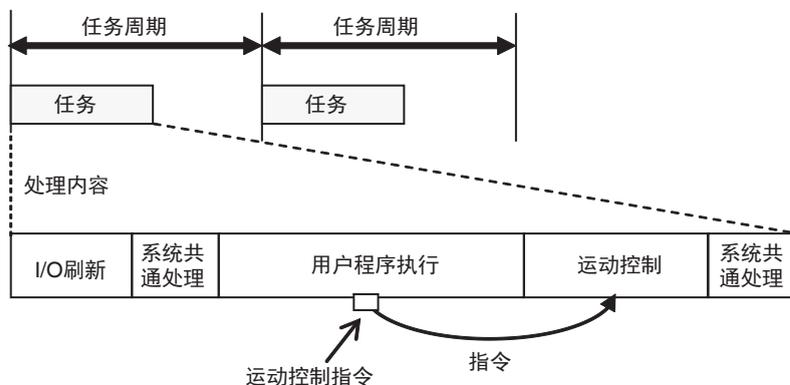
控制器将优先执行优先度较高的任务。下图表示主固定周期任务和固定周期任务的动作示例。正在执行固定周期任务时，若执行了主固定周期任务，将优先执行主固定周期任务。



任务动作的详情请参考 □□「5-2 任务的概要」(P.5-4)。

### ● 任务中执行的处理动作

任务中分配的处理将在任务中按下图顺序执行。程序中记述了运动控制指令时，通过任务中的“用户程序执行”向“运动控制”发行指令。“运动控制”将根据指令进行运动控制处理。



(注) “运动控制”处理在主固定周期任务中执行。记述了运动控制指令的程序分配到固定周期任务时，通过固定周期任务中的“用户程序执行”向主固定周期任务的“运动控制”发行指令。

详情请参考 □「5-9-3 系统输入输出响应时间」(P.5-58)。

## 标签数据链接服务

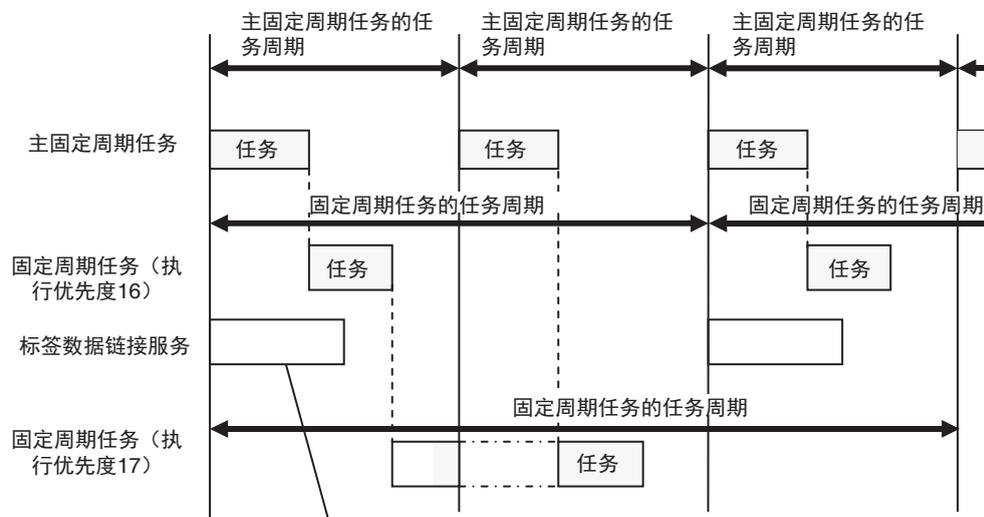
### ● 标签数据链接服务的处理

标签数据链接服务是指在 EtherNet/IP 网络上，使用其他控制器、设备和标签进行通信处理。使用控制器的内置 EtherNet/IP 端口。

### ● 标签数据链接服务的动作

标签数据链接服务以恒定周期执行。周期或 1 次处理的执行时间因控制器的型号或标签数据链接设定不同而异。

此外，标签数据链接服务与任务的执行并列执行。



与任务的执行并列执行。

标签数据链接服务的详情请参考 □「5-4 标签数据链接服务和系统服务」(P.5-25)。

## 系统服务

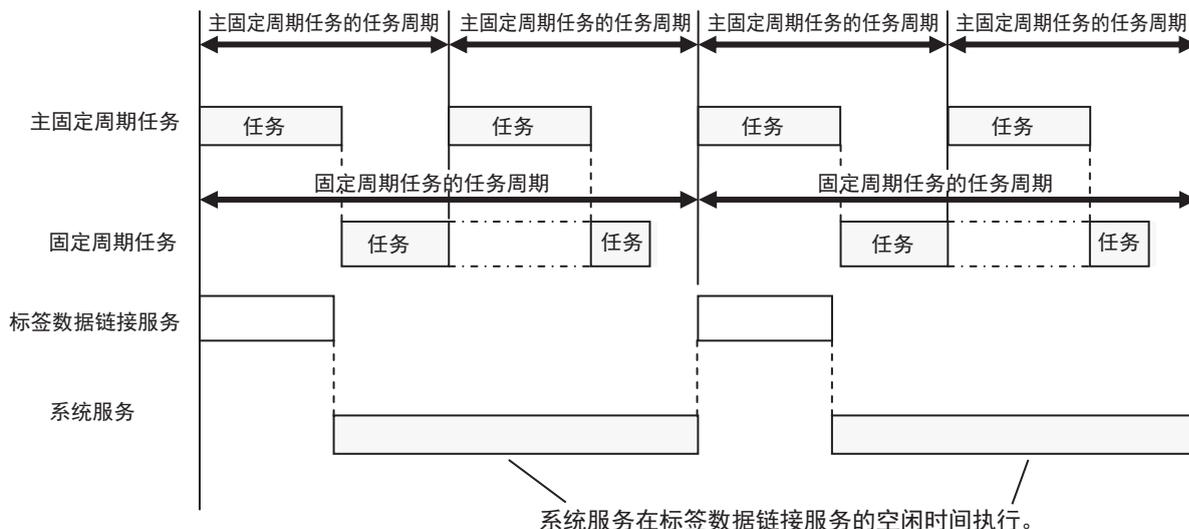
### ● 系统服务的种类

系统服务中包括以下处理。

处理	内容
内置 EtherNet/IP 端口服务	<ul style="list-style-type: none"> <li>对来自编程设备、触摸屏、上位计算机或其他控制器的服务请求（CIP 等）进行处理</li> <li>通信指令（CIP、套接字）的执行</li> </ul>
内置 EtherCAT 端口服务	<ul style="list-style-type: none"> <li>EtherCAT 信息通信的执行</li> </ul>
虚拟 SD 存储卡服务	<ul style="list-style-type: none"> <li>从 FTP 客户端访问</li> <li>通过 Sysmac Studio 对 SD 存储卡进行操作</li> <li>SD 存储卡指令的执行</li> </ul>
自诊断处理	<ul style="list-style-type: none"> <li>硬件的异常检测</li> <li>Windows 的状态监视</li> </ul>

### ● 系统服务的动作

NY 系列控制器中，有来自硬件或控制器以外的请求时，系统服务将与其他处理并列执行。不受任务执行的影响，始终执行。但是，正在执行标签数据链接服务时，不会执行系统服务。



系统服务的详情请参考 □□「5-4 标签数据链接服务和系统服务」(P.5-25)。

## 2-3 与 I/O 的访问原理

NY 系列控制器利用变量来访问 I/O。下面介绍变量与 I/O 的访问原理。

本手册中，将 EtherCAT 从站作为 I/O 处理。关于使用标签数据链接与其他控制器进行的访问，请参考

□《NY 系列 工业平板电脑 / 工业 BOX PC 用户手册 内置 EtherNet/IP 端口篇 用户手册 (SBCD-CN5-369)》。

### 2-3-1 变量的种类

NY 系列控制器中，通过用户程序，介由变量来访问 I/O 或控制器内部。

因是否可由用户设定属性或访问对象不同，变量分为以下几种。

变量		可否设定属性	访问对象	
用户定义变量		可设定所有的属性	控制器内部	
准用户定义变量	设备变量	可设定部分属性*1	EtherCAT 从站*2	
	凸轮数据变量		伺服驱动器、编码器输入从站、 控制器内部	
系统定义变量	PLC 系统变量	所有的属性都不可设定	控制器内部	
	运动控制系统变量		MC 共通变量	伺服驱动器、编码器输入从站、 控制器内部
			轴变量	
			轴组变量	
EtherNet/IP 系统变量	内置 EtherNet/IP 端口			
EtherCAT 主机系统变量	内置 EtherCAT 主机端口			

\*1. 关于可设定的属性，请参考 □「设备变量的属性」(P.3-7)。

\*2. EtherCAT 从站包括 EtherCAT 从站终端上的 NX 单元。

### 用户定义变量

所有属性由用户定义的变量。用户定义变量的详情请参考 □「6-3 变量」(P.6-27)。

## 准用户定义变量

可设定部分属性的变量。用于访问特定的对象数据。因访问对象不同，准用户定义变量分为设备变量和凸轮数据变量。

### ● 设备变量

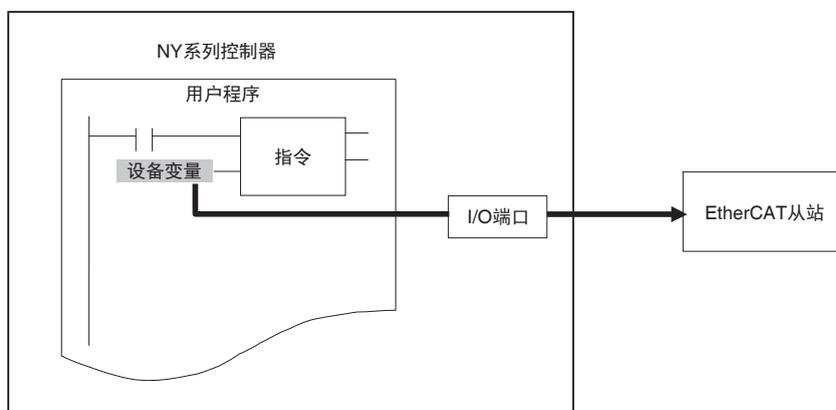
设备变量是指可访问设备数据的变量。设备是指通过控制器进行 I/O 刷新的对象的总称。表示 EtherCAT 从站。

因设备及设备访问对象的数据不同，分为以下几种。

设备变量的种类	设备	访问对象的数据
EtherCAT 从站设备变量	EtherCAT 从站	EtherCAT 上的从站的过程数据*1

\*1. 主机和从站之间以过程数据通信周期进行更换的 I/O 数据。

设备变量通过 I/O 端口访问 EtherCAT 从站。I/O 端口是指用于访问设备的逻辑端口。



I/O 端口及设备变量的详情请参考 □□「3-3-1 I/O 端口」(P.3-5)。

### ● 凸轮数据变量

在运动控制中，用于访问凸轮表数据的变量。

详情请参考 □□《NY 系列 工业平板电脑 / 工业 BOX PC 用户手册 运动控制篇 (SBCE-CN5-379)》。

## 系统定义变量

系统定义变量是指名称等所有属性由 NY 系统事先定义，且分配有特定功能的变量。用户无法变更名称等属性。

特定功能在各个功能模块中，因此各个功能模块中有系统定义变量。系统定义变量的种类如下所示。

功能模块	系统定义变量的种类
PLC 功能模块	PLC 系统变量
运动控制功能模块	运动控制系统变量
EtherNet/IP 功能模块	EtherNet/IP 系统变量
EtherCAT 主机功能模块	EtherCAT 主机系统变量

运动控制系统变量根据运动控制功能的内容分类如下。

运动控制系统变量	功能的内容
MC 共通变量	运动控制功能模块共通处理
轴变量	“轴”的控制
轴组变量	轴组 <sup>*1</sup> 的控制

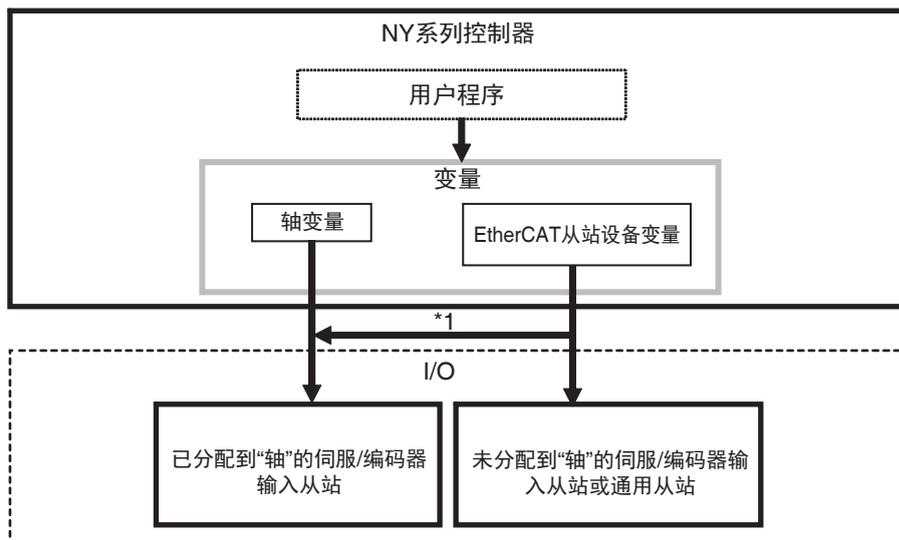
\*1. 轴组是指整合了多个“轴”的组。用于插补动作。

关于系统定义变量的详情，请参考 □「A-3 系统定义变量一览」(P.A-22)。

### 2-3-2 变量和 I/O 的访问

NY 系列控制器在用户程序中使用变量。变量可访问所分配 I/O 的数据。NY 系列控制器中，如下分配 I/O 和变量。

I/O		变量
EtherCAT 从站	EtherCAT 从站	EtherCAT 从站设备变量
	“轴”中分配的 EtherCAT 从站	轴变量或 EtherCAT 从站设备变量



### 与 EtherCAT 从站的访问

与 EtherCAT 从站的访问方法因 EtherCAT 从站的种类不同而异。

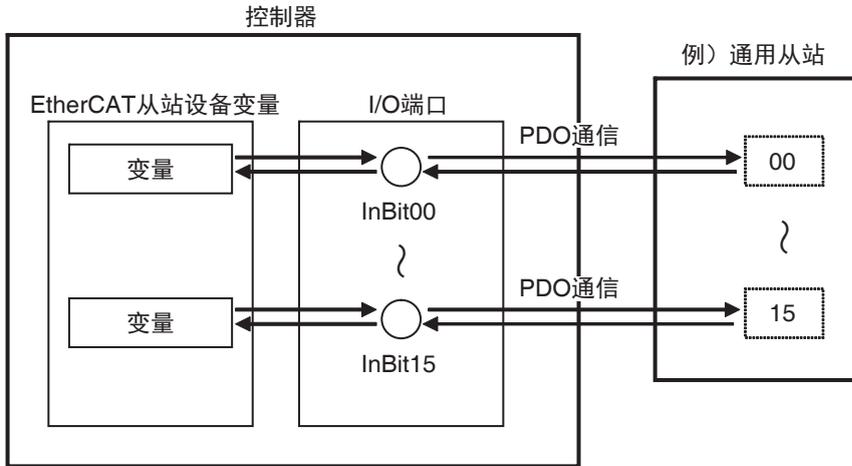
EtherCAT 从站的种类	访问方法
<ul style="list-style-type: none"> <li>• 伺服 / 编码器输入从站</li> <li>• 通用从站</li> </ul>	根据 EtherCAT 从站设备变量通过 I/O 端口访问
已分配到“轴”的伺服 / 编码器输入从站	通过轴变量直接访问 <sup>*1</sup>

\*1. 伺服驱动器时，将 1 台伺服电机作为 1 个“轴”分配到轴变量中使用；编码器输入从站时，则为 1 个计数器。

(注) 不可分配到“轴”的 EtherCAT 从站称为“通用从站”，可分配到“轴”的 EtherCAT 从站称为“伺服 / 编码器输入从站”。伺服驱动器 / 编码器输入从站请参考 □□《NY 系列 工业平板电脑 / 工业 BOX PC 用户手册 运动控制篇 (SBCE-CN5-379)》。

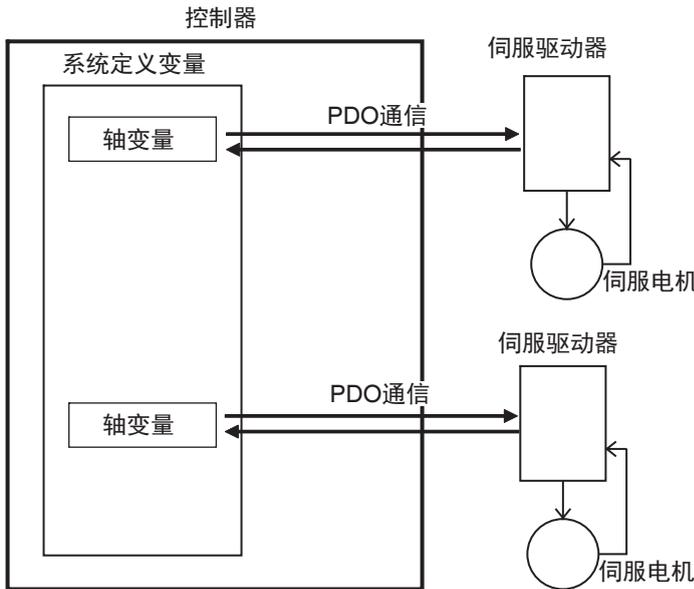
● 与未分配到“轴”的伺服 / 编码器输入从站或通用从站的访问

根据 EtherCAT 从站设备变量通过 I/O 端口访问。通过 I/O 端口访问时，利用 PDO 通信进行。



● 与已分配到“轴”的伺服 / 编码器输入从站的访问

轴变量可直接访问已分配到“轴”的伺服 / 编码器输入从站。通过轴变量访问时，利用 PDO 通信进行。例如，使用伺服驱动器控制伺服电机时，针对轴变量中分配的伺服电机发出的控制指示将发送到伺服驱动器中。来自伺服电机的反馈将通过轴变量从伺服驱动器发送到控制器中。



轴变量的详情请参考 □□「3-5-2 轴变量 / 轴组变量」(P.3-11)。



### 使用注意事项

---

可将设备变量分配到“轴”中分配的伺服 / 编码器输入从站的 I/O 端口上。

可分配设备变量的 I/O 端口应满足以下任一条件。

- R/W 属性的值为 R（读取专用）。
  - R/W 属性的值为 W（写入专用），Sysmac Studio 的[轴基本设定]画面的[详细设定]中，过程数据的设定为 < 不分配 >。
- 



### 参考

---

EtherCAT 通信分为 PDO 通信和 SDO 通信。对于伺服电机的位置控制等以固定控制周期更新输入输出数据的指令，使用 PDO 通信。对于参数传送等以指定时间读取或写入数据的指令，则使用 SDO 通信。详情请参考  《NY 系列 工业平板电脑 / 工业 BOX PC 用户手册 内置 EtherCAT 端口篇（SBCD-CN5-368）》。

---

## 2-4 顺序控制和运动控制的原理

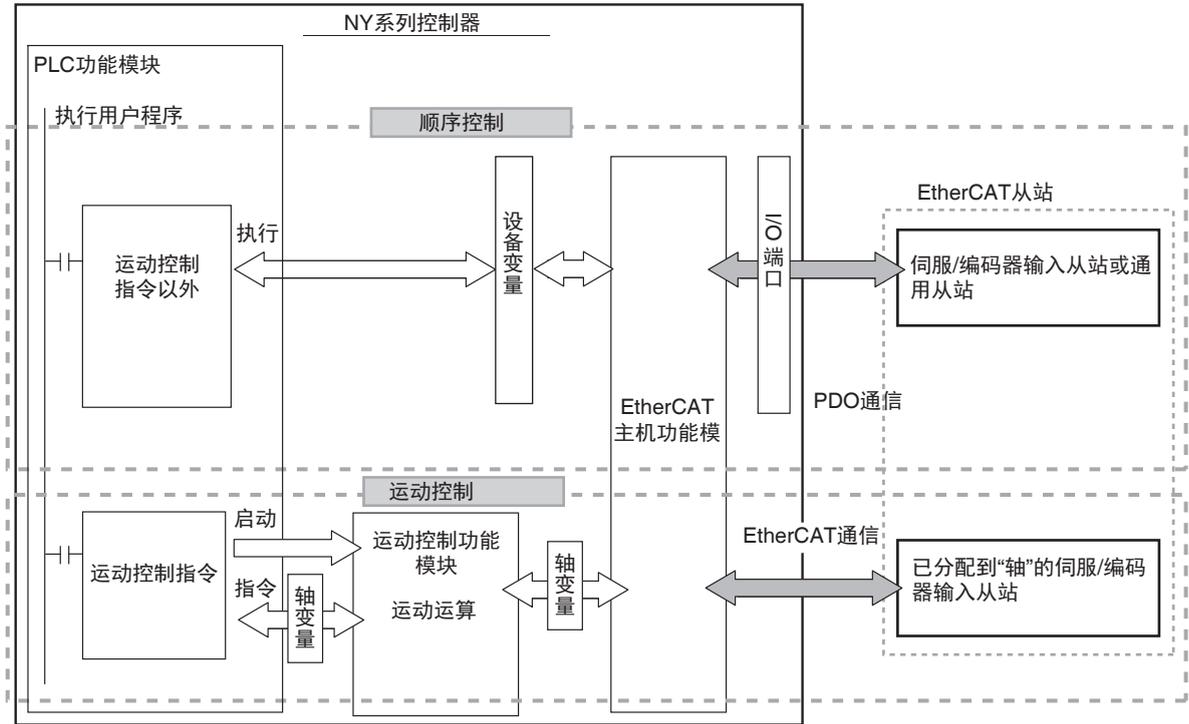
下面介绍 NY 系列控制器的顺序控制和运动控制的原理。

### 2-4-1 控制的概要

NY 系列控制器中可执行顺序控制和运动控制。

顺序控制是指执行用户程序中记述的运动控制指令以外的指令，对 EtherCAT 从站进行控制。控制时使用 PLC 功能模块、EtherCAT 主机功能模块。

运动控制是指执行用户程序中记述的运动控制指令，对“轴”中分配的 EtherCAT 从站上的伺服驱动器 / 编码器输入从站进行控制。控制时使用 PLC 功能模块、运动控制功能模块、EtherCAT 主机功能模块。



#### 参考

#### 控制原理对应的指令分类

根据控制原理，指令大致可分为以下两类。

指令的分类	定义
运动控制指令以外 (顺序控制)	执行 PLC 功能模块中的用户程序，结束处理的指令。
运动控制指令	通过执行 PLC 功能模块中的用户程序来启动，对运动控制功能模块发出指令。 例)MC_Home (原点复位)、MC_Move (定位)、MC_CamIn (凸轮动作开始) 等运动控制相关的各种操作

运动控制指令的详情请参考 □ 《NY 系列 指令基准手册 运动篇 (SBCE-CN5-380)》。运动控制指令以外的指令详情请参考 □ 《NY 系列 指令基准手册 基本篇 (SBCE-CN5-437)》。



### 使用注意事项

---

可将设备变量分配到“轴”中分配的伺服 / 编码器输入从站的 I/O 端口上。

可分配设备变量的 I/O 端口应满足以下任一条件。

- R/W 属性的值为 R（读取专用）。
- R/W 属性的值为 W（写入专用），Sysmac Studio 的[轴基本设定]画面的[详细设定]中，过程数据的设定为 < 不分配 >。

将设备变量分配到“轴”中分配的伺服 / 编码器输入从站的 I/O 端口时，若进行以下操作，将解除设备变量的分配。敬请注意。

- (1) Sysmac Studio Ver.1.09 以上中，将设备变量分配到“轴”中分配的伺服 / 编码器输入从站的 I/O 端口上。
  - (2) 保存项目数据。
  - (3) 在 Sysmac Studio Ver.1.08 或以下版本中打开该项目数据。
-

2-4-2 顺序控制的原理

因顺序控制的对象设备不同而异。下面以功能模块的动作、控制周期为例对顺序控制的原理进行说明。

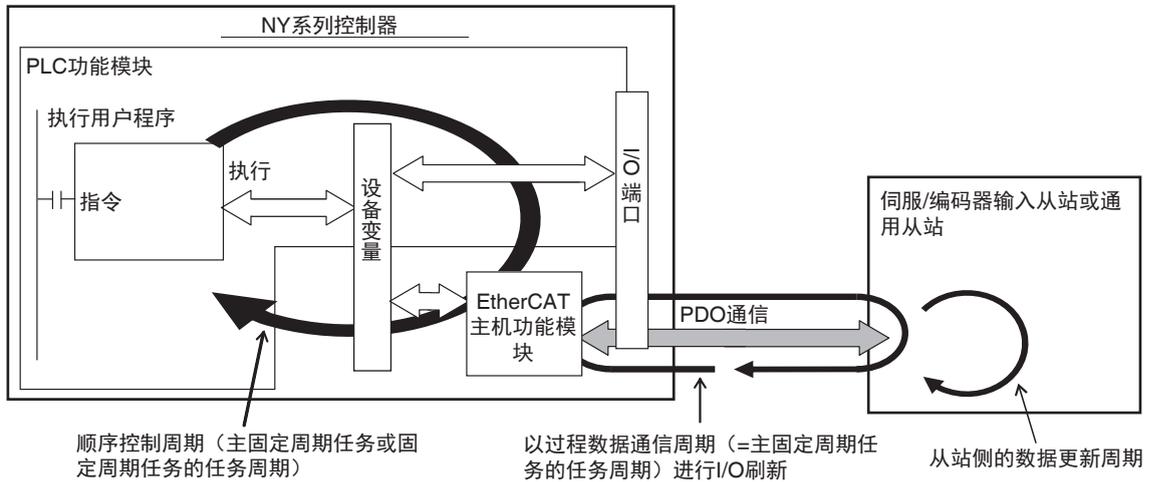
设备	顺序控制的原理	
	功能模块的动作	控制周期
伺服 / 编码器输入从站或通用从站	<ul style="list-style-type: none"> <li>• PLC 功能模块将执行用户程序，更新设备变量。</li> <li>• EtherCAT 主机功能模块使用设备变量，通过 I/O 端口与从站进行数据交换。</li> </ul>	分配了程序的任务之任务周期（主固定周期任务或固定周期任务的周期）*1

\*1. 从站侧的数据更新周期取决于从站的设定。

伺服 / 编码器输入从站或通用从站，以过程数据通信周期，即主固定周期任务的任务周期进行 I/O 刷新。但是，用户程序的执行及设备变量的更新按分配了程序的任务之任务周期执行。因此，将从站的值反映到设备变量中并进行控制的周期为分配了程序的任务之任务周期。

希望以过程数据通信周期控制从站时，需要将控制从站的程序分配到主固定周期任务中。

详情请参考 □「5-9-3 系统输入输出响应时间」(P.5-58)。



### 2-4-3 运动控制的原理

下面以功能模块的动作、控制周期为例对运动控制的原理进行说明。

#### ● 功能模块的动作

- PLC 功能模块通过启动用户程序中的运动控制指令，向运动控制功能模块发行运动控制指令。本指令中使用轴变量。
- 运动控制功能模块根据 PLC 功能模块的指令进行运动计算，然后按计算结果的指令值更新轴变量的值。
- EtherCAT 主机功能模块通过 EtherCAT 通信将轴变量的指令值发送到伺服驱动器等。

#### ● 控制周期

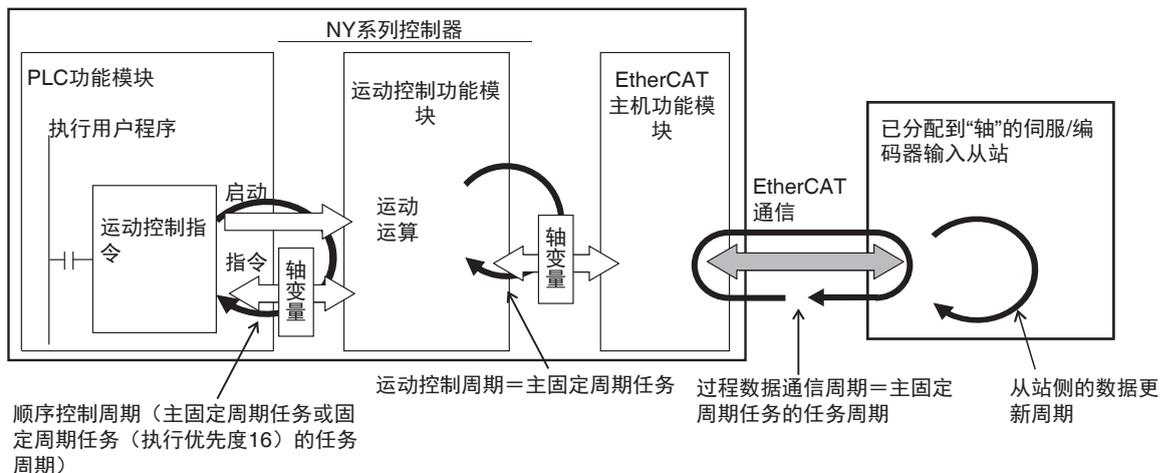
运动控制周期为主固定周期任务的任务周期。

运动控制功能模块的运动计算周期为主固定周期任务的任务周期。此外，与已分配到控制对象“轴”的伺服 / 编码器输入从站之间，以过程数据通信周期进行数据交换。过程数据通信周期与主固定周期任务同步。

因此，运动控制周期为主固定周期任务的任务周期，可完全同步地运行多个轴。

但存在以下限制。

- 运动控制指令的启动及运动控制指令的发行以顺序控制周期执行。
- EtherCAT 从站侧的数据更新周期取决于从站的设定。



#### 参考

- 用运动控制功能模块控制的EtherCAT从站需要在Sysmac Studio上进行“轴”的分配设定。因此，PLC 功能模块通过启动用户程序中的“运动控制指令”，向运动控制功能模块发出指令。而且，可利用运动控制功能模块，通过轴变量获取信息。
- 运动控制系统的输入输出响应时间因分配（记述有运动控制指令的）程序的任务种类不同而异。详情请参考 □「5-9-3 系统输入输出响应时间」（P.5-58）。

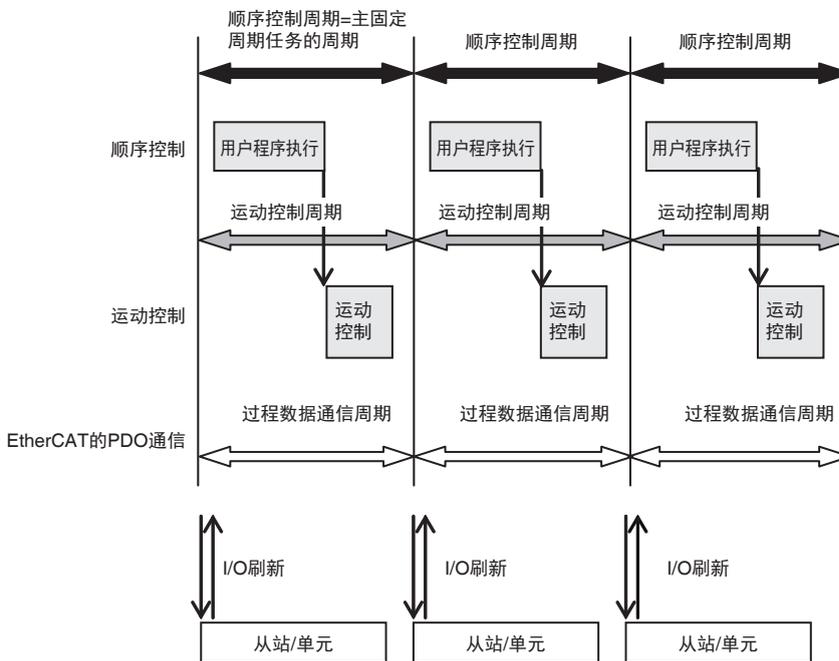
### 2-4-4 顺序控制和运动控制的同步

顺序控制的控制周期为分配了程序或 I/O 刷新的任务之任务周期。而运动控制始终以主固定周期任务的任  
务周期执行。此外，运动控制中使用的 EtherCAT 从站的过程数据通信周期与主固定周期任务同步。

在 NY 系列控制器上，按主固定周期任务的周期执行轴和轴组的运动控制和过程数据通信。

因此，若将顺序控制程序分配到主固定周期任务中，可使顺序控制和运动控制的控制周期及 EtherCAT 的  
过程数据通信周期同步。

将程序分配到主固定周期任务时的图如下所示。在下图中，顺序控制周期、运动控制的周期、EtherCAT  
的过程数据通信周期全部同步。



## 2-5 控制器内的数据概要

NY 系列控制器中包含各种设定、用户程序、当前值等数据。主要的数据及其内容如下所示。  
关于 NY 系列控制器的所有数据，请参考 □「A-5 控制器内数据的属性」(P.A-79)。

数据类型		内容	
各种设定	EtherCAT 构成	EtherCAT 从站构成	EtherCAT 从站的构成信息。
		EtherCAT 主机设定	EtherCAT 主机的通信周期等参数设定。
	单元构成和单元设定		识别控制器所需的单元构成信息及高性能单元的初始设定信息。
	I/O 映射		对根据单元构成自动生成的 I/O 端口变量进行的分配信息。
	控制器设定	动作设定	接通电源时的动作模式、安全等设定或系统服务设定。
		内置 EtherNet/IP 端口设定	EtherNet/IP 端口的以下设定。 TCP/IP 设定、Ethernet 设定、DHCP 设定、DNS 设定、FTP 设定、SNMP 设定
	运动控制设定		“轴” / “轴组”的轴轴变量 / 轴组变量及运动控制参数设定。
	凸轮数据设定		由运动控制指令的凸轮动作中使用的相位 / 变位数据构成的凸轮表。
	事件设定		用户定义的异常 / 信息的设定。
	任务设置		任务的种类或数量、任务的执行条件、任务名称、任务中执行的程序等设定。
	数据追踪设定		数据追踪触发条件等设定。
	标签数据链接表		EtherNet/IP 上的标签数据链接设定。
	控制器名称		控制器的名称。
	操作权限认证		Sysmac Studio 对控制器进行操作所需的权限密码。
用户程序	POU (程序构成单位)	数据类型	数据类型的定义。
		全局变量	全局变量的属性信息。初始值属性和保持属性等。
	数据	变量的值。	
其它	事件日志		控制器的异常履历及电源 ON/OFF 或运行开始等异常以外的现象的履历。
	绝对编码器原点位置偏置		运动控制中，原点确定后的指令位置和从绝对编码器读取的绝对值数据的差，用于恢复带绝对编码器的伺服驱动器的当前位置。

## 2-6 控制器在各种状态下的动作

介绍 NY 系列控制器在各种状态下的用户程序执行、I/O 刷新、与外部的通信处理动作。此外，还对 NY 系列控制器处于“正常运行中”时，介绍用户程序执行状态的切换和动作模式。

### 2-6-1 控制器的状态

NY 系列控制器有“启动中”、“正常运行中”、“异常状态”三种状态。各状态的定义如下所示。

状态	定义
启动中	软件正在执行启动处理的状态
正常运行中	由软件执行任务中执行的处理或系统服务，未发生控制器异常的状态
异常状态	由软件执行任务中执行的处理或系统服务，发生控制器异常的状态

此外，“正常运行中”包括“程序模式动作中”、“运行模式动作中”、“下载中”三种状态。“正常运行中”的控制器将根据用户操作变为各种状态。各状态的定义如下所示。

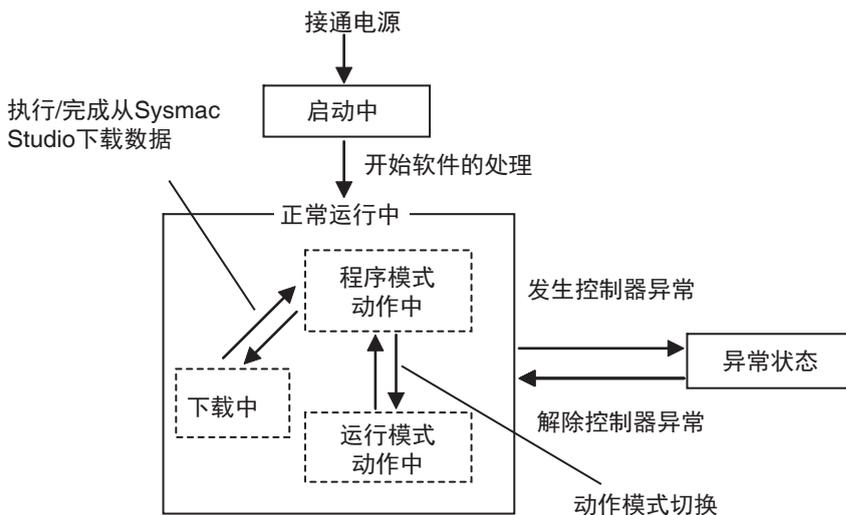
状态	定义
程序模式动作中	动作模式为程序模式时
运行模式动作中	动作模式为运行模式时
下载中	正在从 Sysmac Studio 下载数据的状态

(注) 程序模式和运行模式请参考 □□「2-6-3 动作模式」(P.2-25)。

## ● 控制器的状态变化

接通电源后，控制器变为“启动中”。控制器变为“启动中”并经过 10 ~ 20 秒后，开始软件处理，变为“正常运行中”。“正常运行中”若发生控制器异常，将变为“异常状态”。解除控制器异常后，恢复为“正常运行中”。

从“启动中”变为“正常运行中”后，控制器将根据控制设定中设定的动作模式，变为“程序模式动作中”或“运行模式动作中”。然后，通过动作模式的切换，“程序模式动作中”时变为“运行模式动作中”，“运行模式动作中”时，变为“程序模式动作中”。“程序模式动作中”若从 Sysmac Studio 下载数据，将变为“下载中”。下载完成后，恢复为“程序模式动作中”。



### 参考

- 控制器的状态可通过工业平板电脑 / 工业 BOX PC 的状态指示 LED 确认。通过状态指示 LED 确认的方法请参考 □□《NY 系列 故障排除手册 (SBCA-CN5-438)》。
- 关于控制器状态发生变化时的数据动作，请参考 □□「A-5 控制器内数据的属性」(P.A-79)。
- 关于控制器状态发生变化时的变量值，请参考 □□「6-3-9 与状态变化对应的变量值的变化」(P.6-62)。

## 2-6-2 各种状态下的动作

因 NY 系列控制器的状态变化，用户程序、I/O 刷新、与外部的通信动作也会变化。“启动中”及“正常运行中”各处理的动作如下所示。

“异常状态”请参考 □□《NY 系列 故障排除手册（SBCA-CN5-438）》。

控制器的处理	启动中的动作	正常运行中的动作		
		程序模式动作中	运行模式动作中	下载中
用户程序	停止。	停止。	执行。	停止。
EtherCAT 从站的 I/O 刷新	停止。	执行。		EtherCAT 通信变为可安全操作状态。 <sup>*1</sup>
与外部的通信	停止。	执行。		执行。 <sup>*2</sup>

\*1. 仅更新输入的值。

\*2. 标签数据链接会继续，但标签数据链接的值不更新。输出标签的值保持下载前的值。输入标签的值不反映到变量中。

## ● I/O 输出接点的值

通过 I/O 刷新处理，各状态下 I/O 输出接点的值如下。

I/O 的输出接点	启动中的动作	正常运行中的动作		
		程序模式动作中	运行模式动作中	下载中
EtherCAT 从站的输出接点	遵照各从站的设定。 <sup>*1</sup>	变为 EtherCAT 从站设备变量的值。 <sup>*2</sup>		遵照各从站的设定。

\*1. 接通电源后，开始 EtherCAT 通信所需的时间等从站设定请参考各从站的手册。

\*2. 下载完成后，EtherCAT 从站将进行初始化处理。初始化处理中的输出遵照各从站的设定。

关于变量的初始值属性，请参考 □□「6-3-8 变量的各属性说明」（P.6-56）。



## 参考

**动作模式切换后伺服驱动器的动作**

运动控制中，若从运行模式切换为程序模式，将以各轴参数中设定的最大减速度减速停止。

**EtherCAT 从站初始化处理中的动作模式切换**

EtherCAT 从站正在进行初始化处理时，可将控制器的动作模式变更为运行模式。此时，请确认通信建立后，再创建用于控制从站数据的程序。通过用户程序，可使用系统定义变量“\_EC\_PDslavTbl”（过程数据通信中从站表），确认所有从站的过程数据输出和输入是否有效。

### 2-6-3 动作模式

动作模式是指可根据动作确认或正式运行等目的进行切换的模式。根据不同目的，动作模式分为运行模式和程序模式。各动作模式中，用户程序的执行状态不同。以下表示各动作模式的目的和用户程序的执行状态。

动作模式	目的	用户程序的执行状态
运行模式 <sup>*1</sup>	执行用户程序的试运行或正式运行	执行。
程序模式	I/O 配线检查等，不执行用户程序即可进行的动作确认	不执行。

\*1. 初始设定中，若控制器从“启动中”变为“正常运行中”，将以运行模式进行动作。



#### 参考

通过动作模式的切换，执行轴的停止、运动控制指令的中断等。动作模式切换时运动控制功能模块的动作请参考 □□《NY 系列 指令基准手册 运动篇 (SBCE-CN5-380)》。

### 各动作模式下是否可通过 Sysmac Studio 及触摸屏操作

各动作模式下，可否通过 Sysmac Studio 及触摸屏进行的主要操作如下。

操作		运行模式	程序模式
Sysmac Studio	同步	不支持	支持
	在线编辑	支持	
	强制值刷新	支持	
	变更变量的值	支持	
触摸屏	变更变量的值	支持	

### 动作模式切换时变量值的保持

若从运行模式切换为程序模式或从程序模式切换为运行模式，根据变量的保持属性不同，变量值如下所示。

变量的保持属性	变量的值
非保持	有初始值设定时：指定的初始值 无初始值设定时：系统定义的初始值 <sup>*1</sup>
保持	保持动作模式变更前的值

\*1. 因变量的数据类型不同而异。□□「初始值指定为空白时」(P.6-59)を参照してください。

关于控制器状态发生变化时的变量值，请参考 □□「6-3-9 与状态变化对应的变量值的变化」(P.6-62)。

### 动作模式的设定及切换

接通电源后开始运行时，控制器将按照控制器设定的动作模式动作。此外，正常运行中时，用户可根据目的切换动作模式。设定及切换可在 Sysmac Studio 上进行。

### ● 接通电源后开始运行时的动作模式设定

接通电源后开始运行时，按照“电源接通时模式”中设定的动作模式来运行控制器。在控制器设定的“动作设定”中，将“电源接通时模式”设定为运行模式或程序模式。关于“电源接通时模式”的设定，请参考 □□「4-2-2 控制器设定」(P.4-4)。

### ● 运行中的动作模式切换

动作模式通过 Sysmac Studio 切换。从菜单栏的 [控制器] - [动作模式] 中选择运行模式或程序模式。



#### 使用注意事项

进行接通电源时的动作模式设定及动作模式的切换前，请确认其不会对设备造成不良影响。

## 动作模式的确认

动作模式可通过工业电脑的 RUN LED 或 Sysmac Studio 确认。

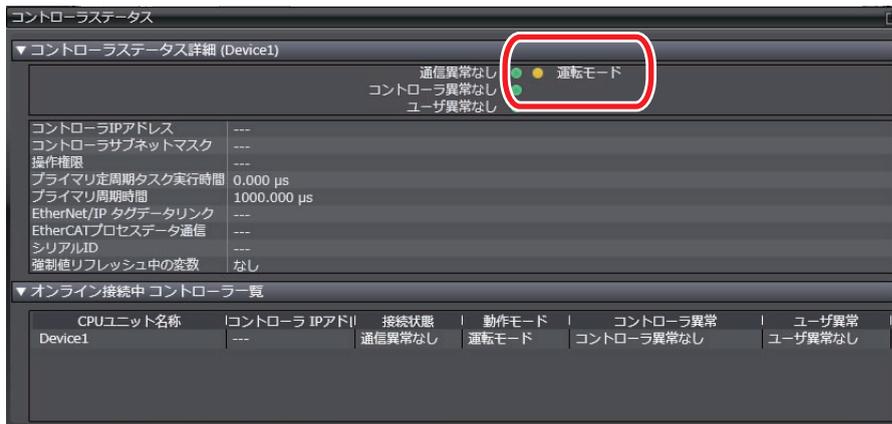
### ● 通过 RUN LED 确认

根据工业电脑的 RUN LED 显示，动作模式如下。

RUN LED 的显示	动作模式
熄灭	程序模式
点亮	运行模式

### ● 通过 Sysmac Studio 确认

动作模式可通过 Sysmac Studio 的控制器状态窗口确认。以下控制器状态窗口中，显示“运行模式”。



## 2-7 Windows 状态的监视及变更

NY 系列中，可通过控制器监视 Windows 的状态，或变更状态。

### 2-7-1 Windows 的状态

NY 系列中安装的 Windows 有以下 5 种状态。

状态	说明
启动中 (Booting)	Windows 正在执行启动处理的状态
运行中 (Running)	Windows 正在运行的状态
关机中 (ShuttingDown)	Windows 正在执行关机处理的状态
停止中 (Halted)	Windows 处于停止状态 (完成关机处理，未运行的状态)
异常发生中 (Error)	Windows 正在发生异常的状态



#### 参考

可通过系统定义变量及指令来识别状态。  
可识别“运行中”、“停止中”、“异常发生中”。无法识别“启动中”和“关机中”的状态。

### 2-7-2 Windows 状态的监视

Windows 状态的监视中使用系统定义变量或指令。

#### Windows 状态监视相关的系统定义变量

系统定义变量中，显示 Windows 状态中的 3 种，即“运行中”、“停止中”及“异常发生中”。  
系统定义变量的详情请参考 □「功能分类：OS(Windows) 相关」(P.A-53)。

变量名称	名称	功能	数据类型	R/W
_OSRunning	OS 运行中标志	控制器观测到 OS(Windows) 的状态为运行中时变为 TRUE。	BOOL	R
_OSHalted	OS 停止中标志	控制器观测到 OS(Windows) 的状态为停止中时变为 TRUE。	BOOL	R
_OSErrorState	OS 异常状态标志	控制器判断 OS(Windows) 的状态为异常发生中时变为 TRUE。	BOOL	R

#### Windows 状态监视相关的指令

作为监视 Windows 状态的指令，有以下指令。

指令的详情请参考 □《NY 系列 指令基准手册 基本篇 (SBCA-CN5-437)》。

指令	指令名称	概要
IPC_GetOSStatus	读取 OS 状态	读取 OS(Windows) 的状态。读取的状态为 Windows 状态中的 3 种，即“运行中”、“停止中”及“异常发生中”。

### 2-7-3 Windows 状态的变更

要变更 Windows 的状态，需要使用指令。

#### Windows 状态变更相关的指令

作为变更 Windows 状态的指令，有以下指令。

指令的详情请参考 □□《NY 系列 指令基准手册 基本篇（SBCA-CN5-437）》。

指令	指令名称	概要
IPC_RebootOS	OS 重新启动	重新启动 OS(Windows)。
IPC_Shutdown	关机	向 OS(Windows) 发出关机请求通知。

### 2-7-4 Windows 状态的变化

启动工业电脑后，Windows 将变为“启动中”。启动处理结束后，变为“运行中”。

运行过程中如果因 IPC\_Shutdown 指令、UPS 的断电检测等而停止供电、按下电源按钮等，将变为“关机中”。关机处理结束后，变为“停止中”。

此外，如果在运行过程中执行 IPC\_RebootOS 指令，将变为“关机中”。关机处理结束后，变为“启动中”，启动处理结束后，变为“运行中”。

要从“异常发生中”进行修复时，也执行 IPC\_RebootOS 指令。

Windows 的状态变为“运行中”、“停止中”及“异常发生中”时，将登录控制器事件，表示发生变化。但是，通过执行 IPC\_Shutdown 指令，控制器向 Windows 发出关机请求通知后，如果经过 [ 关机处理等待时间设定 ] 的 [OS] 中设定的时间后，Windows 仍不变为“停止中”，则不会登录控制器事件。

## 2-8 关机处理

NY 系列工业电脑请务必连接 UPS（不间断电源）使用。连接 UPS 后，即使在停电等意外断电时，也不会造成数据或文件损失，确保正确关机。

UPS 检测到断电或表示要按下电源按钮的系统定义变量变为 TRUE 时，可利用控制器的用户程序执行处理，为断电做好准备。

请根据需要编写以下处理程序。

- 将控制器的用户程序中使用的、无保持属性的变量，复制为有保持属性的变量。
- 停止对控制器的虚拟 SD 存储卡进行文件操作。
- 安全停止控制器正在执行的运动控制动作。

用户程序的处理完成后，控制器系统将执行关机处理，如保持有保持属性的变量等。控制器系统完成关机处理后，控制器系统将向 Windows 发出关机请求通知。

Windows 可将来自控制器的通知作为触发，进行关机处理，如结束应用程序等。

关机处理的详情请参考「关机处理的详情」（P.2-33）。



### 使用注意事项

- 未连接 UPS 时或 UPS 的电池电压下降时，断电时不会执行关机处理，因此无法保持有保持属性的变量、绝对编码器原点位置偏置及事件日志。
- 请选择容量足够的 UPS，确保后备时间大于控制器端及 Windows 端的关机处理时间。如果后备时间较短，关机处理将无法正常结束。



### 参考

关于 UPS 的连接方法，请参考 □□《NY 系列 工业平板电脑 用户手册 硬件篇（SBCA-CN5-435）》、□□《NY 系列 工业 BOX PC 用户手册 硬件篇（SBCA-CN5-434）》。

## 2-8-1 断电及按下电源按钮时的处理

下面对控制器动作过程中，发生断电、按下电源按钮或执行了 IPC\_Shutdown 指令时的控制器处理进行说明。因 UPS 的连接有无和电源按钮的按下时间不同，处理也不同。

操作	UPS 连接	控制器的处理
断开电源	有 UPS 连接	将来自 UPS 的信号作为触发，执行关机处理。 <sup>*1 *2</sup>
	无 UPS 连接，或 UPS 的电池电压降低	不执行关机处理。 <sup>*3</sup>
按下电源按钮（5 秒以内）	—	将按下电源按钮作为触发，执行关机处理。 <sup>*1 *2</sup>
执行 IPC_Shutdown 指令	—	控制器系统执行关机处理。 <sup>*1</sup>
按下电源按钮（5 秒以上）	—	强制关机。不执行关机处理。 <sup>*3</sup>

\*1. 关机处理的内容请参考 □□「2-8-3 关机处理的内容」（P.2-32）。

\*2. 执行 IPC\_Shutdown 指令或经过 [ 关机处理等待时间设定 ] 的 [ 用户程序 ] 中设定的时间后，控制器系统将执行关机处理。

\*3. 未执行关机处理的情况下，下次接通电源时，将发生监视信息的“事件日志保存失败”及全部停止故障“未保存保持变量的当前值”。

此外，控制器中的数据将变为以下状态。

- 有保持属性的变量及绝对编码器原点位置偏置变为上次接通电源时的值。
- 从上次启动到未执行关机处理时为止的事件日志会丢失。



## 参考

若在控制器未运行的状态下按下电源按钮，工业电脑将以通常模式启动。如果在拆下电源连接器的状态下按下电源按钮，然后在按下的状态下连接电源连接器且经过 10 秒，无论控制器设定的 [ 电源接通时模式 ] 设定如何，控制器均以程序模式开始运行。此时，会发生监视信息“安全模式”。关于启动和结束，请参考 □□《NY 系列 工业平板电脑 / 工业 BOX PC 用户手册 安装篇 (SBCA-CN5-441)》。

## 2-8-2 关机处理的设定

在控制器设定中设定以下项目。

设定项目	设定中项目	功能说明	设定值	默认	反映时序	运行模式中改写
关机处理等待时间设定	用户程序	设定等待时间，即表示断电检测的 UPS 信号变为 ON 或通过电源按钮接收到关机指示后，控制器开始执行关机处理为止的时间。	1 ~ 30s	5s	设定传送时	不支持
	OS	设定等待时间，即从控制器向 Windows 发出关机请求通知后，到 Windows 检测到关机完成为止的时间。	30 ~ 300s	60s	设定传送时	不支持

### ● 用户程序

表示断电检测的 UPS 信号由控制器的用户程序监视。将来自 UPS 的信号作为触发，由用户程序执行断电时所需的处理。

控制器系统会按 [用户程序] 中设定的时间，等待用户程序处理完成。因此，[用户程序] 中设定的时间应长于用户程序的处理时间。如果设定的时间较短，用户程序的处理将中断。

### ● OS

控制器系统完成关机处理后，控制器系统将向 Windows 发出关机请求通知。Windows 可将来自控制器的通知作为触发，进行关机处理，如结束应用程序等。

控制器系统通过监视 Windows 的状态，可检测 Windows 是否已正常完成关机。

如果在 [OS] 中设定的时间之内 Windows 完成关机处理，将登录表示已变为“停止中”的控制器事件。因此，[OS] 中设定的时间应长于 Windows 的处理时间。如果设定的时间较短，Windows 的处理将中断，因此无法正常结束 Windows 的应用程序。

未登录表示已变为“停止中”的控制器事件时，可能是因为 [OS] 中设定的时间短于 Windows 的处理时间。

### 2-8-3 关机处理的内容

UPS 检测到断电时或按下电源按钮时，与各状态对应的系统定义变量将变为 TRUE。通过监视这些系统定义变量的状态，可将需要的数据复制为有保持属性的变量，或者安全停止正在执行的运动控制动作。系统定义变量的监视或断电时所需的处理应先写入控制器的用户程序中。

用户程序中断电时所需的处理结束后，请执行 IPC\_Shutdown 指令。执行 IPC\_Shutdown 指令后，控制器系统将执行关机处理。

#### 作为关机处理的触发使用的系统定义变量

用户程序中，作为开始处理的触发使用的系统定义变量如下。

变量名称	名称	功能	数据类型	R/W
_SelfTest_UPSSignal	UPS 信号检测标志	检测到 UPS 发出的临时停电信号时变为 TRUE。	BOOL	R
_RequestShutdown	关机请求标志	如果在运行过程中按下电源按钮，将变为 TRUE。	BOOL	R

系统定义变量的详情请参考 □□「功能分类：电源管理相关」（P.A-52）。

#### 发出关机请求通知的指令

向控制器系统通知用户程序已完成处理，从控制器系统向 Windows 发出关机请求通知的指令如下。

指令	指令名称	概要
IPC_Shutdown	关机	开始控制器系统的关机处理，完成后，向 Windows 发出关机请求通知。

指令的详情请参考 □□《NY 系列 指令基准手册 基本篇（SBCA-CN5-437）》。

## 关机处理的详情

下面介绍控制器在“运行模式动作中”、“程序模式运行中”及“启动中”等状态下的关机处理详情。

### ● 运行模式动作中

正在执行控制器的用户程序。

运行模式动作中时，为了避免控制器所管理设备的运行数据丢失，应执行用户程序中已编写的、系统定义变量的监视或断电时所需的处理。用户程序的处理完成后，控制器系统将执行关机处理，然后，在 Windows 上执行关机处理，如结束应用程序等。

下面介绍运行模式动作中时的关机处理详情。

- 1** 请通过用户程序监视系统定义变量 “\_SelfTest\_UPSSignal (UPS 信号检测标志)” 及 “\_RequestShutdown (关机请求标志)”。
- 2** 系统定义变量 “\_SelfTest\_UPSSignal (UPS 信号检测标志)” 或 “\_RequestShutdown (关机请求标志)” 变为 TRUE 时，请根据需要通过用户程序执行断电时所需的处理。
- 3** 断电时所需的处理结束后，请通过用户程序执行 IPC\_Shutdown 指令。
  - 执行 IPC\_Shutdown 指令后，控制器系统将开始关机处理。
  - 将保持有保持属性的变量、绝对编码器原点位置偏置及事件日志等。
  - 控制器向 Windows 发出关机请求通知，Windows 开始关机处理。
  - 即使未执行 IPC\_Shutdown 指令，在经过 [关机处理等待时间设定] 的 [用户程序] 中设定的时间后，控制器系统将开始关机处理。
  - Windows 完成关机后，将登录表示 Windows 已变为“停止中”的控制器事件。  
但是，控制器向 Windows 发出关机请求通知后，如果经过 [关机处理等待时间设定] 的 [OS] 中设定的时间后，Windows 仍不变为“停止中”，则不会登录控制器事件。
  - 将来自 UPS 的信号作为触发开始关机处理时，在关机处理完成后，工业电脑 I/O 连接器的“电源状态输出”将变为 ON。
  - “电源状态输出”变为 ON 后，UPS 将停止供电。

### ● 程序模式动作中

未执行控制器的用户程序。

程序模式动作中时，由于控制器未执行设备的控制，所以无需通过用户程序执行断电时所需的处理。控制器系统的关机处理完成后，在 Windows 上执行关机处理，如结束应用程序等。

下面介绍程序模式动作中时的关机处理详情。

- 系统定义变量“\_SelfTest\_UPSSignal（UPS 信号检测标志）”或“\_RequestShutdown（关机请求标志）”变为 TRUE 时，控制器系统将立即开始关机处理。
- 将保持有保持属性的变量、绝对编码器原点位置偏置及事件日志等。
- 控制器向 Windows 发出关机请求通知，Windows 开始关机处理。
- Windows 完成关机后，将登录表示 Windows 已变为“停止中”的控制器事件。  
但是，控制器向 Windows 发出关机请求通知后，如果经过 [ 关机处理等待时间设定 ] 的 [OS] 中设定的时间后，Windows 仍不变为“停止中”，则不会登录控制器事件。
- 将来自 UPS 的信号作为触发开始关机处理时，在关机处理完成后，工业电脑 I/O 连接器的“电源状态输出”将变为 ON。
- “电源状态输出”变为 ON 后，UPS 将停止供电。

### ● 启动中

控制器正在执行启动处理的状态。

动作中时，由于控制器未执行设备的控制，所以无需通过用户程序执行断电时所需的处理。

控制器系统的关机处理完成后，在 Windows 上执行关机处理，如结束应用程序等。

下面介绍启动中时的关机处理详情。

- 系统定义变量“\_SelfTest\_UPSSignal（UPS 信号检测标志）”或“\_RequestShutdown（关机请求标志）”变为 TRUE 时，控制器系统将立即开始关机处理。
- 将保持有保持属性的变量、绝对编码器原点位置偏置及事件日志等。
- 控制器向 Windows 发出关机请求通知，Windows 开始关机处理。
- Windows 完成关机后，将登录表示 Windows 已变为“停止中”的控制器事件。  
但是，控制器向 Windows 发出关机请求通知后，如果经过 [ 关机处理等待时间设定 ] 的 [OS] 中设定的时间后，Windows 仍不变为“停止中”，则不会登录控制器事件。
- 将来自 UPS 的信号作为触发开始关机处理时，在关机处理完成后，工业电脑 I/O 连接器的“电源状态输出”将变为 ON。
- “电源状态输出”变为 ON 后，UPS 将停止供电。

## 2-9 控制器重置

通过 Sysmac Studio 或 Industrial PC Support Utility 执行控制器重置，可以只重置控制器。

- **通过 Sysmac Studio 进行控制器重置**

控制器的动作模式为程序模式时，可执行控制器重置。

关于通过 Sysmac Studio 进行控制器重置，请参考□□《Sysmac Studio Version 1 操作手册（SBCA-CN5-362）》。

- **通过 Industrial PC Support Utility 进行控制器重置**

控制器的动作模式为程序模式时，可执行控制器重置。

关于通过 Industrial PC Support Utility 进行控制器重置，请参考□□《NY 系列 工业平板电脑 / 工业 BOX PC 用户手册 安装篇（SBCA-CN5-441）》。



# 3

## I/O 单元和从站 / 单元构成

本章介绍 I/O 端口的使用方法、从站 / 单元构成的创建和步骤、各功能的分配。

---

<b>3-1</b>	<b>从站 / 单元构成的步骤</b> .....	<b>3-2</b>
<b>3-2</b>	<b>EtherCAT 从站构成的创建</b> .....	<b>3-4</b>
<b>3-3</b>	<b>I/O 端口和设备变量</b> .....	<b>3-5</b>
3-3-1	I/O 端口 .....	3-5
3-3-2	I/O 端口名称 .....	3-6
3-3-3	设备变量 .....	3-6
<b>3-4</b>	<b>向单元分配变量</b> .....	<b>3-8</b>
3-4-1	向单元分配变量的方法 .....	3-8
3-4-2	单元中分配的变量的使用方法 .....	3-9
<b>3-5</b>	<b>“轴”的创建和伺服驱动器 / 编码器输入从站的分配</b> .....	<b>3-11</b>
3-5-1	概要 .....	3-11
3-5-2	轴变量 / 轴组变量 .....	3-11
3-5-3	“轴”及轴变量的创建和使用方法 .....	3-13

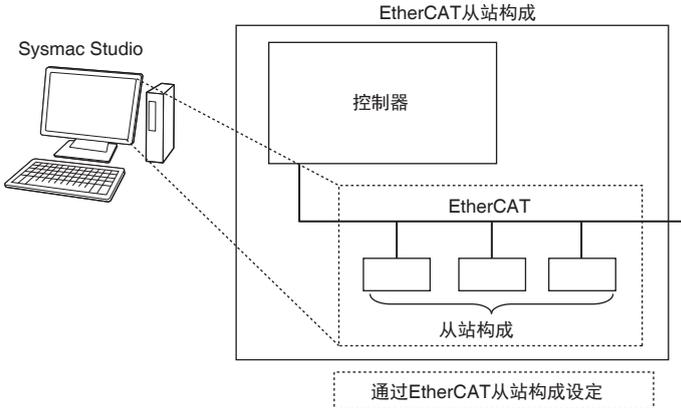
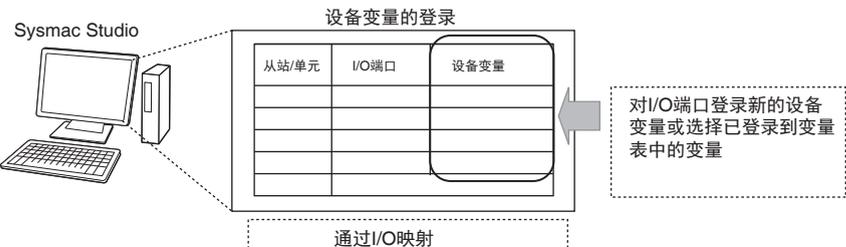
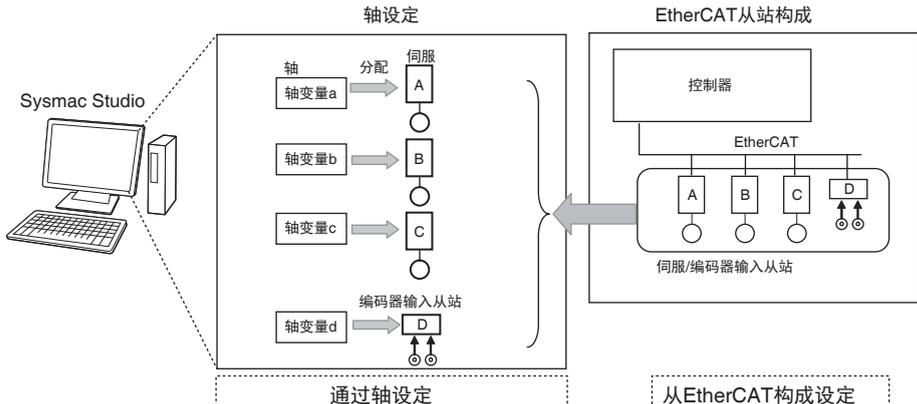
## 3-1 从站 / 单元构成的步骤

下面介绍从站 / 单元构成的步骤。

在整体步骤中，从站 / 单元构成与以下阴影标示的位置对应。

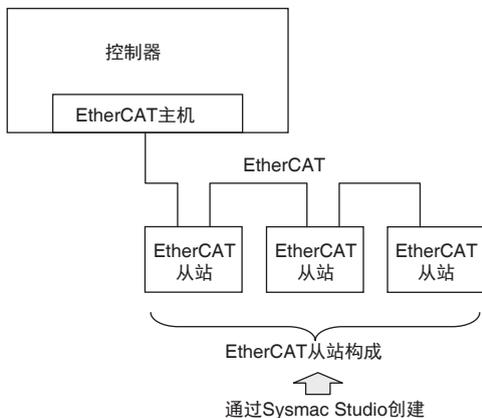


详情请参考 □「1-4 NY 系列的整体使用步骤」(P.1-10)。

STEP1 EtherCAT 从站构成的创建 (使用 EtherCAT 时)	参考
<p>• EtherCAT 从站构成的创建</p>  <p>通过EtherCAT从站构成设定</p>	<p>☐「3-2 EtherCAT 从站构成的创建」(P.3-4)</p>
<p>STEP2 向 I/O 端口分配设备变量</p> <p>• 设备变量的登录</p>  <p>通过I/O映射</p>	<p>☐「2-3-1 变量的种类」(P.2-10) ☐「3-3 I/O 端口和设备变量」(P.3-5)</p>
<p>STEP3 “轴”的创建和向伺服 / 编码器输入从站的分配 (运动控制时)</p> <p>1. “轴”的创建 2. 向 EtherCAT 构成上的伺服 / 编码器输入从站分配</p>  <p>通过轴设定</p> <p>从EtherCAT构成设定</p>	<p>☐「3-5 “轴”的创建和伺服驱动器 / 编码器输入从站的分配」(P.3-11)</p>

## 3-2 EtherCAT 从站构成的创建

为了访问与 NY 系列连接的各从站，在 Sysmac Studio 上创建从站构成。在 Sysmac Studio 的 EtherCAT 构成编辑画面中创建将 NY 系列识别为“正”的 EtherCAT 从站构成。



将自动登录构成从站的“I/O 端口”。有后面的用户对“I/O 端口”分配“设备变量”。  
在用户程序中指定“设备变量”，即可访问各从站。

EtherCAT 从站构成的创建方法请参考□□《Sysmac Studio Version 1 操作手册 (SBCA-CN5-362)》中的“[EtherCAT] 的构成和设定”。



### 参考

连接 EtherCAT 从站终端时，除了创建 EtherCAT 从站构成外，还需要进行从站终端的构成和动作设定。从站终端的构成和动作设定的详情请参考□□《NX 系列 EtherCAT 耦合器单元 用户手册 (SBCD-CN5-361)》。



### 参考

EtherCAT 从站为伺服 / 编码器输入从站时，登录到 EtherCAT 从站构成后，再创建“轴”，将自动登录轴变量。

请参考□□「3-5“轴”的创建和伺服驱动器 / 编码器输入从站的分配」(P.3-11)。

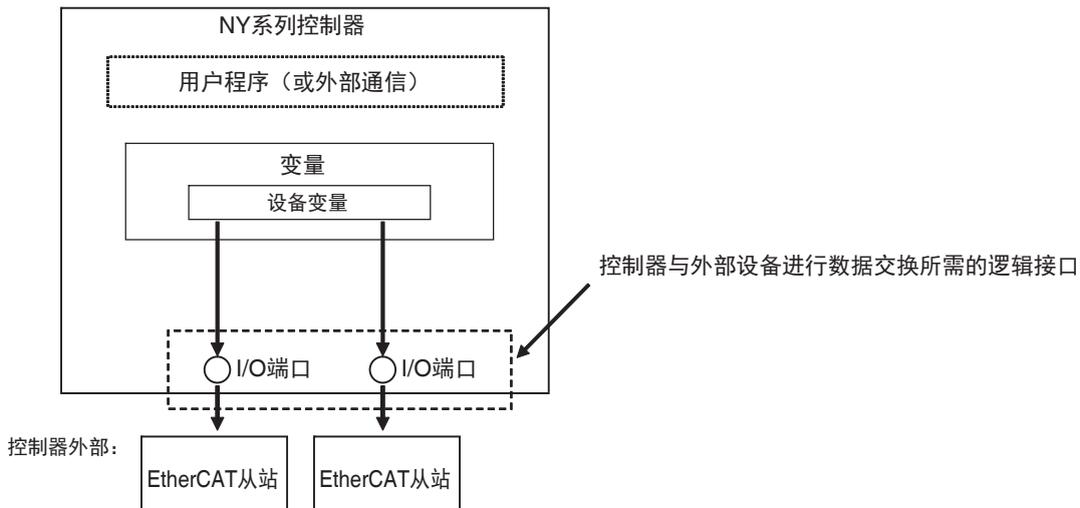
## 3-3 I/O 端口和设备变量

下面介绍访问 NY 系列控制器的 EtherCAT 从站所需的 I/O 端口及设备变量。

### 3-3-1 I/O 端口

“I/O 端口”是指 NY 系列控制器与外部设备（从站 / 单元）进行数据交换所需的逻辑接口。

在 Sysmac Studio 上创建从站 / 单元构成后，可自动生成 I/O 端口。在 I/O 端口上分配设备变量后，通过用户程序访问。



在 Sysmac Studio 上创建 EtherCAT 从站构成及单元构成或从实际控制器中读取这些信息后，“I/O 端口”将自动登录到 I/O 映射中。

登录的“I/O 端口”可在 Sysmac Studio 的“I/O 映射”中确认。

I/O 映射：

位置	ポート	説明	R/W	データ型	変数	変数コメント
CF	CPU・増設ラック					
[0]	CPUラック 0					
	CJ1W-SCU32 (シリアル通信ユニット)					
	Com_UnitSta	SCU ユニットステータス	R	WORD	J05_Com_UnitSta	
	Com_UnitLogMemErr	異常履歴データ異常	R	BOOL	J05_Com_UnitLogMemErr	
	Com_UnitPmrDatErr	プロトコルデータ異常	R	BOOL	J05_Com_UnitPmrDatErr	
	P1_PortCfg	ポート1:設定	RW	WORD	J05_P1_PortCfg	
	P1_SerSetCfg	ポート1:任意設定の有無	RW	BOOL	J05_P1_SerSetCfg	
	P1_StartBitCfg	ポート1:スタートビット	RW	BOOL	J05_P1_StartBitCfg	
	P1_DatBitCfg	ポート1:データ長	RW	BOOL	J05_P1_DatBitCfg	
	P1_StopBitCfg	ポート1:ストップビット	RW	BOOL	J05_P1_StopBitCfg	
	P1_ParityYNCfg	ポート1:パリティ	RW	BOOL	J05_P1_ParityYNCfg	
	P1_ParityBitCfg	ポート1:パリティ偶数/奇	RW	BOOL	J05_P1_ParityBitCfg	
	P1_BaudrateCfg	ポート1:伝送速度	RW	USINT		
	P1_SendDelayCfg	ポート1:送信ディレイ設定	RW	WORD		
	P1_SendDelaySetCfg	ポート1:送信ディレイ設定	RW	BOOL		
	P1_HlkCfg	ポート1:上位リンク設定	RW	WORD		
	P1_HlkCTSCfg	ポート1:CTS制御	RW	BOOL		
	P1_HlkProcCfg	ポート1:1:N/1:1手順	RW	BOOL		
	P1_HlkUnitNoCfg	ポート1:上位リンク号機	RW	USINT		
	P1_PrmUnitNoCfg	ポート1:上位リンク号機	RW	WORD		

### 3-3-2 I/O 端口名称

I/O 端口名称会自动登录。设备（单元 / 从站）为 EtherCAT 从站，I/O 端口名称如下所示。

设备为 EtherCAT 从站，且为远程 I/O 终端时，I/O 端口名称如下所示。

例) 16 点远程 I/O 终端时：

Bit00 ~ Bit15

在其他从站中，为 EtherCAT 对象字典中定义的对象名称或其一部分。

例) 模拟输入单元时：

CHO\_input16-bit

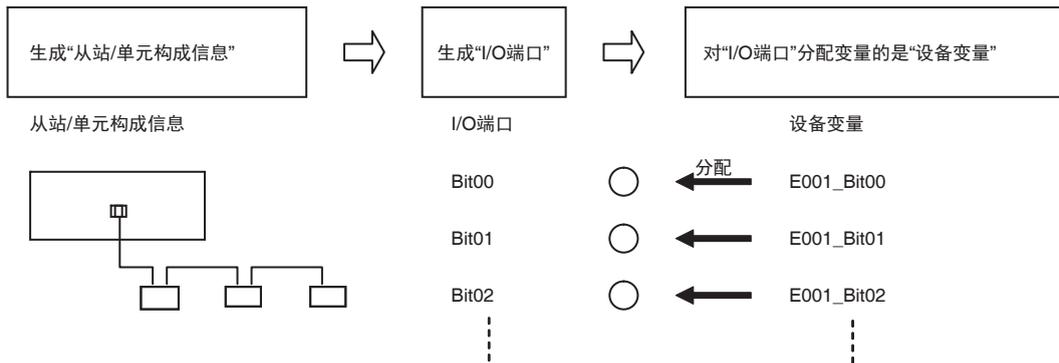
例) R88D-KN50H-ECT 时：

Position actual value、Digital inputs 等

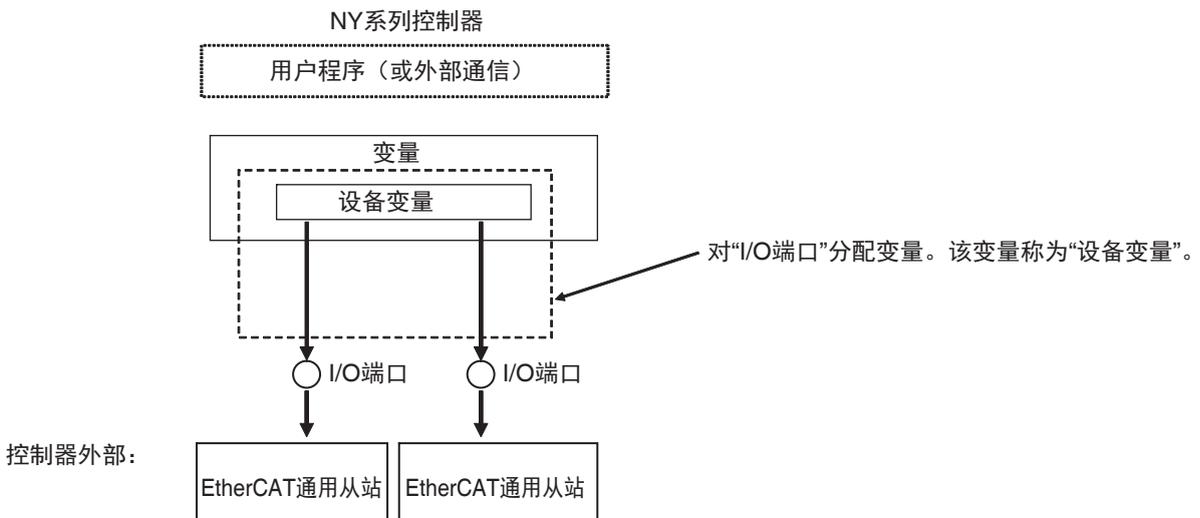
### 3-3-3 设备变量

NY 系列控制器中，不是分配控制器外部设备（从站 / 单元）分配的存储器空间上的地址，而是对该“I/O 端口”分配“变量”。

该变量称为“设备变量”。



用户程序（或外部通信）可利用该“设备变量”来访问设备（从站 / 单元）。



关于整体变量中设备变量的定位，请参考 □□「2-3-1 变量的种类」（P.2-10）。

Sysmac Studio 上的设备变量登录方法请参考 □□《Sysmac Studio Version 1 操作手册（SBCA-CN5-362）》。

## 设备变量的属性

设备变量的各属性如下所示。属性分为可变更设定的属性和不可变更的属性。

属性	设定内容	设定变更
变量名称	自动生成时：[ 设备名称 _ ] + [ I/O 端口名称 ]  EtherCAT 从站的“设备名称”默认为“E” + “001 开始的连号”。  “I/O 端口名称”请参考 □□「3-3-2 I/O 端口名称」(P.3-6)。  手动输入时：任意变量名称	可以
数据类型	遵照“I/O 端口”的数据类型。	可以
AT (分配目标) 指定	<ul style="list-style-type: none"> <li>NX 单元 (通过 EtherCAT 耦合器单元) ECAT://node#[EtherCAT 耦合器单元节点地址 ,NX 单元编号 ]/I/O 单元名称</li> <li>EtherCAT 从站 ECAT://node#[ 节点地址 ]/ [I/O 端口名称 ]</li> </ul>	不支持
保持	EtherCAT 从站设备变量：非保持	不支持
初始值	无	可以
常量	无	可以
网络公开	非公开	可以
边缘	无	不支持

各属性的含义请参考 □□「6-3-4 变量的属性」(P.6-29)。



### 参考

- 在 I/O 映射中，可进行 I/O 端口的强制值刷新。因此，可以强制进行实际 I/O 的 ON/OFF，对配线进行检查。
- 在 I/O 映射中，可以选择将设备变量登录到哪个变量表（全局变量表、各 POU 的本地变量表）中。

## 3-4 向单元分配变量

有些指令中以变量的形式指定 EtherCAT 从站终端的各单元。因此，用户需要事先将变量分配到各单元中。一旦将变量分配到单元后，即使变更单元的连接位置，该信息也会自动反映到变量的值中。因此，不需要在每次变更单元的连接位置时重新分配变量。



### 参考

还可对从站终端以外的 EtherCAT 从站分配变量。其他公司生产的 EtherCAT 从站也一样。向 EtherCAT 从站分配变量的方法与向 EtherCAT 耦合器单元和 NX 单元分配变量的方法相同。

### 3-4-1 向单元分配变量的方法

即使在 Sysmac Studio 上进行 EtherCAT 从站终端的设定，也不会自动生成单元中分配的变量。请按以下步骤设定向单元分配的变量。

- 1** 在 Sysmac Studio 的 [构成和设定]-[EtherCAT] 中，设定 EtherCAT 从站终端的构成。
- 2** 在 [构成和设定]-[I/O 映射] 中，显示 I/O 映射。  
显示设定的 EtherCAT 从站终端的各单元 I/O 映射。
- 3** 右击分配变量的单元的类型，选择 [显示节点位置端口]。  
将在 I/O 映射中追加“Node location information”端口。
- 4** 右击 [Node location information]，选择 [生成设备变量]。  
将在 [Node location information] 端口的 [变量] 栏中写入变量名称。

单元中分配的变量之数据类型为 `_sNXUNIT_ID` 型结构体。`_sNXUNIT_ID` 型结构体的详情如下所示。

变量	名称	含义	数据类型
设定为任意值	指定单元	指定的单元	<code>_sNXUNIT_ID</code>
NodeAdr	节点地址	EtherCAT 耦合器单元的节点地址	UINT
IPAdr	IP 地址 <sup>*1</sup>	EtherCAT 耦合器单元的 IP 地址	BYTE[5]
UnitNo	单元编号	指定单元的单元编号	UDINT
Path	路径 <sup>*1</sup>	指向指定单元的路径	BYTE[64]
PathLength	“Path”的有效数据长度	“Path”的有效数据长度	USINT

\*1. 只在控制器内部使用的信息。用户不会参照或编辑。



### 使用上の注意

单元中分配的变量值将在登录变量时自动设定。变量的值请勿变更。若用户变更了变量的值，控制器可能发生以外动作。



## 参考

向从站终端以外的 EtherCAT 从站分配变量时，变量的数据类型为 `_sECAT_ID` 型结构体。详情如下所示。

变量	名称	含义	数据类型
设定为任意值	指定从站	要指定的从站	<code>_sECAT_ID</code>
	NodeAdr	指定从站的节点地址	UINT

### 3-4-2 单元中分配的变量的使用方法

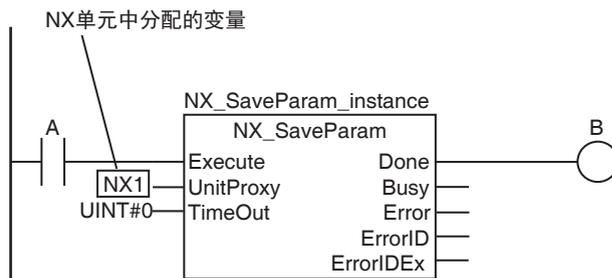
单元中分配的变量在用户程序上如下使用。无论哪种情况，都需要将与 I/O 映射中分配到单元的变量同名的变量，登录到变量表中。该变量的数据类型为 `_sNXUNIT_ID` 型结构体。

#### 单元的指定

将单元中分配的变量直接作为参数传递到指定单元的指令中。

例) `NX_SaveParam` 指令的执行

指定分配了“NX1”这一变量的 NX 单元，执行 `NX_SaveParam` 指令。向变量“UnitProxy”传递“NX1”。



#### 单元属性的指定

指定单元中分配的变量之成员，可以只指定单元的部分参数。

例) 根据 NX 单元的单元编号执行指令

若分配了“NX1”这一变量的 NX 单元的单元编号为 2，则从该单元的数据对象中读取数据。变量“`NX1.UnitNo`”表示单元编号。

```
IF (NX1.UnitNo = UINT#2) THEN
```

```
  NX_ReadObj_instance(Execute:=TRUE, UnitProxy:=NX1, Obj:=S_Obj, ReadDat:=Rdat);
```

```
END_IF;
```

## 多个单元的指定

将单元中分配的变量设为排列指定。使用循环处理，可对多个单元执行相同的处理。

例) 执行将多个 NX 单元设为数据可写入模式的指令

“NX0”、“NX1”、“NX2”为分配到 NX 单元的变量。分别代入排列 NXTable[0..2] 中后，依次执行 NX\_ChangeWriteMode 指令。

- 变量表

名称	数据类型
NXTable	ARRAY[0..2] OF_sNXUNIT_ID

- ST 程序

```
FOR i:= 0 TO 2 DO
```

```
  NX_ChangeWriteMode_instance[i](Execute:=FALSE);
```

```
END_FOR;
```

```
NXTable[0] := NX0;
```

```
NXTable[1] := NX1;
```

```
NXTable[2] := NX2;
```

```
FOR i:= 0 TO 2 DO
```

```
  NX_ChangeWriteMode_instance[i](Execute:=TRUE, UnitProxy:=NXTable[i]);
```

```
END_FOR;
```

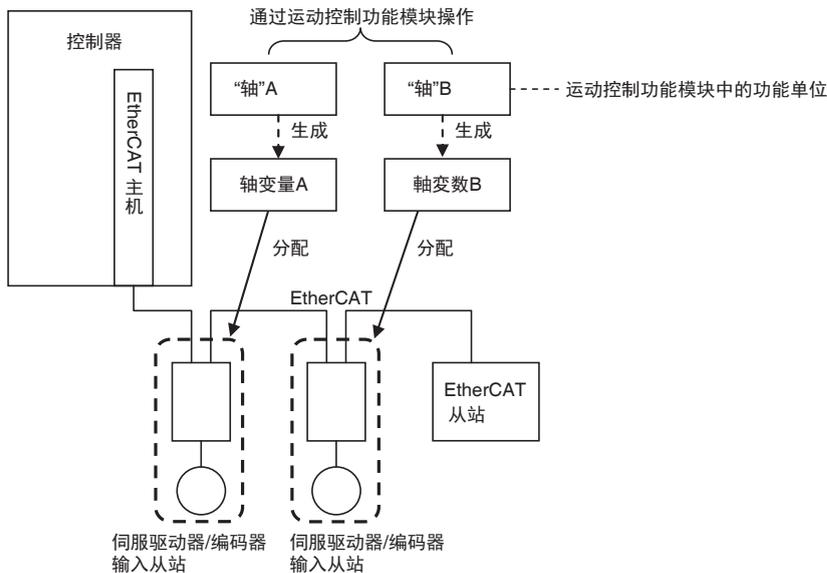
## 3-5 “轴”的创建和伺服驱动器 / 编码器输入从站的分配

下面介绍 NY 系列控制器上的“轴”创建和将伺服驱动器 / 编码器输入从站分配到“轴”。

### 3-5-1 概要

用运动控制功能模块操作 EtherCAT 从站中的伺服驱动器或编码器输入从站时，需要在 Sysmac Studio 中创建“轴”，然后分配为 EtherCAT 上的伺服轴、编码器轴。

将自动创建系统定义变量的轴变量。



利用用户程序中的运动控制指令指定轴变量，可方便地访问伺服驱动器或编码器输入从站并进行操作。

### 3-5-2 轴变量 / 轴组变量

轴变量 / 轴组变量分为以下种类。

变量种类	用途	访问对象的设备	生成方法
轴变量	系统定义的轴变量 用 Sysmac Studio 创建了“轴”时，自动生成的轴变量	用 1 个轴单独控制时	“轴”中分配的 EtherCAT 从站（伺服 / 编码器输入从站） 标配 在 Sysmac Studio 中创建“轴”，并分配轴
轴组变量	系统定义的轴组变量 用 Sysmac Studio 创建了“轴组”时，自动生成的轴组变量	对多个轴进行多轴协调控制时	“轴组”中分配的 EtherCAT 从站（伺服 / 编码器输入从站） 标配 在 Sysmac Studio 中创建“轴组”

轴变量 / 轴组变量的详情请参考 □《NY 系列 指令基准手册 运动篇（SBCE-CN5-380）》。

• 指定方法

可以用以下 Sysmac Studio 创建的变量和系统定义的变量指定。

种类	名称	
	轴变量	轴组变量
Sysmac Studio 创建的变量	MC_Axis*** (***) 按创建顺序从 000 开始升序排列) 用户可任意变更。	MC_Group*** (***) 按创建顺序从 000 开始升序排列) 用户可任意变更。
系统定义的变量	_MC_AX[***] (排列要素编号按创建顺序从 0 开始升序排列)	_MC_GRP[***] (排列要素编号按创建顺序从 0 开始升序排列)

• 用途

有以下两种。

(1) 用运动控制指令进行轴指定或轴组指定：

通过指定为运动控制指令的输入输出变量，可对欧姆龙产的伺服 / 编码器输入从站进行操作。

(2) 轴变量各成员的监视：

可用常用的指令对各伺服 / 编码器输入从站的当前位置或异常信息等进行监视。



参考

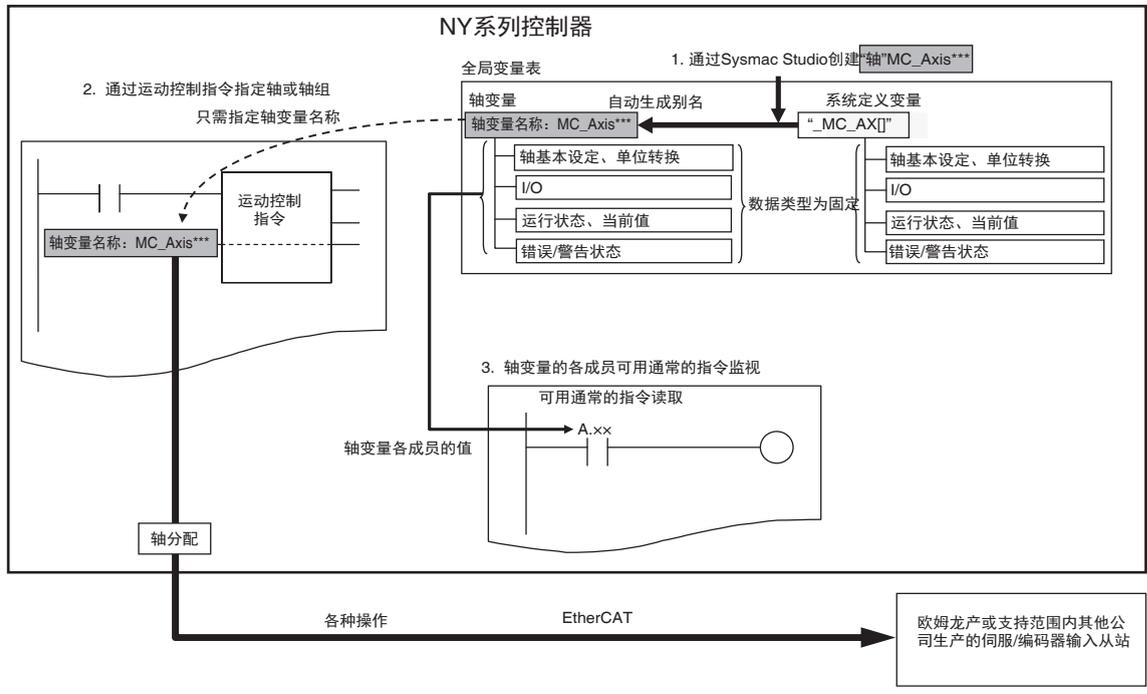
轴变量的详细介绍

1. 在 Sysmac Studio 中创建轴名称为“A”的轴。

根据系统定义变量的轴变量，自动生成变量名称为“A”的“轴变量”。轴变量由轴基本设定、单位转换、I/O、运行状态、当前值、错误 / 警告状态构成。

2. 用运动控制指令的输入输出变量指定轴变量名称“A”。可访问欧姆龙产或支持范围内其他公司生产的伺服 / 编码器输入从站，并进行各种操作。

3. 可根据需要指定轴变量，用常用的指令对各伺服 / 编码器输入从站的当前位置或异常信息等进行监视。

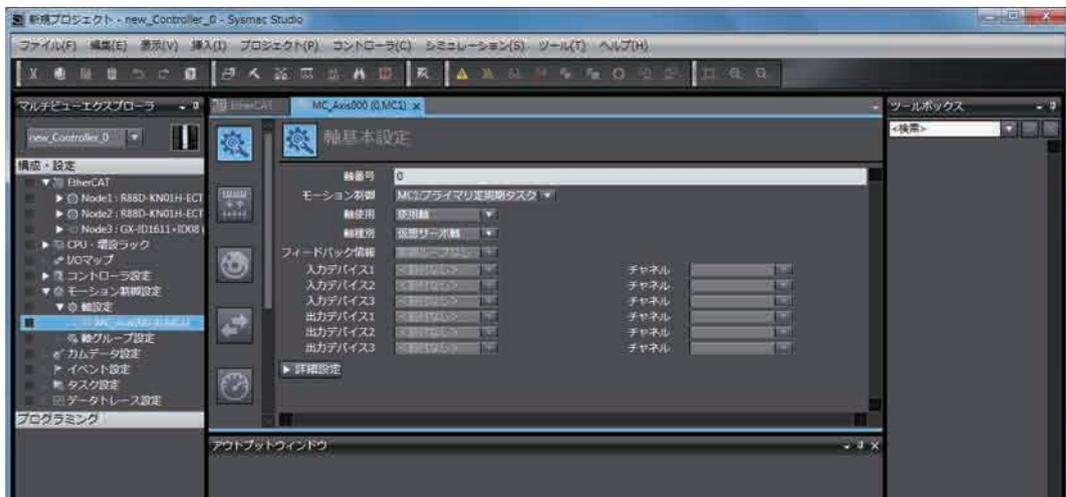


### 3-5-3 “轴”及轴变量的创建和使用方法

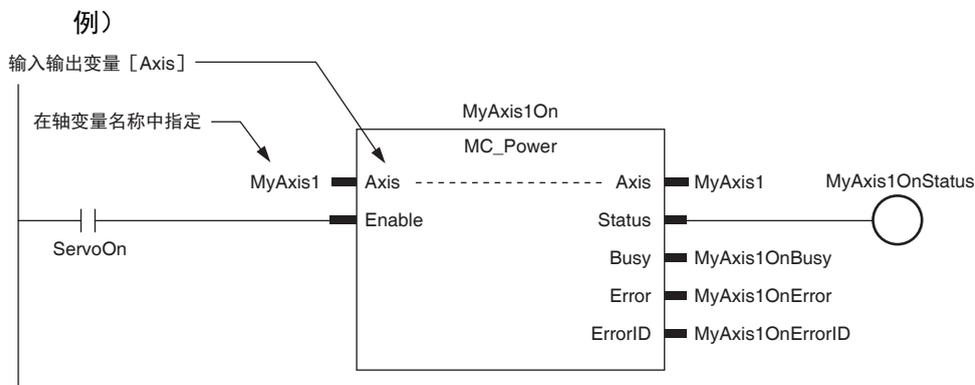
如下创建“轴”及轴变量并使用。

- 1 右击多视图浏览器的[配置和设定]-[运动控制设定]-[轴设定], 从菜单中选择[添加]-[轴设定]。根据需要, 将默认的轴变量名称“MC\_Axis\*\*\*” (\*\*\*) 按创建顺序从 000 开始升序排列) 改为其他名称。
- 2 将Sysmac Studio的EtherCAT从站构成上的伺服驱动器/编码器输入从站分配到创建的“轴”中。用 Sysmac Studio 设定以下“轴基本设定”。

区分	参数名称	设定内容
轴基本设定	轴编号	按轴的创建顺序自动设定。
	轴使用	选择 [使用轴]。
	轴种类	选择伺服轴或编码器轴。
	输入 / 输出设备	选择要分配到轴的 EtherCAT 从站。



- 3 用 Sysmac Studio 进行试运行所需的设定 (单位转换、计数模式、极限等) 及正式运行所需的设定, 并作为项目传送到控制器中。
- 4 在用户程序中, 对运动控制指令的输入输出变量“Axis”指定轴变量名称。轴变量名称指定为“运动控制设定”中插入的“轴名称”(轴变量名称)或系统定义变量。可对相应的伺服驱动器 / 编码器输入从站执行运动控制。以下表示轴变量名称指定为“MyAxis1”时的示例。



关于轴变量的详情, 请参考 □「3-5-2 轴变量 / 轴组变量」(P.3-11)。



# 4

## 控制器设定

本章介绍各功能模块的初始设定。

4

<b>4-1 控制器设定的概要</b> .....	<b>4-2</b>
<b>4-2 PLC 功能模块的初始设定</b> .....	<b>4-4</b>
4-2-1 概要 .....	4-4
4-2-2 控制器设定 .....	4-4
4-2-3 任务设置 .....	4-7
<b>4-3 运动控制功能模块的初始设定</b> .....	<b>4-12</b>
4-3-1 概要 .....	4-12
4-3-2 设定方法 .....	4-13
<b>4-4 EtherCAT 主机功能模块的初始设定</b> .....	<b>4-14</b>
<b>4-5 EtherNet/IP 功能模块的初始设定</b> .....	<b>4-15</b>

## 4-1 控制器设定的概要

下面表示控制器设定的概要。

在整体步骤中，与以下阴影标示的位置对应。



详情请参考□「1-4 NY 系列的整体使用步骤」(P.1-10)。

控制器设定	参考
<p>● <b>PLC 基本模块相关的初始设定：</b>            控制器设定：            接通电源时的动作模式、写入保护的有无、系统服务监视设定等</p>	<p>☐☐「4-2 PLC 功能模块的初始设定」(P.4-4)</p>
<p>● <b>运动控制功能模块相关的初始设定：</b>            ●轴参数：单轴动作时的运动控制用参数            ●轴组参数：多轴协调控制动作时的运动控制用参数            ●凸轮表：凸轮动作中使用的、相位和变位间的设定表</p>	<p>☐☐「4-3 运动控制功能模块的初始设定」(P.4-12)</p>
<p>● <b>EtherCAT 主机功能模块相关的初始设定：</b>            EtherCAT 构成上的 EtherCAT 主机参数：            EtherCAT 主机的过程数据通信周期等参数设定</p>	<p>☐☐「4-4 EtherCAT 主机功能模块的初始设定」(P.4-14)</p>
<p>● <b>EtherNet/IP 功能模块相关的初始设定：</b>            Ethernet 端口设定：            EtherNet/IP 端口的 TCP/IP 设定、Ethernet 设定等</p>	<p>☐☐「4-5 EtherNet/IP 功能模块的初始设定」(P.4-15)</p>

## 4-2 PLC 功能模块的初始设定

下面介绍 PLC 功能模块的初始设定。

### 4-2-1 概要

PLC 功能模块初始设定包括以下内容。

- 控制器设定
- 任务设置

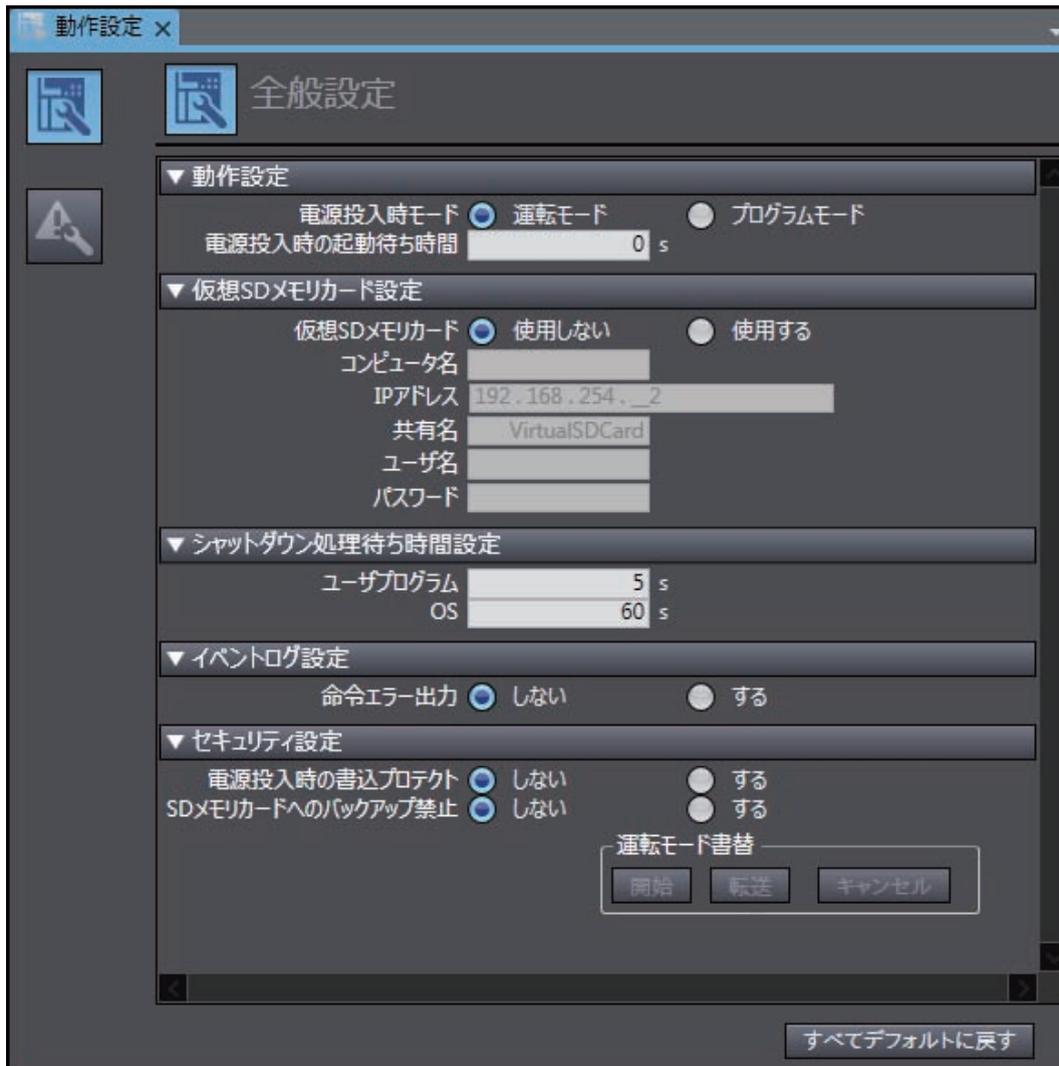
用 Sysmac Studio 分别设定 [ 配置和设定 ]-[ 控制器设定 ]、[ 任务设定 ]。

### 4-2-2 控制器设定

#### 动作设定标签

##### ● 整体设定

电源接通时或动作模式切换时的动作定义等，与控制器支持的功能相关的设定。



设定项目	设定中项目	功能说明	设定值	默认	反映时序	运行模式中改写
动作设定	电源接通时模式	设定电源接通时的控制器动作模式。	运行模式 程序模式	运行模式	设定传送时	不支持
	电源接通时的启动等待时间 <sup>*1</sup>	电源接通后正在启动时，设定等待 NS 系列触摸屏的启动，优先执行标签核对的时间。 <sup>*2</sup>	0 ~ 10s	0s	设定传送时	不支持
虚拟 SD 存储卡设定	虚拟 SD 存储卡	设定是否使用作为虚拟 SD 存储卡的共享文件夹。	使用 不使用	无效	设定传送时	不支持
	电脑名称	设定 Windows 端的电脑名称。	字符串 <sup>*3</sup>	无	设定传送时	不支持
	IP 地址	设定 Windows 端内部通信端口的 IP 地址。	IPv4 <sup>*4</sup>	192.168.254.2	设定传送时	不支持
	共享名称	设定作为虚拟 SD 存储卡使用的共享文件夹的共享名称。	字符串 <sup>*5</sup>	VirtualSDCard	设定传送时	不支持
	用户名	设定作为虚拟 SD 存储卡使用的共享文件夹的共享用户名。 <sup>*6</sup>	字符串 <sup>*7</sup>	无	设定传送时	不支持
	密码	设定作为虚拟 SD 存储卡使用的共享用户的密码。	字符串 <sup>*8</sup>	无	设定传送时	不支持
关机处理等待时间设定	用户程序	设定等待时间，即表示断电检测的 UPS 信号变为 ON 或通过电源按钮接收到关机指示后，控制器开始执行关机处理为止的时间。	1 ~ 30s	5s	设定传送时	不支持
	OS	设定等待时间，即从控制器向 Windows 发出关机指示后，到 Windows 检测到关机完成为止的时间。	30 ~ 300s	60s	设定传送时	不支持
事件日志设定	指令错误输出	发生指令错误时，设定是否将与之对应的事件输出到事件日志中。	否 是	否	设定传送时	不支持
安全设定	电源接通时的写入保护	设定在接通控制器的电源时，是否自动启用写入保护。	否 是	否	电源接通时	支持
	禁止向 SD 存储卡备份 <sup>*9</sup>	设定是否禁止向 SD 存储卡备份。	否 是	否	设定传送时	支持

- \*1. 将 NS 系列触摸屏连接到控制器的内置 EtherNet/IP 端口上，设定为在 NS 系列触摸屏和控制器同时接通电源时有效。
- \*2. 利用本设定，可缩短 NS 系列触摸屏的标签核对处理时间。以标签核对为优先时，请设定为“10”，不优先时，设定为“0”。设定为“10”时，控制器在电源接通后的“启动中”时，将优先执行 NS 触摸屏的标签核对处理约 10 秒钟，然后变为“正常运行中”。在“启动中”优先执行标签核对处理的一部分，可以缩短标签核对所需的时间。设定为“1 ~ 10”时，无论是否有 NS 系列触摸屏，都只按设定的时间优先执行标签核对处理，因此控制器变为“正常运行中”所需的时间会变长。未连接 NS 系列触摸屏时，或不同时接通电源时，请设定为“0”。
- \*3. 可使用的字符为 0 ~ 9、A ~ Z、a ~ z、-，字符数为 1 ~ 15 个字符（不含 NULL）。不区分大小写。
- \*4. 不可设定为以下 IP 地址。  
以 127（10 进制数）开始的 IP 地址  
D 类 IP 地址（224.0.0.0 ~ 239.255.255.255）  
E 类 IP 地址（240.0.0.0 ~ 255.255.255.255）
- \*5. 可使用的字符为 0 ~ 9、A ~ Z、a ~ z、-，字符数为 1 ~ 32 个字符（不含 NULL）。不区分大小写。
- \*6. 指定域时，为 <域名><用户名>。不存在同名的域用户和本地用户时，可省略域名的指定。
- \*7. 可使用的字符为 0 ~ 9、A ~ Z、a ~ z、及半角符号（`~!#\$%^&()\_{}|\'），字符数为 1 ~ 274 个字符（不含 NULL）。不区分大小写。

## 4 控制器设定

- \*8. 可使用的字符为 0 ~ 9、A ~ Z、a ~ z、及半角符号 ( ` ~ ! @ # \$ % ^ & \* ( ) \_ - + = { } [ ] \ | ; : ' < > . ? / )，字符数为 8 ~ 32 个字符（不含 NULL）。区分大小写。
- \*9. NY 系列控制器中的对象为虚拟 SD 存储卡。

操作项目	功能说明
运行模式改写	更改可在运行模式中改写的设定项目之设定值。 关于操作，请参考  《Sysmac Studio Version 1 操作手册（SBCA-CN5-362）》。 开始                   : 变为可改写设定值的状态。 传送                   : 向控制器传送设定值。将改写设定值。 Cancel(取消)       : 变为禁止改写设定值的状态。



### 使用上の注意

将 [ 事件日志设定 ]-[ 指令错误输出 ] 设定为 [ 是 ] 时，若反复执行有异常的指令，将每次输出指令错误。因此，事件日志的记录条数可能超出上限。此时，将从旧到新依次覆盖，敬请注意。

## 4-2-3 任务设置

## ● 任务设置

进行任务的追加和各任务的设定。



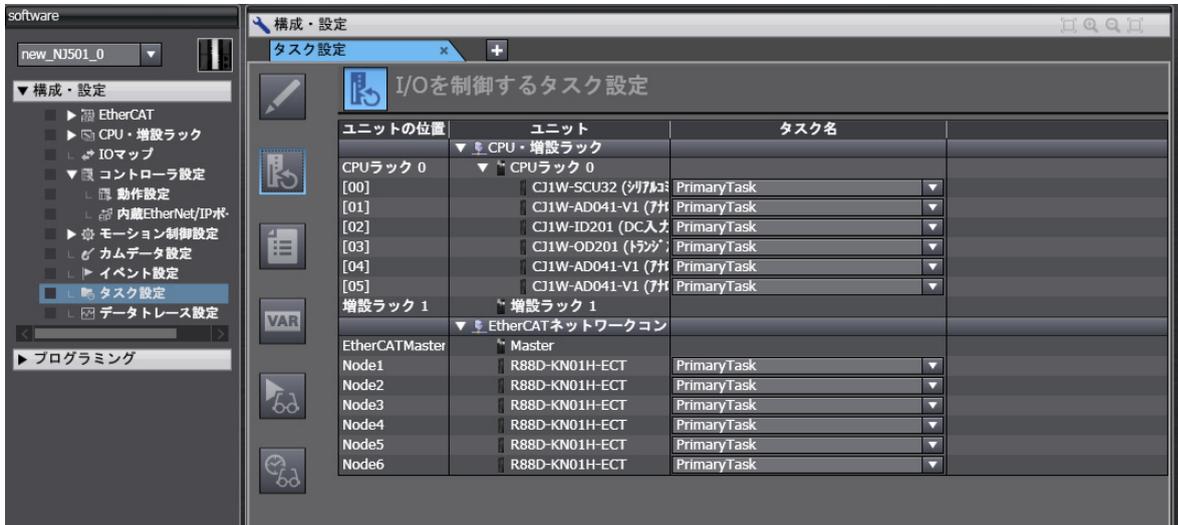
設定項目	設定中項目	功能说明	设定值	默认	反映时序	运行模式中 改写
任务类型		选择任务类型。	主固定周期任务 固定周期任务〔优先级 16〕 固定周期任务〔优先级 17〕 固定周期任务〔优先级 18〕 事件任务〔优先级 8〕 事件任务〔优先级 48〕	主固定周期任务	设定传送时	不支持
	任务的执行 优先级	选择任务执行优先级。	根据“任务类型” 自动选择	主 固定周期任务：4	设定传送时	不支持
任务名称		设定任务名称。	字符串	主 固定周期任务： PrimaryTask 固定周期任务： PeriodicTask0 事件任务： EventTask0	设定传送时	不支持
周期 / 执行条件		选择任务周期。	主固定周期任务： 500μs ~ 8ms (可以 250μs 为 单位设定) 固定周期任务： 1ms ~ 100ms (可以 250μs 为 单位设定) 事件任务： 通过指令执行，变 量的条件式一致	主 固定周期任务： 1ms 固定周期任务： 10ms (优先级 16、17、18) 事件任务： 通过指令执行	设定传送时	不支持

## 4 控制器设定

设定项目	设定中项目	功能说明	设定值	默认	反映时序	运行模式中改写
任务周期超限检测		设定在超过任务周期时，是否检测错误。	<ul style="list-style-type: none"> <li>检测（发生轻度故障等级的控制器异常）</li> <li>不检测（监视信息等级，记录到日志中。）</li> </ul>	主固定周期任务、固定周期任务：检测 事件任务：固定为不检测	设定传送时	不支持
任务执行超时时间		设定任务执行超时时间。若检测到超过超时时间，将发生任务执行超时。	主固定周期任务、固定周期任务：周期时间 ×1 倍～周期时间 ×5 倍 事件任务：执行优先级 8 时 1～500ms 执行优先级 48 时 1ms～10s	主固定周期任务、固定周期任务：周期时间 ×5 倍 事件任务：执行优先级 8 时 200ms 执行优先级 48 时 1s	设定传送时	不支持
变量访问时间 [%]		设定从控制器外部进行变量访问的时间与任务周期的比例。	主固定周期任务、固定周期任务：1%～50% 事件任务：无	3%	设定传送时	不支持

### ● 设定控制 I/O 的任务

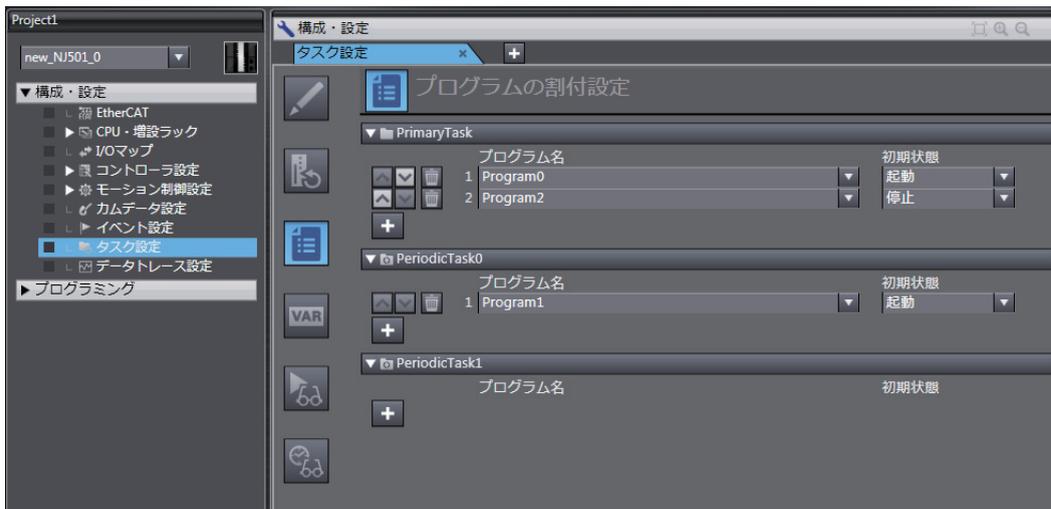
设定各输入输出的刷新执行时间。



设定项目	功能说明	设定值	默认	反映时序	运行模式中改写
任务名称	对在哪个任务中进行单元 / 从站的 I/O 刷新进行分配。	主固定周期任务	主固定周期任务	设定传送时	不支持

## ● 程序的分配设定

设定各任务的程序分配、程序执行顺序及各程序的运行开始时动作。



设定项目	功能说明	设定值	默认	反映时序	运行模式中 改写
程序执行顺序	设定将哪个程序分配到任务中，以及该程序在任务中的执行顺序。	按程序执行顺序，从上到下分配到任务中。	Program0	设定传送时	不支持
程序的初始状态	设定在运行开始时，是否启动该程序。	启动 停止	启动	设定传送时	不支持

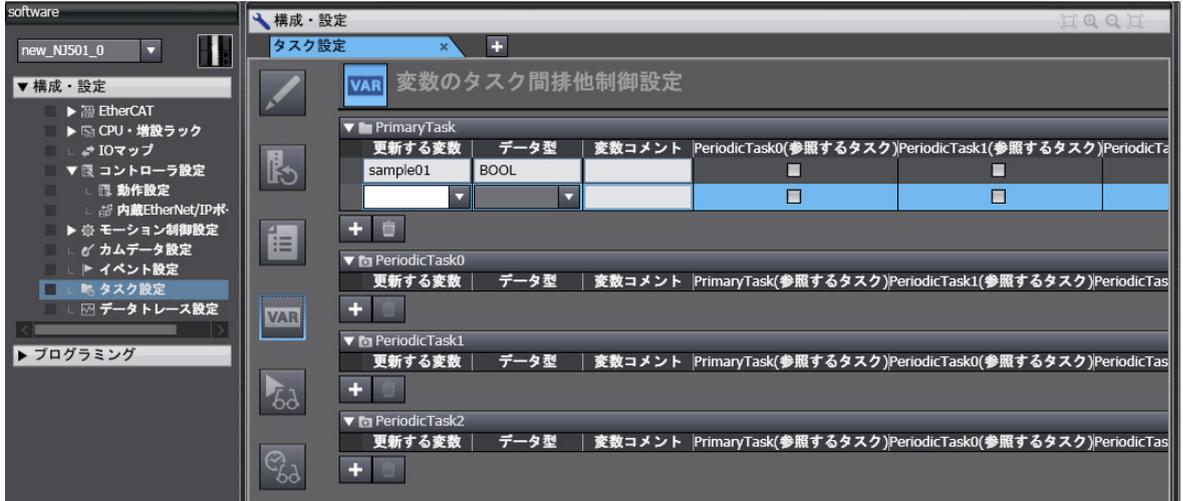


### 使用上の注意

- 记述有访问从站/单元所需的设备变量的程序，请分配到与[设定控制I/O的任务]中分配到对象从站/单元的任务相同的任务中。若分配到不同的任务中，链接时将发生异常。
- [设定控制I/O的任务]的默认设定为主固定周期任务。将程序分配到主固定周期任务以外的任务时，可以事先对[设定控制I/O的任务]进行重新设定，防止链接时发生错误。

● 变量的任务间排他控制设定

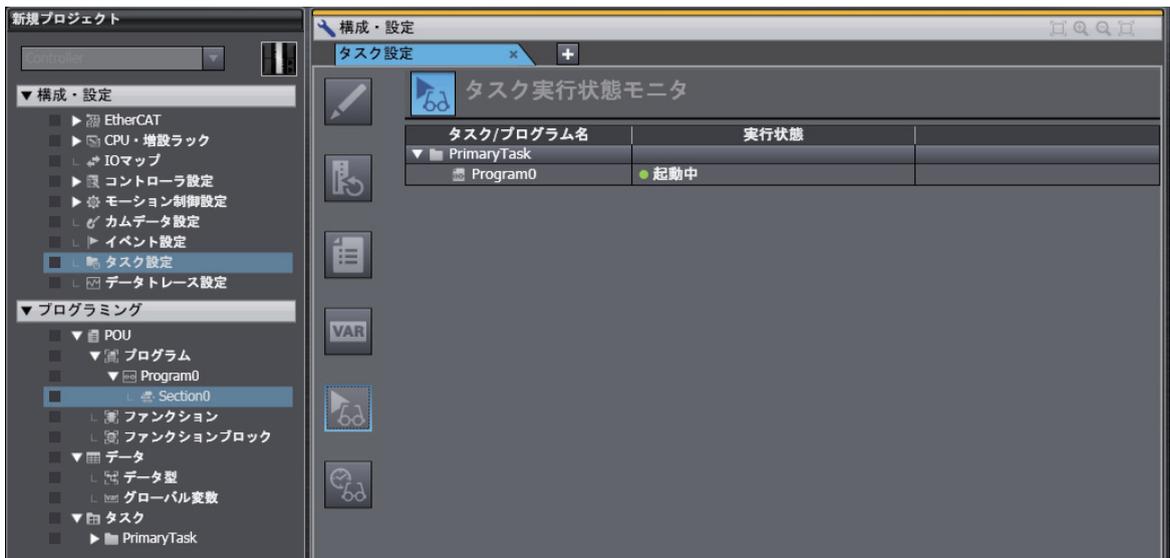
设定更新和参照全局变量的任务。



設定項目	設定中項目	功能说明	设定値	默认	反映时序	运行模式中 改写
各任务	更新的变量	从主固定周期任务或固定周期任务中选择作为更新任务的变量。		无	设定传送时	不支持
	数据类型	该变量的数据类型	无			
	变量注释	该变量的变量注释	无			
	参照任务	选择该变量的参照任务。				

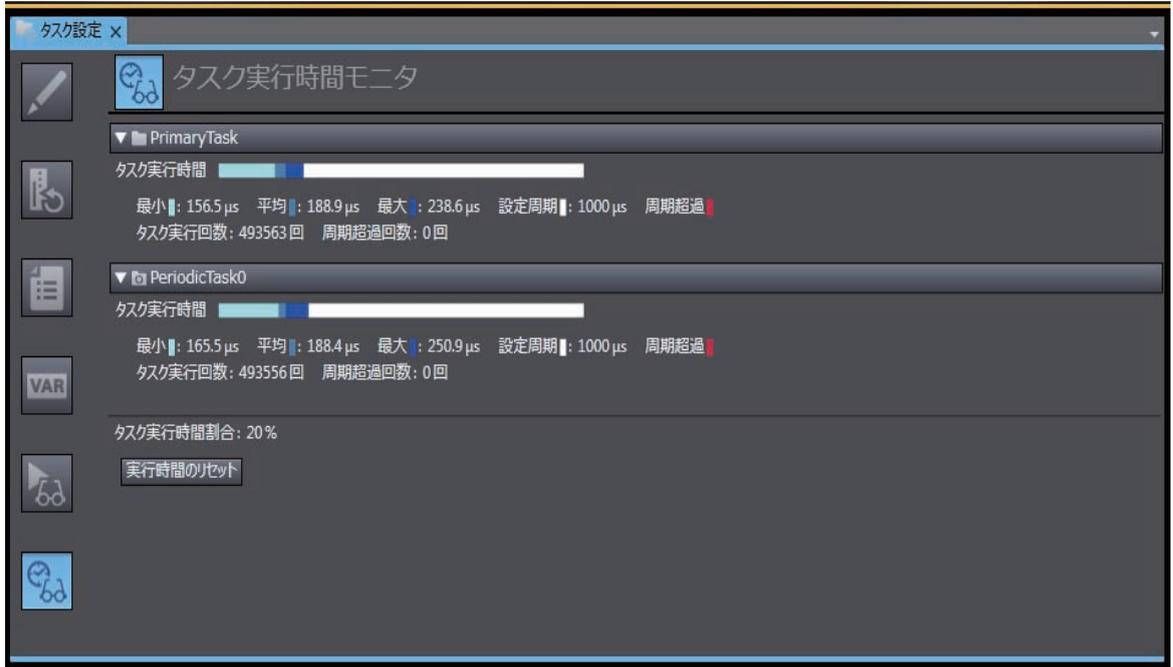
● 任务执行状态监视

显示各程序的执行状态。



## ● 任务执行时间监视

显示各任务的执行时间。



## 4-3 运动控制功能模块的初始设定

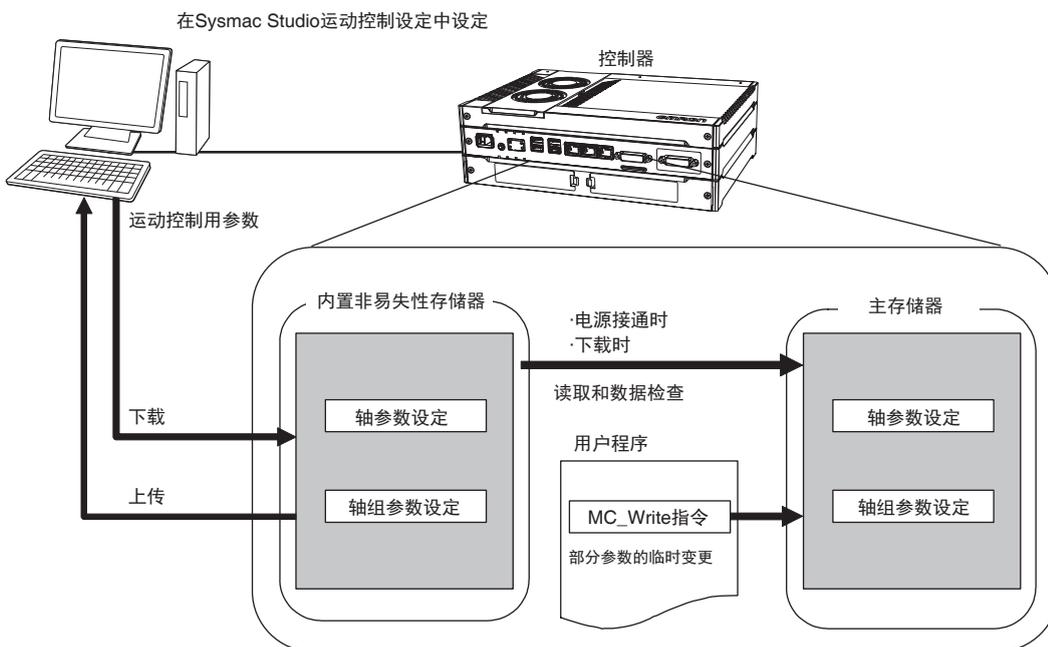
下面介绍运动控制功能模块的初始设定。

### 4-3-1 概要

运动控制功能模块的初始设定称为“运动控制参数”。

“运动控制参数”包括以下内容。

- 轴参数：单轴控制所需的设定
- 轴组参数：多轴协调控制所需的设定



## 4-3-2 设定方法

运动控制用参数的设定通过以下任意方法进行。

## 方法 1：通过 Sysmac Studio 的运动控制设定进行设定

从 Sysmac Studio 中右击以下 [ 配置和设定 ]-[ 运动控制设定 ]-[ 轴设定 ]，在 [ 轴设置表 ] 中进行设定。

轴名称		1 MC_Axis000(0)
▼ 轴基本设定		
轴使用		使用轴
轴种別		仮想サーボ軸
フィードバック情報		制御ループなし
入力デバイス		-
チャンネル		-
出力デバイス		-
チャンネル		-
▼ 単位変換設定		
表示単位		pulse
モータ1回転のパルス数		10000 pulse/rev
モータ1回転の移動量		10000 pulse/rev
▼ 動作設定		
最高速度		40000000 pulse/s
速度警告値		0 %
ジョグ最高速度		1000000 pulse/s
最大加速度		0 pulse/s <sup>2</sup>
加速度警告値		0 %
最大減速度		0 pulse/s <sup>2</sup>
減速度警告値		0 %
加減速オーバー		加減速度を急激にする(ブレンディングをバッファードに切り替える)
反転時動作		減速停止
正方向トルク警告値		0 %
負方向トルク警告値		0 %
インポジション幅		10 pulse
インポジションチェック時間		0 msec
フィードバック速度フィルタ時定数		0 msec
原点位置範囲		10 pulse
▼ 拡張動作設定		
原点停止入力停止方法		指令値を即停止

若将运动控制用参数下载到控制器，将保存到控制器内置的非易失性存储器中。电源接通时或下载时，下载的设定数据变为有效。

## 方法 2：用 MC\_Write 指令设定

部分运动控制用参数可通过 MC\_Write 指令临时替换。

详情请参考□□《NY 系列 指令基准手册 运动篇 (SBCE-CN5-380)》。

## 4-4 EtherCAT 主机功能模块的初始设定

下面介绍 EtherCAT 主机功能模块的初始设定。

EtherCAT 主机功能模块的初始设定包括以下内容。

- 设备名称
- 电缆总长
- 低效运行设定
- 所有从站加入等待时间
- PDO 通信连续超时检测次数
- 修订版本检查基准
- 序列号检查基准

在 Sysmac Studio 中选择 [配置和设定]-[EtherCAT]，然后选择 Master，将显示 EtherCAT 主机功能模块的初始设定画面。



详情请参考 [□](#) 《NY 系列 工业平板电脑 / 工业 BOX PC 用户手册 内置 EtherCAT 端口篇 (SBCD-CN5-368)》。

## 4-5 EtherNet/IP 功能模块的初始设定

下面介绍 EtherNet/IP 功能模块的初始设定。

EtherNet/IP 功能模块的初始设定包括以下内容。

- TCP/IP 设定
- LINK 设定
- FTP 设定
- SNMP 设定
- SNMP 捕捉设定

用 Sysmac Studio 在 [ 配置和设定 ]-[ 控制器设定 ]-[ 内置 EtherNet/IP 端口设定 ] 中设定。

详情请参考□□《NY 系列 工业平板电脑 / 工业 BOX PC 用户手册 内置 EtherNet/IP 端口篇 (SBCD-CN5-369)》。



# 5

## 设计任务

本章介绍任务的原理和种类。

<b>5-1</b>	<b>任务设计步骤的概要</b> .....	<b>5-2</b>
<b>5-2</b>	<b>任务的概要</b> .....	<b>5-4</b>
5-2-1	任务定义 .....	5-4
5-2-2	任务相关的指令 .....	5-6
5-2-3	任务相关的系统定义变量 .....	5-6
<b>5-3</b>	<b>NY 系列的任务规格和基本动作</b> .....	<b>5-8</b>
5-3-1	NY 系列的任务规格 .....	5-8
5-3-2	NY 系列的任务分配说明 .....	5-8
5-3-3	NY 系列的任务基本动作 .....	5-9
5-3-4	NY 系列事件任务的执行条件 .....	5-15
5-3-5	NY 系列事件任务的执行时序 .....	5-20
5-3-6	事件任务完成前，执行条件再次成立时的动作 .....	5-24
<b>5-4</b>	<b>标签数据链接服务和系统服务</b> .....	<b>5-25</b>
5-4-1	标签数据链接服务和系统服务的执行优先度和执行顺序 .....	5-26
5-4-2	标签数据链接服务的处理内容和执行时序 .....	5-28
5-4-3	系统服务的处理内容和执行时序 .....	5-30
<b>5-5</b>	<b>任务相关的分配和设定</b> .....	<b>5-31</b>
5-5-1	分配到 I/O 刷新的任务中 .....	5-31
5-5-2	将程序分配到任务中 .....	5-37
5-5-3	任务的设定项目 .....	5-38
<b>5-6</b>	<b>确保变量的同时性</b> .....	<b>5-40</b>
5-6-1	确保任务间变量值同时性的方法 .....	5-40
5-6-2	从控制器外部进行的变量访问 .....	5-46
<b>5-7</b>	<b>任务相关的异常</b> .....	<b>5-50</b>
<b>5-8</b>	<b>任务执行状态、任务执行时间的监视</b> .....	<b>5-52</b>
<b>5-9</b>	<b>任务设计方法及输入输出响应时间</b> .....	<b>5-56</b>
5-9-1	任务执行时间的确认 .....	5-56
5-9-2	任务设计示例 .....	5-57
5-9-3	系统输入输出响应时间 .....	5-58

## 5-1 任务设计步骤的概要

下面介绍任务设计步骤的概要。

在整体步骤中，与以下阴影标示的位置对应。



详情请参考□□「1-4 NY 系列的整体使用步骤」(P.1-10)。

任务设计	参考
<ul style="list-style-type: none"> <li>● <b>任务构成的设计</b> 根据控制对象装置所需的输入输出响应性能，设计任务构成。</li> </ul>	<ul style="list-style-type: none"> <li>□ 「5-3-3 NY 系列的任务基本动作」 (P.5-9)</li> <li>□ 「5-9 任务设计方法及输入输出响应时间」 (P.5-56)</li> </ul>
<ul style="list-style-type: none"> <li>● <b>将设备（从站/单元）的 I/O 刷新分配到任务中。NY 系列控制器中，可以只分配到主固定周期任务中。</b></li> </ul> <div data-bbox="331 533 683 757" style="text-align: center;"> <p>设备（从站/单元）</p> <p>主固定周期任务</p> <p>分配进行I/O刷新的任务</p> </div>	<ul style="list-style-type: none"> <li>□ 「5-5-1 分配到 I/O 刷新的任务中」 (P.5-31)</li> </ul>
<ul style="list-style-type: none"> <li>● <b>设计将哪个程序分配到主固定周期任务、固定周期任务（执行优先级 16 ~ 18）、事件任务（执行优先级 8、48）中的哪一个</b></li> </ul>	<ul style="list-style-type: none"> <li>□ 「5-5-2 将程序分配到任务中」 (P.5-37)</li> </ul>
<ul style="list-style-type: none"> <li>● <b>设计任务间变量的排他方法</b> 在不同的任务中使用相同的全局变量时，设计任务间变量的排他方法。</li> </ul>	<ul style="list-style-type: none"> <li>□ 「5-6-1 确保任务间变量值同时性的方法」 (P.5-40)</li> </ul>
<ul style="list-style-type: none"> <li>● <b>从控制器外部访问变量时的任务设定</b> 从控制器外部访问控制器内的变量时，需要将变量访问和特定任务的用户程序执行同步时设计。变量访问中包括 EtherNet/IP 标签数据连接。</li> </ul>	<ul style="list-style-type: none"> <li>□ 「5-6-2 从控制器外部进行的变量访问」 (P.5-46)</li> </ul>

## Sysmac Studio 上的任务设定

任务的设定	参考
<ul style="list-style-type: none"> <li>● <b>PLC 基本模块相关的初始设定</b> 任务设定：任务周期、输入输出设定、程序分配 任务间接口设定等</li> </ul>	<ul style="list-style-type: none"> <li>□ 「4-2 PLC 功能模块的初始设定」 (P.4-4)</li> </ul>

## Sysmac Studio 上的离线调试

上机动作确认	参考
<ul style="list-style-type: none"> <li>● <b>用模拟器（虚拟 PLC）对顺序控制和运动控制进行上机调试。</b></li> </ul>	<ul style="list-style-type: none"> <li>□ 「第 7 章 动作确认和运行」 (P.7-1)</li> </ul>

## 5-2 任务的概要

下面介绍任务的概要。

### 5-2-1 任务定义

任务是指在 I/O 刷新或用户程序执行等一系列处理过程中，指定执行条件和执行顺序的功能。

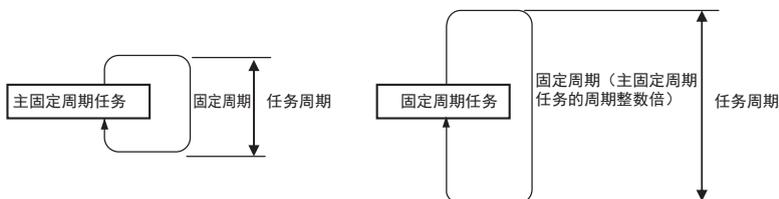
因执行条件及执行优先级不同，任务分为以下 3 种。

任务类型	任务数	任务执行优先级	定义	主要处理内容
主固定周期任务	1	4	按设定的任务周期执行 1 次。优先于任何任务执行。主固定周期任务的运动控制指令及 EtherCAT 的通信按主固定周期任务的周期执行。	I/O 刷新、用户程序执行、运动控制
固定周期任务	0 ~ 3	16、17、18	按设定的任务周期执行 1 次。执行优先级 16 的固定周期任务的运动控制指令及 EtherCAT 的通信按主固定周期任务的周期执行。	处理内容因任务执行优先级不同而异。 执行优先级 16、17、18：用户程序执行
事件任务	0 ~ 32	8、48	在指定的执行条件成立时，仅执行 1 次。	用户程序执行

## ● 主固定周期任务和固定周期任务

主固定周期任务及固定周期任务都按固定周期执行。

主固定周期任务及固定周期任务按固定周期执行时的间隔称为“任务周期”。



各任务中可分配 1 ~ 128 个程序。分配的程序按分配顺序执行。这称为“用户程序执行”。

与 EtherCAT 从站的数据交换称为 I/O 刷新。

I/O 刷新可按从站 / 单元只分配到主固定周期任务中。

## ● 事件任务

事件任务在指定的执行条件成立时，执行 1 次。事件任务的执行条件有以下两种。

执行条件	规格
通过指令执行	若执行事件任务执行指令 ActEventTask，将执行事件任务。
变量的条件一致后执行	若指定的变量与事先设定的条件一致，则执行事件任务。

各任务中可分配 1 ~ 128 个程序。分配的程序按分配顺序执行。



### 使用注意事项

- 事件任务中不会执行 I/O 刷新或运动控制。因此，带 I/O 控制的程序或包括运动控制指令的程序不可分配到事件任务中。
- 事件任务不是每个任务周期重复执行的任务。因此，若程序中包含跨越多个任务周期的指令，将无法分配到事件任务中。跨越多个任务周期执行的指令是指部分基本指令，如 SD 存储卡指令、通信指令，以及所有的运动控制指令和所有的模拟指令。事件任务中无法使用的基本指令详情请参考 □ 《NY 系列 指令基准手册 基本篇 (SBCA-CN5-437)》。

## 5-2-2 任务相关的指令

任务相关的指令如下。

指令	指令名称	概要	
GetMyTaskStatus	读取当前任务状态	读取当前任务的以下任务状态。 上次任务执行时间、任务执行时间最大值、任务执行时间最小值、任务执行次数、任务周期超过标志、任务周期超过次数	
GetMyTaskInterval	读取当前任务设置周期	读取当前任务的周期。	
Task_IsActive	任务执行中判定	返回指定的任务是否正在执行。	
Lock	任务间独占锁定	开始锁定区间。	其他任务中相同编号的锁定区间将无法同时执行。
Unlock	解除任务间独占锁定	结束锁定区间。	
ActEventTask	事件任务启动	启动指定的事件任务。	

## 5-2-3 任务相关的系统定义变量

如下所示，表示任务状态的系统定义变量按不同任务分别准备。

但是，出于实时性及与各功能模块异常状态的同步性考虑，在用户程序中无法使用。仅在通过 Sysmac Studio 的数据追踪功能对任务状态进行采样时使用。

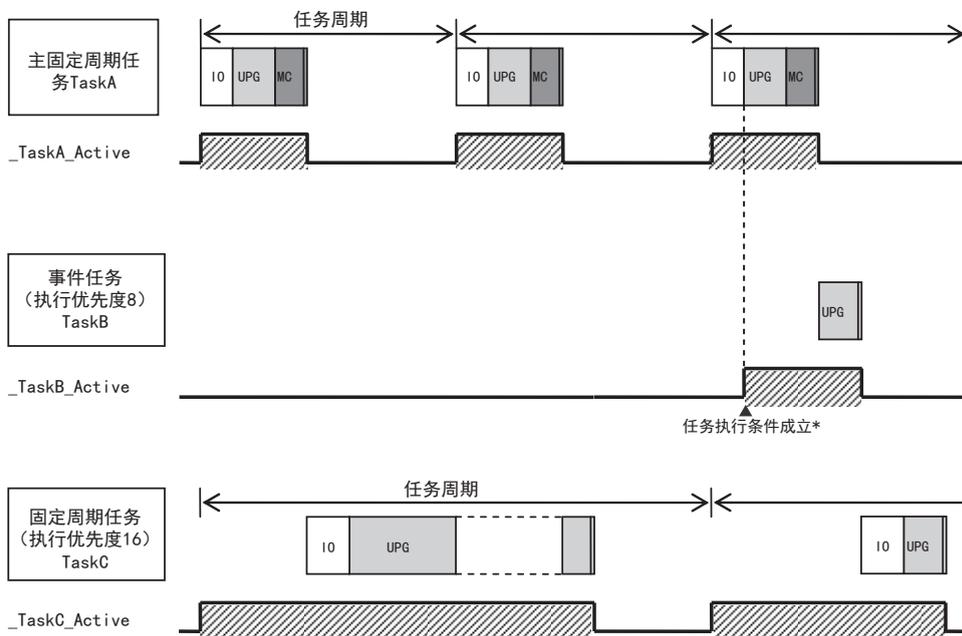
通过用户程序参照任务的状态时，请使用 GetMyTaskStatus 指令、Task\_IsActive 指令。

变量名称	名称	功能	数据类型	R/W
_ < 任务名称 > _Active	任务执行中标志	正在执行任务时为 TRUE。 任务执行完成后到任务周期结束为止，为 FALSE。	BOOL	R
_ < 任务名称 > _LastExecTime	上次任务执行时间	表示上次执行的任务执行时间。	TIME	R
_ < 任务名称 > _MaxExecTime	任务执行时间最大值	表示任务执行时间的最大值。	TIME	R
_ < 任务名称 > _MinExecTime	任务执行时间最小值	表示任务执行时间的最小值。	TIME	R
_ < 任务名称 > _ExecCount	任务执行次数	表示任务的执行次数。 若当前值超出数据类型的最大值，当前值将恢复为 0，然后继续计数。	UDINT	R
_ < 任务名称 > _Exceeded	任务周期超出标志	超出任务周期后，在任务执行完成时变为 TRUE。 若在任务周期内完成任务执行，则变为 FALSE。	BOOL	R
_ < 任务名称 > _ExceedCount	任务周期超出次数	保存超出任务周期的次数。 若当前值超出数据类型的最大值，当前值将恢复为 0，然后继续计数。	UDINT	R

(注) 例：任务名称“MainTask”的任务周期超过标志为“\_MainTask\_Exceeded”。

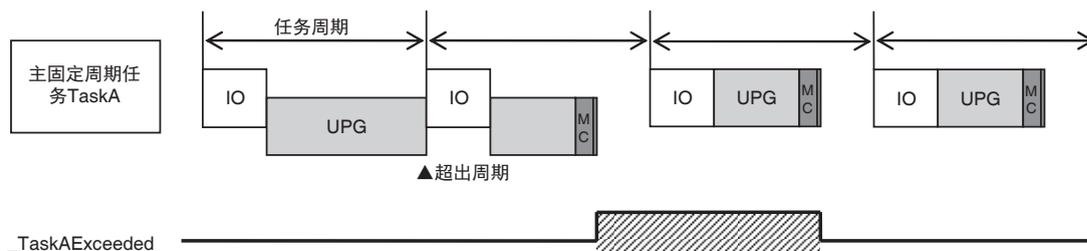
## 标志动作

### ● 任务执行中标志（\_<任务名称>\_Active）



\*用ActEventTask指令启动事件任务时，ActEventTask执行过程中，任务执行中标志将变为ON。

### ● 任务周期超出标志（\_<任务名称>\_Exceeded）



## 5-3 NY 系列的规格和基本动作

下面介绍配备多核处理器的 NY 系列控制器的任务规格和基本动作。

### 5-3-1 NY 系列的任务规格

任务的规格如下所示。

项目	规格
任务类型	<ul style="list-style-type: none"> <li>主固定周期任务</li> <li>固定周期任务（优先度 16、17、18）</li> <li>事件任务（优先度 8、48）</li> </ul>
任务个数	<ul style="list-style-type: none"> <li>主固定周期任务：1</li> <li>固定周期任务：0 ~ 3<sup>*1</sup></li> <li>事件任务：0 ~ 32<sup>*2</sup></li> </ul>
1 个任务的程序数	最大 128
主固定周期任务的周期	500μs ~ 8ms（可以 250μs 为单位设定）
固定周期任务的周期	1ms ~ 100ms（可以 250μs 为单位设定）
	固定周期任务的周期设定为主固定周期任务的整数倍。 不可选择各任务的周期最小公倍数超出 600ms 的组合。

\*1. 执行优先度 16、17、18 分别为 0 或 1 个。

\*2. 执行优先度 8、48 分别为 0 ~ 32 个，两者加起来最多 32 个。

### 5-3-2 NY 系列的任务分配说明

所有程序都需要分配到任意一个任务中。请根据程序的特性，按以下方法决定要分配的任务。

任务	适合分配的程序
主固定周期任务	<ul style="list-style-type: none"> <li>需要严格遵守各周期的 I/O 刷新、用户程序执行、运动控制、系统共通处理的执行周期。</li> <li>包括执行优先度最高、需要高速响应的控制。</li> <li>包括执行优先度最高的运动控制指令。</li> </ul>
固定周期任务（优先度 16）	<ul style="list-style-type: none"> <li>包括将 I/O 刷新分配到主固定周期任务中的部分从站的控制。</li> <li>执行优先度相对较低，但需要按周期执行用户程序、系统共通处理。</li> <li>包括执行优先度较低的运动控制指令。</li> <li>在针对主固定周期任务中控制的从站 / 单元的处理中，将执行优先度相对较低的处理分配到固定周期任务（执行优先度 16）进行控制时使用。</li> </ul>
固定周期任务（优先度 17,18）	<ul style="list-style-type: none"> <li>执行优先度相对较低，但需要按周期执行用户程序、系统共通处理。</li> <li>包括不需要高速响应的运算处理、通信处理。</li> </ul>
事件任务	<ul style="list-style-type: none"> <li>仅在特定条件成立时执行。</li> </ul>

### 5-3-3 NY 系列的任務基本動作

NY 系列控制器中搭載有多核處理器，可並列進行任務的執行和標籤數據鏈接服務或系統服務的執行。各任務按什麼樣的順序執行，取決於各任務中設定的執行優先度。

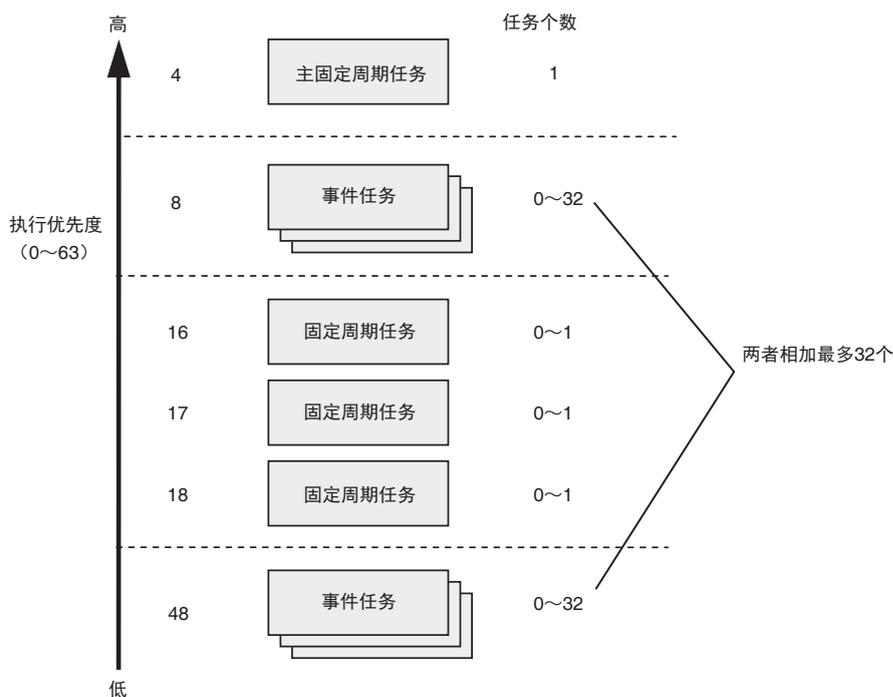
#### 任務執行優先度

任務根據其類型決定執行優先度。NY 系列控制器將優先執行優先度較高的任務。

正在執行某任務 Ta 時，若執行優先度比它高的 Tb 執行條件成立，將優先把 Tb 的處理分配到處理器。

不同任務類型的執行優先度如下所示。執行優先度的值越小，優先度越高。

任務	執行優先度	執行優先度的重複
主固定周期任務	4	—
固定周期任務	16、17、18	不可將多個任務設定為相同的執行優先度。
事件任務	8、48	可將多個任務設定為相同的執行優先度。 此時的執行順序請參考 □ 「5-3-5 NY 系列事件任務的執行時序」 (P.5-20)。



## 主固定周期任务和固定周期任务的周期

主固定周期任务和固定周期任务是指按周期反复执行的任务。

固定周期任务的周期必须设定为主固定周期任务的周期（称为“主周期”）的整数倍。因此，几次后两者的执行时序会重叠。

例如，设定为主周期 1ms、固定周期任务（执行优先级 16）的周期 4ms 时，则主固定周期任务每执行 4 次，主固定周期任务和固定周期任务的执行时序重叠。



### 参考

事件任务是指不按周期执行，而是在指定的执行条件成立时，仅执行 1 次的任务。因此，事件任务的执行时序取决于事件任务的执行条件成立的时间和事件任务的执行优先级。

## 任务的执行顺序示例

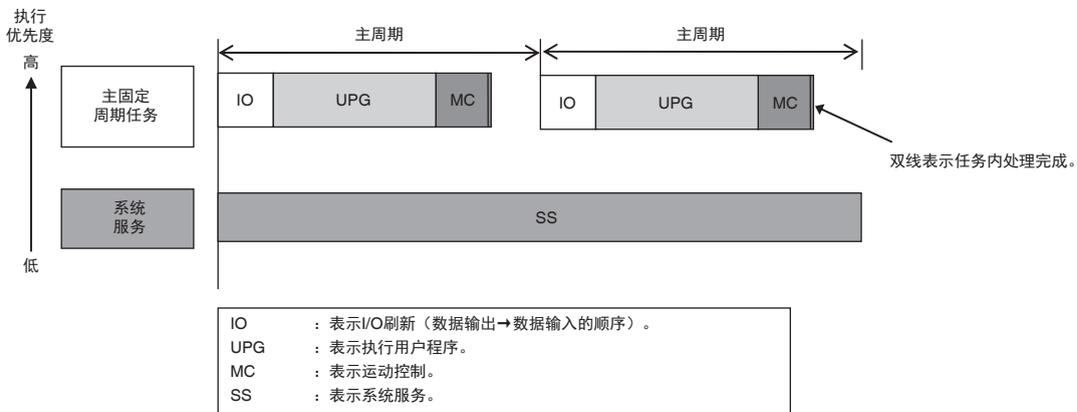
下面举例说明主固定周期任务和固定周期任务的执行顺序。

事件任务的执行顺序请参考 □□「5-3-5 NY 系列事件任务的执行时序」（P.5-20）。

### ● 仅主固定周期任务时

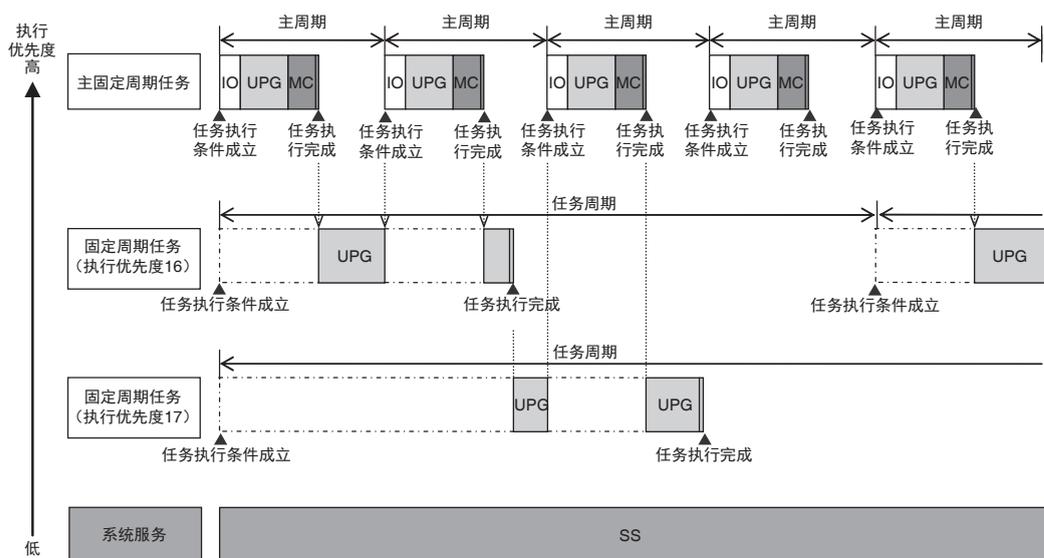
按主固定周期执行主固定周期任务。

图中的系统服务是指通信处理等控制器在任务之外执行的处理。系统服务的详情请参考 □□「系统服务的处理内容」（P.5-30）。



### ● 主固定周期任务、固定周期任务（执行优先级 16）、固定周期任务（执行优先级 17）同时存在时

- 主固定周期任务的执行优先级最高，因此将立即按主周期执行。
- 固定周期任务（执行优先级 16）待主固定周期任务完成后执行。
- 固定周期任务（执行优先级 17）的执行优先级更低，因此在未执行上述两个任务的时间内执行。
- 本示例中，固定周期任务（执行优先级 16）的任务周期为主周期的 4 倍。因此，每经过 4 个主周期，主固定周期任务和固定周期任务（执行优先级 16）的执行开始时间同步。
- 系统服务不受任务的影响，始终执行。



#### 使用注意事项

可能在多个任务间读写同一个变量时，请务必使用“变量的任务间排他控制功能”。否则，在执行某个任务的过程中，变量的值可能被其他任务变更。

详情请参考 □「5-6-1 确保任务间变量值同时性的方法」(P.5-40)。

## 任务和动作模式

NY 系列的动作模式和任务有以下关系。

任务	规格
主固定周期任务	• 无论是运行模式还是程序模式，都执行。 • 但是，执行用户程序的仅限运行模式。
固定周期任务	
事件任务	仅在运行模式下执行。



### 使用注意事项

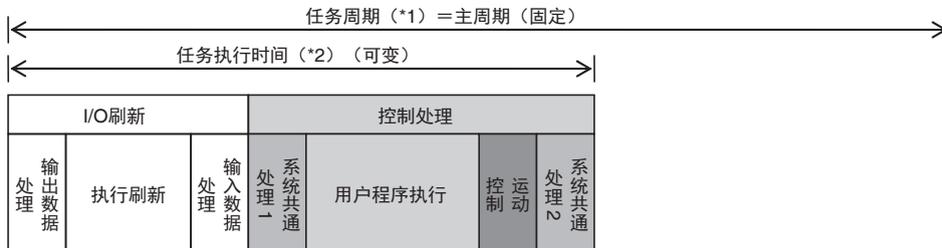
- 动作模式从程序模式切换为运行模式时，若事件任务的执行条件已成立，将不执行事件任务。当动作模式为运行模式，且事件任务的执行条件从不成立变为成立时，执行事件任务。
- 即使动作模式为运行模式，如果发生全部停止故障等级的异常，也不会执行事件任务。

## 各任务类型的处理内容

### ● 主固定周期任务

任务的执行优先级最高，可高速、高精度执行的任务。

按指定的周期执行系统共通处理、I/O 刷新、用户程序、运动控制。



\*1 : 任务周期: 任务按固定周期执行，为设定的固定时间。

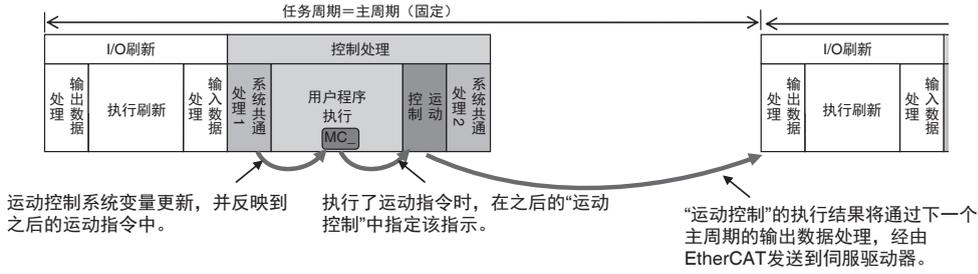
\*2 : 任务执行时间: 从任务的执行条件成立到执行完成为止，实际花费的时间。

处理		处理内容
I/O 刷新	输出数据处理	<ul style="list-style-type: none"> <li>生成向执行 I/O 刷新的 I/O 输出的 OUT 刷新数据。</li> <li>设定为强制值刷新时，将在 OUT 刷新数据中反映强制值（输出）。</li> </ul>
	执行刷新	<ul style="list-style-type: none"> <li>与 I/O 执行数据交换。</li> </ul>
	输入数据处理	<ul style="list-style-type: none"> <li>事件任务的条件式一致判定。</li> <li>导入执行了 I/O 刷新的 I/O（输入）单元之 IN 刷新数据。</li> <li>设定为强制值刷新时，将在导入的 IN 刷新数据中反映强制值（输入）。</li> </ul>
系统共通处理 1		<ul style="list-style-type: none"> <li>变量的任务间排他控制处理（参照任务时）</li> <li>运动输入处理<sup>*1</sup></li> <li>数据追踪处理（采样、触发判定）</li> </ul>
用户程序执行		<ul style="list-style-type: none"> <li>按分配顺序执行任务中分配的程序。</li> </ul>
运动控制 <sup>*2</sup>		<ul style="list-style-type: none"> <li>通过主固定周期任务和固定周期任务（执行优先级 16）中分配的用户程序中的运动控制指令，发出运动控制指示并执行。</li> <li>对下一个主固定周期任务的“I/O 刷新”进行运动输出处理</li> </ul>
系统共通处理 2		<ul style="list-style-type: none"> <li>变量的任务间排他控制处理（更新任务时）</li> <li>为确保与任务执行的同時性，从控制器外部进行的变量访问处理（[任务设定]中设定的“变量访问时间”分）</li> <li>EtherNet/IP 标签数据链接处理时，若对标签（网络公开属性的变量）设定更新任务，将作为上述变量访问处理执行。</li> </ul>

\*1. 运动控制系统变量的轴当前值（位置、速度、转矩）及运动控制系统变量的伺服驱动器状态将更新。

\*2. 若为主固定周期任务的用户程序中的运动控制指令，如下图所示，将在随后的“运动控制”中执行该指令发出的指示，并在下一主固定周期任务的“I/O 刷新”时输出到伺服驱动器。

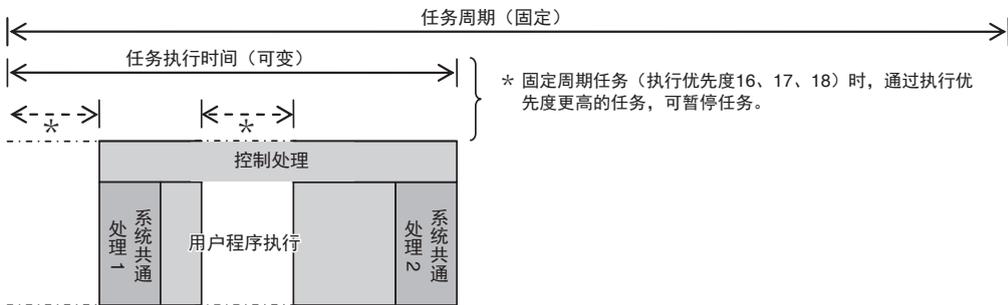
(注) 上表中各单元格的处理将按记载顺序执行。



此外，若为固定周期任务（执行优先级 16）的用户程序中的运动控制指令，将在该固定周期（执行优先级 16）的下一个主固定周期任务的“运动控制（MC）”时，执行该指令发出的指示。详情请参考□□「5-9-3 系统输入输出响应时间」（P.5-58）。

● 固定周期任务（执行优先级 16、17、18）

按指定为主周期整数倍的任务周期，执行用户程序的任务。整个任务中可分配 0 ~ 3 个。

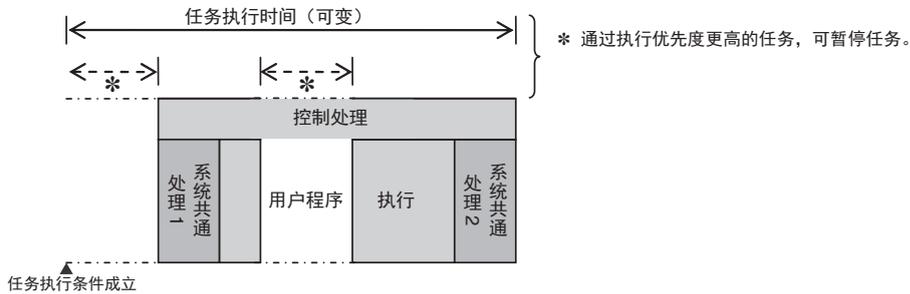


处理	处理内容
系统共通处理 1	<ul style="list-style-type: none"> <li>变量的任务间排他控制处理（参照任务时）</li> <li>数据追踪处理（采样、触发判定）</li> </ul>
用户程序执行	<ul style="list-style-type: none"> <li>按分配顺序执行任务中分配的程序。</li> </ul>
系统共通处理 2	<ul style="list-style-type: none"> <li>变量的任务间排他控制处理（更新任务时）</li> <li>为确保与任务执行的同时性，从控制器外部进行的变量访问处理（[任务设定]中设定的“变量访问时间”分）</li> <li>EtherNet/IP 标签数据链接处理时，若对标签（网络公开属性的变量）设定更新任务，将作为上述变量访问处理执行。</li> </ul>

(注) 上表中各单元格的处理将按记载顺序执行。

## ● 事件任务

在指定的执行条件成立时，仅执行 1 次的任务。整个任务中可分配 0 ~ 32 个。  
事件任务的处理内容如下所示。



处理	处理内容
系统共通处理 1	• 变量的任务间排他控制处理 (参照任务时) <sup>*1</sup>
用户程序执行	• 按分配顺序执行任务中分配的程序。
系统共通处理 2	• 变量的任务间排他控制处理 (更新任务时) <sup>*1</sup>

\*1. 变量的任务间排他控制处理详情请参考□□「5-6-1 确保任务间变量值同时性的方法」(P.5-40)。

### 5-3-4 NY 系列事件任务的执行条件

事件任务是指在指定的执行条件成立时，仅执行 1 次的任务。事件任务的执行条件有以下两种。

执行条件	执行事件任务的时序	使用时
通过 ActEventTask 指令执行	执行 ActEventTask 指令时	<ul style="list-style-type: none"> <li>• 希望在用户程序中写明执行哪个事件任务时</li> <li>• 在变量的条件式一致判定前，事件任务的执行条件可能发生变化时</li> </ul>
变量的条件式一致后执行	指定的变量值与指定的条件式一致时 <sup>*1</sup>	希望省略执行事件任务所需的用户程序，将用户程序简化时

\*1. 关于判定指定的变量值与指定的条件式是否一致的时间，请参考□□「执行条件为变量的条件式一致时的执行时序」(P.5-21)。

## 通过 ActEventTask 指令执行事件任务

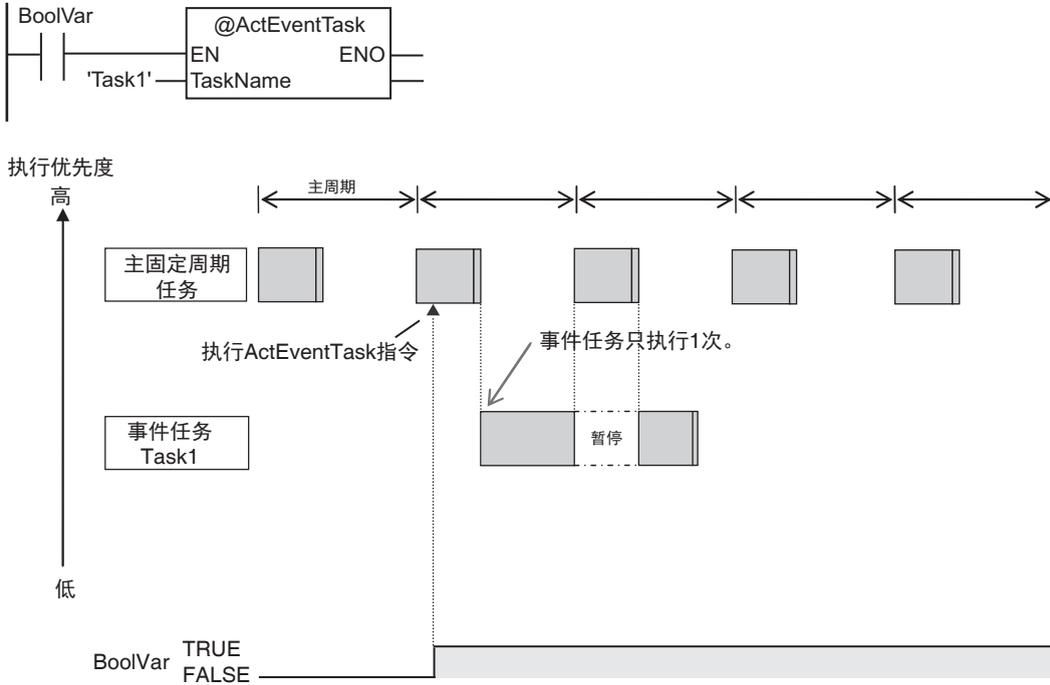
若在程序中执行 ActEventTask 指令，将执行 1 次指定的事件任务。ActEventTask 指令的详细规格请参考□□《NY 系列 指令基准手册 基本篇 (SBCA-CN5-437)》。

若使用 ActEventTask 指令，看到用户程序后，能立即知道是启动哪个事件任务。此外，在变量的条件式一致判定前，即使事件任务的执行条件可能发生变化，仍为有效。

● 使用 ActEventTask 指令的用户程序示例

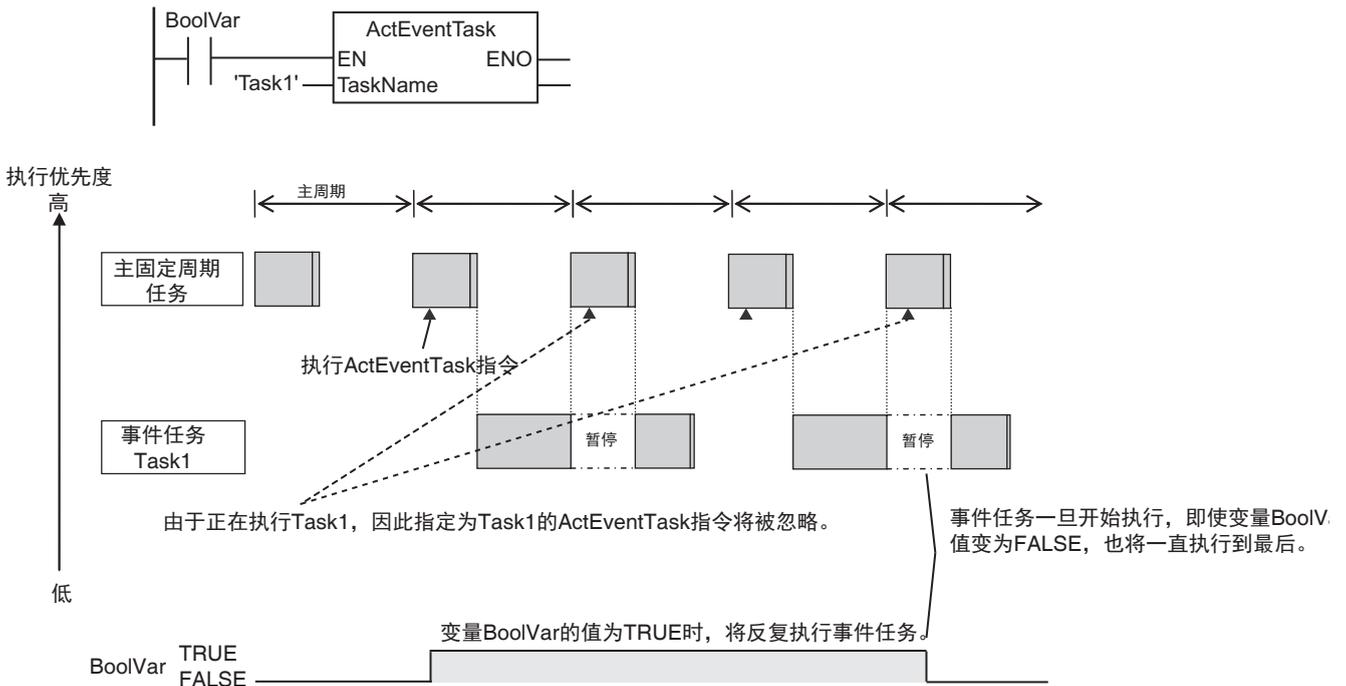
例 1) 变量值发生变化时, 执行 1 次事件任务

如下图所示, 若在 ActEventTask 指令中加入输入上升沿微分的动作选项, 当 BOOL 型变量 BoolVar 的值从 FALSE 变为 TRUE 时, 将执行 1 次事件任务 Task1。



例 2) 变量为指定值时, 反复执行事件任务

如下图所示, 若 ActEventTask 指令中不加入输入上升沿微分的动作选项, 当 BOOL 型变量 BoolVar 的值为 TRUE 时, 将反复执行事件任务 Task1。但是, 在执行 Task1 时, 即使执行指定了 Task1 的 ActEventTask 指令, 也将忽略。此外, 事件任务一旦开始执行, 即使在执行过程中 BoolVar 的值变为 FALSE, 也将一直执行到最后。



## 变量的条件式一致后执行事件任务

变量的条件式一致后执行的方法是指 Sysmac Studio 中指定的变量值与指定的条件式一致时，执行 1 次事件任务。该变量的值与条件式一致期间，不会持续执行事件任务，而是在变量的值从与条件式不一致变为一致时，执行 1 次事件任务。

变量的条件式一致后执行时，不需要执行事件任务所需的用户程序。

### ● 可指定条件式的变量种类

可指定条件式的变量种类如下所示。

变量的种类		可否指定
系统定义变量		支持 <sup>*1</sup>
准用户定义变量		支持
用户定义变量	全局变量	支持
	程序中使用的变量	支持
	功能块中使用的变量	支持 <sup>*2</sup>
	函数中使用的变量	不支持

\*1. 以下变量除外。

EN, ENO, P\_Off, P\_CY, P\_First\_RunMode, P\_First\_Run, P\_PRGER

\*2. 输入输出变量除外。

### ● 可指定条件式的变量的数据类型

可指定条件式的变量的数据类型如下所示。

数据类型的分类	数据类型		可否指定
基本数据类型	布尔型、位列型、整数型、实数型		支持
	持续时间型、日期型、时刻型、日期时刻型、字符串型		不支持
数据类型的指定	排列指定	排列整体	不支持
		要素	支持 <sup>*1</sup>
派生数据类型	结构体型	结构体整体	不支持
		成员	支持 <sup>*2</sup>
	联合体型	联合体整体	不支持
		成员	可 <sup>*2</sup>
列举型		支持	

\*1. 排列要素的数据类型仅限布尔型、位列型、整数型、实数型中的一个。

\*2. 成员的数据类型仅限布尔型、位列型、整数型、实数型中的一个。

● 可指定的条件式

因指定条件式的变量数据类型不同，可指定的条件式如下。  
指定条件式的变量数据类型为位列型、整数型、实数型时，需要设定变量的值和要比较的比较常数。

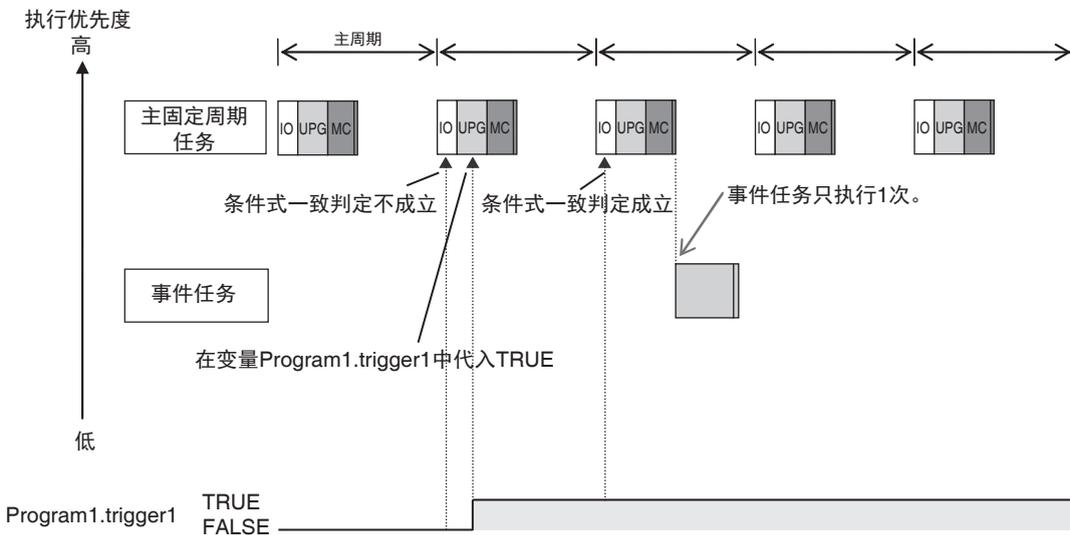
数据类型	可指定的条件式
布尔型、布尔型排列要素、布尔型结构体成员、布尔型联合体成员	上升沿 FALSE → TRUE 下降沿 TRUE → FALSE
位列型、整数型、实数型、或以上数据类型的排列要素、结构体成员、联合体成员	变量 = { 比较常数 } 变量 ≠ { 比较常数 } 变量 > { 比较常数 } 变量 ≥ { 比较常数 } 变量 < { 比较常数 } 变量 ≤ { 比较常数 }

● 比较常数的值的有效范围

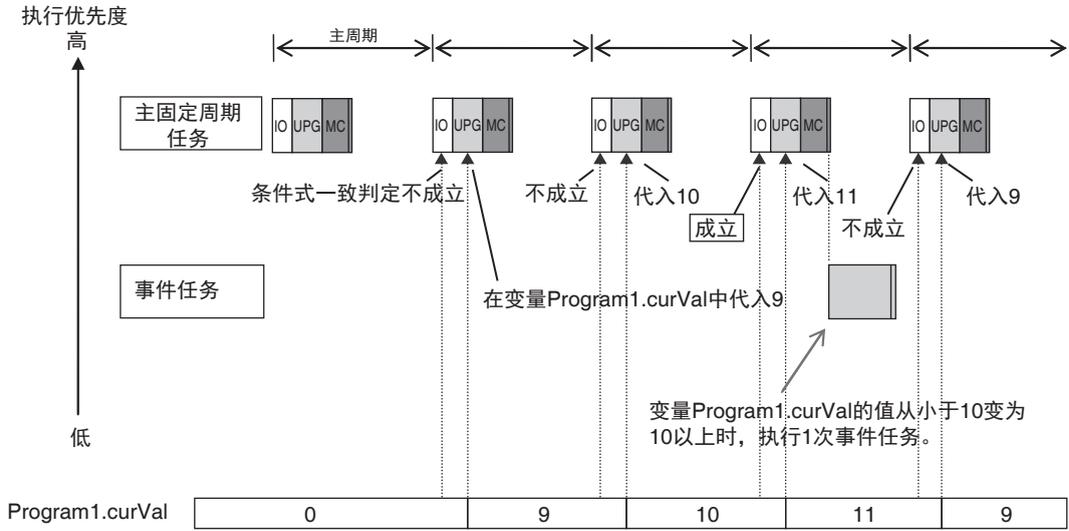
指定条件式的变量数据类型为位列型、整数型、实数型时，需要设定变量的值和要比较的比较常数。比较常数的值的有效范围与指定条件式的变量之数据类型值的有效范围相同。各数据类型的值的有效范围请参考□□「基本数据类型和派生数据类型」(P.6-30)。例如，指定条件式的变量之数据类型为 BYTE 时，比较常数的值的有效范围为 BYTE#16#00 ~ BYTE#16#FF。

● 变量的条件式一致后执行事件任务的示例

例 1) 事件任务的执行条件设定为 BOOL 型变量 Program1.trigger1 的上升沿时  
Program1.trigger1 的值从 FALSE 变为 TRUE 时，执行 1 次事件任务。



例 2) 事件任务的执行条件设定为 INT 型变量 Program1.curVal  $\geq 10$  时  
 Program1.curVal 的值从小于 10 变为 10 以上时，执行 1 次事件任务。



### 使用注意事项

主固定周期任务中，指定变量的值发生变化时，将在下一主固定周期任务中判定是否与条件式一致。因此，执行事件任务的时间为：变量的值满足条件式的周期的下一个主固定周期任务中，进行条件式一致判定后。

### 5-3-5 NY 系列事件任务的执行时序

事件任务的执行优先级为 8 或 48。若正在执行其他任务时事件任务的执行条件成立，将先执行优先级高的任务，优先级低的任务将中断。这与主固定周期任务或固定周期任务相同。

可能在多个任务间读写同一个变量时，请务必使用以下功能，控制主固定周期任务、固定周期任务或执行优先级不同的事件任务的执行。

目的	使用的功能
在事件任务和其他任务中访问相同的全局变量时	变量的任务间排他控制功能 <ul style="list-style-type: none"> <li>• 变量的任务间排他控制设定功能</li> <li>• Lock（任务间独占锁定）指令</li> <li>• Unlock（解除任务间独占锁定）指令</li> </ul>
确认其他任务的执行状态时	Task_IsActive（任务执行中判定）指令

详情请参考□「5-6-1 确保任务间变量值同时性的方法」（P.5-40）。

事件任务的执行时序因事件任务的执行条件不同而异。

此外，事件任务可将多个任务分配为相同的执行优先级。执行优先级相同的多个事件任务的执行条件成立时，需要注意。

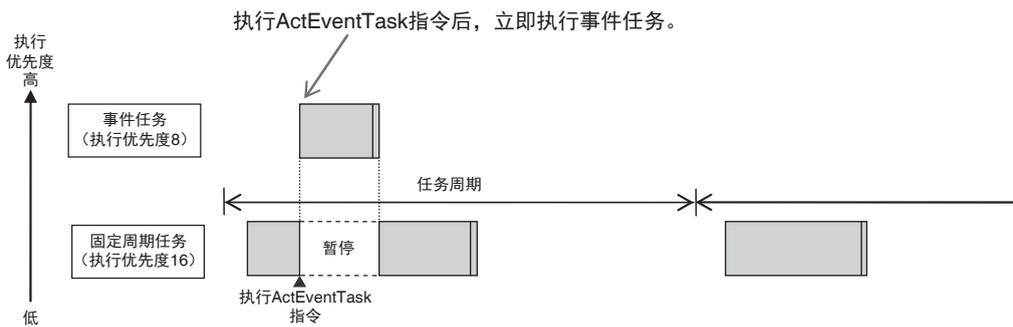
### 执行时序随事件任务执行条件的不同

因事件任务的执行条件为利用 ActEventTask 指令还是利用变量的条件式一致不同，执行条件的成立时间不同。

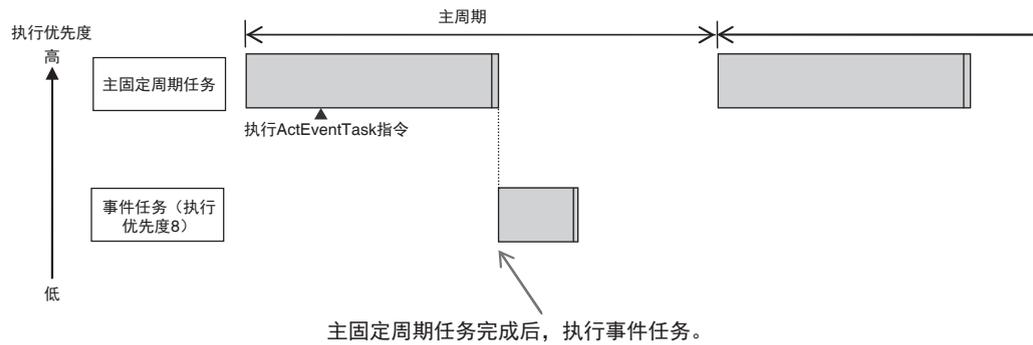
#### ● 执行条件为利用 ActEventTask 指令时的执行时序

事件任务的执行条件为利用 ActEventTask 指令时，执行 ActEventTask 指令后，事件任务的执行条件立即成立。执行条件成立的事件任务将按照任务的执行优先级开始执行。

例 1) 要启动相比执行指令的任务，执行优先级更高的事件任务时



例 2) 要启动相比执行指令的任务，执行优先级更低的事件任务时

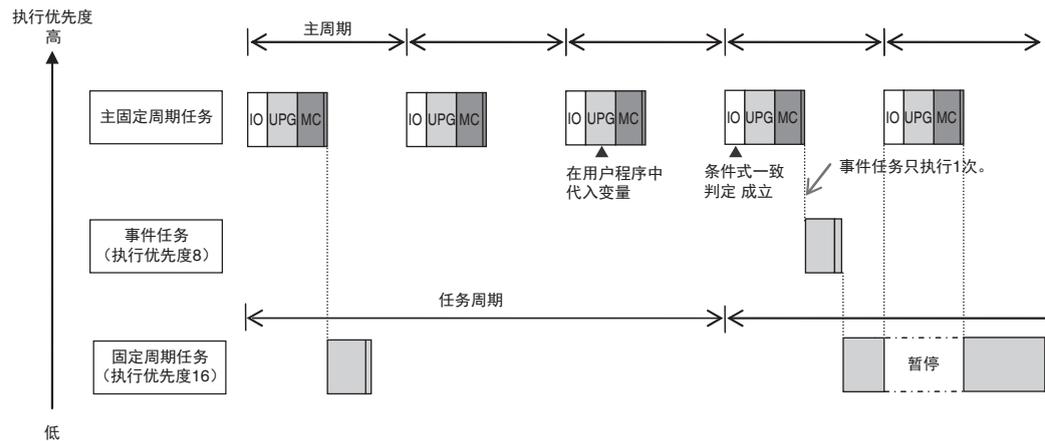


### ● 执行条件为变量的条件式一致时的执行时序

条件式一致判定在主固定周期任务中执行。在条件式的判定处理中，判定为条件一致时，事件任务的执行条件成立。执行条件成立的事件任务将按照任务的执行优先级开始执行。

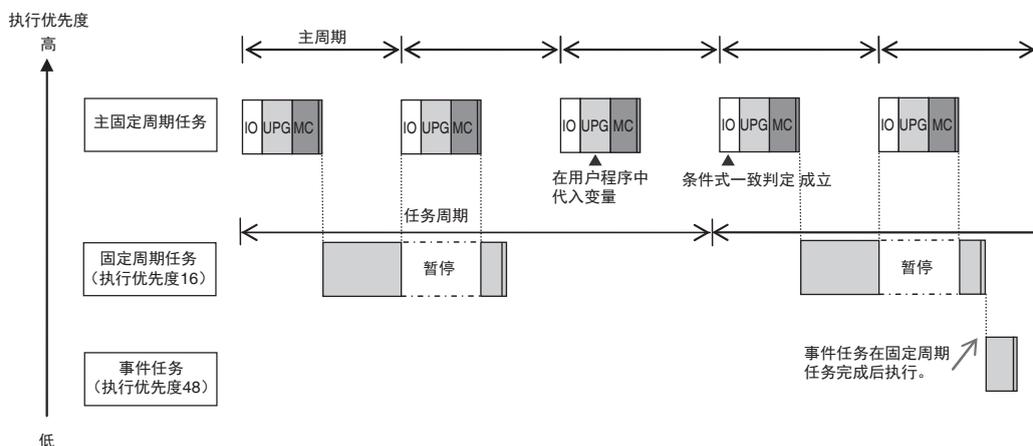
例 1) 存在固定周期任务（执行优先级 16）和事件任务（执行优先级 8）时

事件任务（执行优先级 8）的执行优先级高于固定周期任务（执行优先级 16），因此将在事件任务之后执行固定周期任务（执行优先级 16）。



例 2) 存在固定周期任务（执行优先级 16）和事件任务（执行优先级 48）时

事件任务的执行优先级低于固定周期任务（执行优先级 16），因此将在固定周期任务（执行优先级 16）完成后执行事件任务。





**使用注意事项**

- NY 系列控制器中，由主固定周期任务的 I/O 刷新发生条件式一致时以及由主固定周期任务中分配的用户程序执行发生时，事件任务执行条件的成立时间相同，具体如下。

条件式一致发生原因	事件任务的执行条件成立的时间
主固定周期任务的 I/O 刷新	下一个主固定周期任务的判定处理
主固定周期任务的用户程序执行	下一个主固定周期任务的判定处理

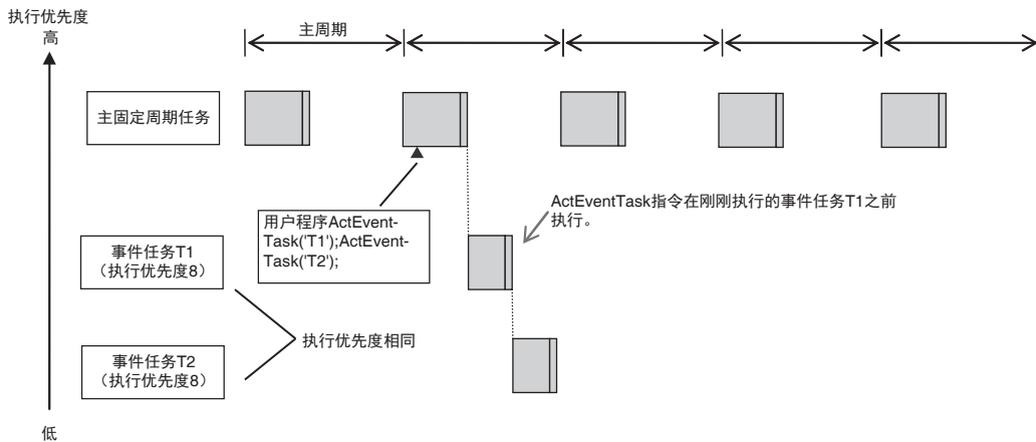
- 执行事件任务的时间：前一次判定时为条件式不一致，下一次判定时为条件式一致时。因此，即使发生从不一致到一致的变化，若在下次判定前恢复为不一致，将不会执行事件任务。

**执行优先级相同的事件任务的执行时序**

事件任务可将多个任务分配为相同的执行优先级。利用 ActEventTask 指令，执行优先级相同的多个事件任务的执行条件成立时，将按执行指令的顺序执行事件任务。

例 1) 执行了 2 个 ActEventTask 指令时

如下图所示，先执行的 ActEventTask 指令所指定的事件任务 T1 将在事件任务 T2 之前执行。

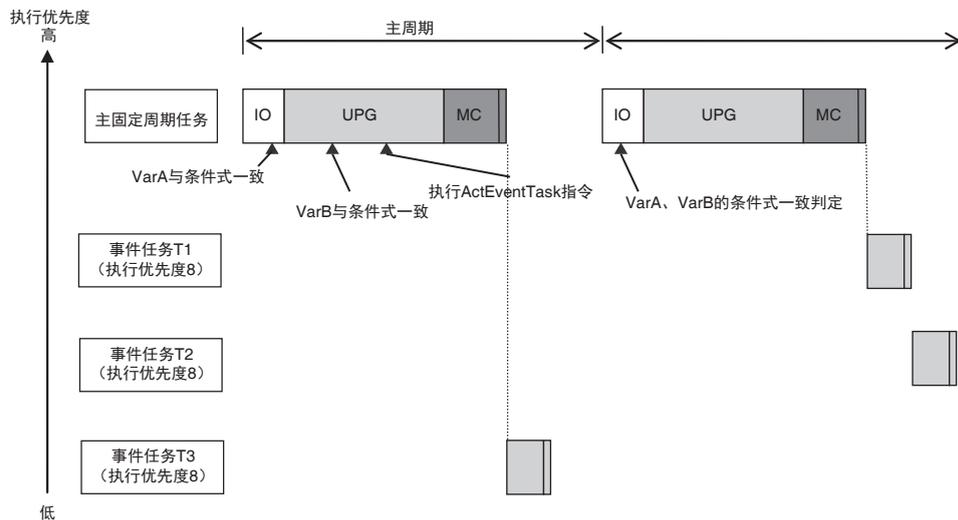


例 2) 变量的条件式一致和 ActEventTask 指令同时存在时  
假设事件任务 T1、T2、T3 的执行条件设定如下。

- T1: 变量 VarA 的条件式一致
- T2: 变量 VarB 的条件式一致
- T3: ActEventTask 指令

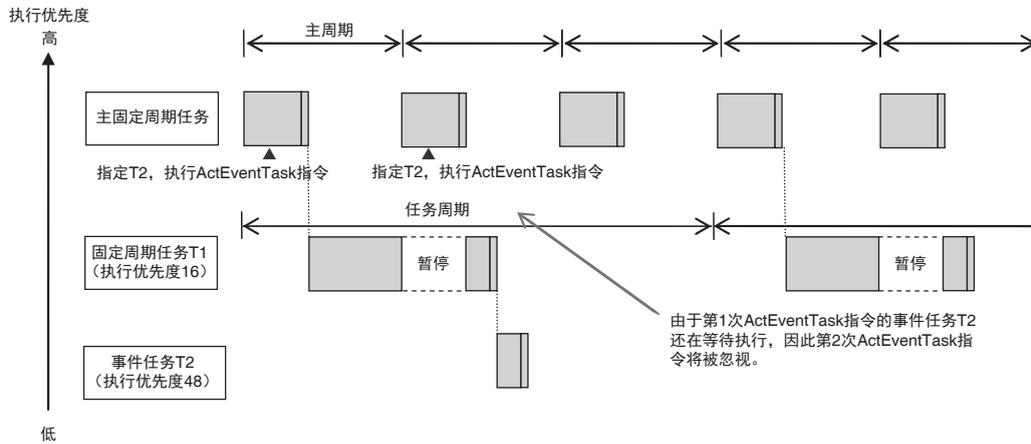
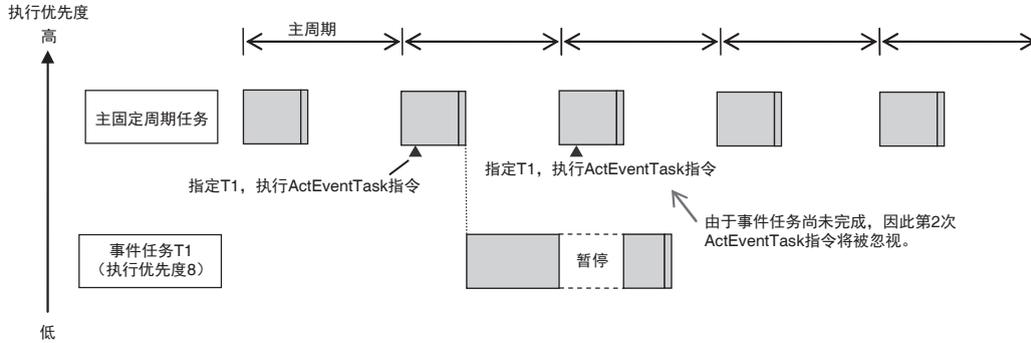
如下图所示，在同一主周期中，I/O 刷新后 VarA 的条件式一致、在用户程序中执行 ActEventTask 指令、用户程序中发生 VarB 的条件式一致时，按以下顺序动作。

- (1) I/O 刷新后 VarA 与条件式一致。
- (2) 用户程序中 VarB 与条件式一致。
- (3) 此时，由于尚未执行条件式的一致判定，因此 T1 和 T2 均不执行。
- (4) 在程序中执行 ActEventTask 指令，执行 T3。
- (5) 在下一主固定周期任务中执行刷新后，VarA 和 VarB 判定为与条件式一致，执行 T1 和 T2。  
若在同一周期内判定有多个条件式一致，则事件任务的执行顺序不确定。下图为先执行 T1 时的情况。



### 5-3-6 事件任务完成前，执行条件再次成立时的动作

事件任务完成前，该事件任务的执行条件再次成立时，第 2 次执行条件的成立将被忽略。事件任务完成前是指事件任务执行中及事件任务执行等待中两种情况。此外，事件任务一旦开始执行，即使执行条件变为不成立，也将一直执行到最后。



## 5-4 标签数据链接服务和系统服务

NY 系类控制器除了主固定周期任务、固定周期任务、事件任务之外，还会执行其他处理。这些处理包括标签数据链接服务和系统服务两种。两者的处理内容、执行优先级、执行时序如下所示。

项目	标签数据链接服务	系统服务
执行优先级	固定周期任务（执行优先级 16）和固定周期任务（执行优先级 17）之间。	执行优先级比所有任务低。
处理内容	在 EtherNet/IP 网络上，使用其他控制器、设备和标签进行数据交换。使用 NY 系列控制器的内置 EtherNet/IP 端口。 <sup>*1</sup>	通信处理等执行优先级较低的处理的总称。
执行时序	执行间隔或 1 次处理的执行时间因 NY 系列的型号或标签数据链接设定不同而异。 <sup>*2</sup>	NY 系列控制器不受任务的影响，始终执行。

\*1. 标签数据链接的处理内容详情请参考 [□□《NY 系列 工业平板电脑 / 工业 BOX PC 用户手册 内置 EtherNet/IP 端口篇（SBCD-CN5-369）》](#)。

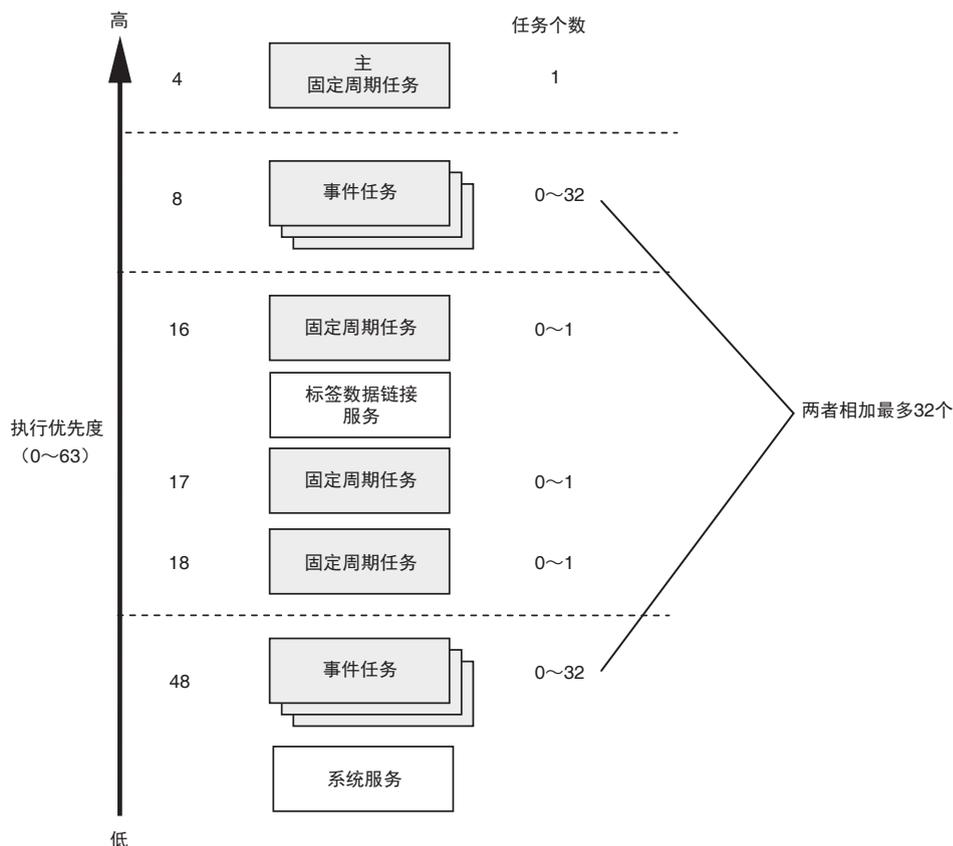
\*2. 标签数据链接的执行间隔和执行时间的详情请参考 [□□「5-4-2 标签数据链接服务的处理内容和执行时序」（P.5-28）](#)。

## 5-4-1 标签数据链接服务和系统服务的执行优先度和执行顺序

下面介绍标签数据链接服务和系统服务的执行优先度和执行顺序示例。

### 标签数据链接服务和系统服务的执行优先度

标签数据链接服务和系统服务的执行优先度如下图所示。标签数据链接服务的执行优先度在固定周期任务（执行优先度 16）和固定周期任务（执行优先度 17）之间。系统服务的执行优先度比所有任务低。

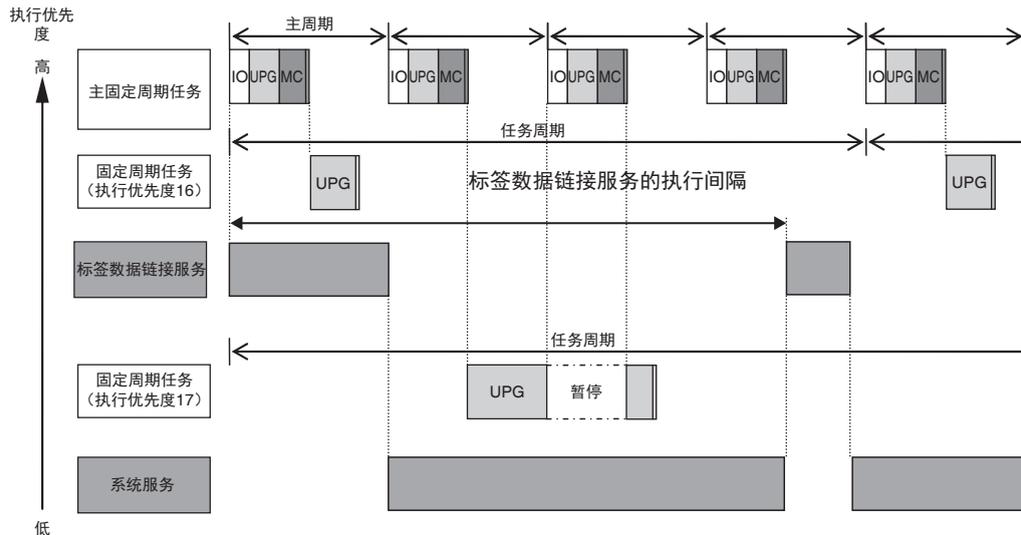


## 标签数据链接服务和系统服务的执行顺序示例

主固定周期任务、固定周期任务（执行优先级 16）、固定周期任务（执行优先级 17）、标签数据链接服务、系统服务同时存在时，各任务的执行顺序示例。

### ● NY 系列控制器时

- 可在执行任务时，并行执行标签数据链接服务和系统服务。
- 正在执行标签数据链接服务时，不会执行系统服务。



标签数据链接服务的执行间隔或 1 次的执行时间因 NY 系列的型号或标签数据链接设定不同而异。详情请参考□「5-4-2 标签数据链接服务的处理内容和执行时序」(P.5-28)。

### 5-4-2 标签数据链接服务的处理内容和执行时序

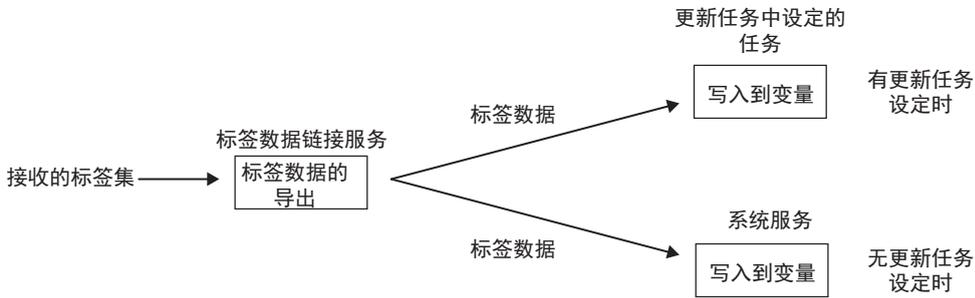
下面介绍标签数据链接服务的处理内容和执行时序。

#### 标签数据链接服务的处理内容

标签数据链接处理细分为多个处理。这些处理通过标签数据链接服务、系统服务、任务来执行。下面表示使用 NY 系列内置 EtherNet/IP 端口进行标签数据链接处理时的示例。

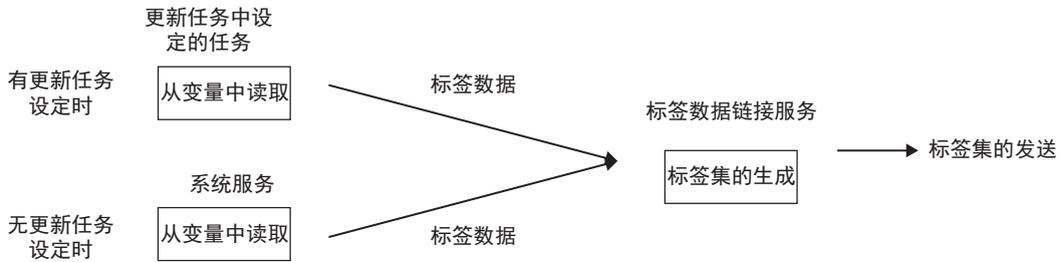
##### ● 标签数据接收处理的流程

- 通过标签数据链接服务，从接收的标签集中导出标签数据。
- 设定有更新任务时，利用该任务，将标签数据的值写入到标签上分配的变量中。
- 未设定更新任务时，利用系统服务，将标签数据的值写入到标签上分配的变量中。



##### ● 标签数据发送处理的流程

- 设定有更新任务时，利用该任务，从标签上分配的变量中读取标签数据的值。
- 未设定更新任务时，利用系统服务，从标签上分配的变量中读取标签数据的值。
- 然后，通过标签数据链接服务，根据标签数据生成标签集，然后发送。





## 参考

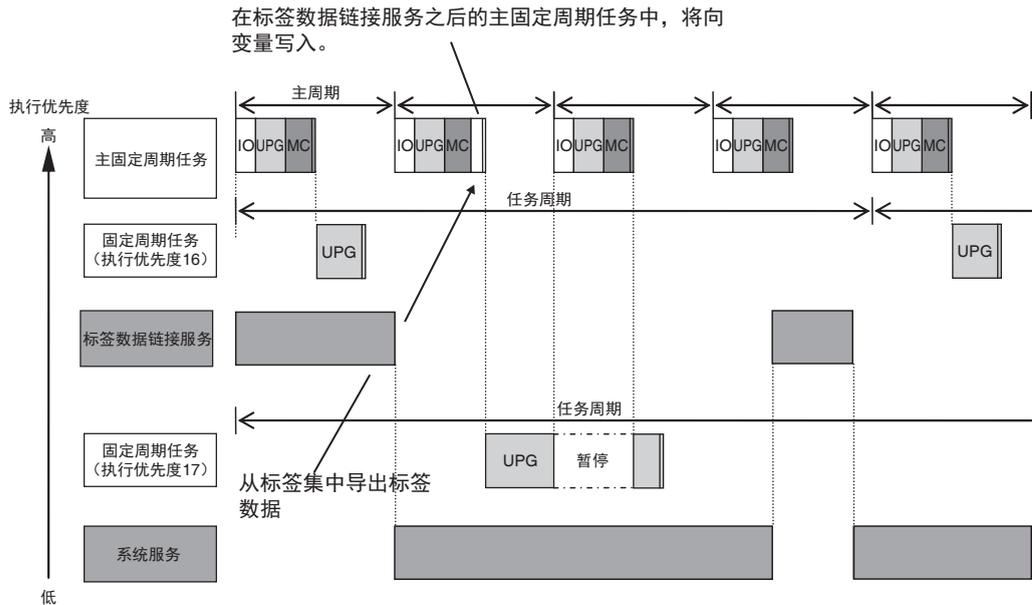
## 有或无更新任务设定时标签数据链接处理时序的不同

有或无更新任务设定时，向变量写入值的处理不同。有更新任务设定时，利用该任务写入值，无时，利用系统服务。因此，标签数据链接处理的执行时序也会不同。

以下表示在 NY 系列控制器中接收标签数据链接时，执行时序的不同。

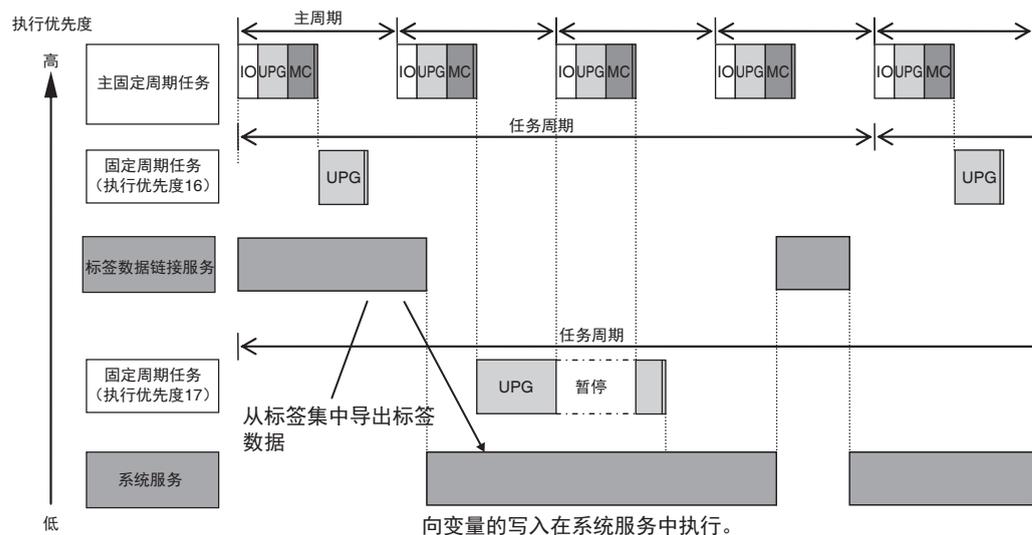
- 有更新任务设定时

将主固定周期任务设定为更新任务时的示例。在标签数据链接服务之后的主固定周期任务中，写入到变量。



- 无更新任务设定时

向变量的写入在系统服务中执行。



### 5-4-3 系统服务的处理内容和执行时序

下面介绍系统服务的处理内容和执行时序。

#### 系统服务的处理内容

系统服务的处理中包括以下内容。

系统服务的处理名称	内容
内置 EtherNet/IP 端口服务	<ul style="list-style-type: none"> <li>对来自编程设备、触摸屏、上位计算机或其他控制器的服务请求（CIP 等）进行处理</li> <li>通信指令（CIP、套接字）的执行</li> </ul>
内部通信支持服务	<ul style="list-style-type: none"> <li>NY 系列内部对来自 Windows 的服务请求（CIP、HTTP）的处理</li> </ul>
内置 EtherCAT 端口服务	<ul style="list-style-type: none"> <li>EtherCAT 信息通信的执行</li> </ul>
虚拟 SD 存储卡服务	<ul style="list-style-type: none"> <li>从 FTP 客户端访问</li> <li>通过 Sysmac Studio 对 SD 存储卡进行操作</li> <li>SD 存储卡指令的执行</li> </ul>
自诊断处理	<ul style="list-style-type: none"> <li>硬件的异常检测</li> </ul>

#### 系统服务的执行时序

NY 系列控制器的系统服务不受任务的影响。但是，正在执行标签数据链接服务时，不会执行系统服务。

系统服务的处理没有优先顺序。各处理在进行时间切片后并列执行。

## 5-5 任务相关的分配和设定

下面介绍任务相关的分配和设定。

### 5-5-1 分配到 I/O 刷新的任务中

将 EtherCAT 从站的 I/O 刷新分配到任务中。分配单位因 I/O 刷新对象的不同而异。分配单位是指可分配到 1 个 I/O 刷新上的 I/O 刷新对象的总和。例如，分配单位为从站终端时，即使 1 台通信耦合器单元上连接了多台 NX 单元，可分配 I/O 刷新的任务仅为 1 个。

想要在未分配 I/O 刷新的任务中进行输入输出时，请参考□□「未分配 I/O 刷新的任务的输入输出」(P.5-32)。

I/O 刷新对象、可分配的任务和分配单位的关系如下。

I/O 刷新对象	可分配的任务	分配单位
通信耦合器单元	主固定周期任务	从站终端
EtherCAT 从站	主固定周期任务	从站

### Sysmac Studio 上的设定操作

未分配到轴的各从站 / 单元应在 Sysmac Studio 的 [任务设定]-[设定控制 I/O 的任务] 中设定要进行 I/O 刷新的任务。

详情请参考□□「设定控制 I/O 的任务」(P.4-8)。

已分配到轴的各从站 / 单元若在 Sysmac Studio 的 [运动控制设定] 中选择要使用的运动控制，将设定要进行 I/O 刷新的任务。

详情请参考□□《NY 系列 工业平板电脑 / 工业 BOX PC 用户手册 运动控制篇 (SBCE-CN5-379)》。

### I/O 刷新的时序

EtherCAT 从站的 I/O 刷新时序如下所示。

I/O 刷新对象	控制 I/O 的任务	I/O 刷新的周期
EtherCAT 从站	主固定周期任务	主固定周期任务的任务周期

### 在用户程序上访问 I/O 的方法

通过用户程序访问 I/O 端口时，需要使用设备变量。

请通过 [设定控制 I/O 的任务] 中所设定任务的程序访问设备变量。

## 未分配 I/O 刷新的任务的输入输出

若试图用未分配 I/O 刷新的任务输出数据，Sysmac Studio 的程序检查时将发生错误。例如，在从站终端中，将各 NX 单元的处理分配到不同的任务时，只能通过分配有 I/O 刷新的任务来输出数据。此时，需要用未分配 I/O 刷新的任务向分配有 I/O 刷新的任务传递数据，然后从分配有 I/O 刷新的任务输出数据。

下面为用未分配 I/O 刷新的任务向分配有 I/O 刷新的任务传递数据并进行输出的示例程序。

此外，本示例程序中，利用未分配 I/O 刷新的任务来输入外部数据。若执行上述处理，Sysmac Studio 的程序检查时会出现警告，但仍可执行。

### ● 单元构成

使用从站终端。从站终端的单元构成如下所示。

型号	产品名称
NX-ECC20□	EtherCAT 耦合器单元
NX-ID3317	DC 输入单元
NX-OD3256	晶体管输出单元

### ● I/O 映射

I/O 映射如下。表中仅标示示例程序中使用的接口。

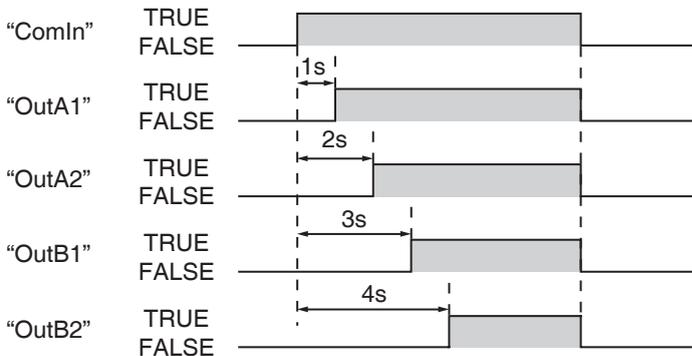
位置	端口	说明	R/W	数据类型	变量	变量注释	变量种类
Unit1	NX-ID3317						
	Input Bit 00	输入接口 00	R	BOOL	ComIn	共通输入值	全局变量
Unit2	NX-OD3256						
	Output Bit 00	输出接口 00	W	BOOL	OutA1	输出值 A1	全局变量
	Output Bit 01	输出接口 01	W	BOOL	OutA2	输出值 A2	全局变量
	Output Bit 02	输出接口 02	W	BOOL	OutB1	输出值 B1	全局变量
	Output Bit 03	输出接口 03	W	BOOL	OutB2	输出值 B2	全局变量

### ● 输入输出规格

输入输出规格如下所示。

- “ComIn”的值从FALSE变为TRUE后，1秒后“OutA1”、2秒后“OutA2”、3秒后“OutB1”、4秒后“OutB2”的值从FALSE变为TRUE。
- “ComIn”的值从TRUE变为FALSE后，“OutA1”、“OutA2”、“OutB1”、“OutB2”的所有值从TRUE变为FALSE。

时序图如下所示。



## ● 任务的处理内容

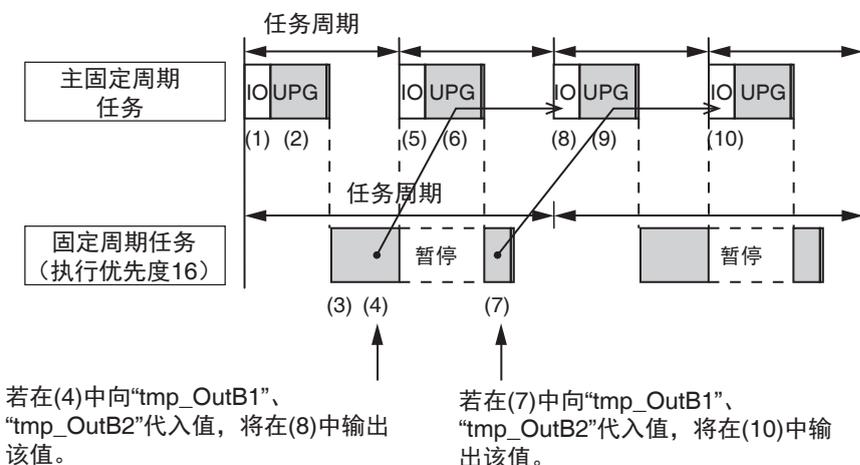
任务中使用主固定周期任务和固定周期任务（执行优先级 16）。将 I/O 刷新分配到主固定周期任务中。

“OutA1”、“OutA2” 的控制主固定周期任务中执行。从 “ComIn” 的输入，到运算、“OutA1”、“OutA2” 的输出等一系列处理全部在主固定周期任务中执行。

“OutB1”、“OutB2” 的控制中，从 “ComIn” 的输入到运算为止在固定周期任务中执行。固定周期任务中的运算结果将代入到临时变量的全局变量 “tmp\_OutB1”、“tmp\_OutB2” 中。主固定周期任务中，将 “tmp\_OutB1”、“tmp\_OutB2” 的值代入 “OutB1”、“OutB2” 后，将输出 “OutB1”、“OutB2”。

固定周期任务可能因主固定周期任务的执行而暂停。将固定周期任务中的运算结果代入 “tmp\_OutB1”、“tmp\_OutB2” 的处理，是在固定周期任务暂停前执行还是在暂停后执行，会影响到主固定周期任务输出固定周期任务运算结果的时序。

处理流程如下图所示。假设固定周期任务的任务周期为主固定周期任务的任务周期的 2 倍。



- (1) 通过主固定周期任务的 I/O 刷新，进行 “ComIn” 的输入。
- (2) 通过执行主固定周期任务的用户程序进行运算，将值代入 “OutA1”、“OutA2” 中。然后，将 “tmp\_OutB1”、“tmp\_OutB2” 的值代入 “OutB1”、“OutB2” 中。但是，由于未通过固定周期任务将值代入 “tmp\_OutB1”、“tmp\_OutB2”，所以 “tmp\_OutB1”、“tmp\_OutB2” 的值为初始值。因此，“OutB1”、“OutB2” 的值也是初始值。
- (3) 通过固定周期任务，进行 “ComIn” 的输入。
- (4) 通过执行固定周期任务的用户程序进行运算。  
正在执行固定周期任务时，若开始执行主固定周期任务，将暂停固定周期任务。将值代入 “tmp\_OutB1”、“tmp\_OutB2” 的处理，是在固定周期任务暂停前还是暂停后，受固定周期任务中的处理时序影响。
- (5) 通过主固定周期任务的 I/O 刷新，进行 “OutA1”、“OutA2”、“OutB1”、“OutB2” 的输出。同时，将进行第 2 次 “ComIn” 输入。

- (6) 通过执行主固定周期任务的用户程序进行运算，将值代入“OutA1”、“OutA2”中。  
然后，将“tmp\_OutB1”、“tmp\_OutB2”的值代入“OutB1”、“OutB2”中。在(4)中将值代入“tmp\_OutB1”、“tmp\_OutB2”后，固定周期任务的运算结果将反映到“OutB1”、“OutB2”中。在(4)中未将值代入“tmp\_OutB1”、“tmp\_OutB2”时，“tmp\_OutB1”、“tmp\_OutB2”的值为初始值。因此，“OutB1”、“OutB2”的值也是初始值。
- (7) 通过执行固定周期任务的用户程序，进行(4)后面的运算。  
在(4)中未将值代入“tmp\_OutB1”、“tmp\_OutB2”时，在这里将值代入“tmp\_OutB1”、“tmp\_OutB2”中。
- (8) 通过主固定周期任务的 I/O 刷新，进行“OutA1”、“OutA2”、“OutB1”、“OutB2”的输出。  
在(4)中将值代入“tmp\_OutB1”、“tmp\_OutB2”后，固定周期任务的运算结果将作为“OutB1”、“OutB2”输出。  
在(7)中将值代入“tmp\_OutB1”、“tmp\_OutB2”后，将输出“OutB1”、“OutB2”的初始值。
- (9) 通过执行主固定周期任务的用户程序进行运算，将值代入“OutA1”、“OutA2”中。  
然后，将“tmp\_OutB1”、“tmp\_OutB2”的值代入“OutB1”、“OutB2”中。在(7)中将值代入“tmp\_OutB1”、“tmp\_OutB2”后，固定周期任务的运算结果将在这里反映到“OutB1”、“OutB2”中。
- (10) 通过主固定周期任务的 I/O 刷新，进行“OutA1”、“OutA2”、“OutB1”、“OutB2”的输出。  
在(7)中将值代入“tmp\_OutB1”、“tmp\_OutB2”后，固定周期任务的运算结果将在这里作为“OutB1”、“OutB2”输出。

此外，为了防止“tmp\_OutB1”、“tmp\_OutB2”的值在主固定周期任务内参照前被固定周期任务覆盖，应使用 Lock、Unlock 指令，进行任务间的排他控制。任务间的排他控制详情请参考□「5-6-1 确保任务间变量值同时性的方法」(P.5-40)。

## ● 全局变量表

全局变量如下所示。

全局变量的变量表

名称	数据类型	初始值	分配目标	注释
ComIn	BOOL	FALSE	ECAT://node#[1,1]/Input Bit 00	共通输入值
OutA1	BOOL	FALSE	ECAT://node#[1,2]/Output Bit 00	输出值 A1
OutA2	BOOL	FALSE	ECAT://node#[1,2]/Output Bit 01	输出值 A2
OutB1	BOOL	FALSE	ECAT://node#[1,2]/Output Bit 02	输出值 B1
OutB2	BOOL	FALSE	ECAT://node#[1,2]/Output Bit 03	输出值 B2
tmp_OutB1	BOOL	FALSE		B1 的临时变量
tmp_OutB2	BOOL	FALSE		B2 的临时变量

## ● 主固定周期任务的梯形图

主固定周期任务的梯形图如下所示。

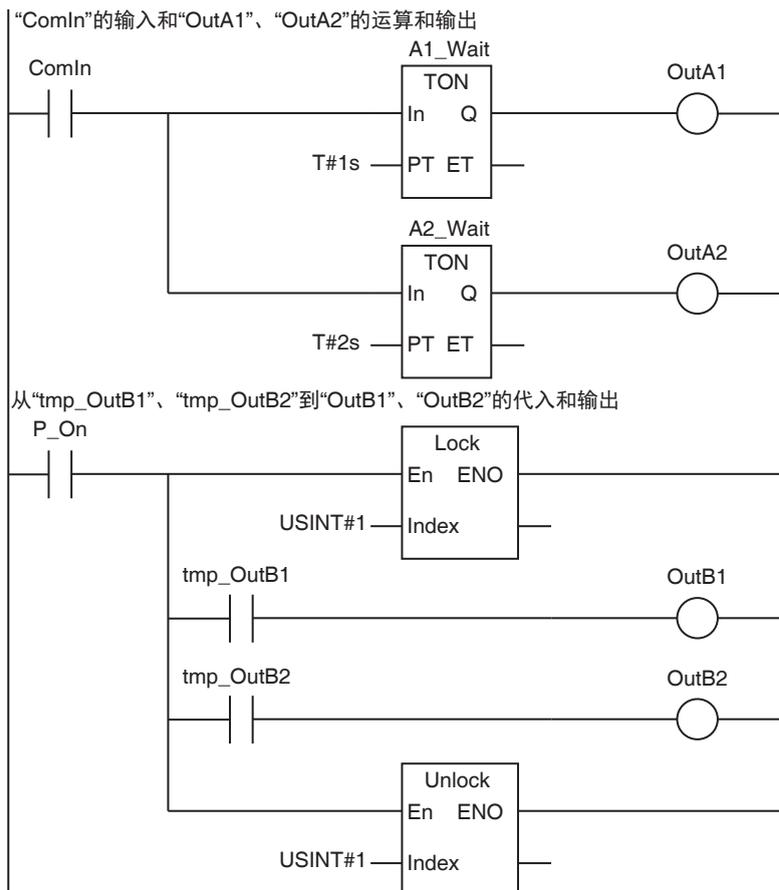
内部变量表

名称	数据类型
A1_Wait	TON
A2_Wait	TON

外部变量表

名称	数据类型	注释
ComIn	BOOL	共通输入值
OutA1	BOOL	输出值 A1
OutA2	BOOL	输出值 A2
OutB1	BOOL	输出值 B1
OutB2	BOOL	输出值 B2
tmp_OutB1	BOOL	B1 的临时变量
tmp_OutB2	BOOL	B2 的临时变量

算法部分



### ● 固定周期任务的梯形图

固定周期任务的梯形图如下所示。

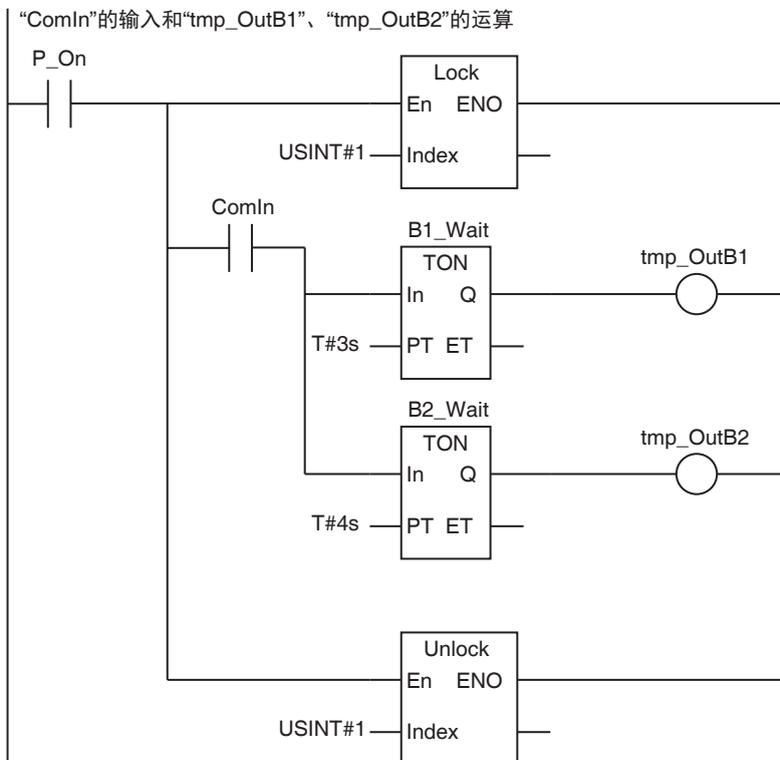
内部变量表

名称	数据类型
B1_Wait	TON
B2_Wait	TON

外部变量表

名称	数据类型	注释
ComIn	BOOL	共通输入值
tmp_OutB1	BOOL	B1 的临时变量
tmp_OutB2	BOOL	B2 的临时变量

固定周期任务的算法部分



## 5-5-2 将程序分配到任务中

将要执行的程序分配到任务中（每个任务最多 128 个）。同时，设定各程序的运行开始时动作。

### 程序的执行顺序

在 Sysmac Studio 中设定任务内程序的执行顺序。

### 程序的运行开始时初始状态

设定各程序的运行开始时动作。运行开始时初始状态是指控制器的动作模式从程序模式变更为运行模式后，首先执行分配有该程序的任务时，是否执行该程序。设定范围为“启动”、“停止”之一。

初始状态为“停止”时，若通过 PrgStart 指令发出执行指示，将在下次执行该程序的时序起执行。初始状态为“启动”时，若通过 PrgStop 指令发出停止指示，将在下次执行该程序的时序起停止。

#### ● Sysmac Studio 上的设定操作

在 Sysmac Studio 的 [配置和设定]-[任务设定]-[程序的分配设定] 中，对向任务分配程序、任务中程序的执行顺序及各程序的初始状态进行设定。详情请参考 [□□「程序的分配设定」](#)（P.4-9）。

### 任务中可分配的 POU

任务中可分配 0 ~ 128 个程序。

可分配的 POU 仅限程序。FB 实例及 FUN 不可直接分配到程序中。同一个程序不可分配到多个任务中。

## 5-5-3 任务的设定项目

各任务的设定项目如下所示。

● 主固定周期任务的设定项目

设定项目		设定范围	默认	反映时序	运行模式中 改写
任务类型		选择“主固定周期任务”。	—	设定传送时	不支持
	执行优先级	固定为“4”	—		
任务名称		字符串	PrimaryTask		
周期 / 执行条件	任务周期 <sup>*1</sup>	500 $\mu$ s ~ 8ms (以 250 $\mu$ s 为单位)	1ms		
任务周期超限检测		设定在超过有任务执行时间设定的任务周期时，是否检测错误。 • 检测 (发生轻度故障等级的控制器异常) • 不检测 (作为监视信息记录到事件日志中) 详情请参考□□「任务周期超限」(P.5-50)。	检测		
任务执行超时时间		设定在无限循环等任务不会完成处理时的超时检测时间。 设定为任务周期的倍数。 1倍 ~ 5倍 详情请参考□□「任务执行超时」(P.5-51)。	5倍		
变量访问时间 [%]		设定变量访问时间与任务周期的比例。 1% ~ 50% 请参考□□「变量访问时间设定」(P.5-48)。	3%		

\*1. EtherCAT 设定的过程数据通信周期 (过程数据通信周期) 与本周期的值相同。

### ● 固定周期任务（执行优先级 16、17、18）的设置项目

设定项目	设定范围	默认	反映时序	运行模式中 改写
任务类型	选择以下任意一个。 “固定周期任务（优先级 16）” “固定周期任务（优先级 17）” “固定周期任务（优先级 18）”	—	设定传送时	不支持
执行优先级	将自动选择为“16”“17”“18”中的一个。	—		
任务名称	字符串	Periodic Task0		
周期 / 执行条件	任务周期 参考□□「5-3-1 NY 系列的规格」 (P.5-8)	10ms		
任务周期超限检测	与主固定周期任务相同。	与主固定周期任务相同。		
任务执行超时时间				
变量访问时间 [%]				

### ● 事件任务的设置项目

设定项目	设定范围	默认	反映时序	运行模式中 改写
任务类型	选择以下任意一个。 “事件任务（优先级 8）” “事件任务（优先级 48）”	—	设定传送时	不支持
任务名称	字符串	Event Task0		
执行条件	选择“通过指令执行”、“变量的条件式一致”中的任意一个。	通过指令执行		
任务执行超时时间	设定在无限循环等任务不会完成处理时的超时检测时间。设定单位为 1ms。 • 执行优先级“8”时：1 ~ 500ms • 执行优先级“48”时：1ms ~ 10s	<ul style="list-style-type: none"> <li>• 执行优先级“8”时：200ms</li> <li>• 执行优先级“48”时：1s</li> </ul>		

### ● Sysmac Studio 上的设置操作

在 Sysmac Studio 的 [配置和设定]-[任务设定] 中，进行任务的追加和各任务的设定。

详情请参考□□「任务设置」(P.4-7)。

## 5-6 确保变量的同时性

下面介绍在任务之间确保变量值同时性的方法和从控制器外部进行变量访问。

### 5-6-1 确保任务间变量值同时性的方法

通过多个不同的任务对同一个全局变量进行读写时，确保任务间全局变量值同时性的方法有以下两种。它们统称为“变量的任务间排他控制功能”。

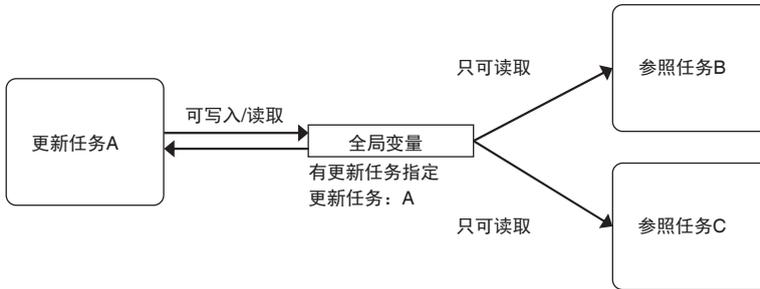
- 方法 1：通过 1 个任务写入到全局变量中，通过其他的多个任务读取时  
使用变量的任务间排他控制设定功能。
- 方法 2：想要通过多个任务写入到全局变量时  
使用任务间排他控制指令。

#### 方法 1) 变量的任务间排他控制设定功能

##### ● 概要

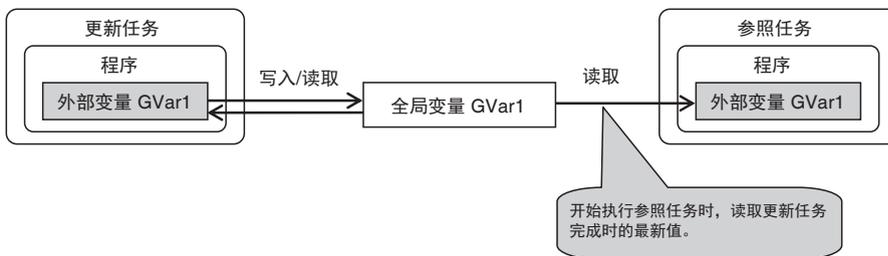
指定更新和参照全局变量的任务，确保对于参照侧任务而言全局变量值的同时性。

指定 1 个对全局变量进行读写的任务（更新任务）。针对该全局变量，指定只进行读取的任务（参照任务）。由此，可确保全局变量值的同时性。



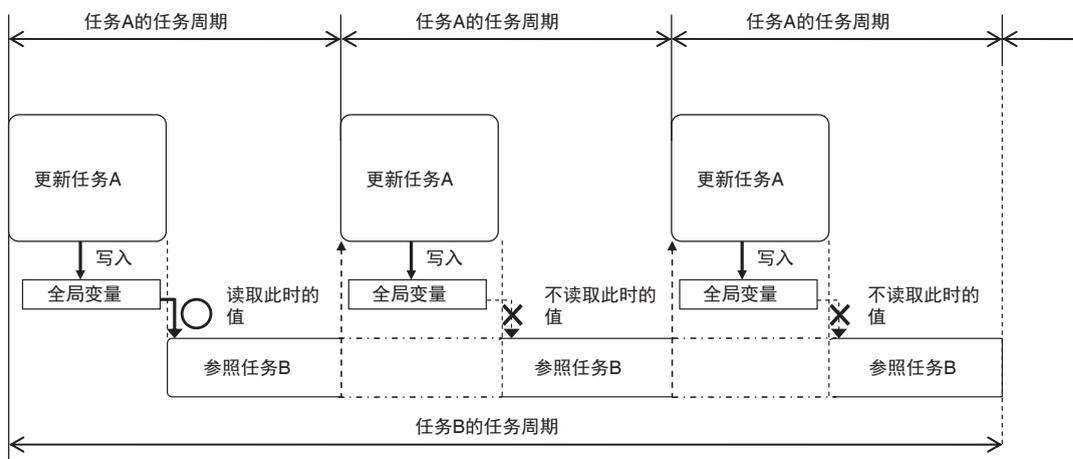
##### ● 用途示例

存在通过主固定周期任务写入的全局变量，在固定周期任务上读取该变量的值时，要在该固定周期任务内确保变量值的同时性时使用。



## ● 原理

对于设定了更新任务的全局变量，待更新任务完成后，参照任务在自己的任务开始执行时，将其写入的值读取出来。



因此，无需特别的用户程序，即可确保任务内全局变量值的同时性。  
若使用通过参照任务将值写入全局变量的指令， Sysmac Studio 的程序检查时将发生错误。

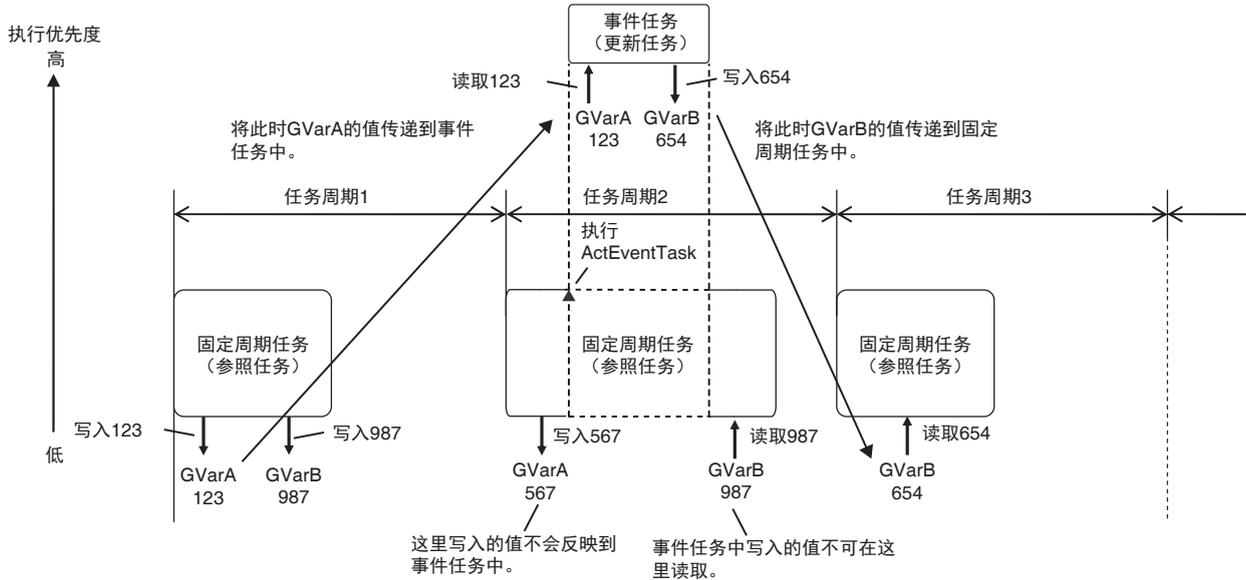


### 使用注意事项

使用 ActEventTask 指令时，请注意在两个任务之间，读取全局变量何时的值、何时更新值。

例如，下图情况下，事件任务读取的全局变量 GVarA 的值为任务周期 1 结束时的值。因此，即使在任务周期 2 中通过固定周期任务写入了 GVarA 的值，也不会反映到事件任务中。

此外，事件任务写入的全局变量 GVarB 的值传递到固定周期任务的时间为任务周期 3 的开始时间。因此，即使在任务周期 2 中通过固定周期任务读取 GVarB 的值，也无法读取事件任务写入的值。



使用 ActEventTask 指令执行事件任务时，若要接收全局变量的值，请勿使用“变量的任务间排他控制功能”。

使用 ActEventTask 指令时，为了确保全局变量的同时性，任务执行中判定指令 Task\_IsActive 为有效。Task\_IsActive 指令是指判定指定的任务是否正在执行或处于待机状态的指令。使用本指令执行事件任务的过程中，请勿从其他任务参照事件任务进行写入所需的参数。Task\_IsActive 指令的详情请参考 □ 《NY 系列 指令基准手册 基本篇 (SBCA-CN5-437)》。

### ● 限制

- 对 1 个全局变量只设定 1 个更新任务。若要通过多个任务写入同一全局变量，请使用后述的任务间排他控制指令。
- 在结构体 / 联合体变量中设定更新任务时，应对结构体 / 联合体变量整体设定 1 个更新任务。不可对每个结构体 / 联合体成员设定更新任务。
- 在排列变量中设定更新任务时，应对排列变量整体设定 1 个更新任务。不可对每个排列要素设定更新任务。



### 使用注意事项

对于需要确保值同时性的全局变量，请勿用参照任务、更新任务以外的任务写入值。对于设定了更新任务的全局变量，若用非更新任务、非参照任务的用户程序写入，将无法保证该全局变量值的同时性。若创建这样的程序，程序检查时会出现警告。



## 参考

- 使用变量的任务间排他控制功能的全局变量中，可将表示该全局变量的外部变量设定为数据追踪的采样对象。由此，可通过数据追踪，分别对更新任务和参照任务的全局变量值进行采样。数据追踪请参考□□「8-4-4 数据追踪」（P.8-35）。

## ● Sysmac Studio 上的设定操作

在 Sysmac Studio 的 [任务设定]-[变量的任务间排他控制设定] 中，设定用于指定更新任务的全局变量（“要更新的变量”）以及与之对应的参照任务（“要参照的任务”）。

详情请参考□□「变量的任务间排他控制设定」（P.4-10）。

## 方法 2) 任务间排他控制指令

通过多个任务写入同一全局变量时，应使用任务间排他控制指令（Lock/Unlock 指令），确保全局变量的同时性。

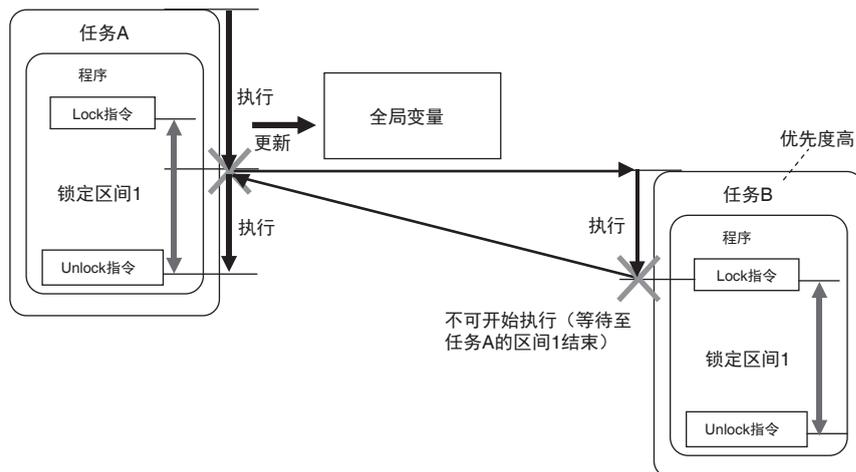
任务间排他控制指令中，正在执行从 Lock 指令到 Unlock 指令之间的区间（锁定区间）时，其他任务中相同锁定编号的锁定区间将无法执行。将写入到全局变量的操作设定在锁定区间内，通过多个任务写入全局变量时，也可确保值的同时性。

详情请参考□□《NY 系列 指令基准手册 基本篇（SBCA-CN5-437）》的 Lock（任务间排他锁定）指令、Unlock（解除任务间排他锁定）指令。

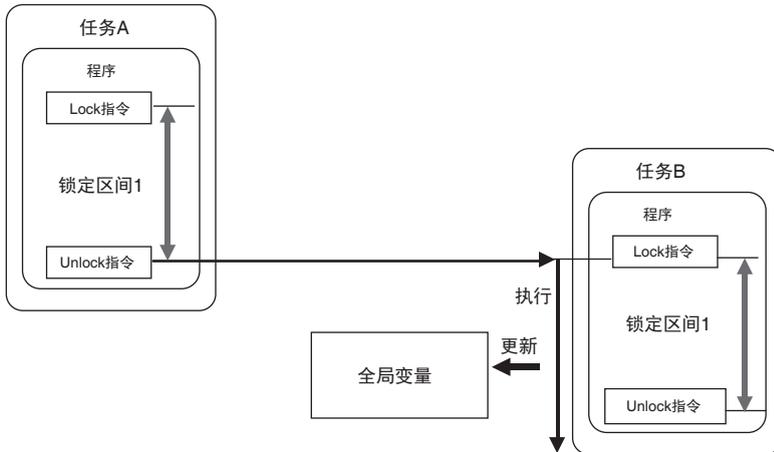
例)

假设任务 A、任务 B 都是拥有锁定区间 1 的任务，且任务 B 的优先度高于任务 A。

正在执行任务 A 的锁定区间 1 时，若任务 B 的执行条件成立，则任务 A 将在锁定区间 1 的中途暂停，执行任务 B。但是，有任务 A 中的锁定区间 1 尚未结束，因此任务 B 会在锁定区间 1 的处理前暂停。任务 B 暂停后，将重新开始任务 A 的锁定区间 1。



任务 A 的锁定区间 1 结束后，任务 A 将再次暂停，执行任务 B 的锁定区间 1 之后的内容。在任务间独占全局变量的写入处理，确保值的同时性。



**使用注意事项**

- 锁定区间请设为最小需要长度。若锁定区间过长，可能超出任务的执行周期。
- Lock 指令和 Unlock 指令请务必在同一 POU 的同一段落中成对使用。

**在任务间访问相同的全局变量时的示例**

下面介绍使用多个全局变量向其他任务发出处理请求的情况。在以下示例程序中，将一个全局变量“gReq”作为处理的排他标志，通过用户程序进行排他控制。此时，对于访问任务间使用的多个全局变量“gPar1”、“gPar2”、“gReq”的逻辑，需要用“变量的任务间排他控制功能”进行排他控制。

下面表示使用任务间排他控制指令（Lock/Unlock 指令）进行排他控制的示例程序。

● **全局变量**

名称	数据类型	注释
gReq	BOOL	请求标志
gPar1	ULINT	参数 1
gPar2	DATA_AND_TIME	参数 2

## ● 请求处理的任务 (MainTask)

- 内部变量

名称	数据类型	注释
ReqTrg	BOOL	请求触发
cCnt	ULINT	100ms 计数值
cTime	DATA_AND_TIME	当前时刻

- ST 程序

```

cCnt:=Get100msCnt();          (* 获取 100ms 计数值 *)
cTime:=GetTime();            (* 获取当前时刻 *)

Lock(1);                      (* 任务间独占锁定开始 *)
IF ReqTrg=TRUE AND gReq=FALSE THEN  (* 独占标志的参照 *)
  gPar1:=cCnt;                (*SubTask 中处理的参数组 *)
  gPar2:=cTime;              (*SubTask 中处理的参数组 *)
  gReq:=TRUE;
  ReqTrg:=FALSE;
END_IF;
Unlock(1);                    (* 解除任务间独占锁定 *)

```

## ● 请求处理的任务 (SubTask)

- 内部变量

名称	数据类型	注释
ReqBusy	BOOL	—
UserDefFB_ins	UserDefFB	执行处理的用户定义 FB 的实例

- ST 程序

```

Lock(1);                      (* 任务间独占锁定开始 *)
IF gReq=TRUE AND ReqBusy=FALSE THEN  (* 独占标志的参照 *)
  ReqBusy:=TRUE;
  UserDefFB_ins.PutData:=gPar1;      (* 通过 MainTask 读取参数 *)
  UserDefFB_ins.PutDate:=gPar2;     (* 通过 MainTask 读取参数 *)
  gReq:=FALSE;                      (* 独占标志的重置 *)

  UserDefFB_ins.Execute:=TRUE;
END_IF;
Unlock(1);                    (* 解除任务间独占锁定 *)

UserDefFB_ins();

IF UserDefFB_ins.Done:=TRUE THEN
  UserDefFB_ins.Execute:=FALSE;
  ReqBusy:=FALSE;
END_IF;

```

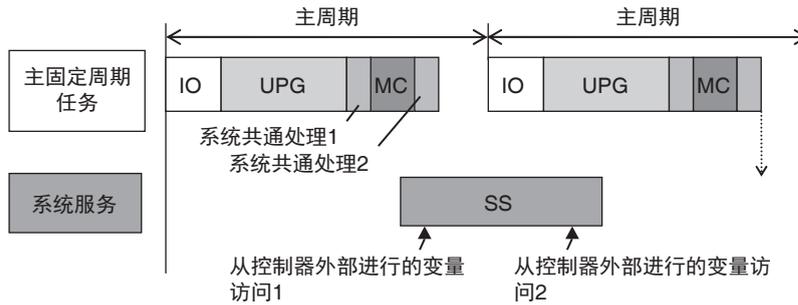
### 5-6-2 从控制器外部进行的变量访问

从控制器外部进行的变量访问通过系统服务执行。系统服务的执行优先度比任务低。因此，从控制器外部访问多个变量时，各变量值的更新可能不在一个任务周期内执行。用户程序中，若同时存在值发生更新的变量和未更新的变量，控制器可能发生意外动作。

为防止这样的情况，应在任务的“系统共通处理 2”中执行控制器外部的变量访问，这样可以确保在同一任务周期内执行多个变量值的更新。

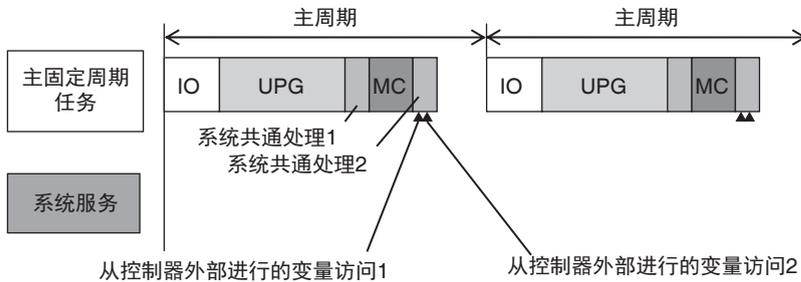
● 从控制器外部进行的变量访问通过系统服务执行时

多个变量的访问可能不在同一任务周期内执行。



● 从控制器外部进行的变量访问通过系统共通处理 2 执行时

多个变量的访问在同一任务周期内执行。



下面介绍通过“系统共通处理 2”执行从控制器外部进行的变量访问的方法。

## 从控制器外部访问变量的方法

从控制器外部访问变量的方法有以下 4 种。

- Sysmac Studio
- NS/NA 系列触摸屏
- EtherNet/IP 的标签数据链接
- 上位计算机等发出的 CIP 通信指令

但是，变量访问方法为 Sysmac Studio 时，任务的“系统共通处理 2”中可执行的处理仅为值的更新处理。值的参照处理在系统服务中执行。

## 通过系统共通处理 2 执行变量访问的任务

从控制器外部进行的变量访问通过“系统共通处理 2”执行时，变量的种类如下。

变量	更新值的任务
变量的任务间排他控制设定中指定的全局变量	Sysmac Studio 的 [ 配置和设定 ]-[ 任务设定 ]-[ 变量的任务间排他控制设定 ] 中指定的更新任务
EtherCAT 从站用设备变量	Sysmac Studio 的 [ 配置和设定 ]-[ 任务设定 ]-[ 设定控制 I/O 的任务 ] 中指定的任务

## 通过系统共通处理 2 执行变量访问所需的设定

要通过任务的“系统共通处理 2”执行从控制器外部进行的变量访问，需要在 Sysmac Studio 上进行以下两个设定。

- 变量的任务间排他控制设定（对象变量为全局变量时）
- 变量访问时间设定

## 变量的任务间排他控制设定

从控制器外部进行的变量访问通过系统共通处理 2 执行时，若对象变量为全局变量，需要进行变量的任务间排他控制设定。变量的任务间排他控制功能是指：指定更新对象全局变量的任务，其他任务或来自控制器外部的访问将无法更新对象全局变量的值。

变量的任务间排他控制功能详情请参考□□「变量的任务间排他控制设定」（P.4-10）。



### 使用注意事项

使用 EtherNet/IP 的标签数据链接时，对于同一标签集中的标签（网络公开属性的变量），请务必将同一任务作为更新任务。否则，标签集内多个标签的更新时序可能跨多个不同的任务周期。

## 变量访问时间设定

若通过任务的“系统共通处理 2”执行从控制器外部进行的变量访问，任务执行时间可能会延长。用户需要在 Sysmac Studio 上设定变量访问时间的上限值。变量访问处理时间的上限值称为“变量访问时间”。

### ● 变量访问时间的计算方法

变量访问时间按以下公式计算。

$$\text{变量访问时间} [\mu\text{s}] = \text{变量的总大小} [\text{字节}] * a + \text{变量的数量} * b + \text{访问数量} * c + d$$

上述表达式中的常数 a ~ d 的值在 NY 系列控制器中如下。

控制器型号	常数的值 [ $\mu\text{s}$ ]			
	a	b	c	d
NY5□2-1□□□	0.0006	0.100	1.40	6.60

### ● 变量访问时间的设定方法

变量访问时间在 Sysmac Studio 的 [配置和设定]-[任务设定] 中设定。设定分任务进行，输入相对于任务周期的比例。默认值为 3%。设定方法的详情请参考 □□《Sysmac Studio Version 1 操作手册 (SBCA-CN5-362)》。

### ● 变量访问时间的设定示例

变量访问时间的设定示例如下所示。

假设针对 NY5□2-1□□□ 中动作的任务，有以下 3 种来自外部控制器的变量访问。

访问 No.	变量访问源	访问变量的总大小 [字节]	访问变量的数量	访问数量
1	EtherNet/IP 的标签数据链接	600	8	1
2	EtherNet/IP 的标签数据链接	200	4	1
3	CIP 通信指令	1,000	1	1

访问 No. 为 1 的变量访问时间根据以下公式计算。

$$\begin{aligned} \text{访问 No. 为 1 的变量访问时间} &= 600 * 0.0006 + 8 * 0.100 + 1 * 1.40 + 6.60 \\ &= 9.16 [\mu\text{s}] \end{aligned}$$

同样，各访问 No. 的变量访问时间为以下值。

访问 No.	变量访问时间 [ $\mu\text{s}$ ]
1	9.16
2	8.52
3	8.7

1 个任务周期中，若只发生这些访问中的一个，将以时间最长的访问 No.1 为准，设定变量访问时间。变量 No.1 的变量访问时间为 9.16 $\mu\text{s}$ ，所以假设任务周期为 500 $\mu\text{s}$ ，变量访问时间应设定为 9.16/500  $\doteq$  2%。

1 个任务周期中，若这些访问逐个发生，则变量访问时间根据以下公式计算。

$$\begin{aligned} \text{变量访问时间} &= (600+200+1000) \times 0.0006 + (8+4+1) \times 0.100 + (1+1+1) \times 1.40 + 6.60 \\ &= 13.18[\mu\text{s}] \end{aligned}$$

假设任务周期为 500 $\mu\text{s}$ ，则变量访问时间设定为  $13.18/500 \approx 3\%$ 。

### ● 实际的访问时间大于变量访问时间的设定值时的处理

实际的访问时间大于变量访问时间的设定值时，根据 1 个周期中的变量访问次数，执行以下处理。请设定为有足够余量的变量访问时间，确保多个变量的访问在同一任务周期内执行。

1 个任务周期中的 变量访问次数	处理
多次	执行变量访问时间的设定值以内的变量访问。未能执行的变量访问将在下一个任务周期内执行。 因此，多个变量的访问无法在一个任务周期内执行。
1 次	超过变量访问时间的设定值执行变量访问。因此，任务执行时间会变长。

## 5-7 任务相关的异常

下面介绍以下异常。

- 任务周期超限
- 运动控制周期超限
- 任务执行超时
- I/O 刷新周期超限

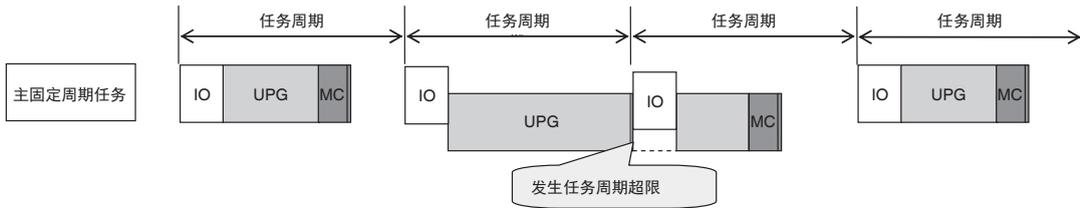
### 任务周期超限

任务的执行时间超过设定的任务周期时，发生“任务周期超限”。

轻度故障等级的控制器异常。发生错误后仍继续运行。

对象为主固定周期任务、固定周期任务。

此外，可通过设定而不发生任务周期超限。在 Sysmac Studio 的“任务设定” — “任务设定”的“任务周期超限检测”中设定。默认为“检测”。



异常名称	异常等级	处理
任务周期超限	轻度故障	修改任务设定及程序，重新下载。请通过 Sysmac Studio 解除异常。

无论“任务周期超限检测”设定如何，任务的处理未能在周期内完成时，将向系统定义变量“<任务名称>\_Exceeded”（任务周期超过标志）、“<任务名称>\_ExceedCount”（任务周期超过次数）、“\_ErrSta”（控制器异常状态）及事件日志输出信息。

任务处理未在周期内完成时的 I/O 刷新动作因输入输出对象的不同有以下差异。

输入输出的对象	任务处理未在周期内完成时的 I/O 刷新动作
EtherCAT 从站*1	输出：输出上一周期的值。 输入：执行 I/O 刷新，但输入数据不反映到正在执行的用户程序中。

\*1. 包括 EtherCAT 从站终端上的 NX 单元。



#### 使用注意事项

发生“任务周期超限”时，请在创建程序时确保在任务周期内，或变更任务周期。

## 运动控制周期超限

“运动控制（MC）”处理连续未能以主周期（=运动控制周期）完成时发生。运动控制功能模块发生部分停止故障等级的控制器异常。此时，将同时发生“任务周期超限”。

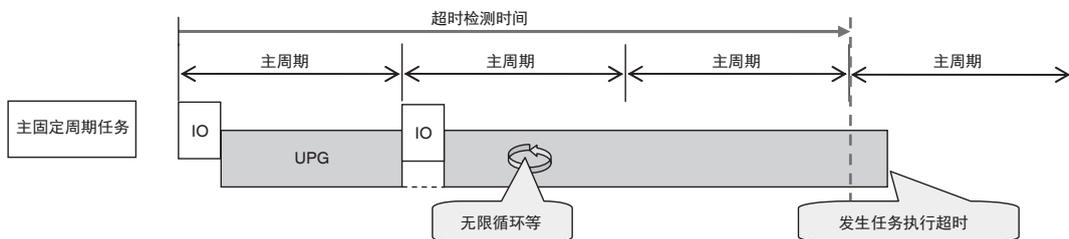
异常名称	异常等级	处理
运动控制周期超限	部分停止故障	请减少程序的处理内容，或在动作不会发生问题的范围内，将控制周期设定为更大的值。

## 任务执行超时

经过指定时间（“超时检测时间”）后任务仍未完成时，将发生“任务执行超时”。

全部停止故障等级的控制器异常。发生错误时，停止执行用户程序。

用户程序中的无限循环等，因逻辑错误而导致任务无法正常动作时也会发生。

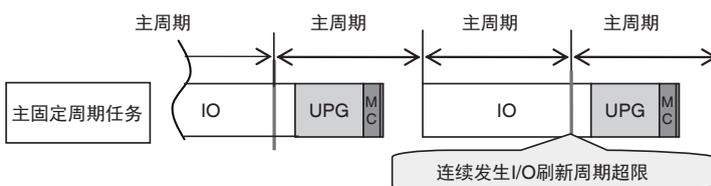


异常名称	异常等级	处理
任务执行超时	全部停止故障	修改任务设定，重新下载用户程序。 需要重新接通电源或进行复位操作。

## I/O 刷新周期超限

主固定周期任务中，若连续出现任务周期内无法完成 I/O 刷新的状态，将发生“I/O 刷新周期超限”。

全部停止故障等级的控制器异常。发生错误时，停止执行用户程序。



异常名称	异常等级	处理
I/O 刷新周期超限	全部停止故障	修改任务设定，重新下载。 需要重新接通电源或进行复位操作。

## 5-8 任务执行状态、任务执行时间的监视

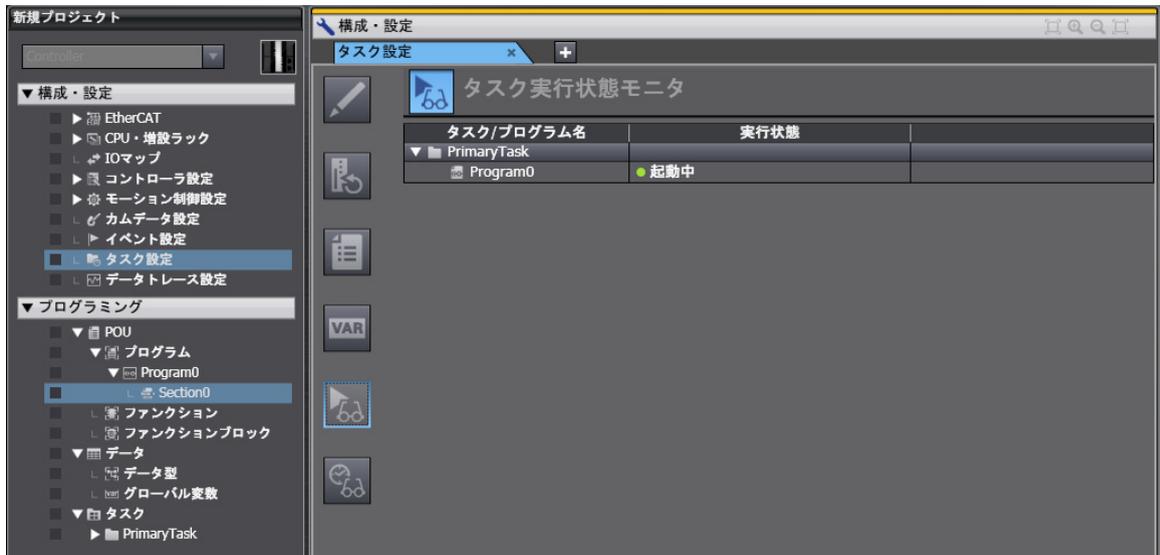
通过 Sysmac Studio 进行在线操作，可监视任务的执行状态、任务的执行时间。

### 任务执行状态的监视

可通过 Sysmac Studio 监视各任务中程序的执行状态（启动中 / 停止中）。

#### ● Sysmac Studio 上的操作

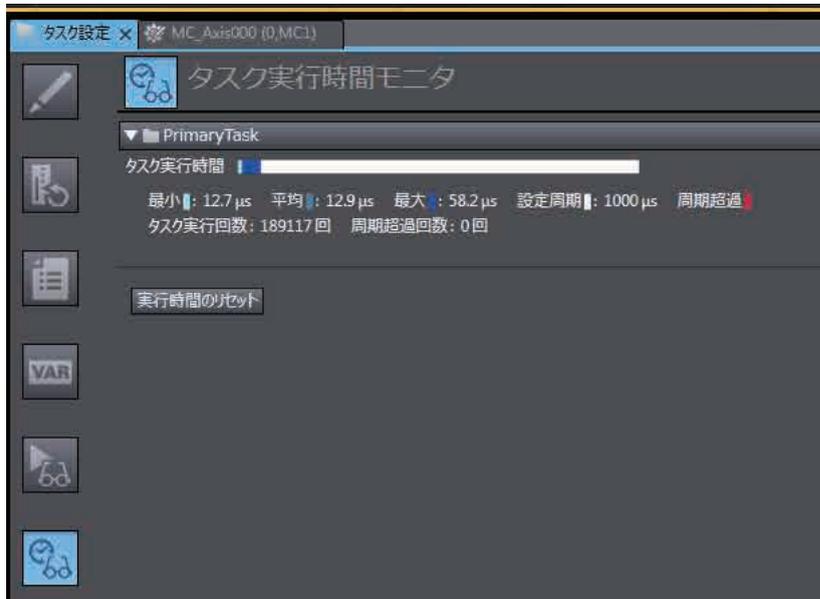
通过 Sysmac Studio 与控制器在线连接，选择 [ 配置和设定 ]-[ 任务设定 ]。单击任务执行状态监视按钮，将显示以下画面。



## 任务执行时间的监视

可通过 Sysmac Studio 监视各任务的执行时间。

- Sysmac Studio 中可监视的值  
连接控制器时



## 连接模拟器时的内置 EtherNet/IP 画面



设定项目如下所示。

使用的端口	设定项目	功能说明	设定值	默认
NY 系列内置 EtherNet/IP 端口	连接数	标签数据链接的连接数。	0 ~ 128	0
	最小 RPI	标签数据链接的连接中，最小的数据包间隔 (RPI) 设定值。	1ms ~ 10s (最小单位 1ms)	—

可监视的项目如下所示。

监视项目	说明	连接控制器时	
任务执行时间*1	最小	任务执行时间的最小值。	有
	平均	任务执行时间的平均值。	
	最大	任务执行时间的最大值。	
设定周期	设定的任务周期。*2		
超出周期	任务执行时间超过任务周期时 (系统定义变量“任务周期超过标志”为 TRUE 时)，在上面的栏中显示其程度。*2		
任务执行次数	表示任务的执行次数。 系统定义变量“任务执行次数”的值。		

\*1. 从任务开始执行到完成为止，实际花费的时间。包括开始执行到完成期间执行的其他任务和系统服务的时间。

\*2. 事件任务时不显示。



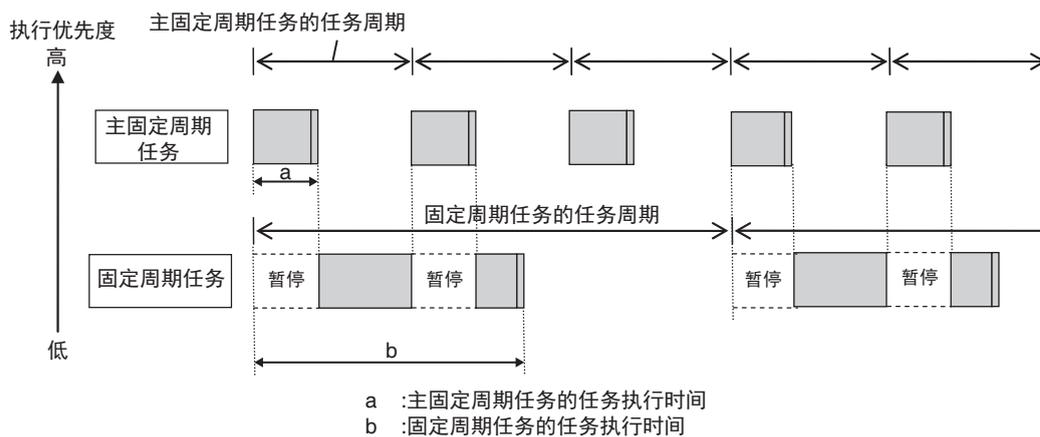
### 使用注意事项

请务必连接实际控制器并确认，再进行设计研究和正式运行。

## 任务实际时间和任务实际处理时间的含义

任务的执行时间监视器中显示的“任务执行时间”、“任务执行处理时间”的含义如下所示。

显示的时间	含义
任务执行时间	从任务周期开始到任务处理结束为止的时间。 但是，固定周期任务包含执行优先级更高的任务时的暂停时间。



## 5-9 任务设计方法及输入输出响应时间

下面介绍任务设计的指南、任务执行时间的预估方法、任务设计示例及 I/O 响应时间。

NY 系列控制器中，主固定周期任务或固定周期任务按设定的周期动作。

实际的执行时间超过设定的周期时，执行各种错误处理。

下面以 1 个主固定周期任务（单任务）构成时为例，介绍预估方法和评估方法。



### 使用注意事项

在实际的控制器上，因用户程序中搭载的逻辑是否动作、是否发生通信指令、数据链接动作、是否使用数据追踪等各种功能不同，任务执行时间会发生变动。

请务必使用实际机器在各种条件下确认性能，确认不会发生任务周期超限，确保良好的通信响应性。

### 5-9-1 任务执行时间的确认

请确保这里预估的任务执行时间平均值和最大值在设定的任务周期内。

#### ● 上机计算时

参考□□「A-2 NY 系列中任务实际处理时间的计算标准」（P.A-12），上机估算“任务执行时间的平均值”。

此外，上机时无法预估最大值。

#### ● 在实际控制器上计算时

连接实际控制器时，在 Sysmac Studio 的“任务执行时间监视功能”中，可以确认以下值。

- 任务执行时间（最小值、平均值、最大值）
- 设定周期
- 任务执行次数
- 周期超过次数

Sysmac Studio 中显示的最大值为在实际控制器上动作时的最大值。

如上所述，任务执行时间因实际控制器的内部状态而变化。



### 参考

“任务执行时间监视功能”中显示的任务执行时间平均值为 10 次任务执行时间的移动平均值。

## 5-9-2 任务设计示例

本节以 1 个主固定周期任务（单任务）构成时为例，介绍设计步骤。

但是，因实际的应用和各种条件不同，设计步骤或必须考虑的要素会发生变化，所以仅供参考。

**1** 根据装置规格计算必要系统的各输入输出响应时间。

**2** 根据系统的各输入输出响应时间，决定主固定周期任务的“任务周期”。

**3** 首先确认是否在“任务执行时间”所确定的“任务周期”内。

上机估算任务执行时间的预计平均值。

**4** 在实际控制器上，确认任务执行时间是否在任务周期内。

将 Sysmac Studio 与实际控制器在线连接，利用“任务执行时间监视功能”，验证任务执行时间。

● **1 个主固定周期任务（单任务）无法控制在设定的任务周期内时，请从以下角度出发，研究对固定周期任务的任务分配。**

- 为了缩短任务执行时间，应使用程序执行指令（PrgStart）及程序停止指令（PrgStop），仅在需要主固定周期任务中分配的程序时执行。
- 实施上述处理后，仍超出主固定周期任务的任务周期时，将主固定周期任务的处理中，不需要高速、高精度处理的部分分配到固定周期任务（执行优先度 16）中。  
但是，在主固定周期任务和固定周期任务（执行优先度 16）之间，如果要参照轴参数等数据，主固定周期任务的任务执行时间可能会变长。详情请参考□□《NY 系列 工业平板电脑 / 工业 BOX PC 用户手册 运动控制篇（SBCE-CN5-379）》。
- 执行上述处理后，仍超出主固定周期任务的任务周期时，请修改主固定周期任务的任务周期。

### 5-9-3 系统输入输出响应时间

系统中从输入到输出为止的时间如下所示。

根据不同情况而不同。

此外，实际情况下，这个时间包括从站 / 单元与外部的输入响应时间、与外部的输出响应时间。

#### 对基本 I/O 单元进行顺序控制时

基本 I/O 单元通过分配了 I/O 刷新的任务与外部进行 I/O 刷新。

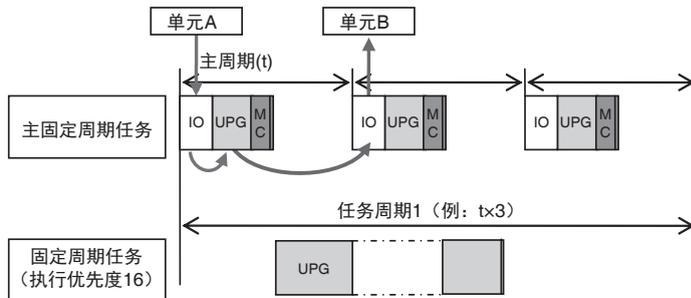
包含 EtherCAT 通信的输入输出响应时间如下所示。

##### ● 用主固定周期任务的用户程序控制时

按以下输入输出响应时间响应。

最小输入输出响应时间 = “主周期”

例) 用主固定周期任务控制单元 A、单元 B 时



注. 上图只记述1次输入输出。

但是，在某些单元的输入时序下，输入输出响应时间如下。

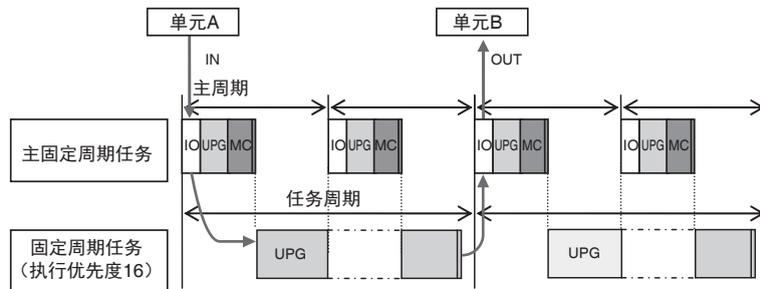
最小输入输出响应时间 = “主周期” × 2

##### ● 用固定周期任务（执行优先级 16）的用户程序控制时

按以下输入输出响应时间响应。

最小输入输出响应时间 = “固定周期任务（执行优先级 16）的周期”

例) 用固定周期任务（执行优先级 16）控制单元 A、单元 B 时



注. 上图只记述1次输入输出。

但是，在某些单元的输入时序下，输入输出响应时间如下。

最大输入输出响应时间 = “固定周期任务（执行优先级 16）的周期” × 2

## 对 EtherCAT 从站进行顺序控制时

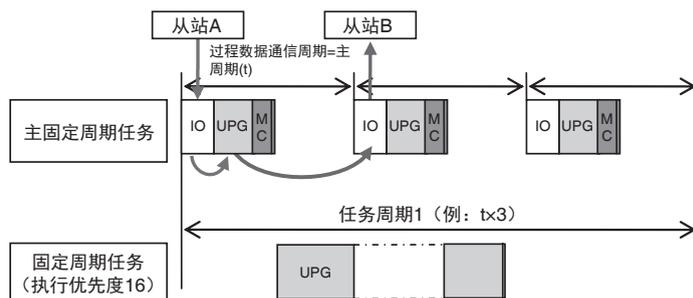
EtherCAT 从站通过主固定周期任务的 I/O 刷新与外部进行 EtherCAT 通信。  
包含 EtherCAT 通信的输入输出响应时间如下所示。

### ● 用主固定周期任务的用户程序控制时

按以下输入输出响应时间响应。

最小输入输出响应时间 = “主周期 (= 过程数据通信周期)”

例) 用主固定周期任务控制 EtherCAT IN 从站 A、EtherCAT OUT 从站 B 时



注. 上图只记述1次输入输出。

但是, 在某些从站的输入时序下, 输入输出响应时间如下。

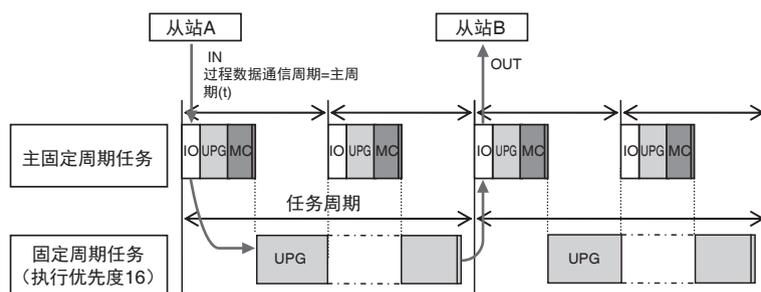
最大输入输出响应时间 = “主周期 (= 过程数据通信周期)” × 2

### ● 用固定周期任务 (执行优先级 16) 的用户程序控制时

按以下输入输出响应时间响应。

输入输出响应时间 = “固定周期任务 (执行优先级 16) 的周期”

例) 用固定周期任务 (执行优先级 16) 控制 EtherCAT IN 从站 A、EtherCAT OUT 从站 B 时



注. 上图只记述1次输入输出。

但是, 在某些从站的输入时序下, 输入输出响应时间如下。

最大输入输出响应时间 = “固定周期任务 (执行优先级 16) 的周期” × 2

## 用运动控制指令执行运动控制时

运动控制指令对已分配到“轴”的伺服/编码器输入从站进行访问。

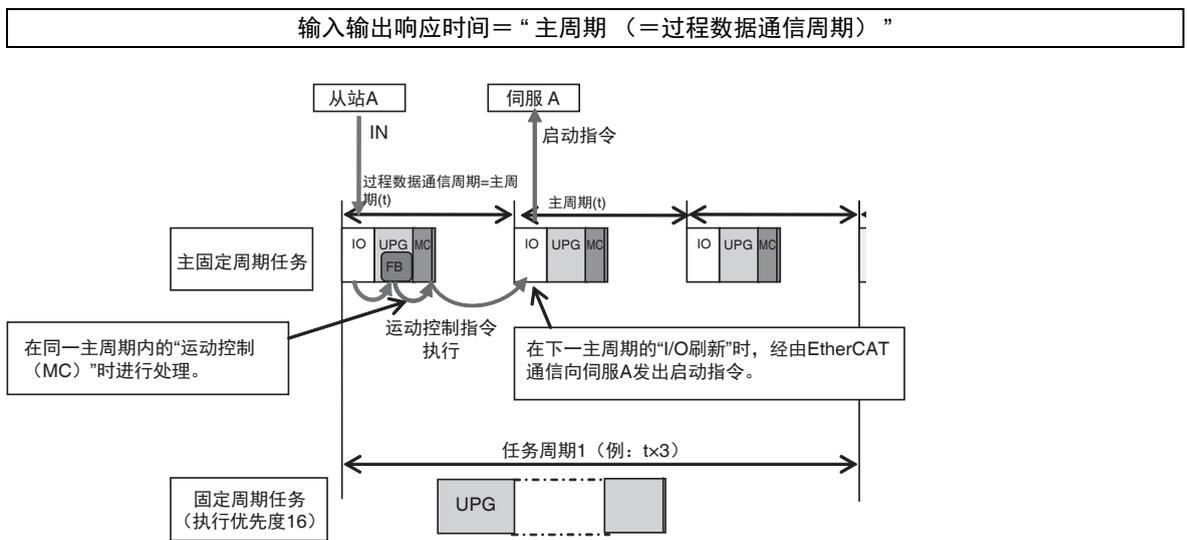
NY 系列控制器的运动控制指令可在主固定周期任务或固定周期任务（执行优先级 16）中记述。无论是哪个任务，运动控制指令均通过主固定周期任务中的“运动控制（MC）”执行。

包含 EtherCAT 通信的输入输出响应时间如下所示。

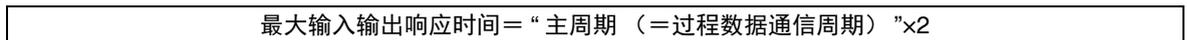
### ● 在主固定周期任务中记述了运动控制指令时

运动控制指令的处理通过主固定周期任务中紧接的“运动控制（MC）”执行。处理结果在下一主固定周期任务的“I/O 刷新”时，经由 EtherCAT 通信输出到 [ 轴 ] 中分配的伺服驱动器中。

按以下输入输出响应时间响应。



但是，在某些从站的输入时序下，输入输出响应时间如下。

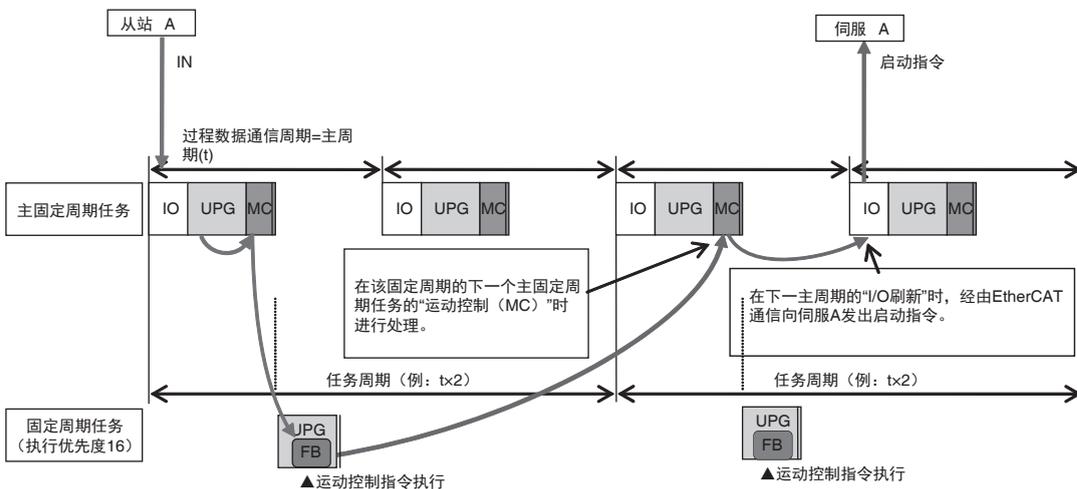


### ● 在固定周期任务（执行优先级 16）中记述了运动控制指令时

运动控制指令的处理通过固定周期任务（执行优先级 16）的下一主固定周期任务中的“运动控制（MC）”执行。处理结果在下一主固定周期任务的“I/O 刷新”时，经由 EtherCAT 通信输出到 [ 轴 ] 中分配的伺服驱动器中。

因此，无论运动控制指令的执行时序如何，都按以下输入输出响应时间响应。

$$\text{最小 I/O 响应时间} = \text{“固定周期任务（执行优先级 16）的周期} + \text{“主周期（=过程数据通信周期）”}$$



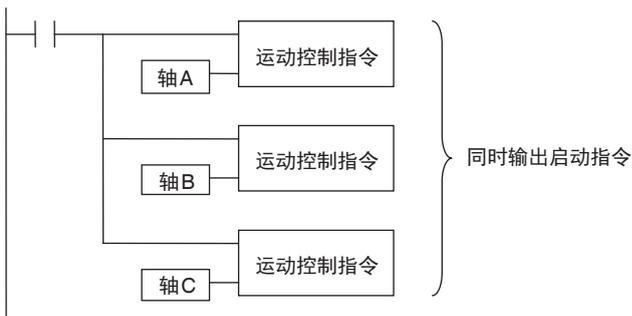
注. 上图只记述1次输入输出。

但是，在某些从站的输入时序下，响应时间如下。

$$\text{最大 I/O 响应时间} = \text{“固定周期任务（执行优先级 16）的周期} + \text{“主周期（=过程数据通信周期）”} \times 2$$

### ● 多个轴的同时启动

用同一任务中的用户程序在同一任务周期内启动多个轴时，可以同时启动。



#### 参考

在控制轴的任务以外的任务中，可以参照轴变量的值。

使用方法的详情及注意事项请参考《NY 系列 工业平板电脑 / 工业 BOX PC 用户手册 运动控制篇（SBCE-CN5-379）》。



# 6

## 编程

本章介绍程序的详情，如程序的种类、构成程序的变量、指令等。

<b>6-1 编程步骤概要</b> .....	<b>6-3</b>
<b>6-2 POU (程序构成单位)</b> .....	<b>6-5</b>
6-2-1 POU 定义 .....	6-5
6-2-2 三个类型 POU 的概要 .....	6-6
6-2-3 程序、功能块、函数的区别 .....	6-7
6-2-4 程序的详情 .....	6-8
6-2-5 功能块 (FB) 的详情 .....	6-9
6-2-6 函数 (FUN) 的详情 .....	6-17
6-2-7 功能块 (FB)、函数 (FUN) 共通动作 .....	6-23
6-2-8 POU 的限制事项 .....	6-25
<b>6-3 变量</b> .....	<b>6-27</b>
6-3-1 变量定义 .....	6-27
6-3-2 变量的种类 .....	6-27
6-3-3 POU 对应的用户定义变量的种类 .....	6-28
6-3-4 变量的属性 .....	6-29
6-3-5 数据类型 .....	6-30
6-3-6 派生数据类型 .....	6-40
6-3-7 数据类型的排列 / 范围指定 .....	6-50
6-3-8 变量的各属性说明 .....	6-56
6-3-9 与状态变化对应的变量值的变化 .....	6-62
6-3-10 功能块 (FB) 的实例 .....	6-72
6-3-11 变量值的监视 .....	6-72
6-3-12 变量名称等程序相关名称的限制 .....	6-73
<b>6-4 常数 (文本)</b> .....	<b>6-75</b>
6-4-1 常数定义 .....	6-75
6-4-2 各数据类型的记述形式 .....	6-75
<b>6-5 编程语言</b> .....	<b>6-80</b>
6-5-1 编程语言定义 .....	6-80
6-5-2 梯形图语言 (LD) .....	6-80
6-5-3 ST 语言 .....	6-86
<b>6-6 指令</b> .....	<b>6-124</b>
6-6-1 指令定义 .....	6-124
6-6-2 指令的基本知识 .....	6-124
6-6-3 指令错误 .....	6-127

<b>6-7 名称空间</b> .....	<b>6-133</b>
6-7-1 名称空间定义 .....	6-133
6-7-2 名称空间的规格 .....	6-134
6-7-3 名称空间的使用步骤 .....	6-137
<b>6-8 库</b> .....	<b>6-138</b>
6-8-1 库的概要 .....	6-138
6-8-2 库的规格 .....	6-139
6-8-3 库部件的规格 .....	6-140
6-8-4 库的使用步骤 .....	6-141
<b>6-9 创建用户程序时的注意事项</b> .....	<b>6-142</b>
6-9-1 输入变量 / 输出变量 / 输入输出变量为排列指定的变量时 .....	6-142
6-9-2 输入变量 / 输出变量 / 输入输出变量为结构体型变量时 .....	6-142
6-9-3 主控 .....	6-143

## 6-1 编程步骤概要

下面介绍编程步骤的概要。  
在整体步骤中，与以下阴影标示的位置对应。



详情请参考□□「1-4 NY 系列的整体使用步骤」(P.1-10)。

设计 POU（处理的分割单位）	参考
<ul style="list-style-type: none"> <li>● 研究和设计用哪个 POU 执行各种处理</li> </ul> 注：函数中不可记述功能块型指令和功能块。	□□「6-2 POU（程序构成单位）」（P.6-5）
<ul style="list-style-type: none"> <li>● 设计将哪个处理用梯形图、ST、联机 ST 中的哪种语言创建</li> </ul> 注：“联机 ST”为梯形图中描述 ST 语言的梯形图语言的要素	□□「6-5 编程语言」（P.6-80）



设计变量	参考
<ul style="list-style-type: none"> <li>● 设计创建哪种用户定义变量</li> </ul>	□□「6-3-1 变量定义」（P.6-27） □□「6-3-2 变量的种类」（P.6-27）
<ul style="list-style-type: none"> <li>● 区分 POU 共同使用的变量（全局变量）和各 POU 单独使用的变量（本地变量）</li> </ul>	□□「6-3-3 POU 对应的用户定义变量的种类」（P.6-28）
<ul style="list-style-type: none"> <li>● 设计是自动生成用于访问从站 / 单元的设备变量之变量名称还是由用户定义</li> </ul>	□□「3-3 I/O 端口和设备变量」（P.3-5）
<ul style="list-style-type: none"> <li>● 设计变量的以下属性</li> </ul> 设定变量名称、数据类型、AT（分配目标）指定、初始值、保持、常量、网络公开等变量的数据类型（排列指定、范围指定、结构体型、列举型）	□□「6-3-4 变量的属性」（P.6-29） □□「6-3-5 数据类型」（P.6-30） □□「6-3-6 派生数据类型」（P.6-40）
<ul style="list-style-type: none"> <li>● 设计变量时，请注意以下事项。               <ul style="list-style-type: none"> <li>• 保持的设计： 接通电源时、变更动作模式时，根据变量值的设定，设计保持属性。</li> <li>• 结构体型的设计： 在所用指令的变量中使用结构体数据类型时，设计为在输入、输出或输入输出参数中使用该结构体数据类型。 例) 通信指令</li> <li>• 排列指定的设计： 在所用指令的变量中使用排列指定的变量时，设计为在输入、输出或输入输出参数中使用排列指定的变量。 例) 移位指令、堆叠 / 表格指令</li> <li>• AT 指定的设计：                   <ul style="list-style-type: none"> <li>• 对部分指令的输入参数变量进行 AT 指定。 例) 自由分配的 I/O 分配</li> </ul> </li> <li>• 网络公开的设计： 设计进行 EtherNet/IP 标签数据链接的变量。</li> </ul> </li> </ul>	□□「6-3-4 变量的属性」（P.6-29） □□「6-3-5 数据类型」（P.6-30） □□「6-3-6 派生数据类型」（P.6-40）

## 6-2 POU（程序构成单位）

NY 系列控制器上动作的用户程序全部以 POU（程序构成单位）的形式创建，然后将它们组合起来使用。下面详细介绍 POU 构成、规格等。

在 Sysmac Studio 上创建 POU 的方法请参考 □□《Sysmac Studio Version 1 操作手册（SBCA-CN5-362）》。

### 6-2-1 POU 定义

POU（Program Organization Unit）是指 IEC 61131-3 中用户程序执行模式的单位。作为构成用户程序的基本要素，对本地变量表和算法（一系列代码 / 逻辑）进行记述。

组合多个 POU，构成整个用户程序。

POU 的构成要素有以下三种。

- 程序  
相当于主例程的、记载有算法的主要要素。  
算法中可记述所有指令、函数、功能块。
- 功能块（以下简称“FB”）  
即使输入值相同，也可输出不同值的要素。通过程序或其他 FB 的调用来启动。
- 函数（以下简称“FUN”）  
如果输入值相同，即可始终输出相同值的要素。通过程序、FB 或 FUN 的调用来启动。

POU 由以上三个要素组合构成。此外，还可创建多个 POU。  
创建的用户程序将分配到任务后启动。

## 6-2-2 三个类型 POU 的概要

### 程序定义

#### ● 启动方法 / 执行条件

- 通过启动任务，启动该任务中分配的程序。
- 始终执行。

#### ● 记述

- POU 中必须记述 1 个以上的程序。1 个任务的程序数最多为 128 个。

### 功能块 (FB) 定义

#### ● 启动方法 / 执行条件

- 通过程序、FB 的调用来启动。
- 始终执行。
- 若要只在条件成立时执行，请定义表示执行条件的输入变量。

#### ● 记述

- 算法中可记述所有指令、用户定义 FUN 及用户定义 FB。
- 可保持内部变量的值。因此，如计时器、计数器等，拥有状态。
- 包括用户定义的功能块和系统定义的功能块。  
用户定义的 FB 称为“用户定义 FB”。系统定义的 FB 称为“FB 型指令”。

FB 的详情请参考□□「6-2-5 功能块 (FB) 的详情」(P.6-9)。

### 函数 (FUN) 定义

#### ● 启动方法 / 执行条件

- 通过程序、FB 或其他 FUN 的调用来启动。
- 执行条件用输入变量 (EN) 指定。EN 为 TURE 时执行一次。

#### ● 记述

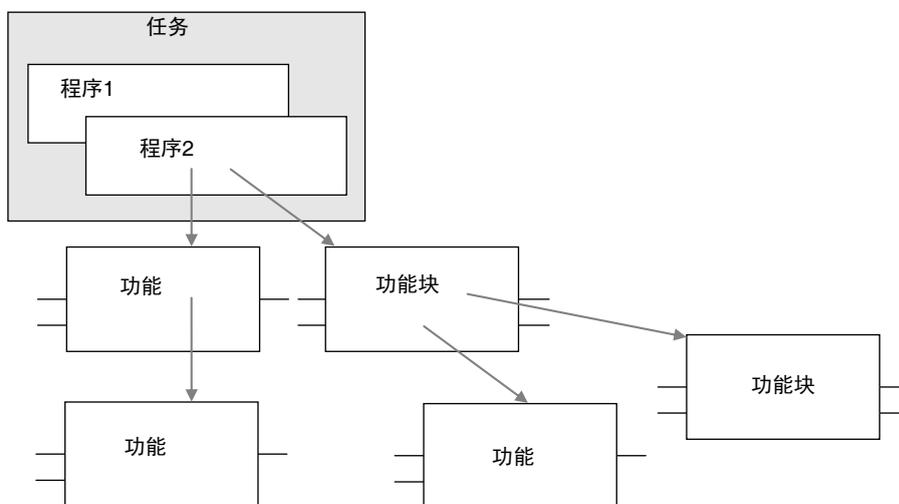
- 算法中不可记述 FB 型指令及用户定义 FB。
- 不可保持内部变量的值。因此，输入值相同时，输出值始终保持一致。
- 包括用户定义的功能块和系统定义的功能块。  
用户定义的 FUN 称为“用户定义 FUN”。系统定义的 FUN 称为“FUN 型指令”。

FUN 的详情请参考□□「6-2-6 函数 (FUN) 的详情」(P.6-17)。

## 6-2-3 程序、功能块、函数的区别

POU 种类		程序	功能块 (FB)	函数 (FUN)
项目				
启动方法		通过启动分配的任务来启动	程序、其他 FB 的调用	程序、FB、FUN 的调用
算法内	所有指令	支持	支持	不支持
	用户定义 FUN	支持	支持	支持
	用户定义 FB	支持	支持	不支持
执行条件		始终执行	始终执行 执行条件用输入变量指定	通过 EN 输入指定

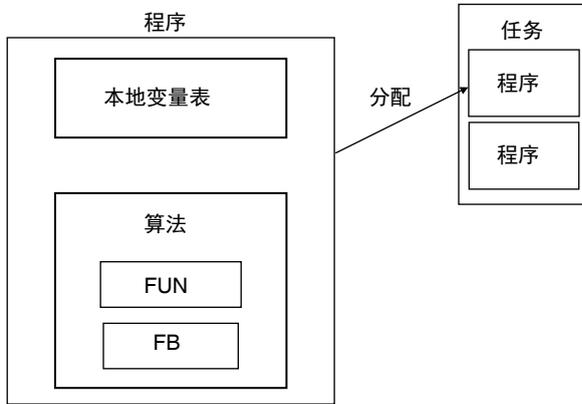
程序、FB、FUN 的层级关系如下图所示。



## 6-2-4 程序的详情

### 程序的构成

程序由本地变量表和算法部分构成。  
算法部分中可记述 FUN 或 FB。



程序不会通过其他 POU 调用。

### 程序的启动条件

通过启动分配了程序的任务来启动程序。

#### ● 执行顺序

对于任务中分配的 program，可设定在任务中的执行顺序。在 Sysmac Studio 的 [ 任务设定 ]-[ 程序的分配设定 ] 中指定。

#### ● 相关系统定义变量

程序分配包括以下本地变量的系统定义变量。

变量名称	名称	功能	数据类型	R/W
P_First_Run Mode	运行开始时 1 个周期 ON 标志	控制器的动作模式从程序模式变更为运行模式时，若该程序正在执行，则在 1 个任务周期内为 TRUE。 若该程序不是正在执行，则保持 FALSE。 用于在控制器开始运行时进行初始处理等情况。	BOOL	R
P_First_Run	程序启动时 1 个周期 ON 标志	该程序开始执行后，仅在 1 个任务周期内为 TRUE。 <sup>*1</sup> 用于每次开始执行程序时需要进行初始处理的情况。	BOOL	R
P_PRGER	指令错误发生标志	相应的程序或通过该程序调用的 FB 或 FUN 发生指令错误时，持续变为 TRUE 的标志。一旦变为 TRUE，在通过用户程序变为 FALSE 前，将保持 TRUE。	BOOL	RW
P_CY	进位标志	在部分指令中更新的标志。	BOOL	R

\*1. 程序的执行 / 停止指令中使用 PrgStart 指令 / PrgStop 指令。PrgStart 指令可设定为 P\_First\_Run 无需变为 TRUE，即可执行程序。

## 6-2-5 功能块 (FB) 的详情

### 功能块 (FB) 的创建步骤

FB 由预先创建的“FB 定义”和实际在程序中使用 FB 所需的“实例”构成。

FB 按以下步骤创建。

#### 1 创建 FB 定义

创建算法。

#### 2 将 FB 定义作为实例配置到程序中

通过程序或其他 FB 调用 FB 定义并实际配置。

1 个 FB 定义可通过多个程序或其他 FB 调用并配置。

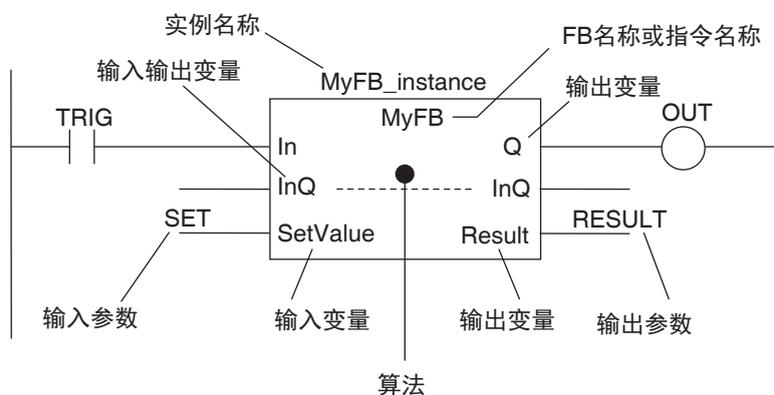
FB 定义在配置到程序或其他 FB 后，将作为实例进行单独处理，可在程序或其他 FB 上进行实际动作。

### 功能块 (FB) 的构成

梯形图语言中，FB 用以下长方形表述。ST 语言中的表述请参考后述的「用 ST 语言调用功能块 (FB) 的表述」(P.6-11)。

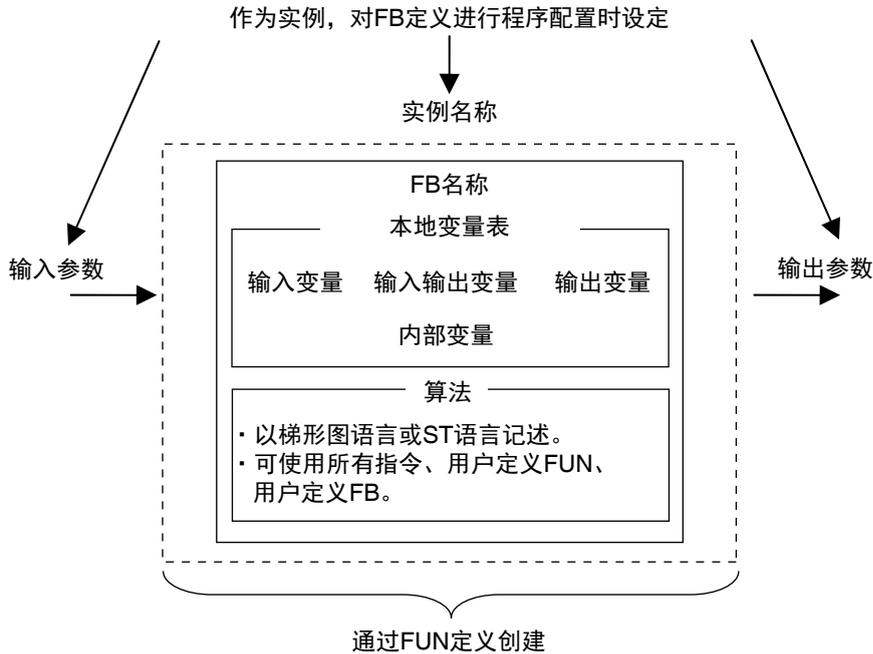
FB 由以下内容构成。

- 用梯形图语言表述的 FB:



- FB 设定的项目

在 FB 定义和实例中配置时，分别进行如下设定。



- **FB 名称或指令名称**

创建时命名的 FB 定义名称或指令名称。

- **实例名称**

为管理 FB，对各程序中配置的 FB 单独命名的名称。  
通过程序或其他 FB 调用 FB 时，设定实例名称。

- **算法**

可以用梯形图语言或 ST 语言记述。  
算法部分中可记述所有指令、用户定义 FUN 及用户定义 FB。

- **本地变量表**

定义输入变量、输出变量、输入输出变量、内部变量、外部变量。  
请参考□□「功能块（FB）内变量的规格」（P.6-12）。

- **参数**

**对输入变量的输入参数**

开始执行 FB 时，向 FB 内的输入变量传递值。记述变量或常数。

**输出变量的输出参数**

FB 执行结束时，接收 FB 中输出变量的值。记述变量。

**输入输出变量？输入输出变量间共享的输入输出参数**

在 FB 内变更值。在输入和输出中记述相同的变量。



**参考**

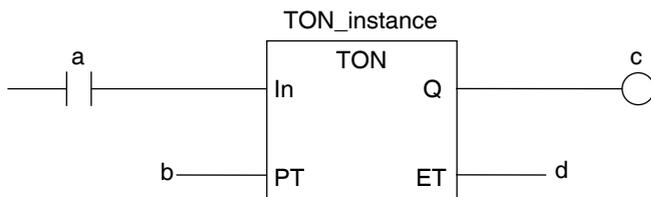
输入参数、输出参数也可省略。动作的详情请参考□□「参数省略时的动作」（P.6-23）。

## 用 ST 语言调用功能块（FB）的表述

ST 语言中，FB 的调用表述如下。

实例名称（输入变量 1:= 输入参数 1, … 输入变量 N:= 输入参数 N, 输入输出变量 1:= 输入输出变量 1, … 输入输出变量 N:= 输入输出参数 N, 输出变量 1=> 输出参数 1, … 输出变量 N=> 输出参数 N）；

按 FB 定义中规定的顺序记述参数，可省略输入变量名称等，只记述参数。  
此时，参数的数量需要与 FB 定义的输入变量的数量一致。



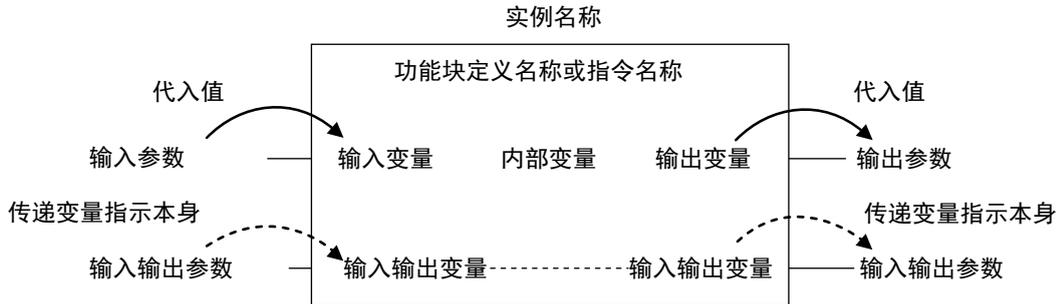
用 ST 语言表述的 FB：

实例名称

```
TON_instance(In:=a, PT:=b, Q=>c, ET=>d);
TON_instance(In:=a, PT:=b, Q=>c); (* 省略了输出ET时 *)
TON_instance(a,b,c,d); (* 省略了输入变量、输出变量时 *)
```

详情请参考后述的 □□「FB 调用」(P.6-111)。

## 功能块（FB）内变量的规格



FB 变量的规格如下所示。

变量	个数	规格
输入变量	1 ~ 64	作为 FB 的输入参数，在 FB 内部使用的（值无法变更）变量。 <ul style="list-style-type: none"> <li>开始执行 FB 时，将输入参数的值设置为输入变量。</li> <li>输入参数中可记述常数或变量。</li> <li>输入参数的省略： 请参考□□「参数省略时的动作」（P.6-23）。</li> <li>可指定上升沿 / 下降沿。</li> <li>可从 FB 外部参照或变更值。在“实例名称. 输入变量名称”中指定。</li> </ul>
输出变量*1	1 ~ 64	作为 FB 的输出参数，在 FB 内部使用的变量。 <ul style="list-style-type: none"> <li>FB 执行结束时，将输出变量的值设置为输出参数。</li> <li>输出参数不可指定为常数或带常量属性的变量。</li> <li>输出参数连接可省略。省略时，输出变量的值将不会代入到任何参数中。</li> <li>可从 FB 外部参照值。可通过“实例名称. 输出变量名称”读取。但是不可写入。</li> </ul>
输入输出变量	0 ~ 64	作为 FB 的输入输出，在 FB 内部使用的（值可变更）变量。 <ul style="list-style-type: none"> <li>传递或接收“输入输出参数”。</li> <li>输入输出参数不可指定为常数或带常量属性的变量。</li> <li>若在 FB 内变更值，输入输出参数的值将在变更时发生变化。</li> <li>不可省略输入输出参数。</li> </ul>
内部变量	无限制	FB 内的临时保存用参数。 <ul style="list-style-type: none"> <li>无论是非执行时还是执行后，都会保持值。</li> <li>可以有保持属性。</li> <li>不可从 FB 外部参照值。</li> </ul>
外部变量	无限制	参照全局变量的变量。
EN	0	FB 中无法使用（用户定义 FB、FB 型指令亦同）。
ENO	0 或 1	作为 BOOL 型输出变量使用，一般在正常结束时变为 TRUE，异常结束或未启动时变为 FALSE。 <ul style="list-style-type: none"> <li>用户定义 FB 中可省略。</li> <li>其他详情请参考后述的□□「ENO」（P.6-20）。</li> </ul>

\*1. 以梯形图语言使用 FB 时，包括 ENO 在内，需要 1 个以上的 BOOL 型输出变量。

各变量的变量属性是否可以设定，请参考□□「6-3-4 变量的属性」（P.6-29）。



### 参考

对于全局变量中外部变量的定义，若用 FB 内的外部变量定义与全局变量同名的变量，将自动设定。

### ● ENO

ENO 为 FALSE 时，其他输出变量的值将保持上次的值。

## 功能块 (FB) 定义和实例

FB 由预先创建的“FB 定义”和实际在程序中使用 FB 所需的“实例”构成。

重复利用的来源称为“FB 定义”。

“FB 定义”中记载有算法和本地变量表。

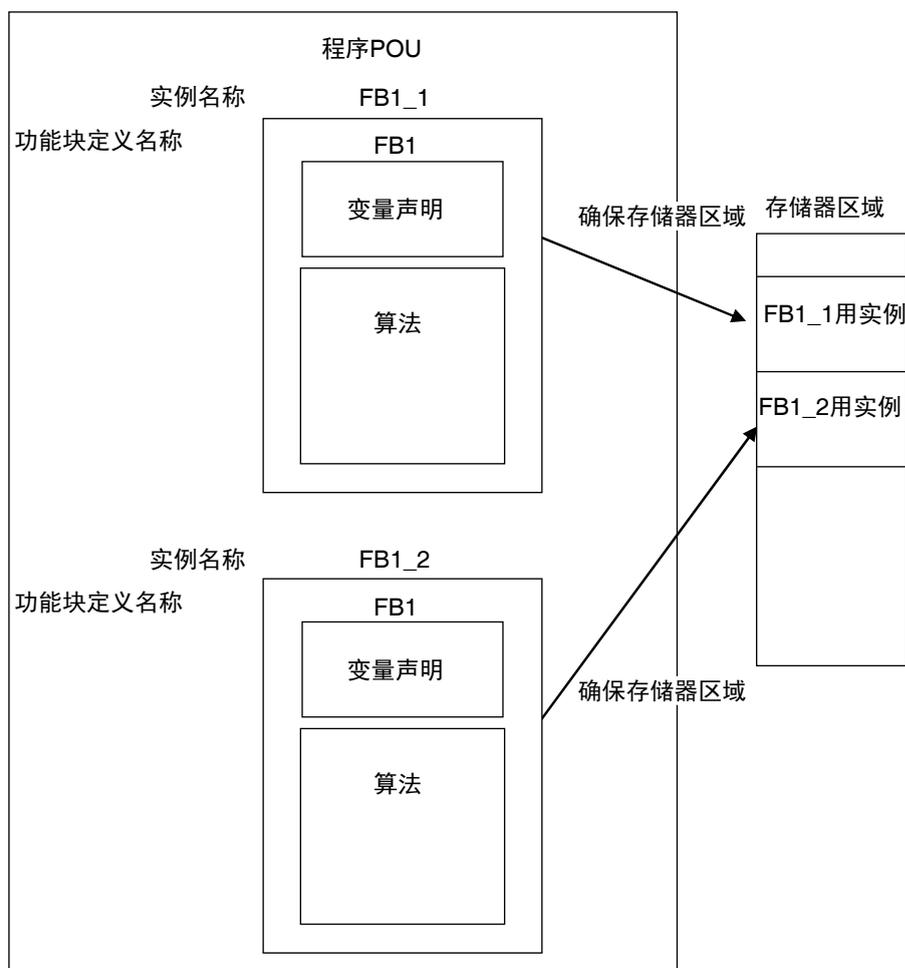
### ● FB 的实例

若将 FB 定义配置到程序或其他 FB 上，FB 定义将作为实际程序或 FB 的一部分处理。

将 FB 定义配置到程序或其他 FB 后，称为“实例”。

各实例拥有实例名称作为识别符，分别使用存储区域。

以 1 个 FB 定义为基础，生成多个“实例”后，可在同一功能中进行不同输入输出数据的处理。



“实例”不可通过不同的程序或其他 FB 读取。实例名称相同，但配置到不同的程序或其他 FB 时，将视为不同的实例进行动作。

### 实例的排列指定化

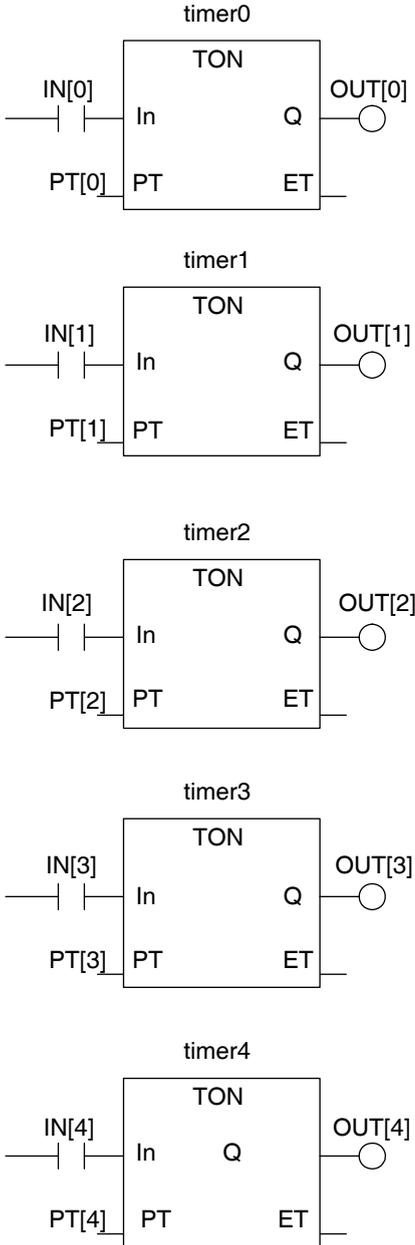
可对实例进行排列指定。

用变量值间接指定要素编号，可用一个实例名称执行多个实例。

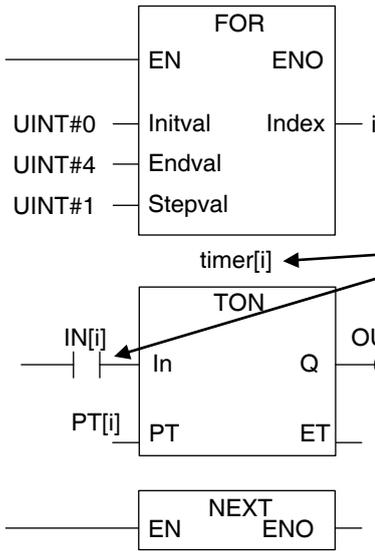
此外，输入参数及输出参数也作为排列指定，用共通的变量指定要素编号，可用一个实例名称切换输入源、输出位置，指定多个实例。

例)

未对实例进行排列指定时



对实例进行了排列指定时



功能块定义TON的实例及输入参数/输出参数，全部作为排列变量，将各要素编号+1移位，并重复执行5次。

变量表

变量名称	数据类型
IN	ARRAY [0..4] OF BOOL
OUT	ARRAY [0..4] OF BOOL
PT	ARRAY [0..4] OF TIME
timer	ARRAY [0..4] OF TON
i	UINT

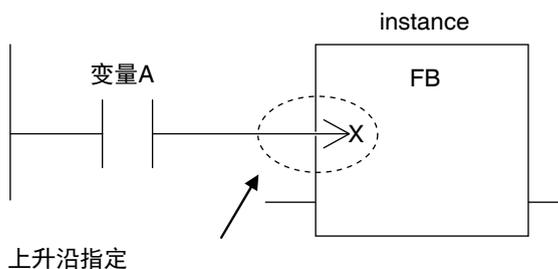
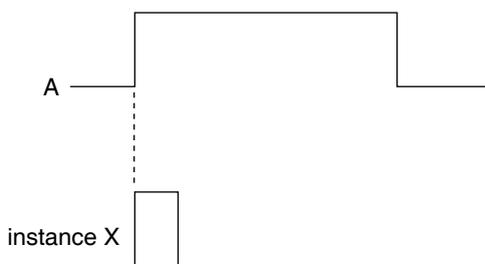
## 功能块 (FB) 的启动条件

FB 与 FUN 不同，没有 EN 输入，始终（每个任务周期）执行。

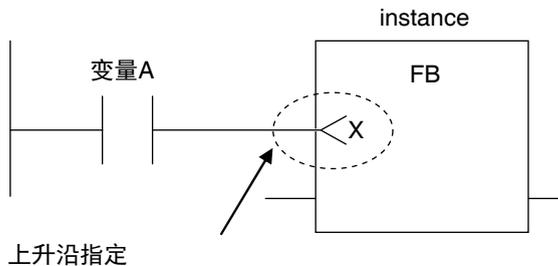
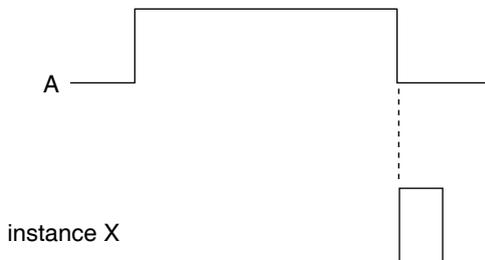
情况	FB 内算法		ENO	ENO 以外的动作
通常时	执行。	正常结束时	TRUE	输出参数： 根据内部算法更新值 输入输出参数： 根据内部算法更新值
		异常结束时	FALSE	输出参数： 保持 输入输出参数： 根据内部算法更新值
主控区域内	在功率流输入为 FALSE 的状态下执行。		用户任意	根据 ENO 的值为上述之一

关于功率流输出、参数输出，请参考□「6-5-2 梯形图语言 (LD)」(P.6-80)。

为输入变量指定边缘，该变量在输入参数的上升沿时变为 TRUE。



也可指定下降沿。



## 从外部参照功能块（FB）的变量

FB 的输入变量及输出变量可从 FB 外部参照。变量的表述如下所示。

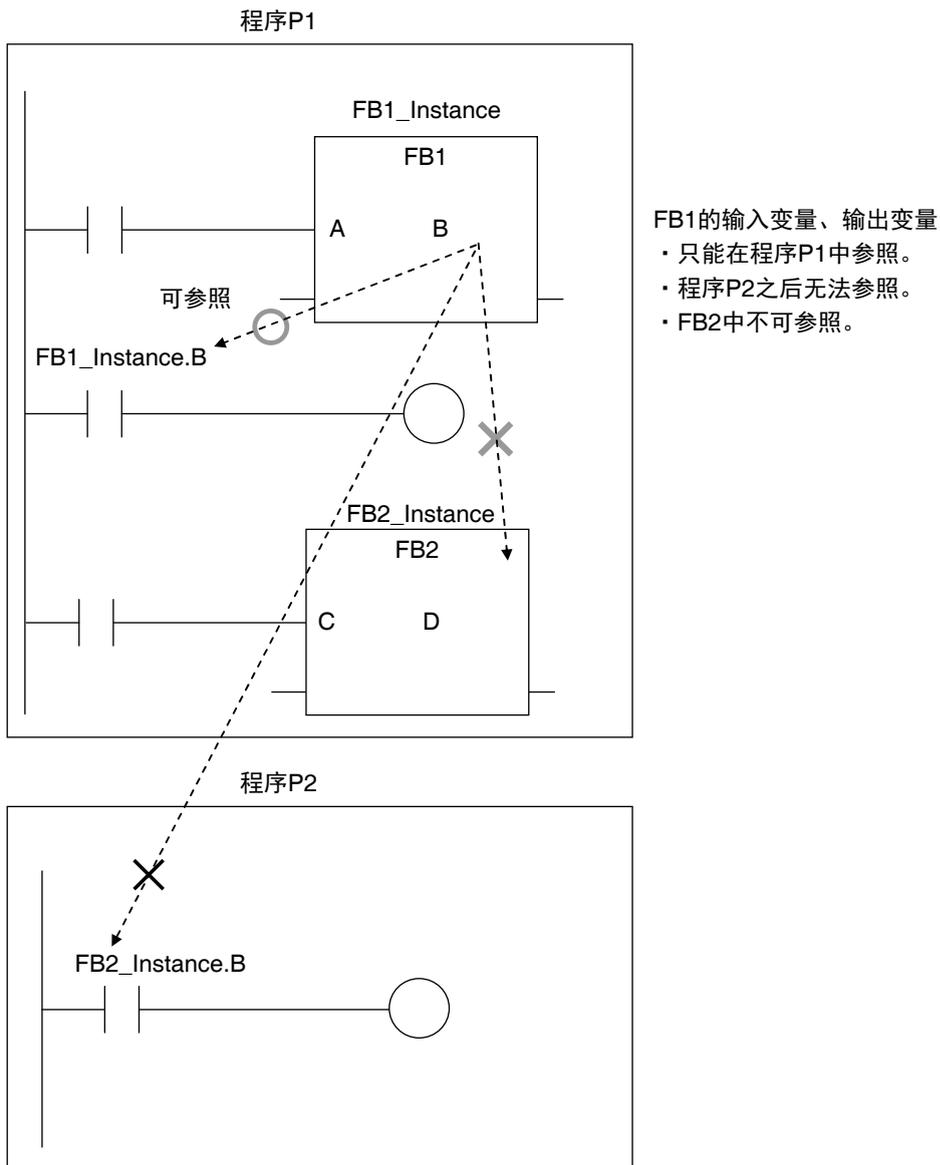
实例名称 . 变量名称

（例）FB1 的实例“FB1\_Instance”内的输出变量“B”时

FB1\_Instance.B

可参照的范围仅限包括该实例的程序内。

不可从该程序中的其他 FB 实例参照。也无法从其他程序参照。



以下参数不可从外部参照。若参照，链接时将发生异常。

- FB 的输入输出变量
- 省略了输入参数时，不应用初始值的 FB 型指令的输入变量

## 6-2-6 函数 (FUN) 的详情

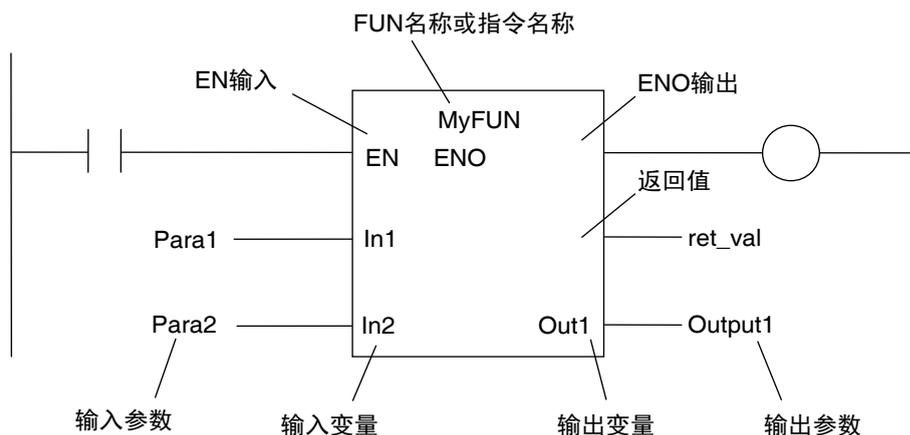
### 函数 (FUN) 的构成

梯形图语言中，FUN 用以下长方形表述。

ST 语言中的表述请参考后述的 □「ST 语言中函数 (FUN) 的表述」(P.6-18)。

FUN 由以下内容构成。

用梯形图语言表述的 FUN：



- **FUN 定义名称或指令名称**

创建时命名的 FUN 定义名称或指令名称。

- **实例名称**

无。

- **算法**

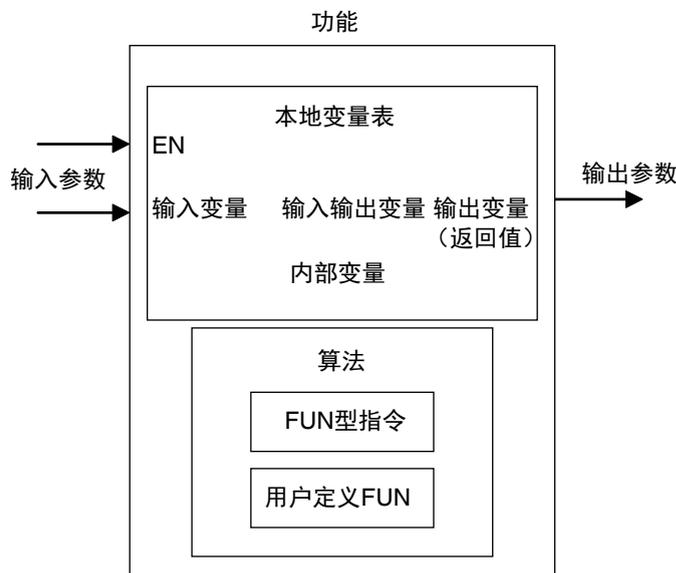
可以用梯形图语言或 ST 语言记述。

算法部分中可记述 FUN 型指令、用户定义 FUN。

不可记述 FB 型指令及用户定义 FB。

不可记述微分指令 (R\_TRIG、UP)。

不可使用用户定义变量 P\_First\_RunMode、P\_First\_Run。



### ● 本地变量表

登录输入变量、输出变量、输入输出变量、内部变量、外部变量。  
请参考□□「函数（FUN）内变量的规格」（P.6-19）。

### ● 参数

#### 对输入变量的输入参数

开始执行 FUN 时，向 FUN 内的输入变量传递值。记述变量或常数。

#### 输出变量的输出参数

FUN 执行结束时，接收 FUN 中输出变量的值。记述变量。

#### 输入输出变量 - 输入输出变量间共享的输入输出参数

在 FUN 内变更值。在输入和输出中记述相同的变量。

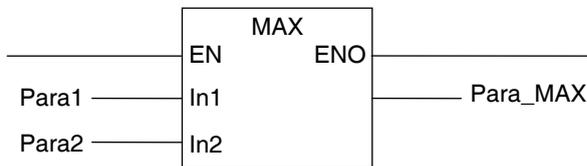
## ST 语言中函数（FUN）的表述

ST 语言中，FUN 表述如下。

返回值 :=FUN 名称（输入变量 1:= 输入参数 1, … 输入变量 N:= 输入参数 N, 输入输出变量 1:= 输入输出变量 1, … 输入输出变量 N:= 输入输出参数 N, 输出变量 1=> 输出参数 1, … 输出变量 N=> 输出参数 N）；  
但是，返回值也可省略。

按 FUN 定义中规定的顺序记述参数，可省略输入变量名称等，只记述参数。此时，参数的数量需要与 FUN 定义的输入变量的数量一致。

用 ST 语言表述的 FUN：

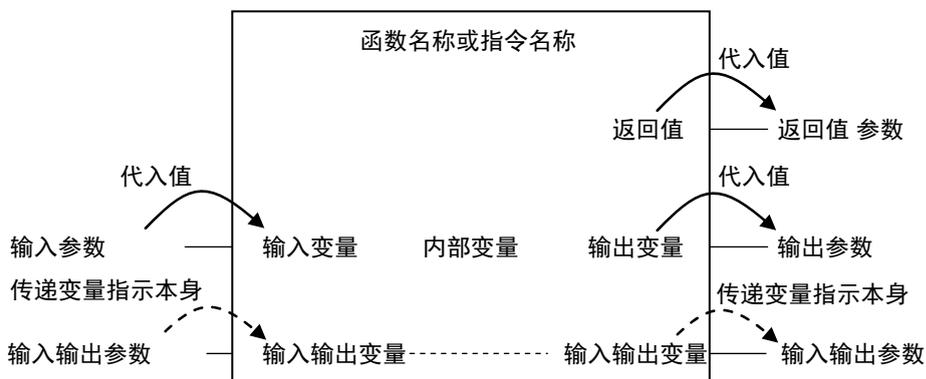


FUN名称

```
Para_MAX := MAX(In1:=Para1, In2:=Para2);
Para_MAX := MAX(Para1, Para2);          (* 省略了输入变量时 *)
```

详情请参考后述的□□「FUN 调用」（P.6-114）。

## 函数（FUN）内变量的规格



FUN 变量的规格如下所示。

变量	个数	规格
输入变量	0 ~ 64	<p>作为 FUN 的输入参数，在 FUN 内部使用的（值无法变更）变量。</p> <ul style="list-style-type: none"> <li>开始执行 FUN 时，将输入参数的值设置为输入变量。</li> <li>输入参数中可记述常数或变量。</li> <li>输入参数的省略： 请参考□□「参数省略时的动作」（P.6-23）。</li> <li>与 FB 不同，不可指定上升沿 / 下降沿。</li> <li>不可从 FUN 外部参照值。</li> <li>欧姆龙提供的指令中包括输入变量的数量可变的指令，但用户创建的 FUN 不可变更输入变量的数量。</li> </ul>
输出变量	0 ~ 64	<p>作为 FUN 的输出参数，在 FUN 内部使用的变量。</p> <ul style="list-style-type: none"> <li>FUN 执行结束时，将输出变量的值设置为输出参数。</li> <li>输出参数不可指定为常数或带常量属性的变量。</li> <li>包括 ENO、返回值在内，至少需要 1 个 BOOL 型输出变量。</li> <li>输出参数连接可省略。 省略时，输出变量的值将不会代入到任何参数中。</li> <li>不可从 FUN 外部参照值。</li> <li>如果是用户定义的 FUN，在 FUN 内的算法中请务必设定输出变量的值。若未在 FUN 内的算法中设定输出变量，输出变量的值将变得不确定。</li> </ul>
输入输出变量	0 ~ 64	<p>作为 FUN 的输入输出，在 FUN 内部使用的（值可变更）变量。</p> <ul style="list-style-type: none"> <li>传递或接收“输入输出参数”本身（变量指示本身）。</li> <li>输入输出参数不可指定为常数或带常量属性的变量。</li> <li>若在 FUN 内变更值，输入输出参数的值将在变更时发生变化。</li> <li>不可省略输入输出参数。</li> <li>不可从 FUN 外部参照值。</li> </ul>
内部变量	无限制	<p>FUN 内的临时保存用参数。</p> <ul style="list-style-type: none"> <li>执行结束后，值不保持。</li> <li>不可从 FUN 外部参照值。</li> </ul>
外部变量	无限制	参照全局变量的变量。
EN	1	<p>启动 FUN 所需的 BOOL 型输入变量。</p> <ul style="list-style-type: none"> <li>EN 的值为 TRUE 时，执行 FUN。</li> <li>必须存在 1 个（用户定义 FUN、FUN 型指令亦同）。</li> </ul>

变量	个数	规格
ENO	0 或 1	作为 BOOL 型输出变量使用，一般在正常结束时变为 TRUE，异常结束或未启动时变为 FALSE。 <ul style="list-style-type: none"> <li>• 用户定义 FUN 中可省略。</li> <li>• 其他详情请参考后述的□□「ENO」(P.6-20)。</li> </ul>
返回值	1	FUN 结束算法，向调用源返回作为处理结果的值。 <ul style="list-style-type: none"> <li>• 必须存在 1 个。</li> <li>• 可指定为所有基本数据类型、枚举型。不可指定为排列、结构体整体、联合体整体。</li> <li>• 如果是用户定义的 FUN，在 FUN 内的算法中请务必设定返回值。若未在 FUN 内的算法中设定返回值，返回值将变得不确定。</li> <li>• 其他详情请参考后述的□□「返回值」(P.6-21)。</li> </ul>

各变量的变量属性是否可以设定，请参考□□「6-3-4 变量的属性」(P.6-29)。



### 参考

在 FUN 内变量表中将全局变量登录为外部变量后，可参照全局变量。

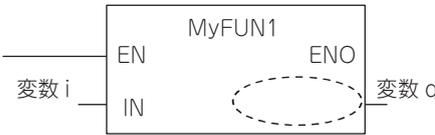
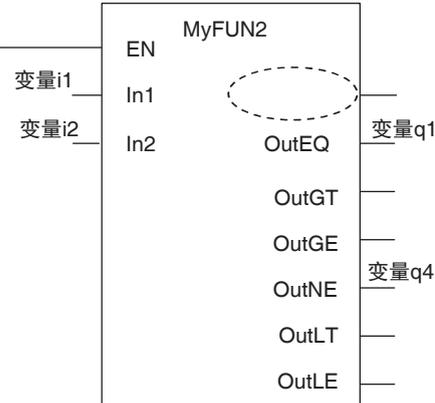
一般情况下，建议创建函数时，确保输入参数的值与输出值一一对应。在函数内参照全局变量，输出值受此影响的算法并不推荐。因此，用 Sysmac Studio 检查程序时，将显示“不推荐使用函数内的全局变量。”的警告。请在确认后采取必要措施。

## ● ENO

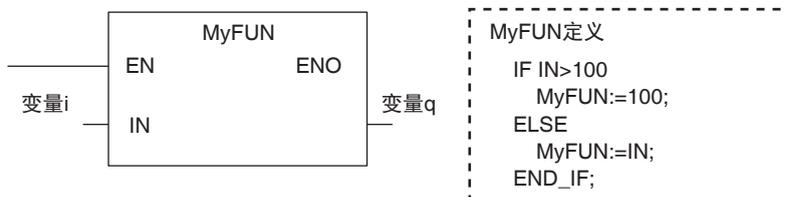
- ENO 为 FALSE 时，其他输出变量的值将保持上次的值。

## ● 返回值

- 在梯形图语言中，返回值显示为空白。

情况	梯形图语言表述	ST 语言表述
使用返回值时		变量 q := MyFUN1 ( 变量 i ) ;
不使用返回值时		MyFUN2 (In1:= 变量 i1, In2:= 变量 i2, OutEQ=> 变量 q1, OutNE=> 变量 q4 ) ;

- 梯形图语言、ST 语言的调用侧都可以不使用返回值。
- 在函数的算法中设置返回值的值时，请将值设置到与函数名称同名的变量中。  
例如，函数为“MyFUN”时，返回值为“MyFUN”。

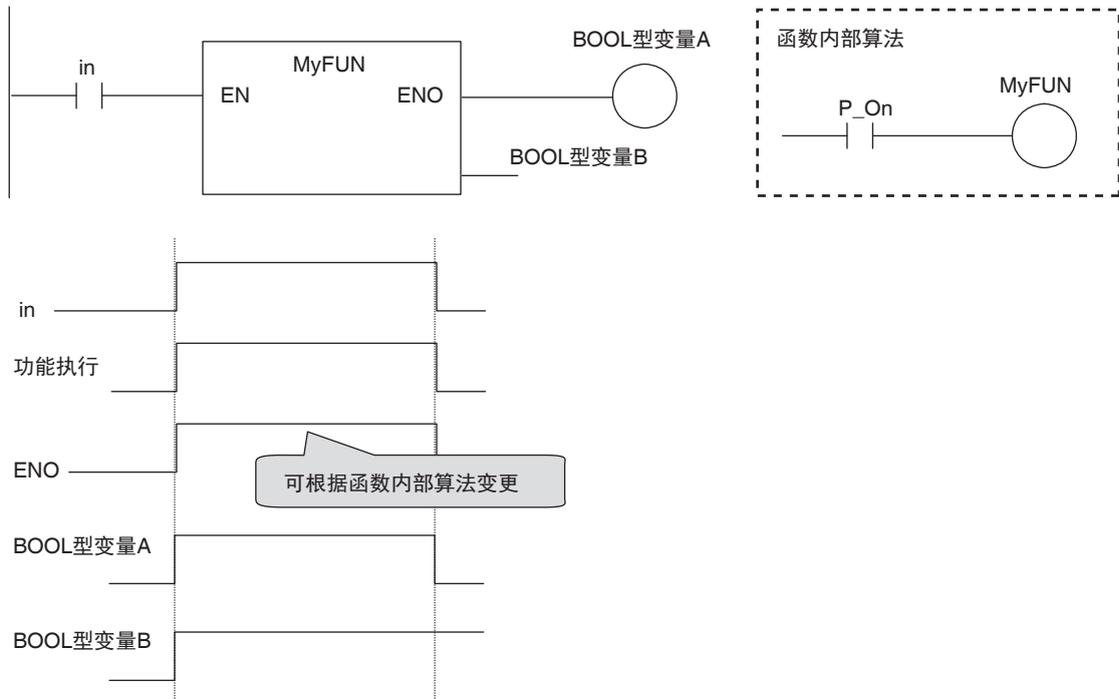


## 函数（FUN）的启动条件

EN 变为 TRUE，则 FUN 启动。EN 变为 FALSE，则 FUN 停止处理。

输入变量	FUN 内算法		ENO	ENO 以外的动作
EN=TRUE 时	执行	正常结束时	TRUE	输出参数： 根据内部算法更新值 输入输出参数： 根据内部算法更新值
		异常结束时	FALSE	输出参数： 保持 输入输出参数： 根据内部算法更新值
EN=FALSE 时	不执行		FALSE	输出参数、输入输出参数： 保持值
主控区域内	不执行		FALSE	输出参数、输入输出参数： 保持值

例)



## 6-2-7 功能块 (FB)、函数 (FUN) 共通动作

## EN、ENO 的使用 / 省略可否

FB/FUN 中 EN、ENO 的使用 / 省略可否情况如下所示。

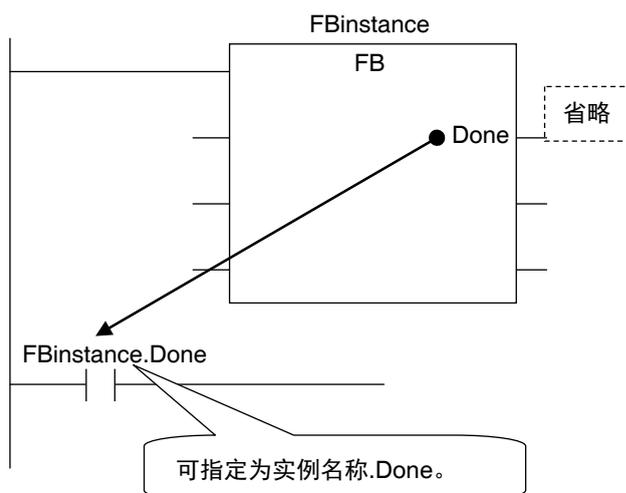
POU		EN	ENO
FB	用户定义	不可使用。 若要通过 Sysmac Studio 定义到变量表中，链接时将发生异常。	可使用（可省略）。 用户通过 Sysmac Studio 定义为输出变量。
	指令	所有 FB 型指令均不使用 EN。	包括使用 ENO 的指令和不使用的指令。
FUN	用户定义	必须使用。 创建 FUN 时，Sysmac Studio 默认自动定义到变量表中。	可使用（可省略）。 用户通过 Sysmac Studio 定义为输出变量。
	指令	所有 FUN 型指令均使用 EN。	包括使用 ENO 的指令和不使用的指令。

## 参数省略时的动作

输入参数、输出参数可以都省略。

省略对象的参数	省略时的动作	
	FB	FUN
对输入变量的输入参数	<ul style="list-style-type: none"> <li>实例首次启动时，按初始值动作。</li> <li>之后按上次值动作（始终省略输入变量时，按初始值继续动作）。</li> </ul>	按初始值动作。
输出变量的输出参数	可省略。 该指令之外，可通过“实例名称. 输出变量名称”访问 (*)	可省略，但省略时无法获取动作结果。
对输入输出变量和来自输入输出变量的输入输出参数	不可省略。	不可省略。

\* FB 的输入变量 / 输出变量可通过“实例名称. 变量名称”从 FB 外部（但仅限该程序）直接参照。FUN 中不可参照。



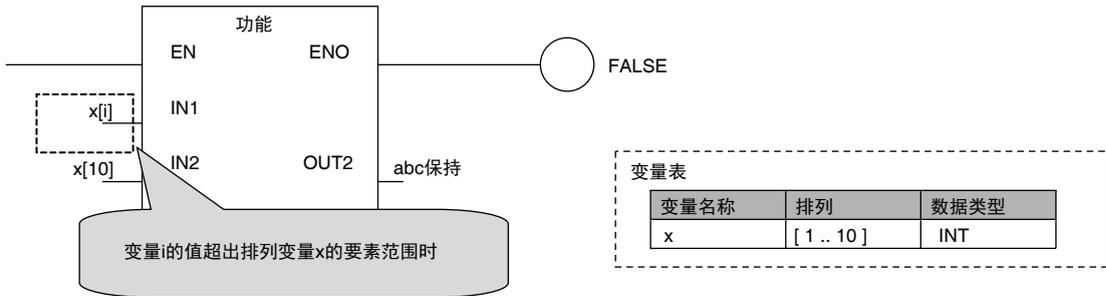
## 参数异常时的动作

输入参数、输出参数或输入输出参数发生异常时，动作如下。

### ● 输入发生异常时

检测到输入参数异常时，将不执行 FB/FUN，ENO 变为 FALSE。同时，功率流输出变为 FALSE，其他保持原来的值。

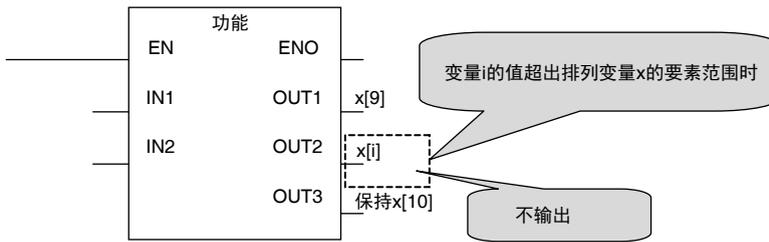
例)



### ● 输出参数发生异常时

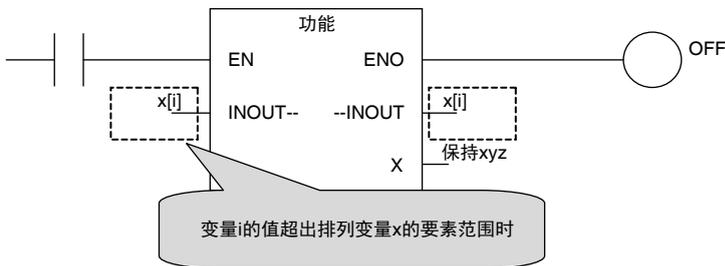
检测到输出参数异常时，该参数之后不输出值，并保持值。

例)



### ● 输入输出发生异常时

检测到输入输出参数异常时，将不执行 FB/FUN，ENO 变为 FALSE。同时，功率流输出变为 FALSE，其他保持原来的值。



## 递归调用

FB 及 FUN 均不可进行以下递归调用。此时，用 Sysmac Studio 链接时将发生异常。

- 自身调用
- 在调用的 FB 或 FUN 上，调用自己本身

## 6-2-8 POU 的限制事项

下面介绍创建 POU 时的各种限制事项。

### 名称

POU 名称、FB 的实例名称限制请参考 □「6-3-12 变量名称等程序相关名称的限制」(P.6-73)。

### 多个参数的传递

向 FB 或 FUN 传递多个参数时，若以排列指定或结构体型传递数据，程序会变得简洁。

此时，与其用输入变量传递，用输入输出变量传递的处理时间更短。但是，使用输入输出变量后，若作为参数传递到 FB 或 FUN 的数据内容发生替换，原来的值不会保留，敬请注意。



#### 参考

#### 将排列指定的变量和结构体型变量的所有成员作为参数时

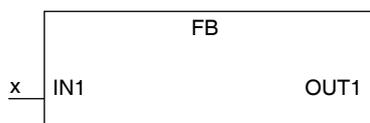
输入参数、输出参数可指定为排列指定的变量或结构体型变量的所有成员。但是，与基本数据型的变量相比，数据的收发时间更长（取决于大小）。

因此，如果是 FB，收发排列指定的变量和结构体型变量的所有成员时，建议设计为对输入输出变量进行收发。

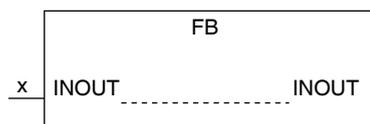
例 1) 指定排列指定的所有要素时

程序的变量表

变量名称	数据类型
x	ARRAY[0..199] OF INT



建议不要将INT型变量x [0..199] 的区域分配到FB输入变量IN1中，  
(FB输入变量IN1的数据类型：ARRAY[0..199] OF INT)

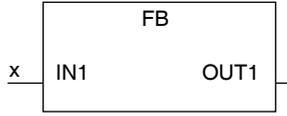


而是将INT型变量x [0..199] 的区域分配到FB输入输出变量INOUT中。  
(FB输入输出变量INOUT的数据类型：ARRAY[0..199] OF INT)

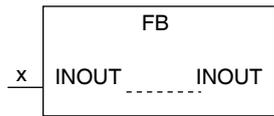
例 2) 指定结构体型变量的所有成员时

程序的变量表

变量名称	数据类型
x	MyStructure



建议不要将MyStructure型变量x的整体成员区域分配到FB输入变量IN1中，（FB输入变量IN1的数据类型：MyStructure）



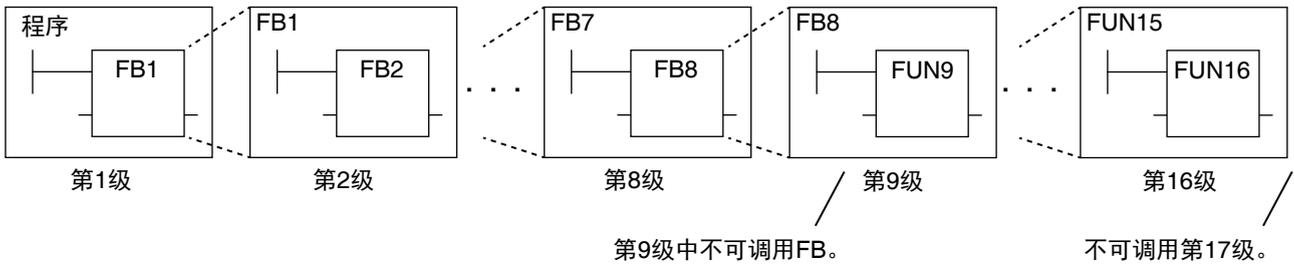
而是将MyStructure型变量x的的整体成员区域分配到FB输入输出变量INOUT中。（FB输入输出变量INOUT的数据类型：MyStructure）

### 嵌套的层级

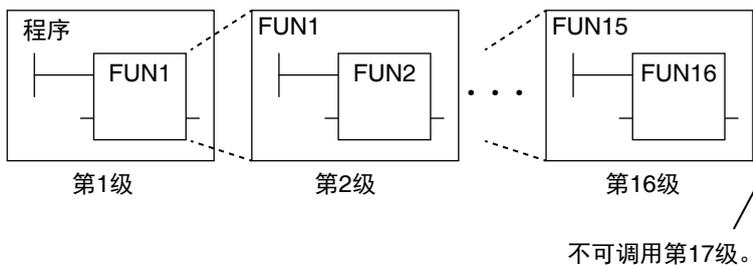
用程序调用的 FB 或 FUN，接着调用其他 FB 或 FUN 的操作称为嵌套。通过用户定义的 FB 或 FUN 可调用的 POU 和嵌套层级的深度有以下限制。若超出限制，链接时将发生异常。

POU	可调用的 POU	层级深度
FB	FB、FUN	最多 8 层
FUN	FUN	最多 16 层

例 1) 可用程序最多调用第 8 层 FB。从这里开始，可以继续调用到第 16 层 FUN。



例 2) 可用程序最多调用第 16 层 FUN。



## 6-3 变量

NY 系列系统中，与外部的输入输出信息交换或数据运算等处理全部通过变量执行。

下面详细介绍变量的规格。

在 Sysmac Studio 上设定变量的方法请参考□□《Sysmac Studio Version 1 操作手册 (SBCA-CN5-362)》。

### 6-3-1 变量定义

变量是指保存与外部进行过交换的输入输出数据和 POU 内部处理时的临时数据的容器。换言之，就是带名称或数据类型等属性的数据之容器。

无需指定变量和存储器地址的分配。NY 系列控制器将自动分配变量所需的存取区域。

### 6-3-2 变量的种类

变量大致可分为以下三种。

- **用户定义变量**

所有属性由用户定义的变量。

本章后面的说明中，将介绍“用户定义变量”。

- **准用户定义变量**

访问特定设备 / 数据所需的变量。

包括设备变量和凸轮数据变量。

设备变量请参考□□「2-3-1 变量的种类」(P.2-10) 及□□「3-3-1 I/O 端口」(P.3-5)。

- **系统定义变量**

名称等所有属性由 NY 系统事先定义，且分配有特定功能的变量。各功能模块分别准备。详情请参考

□□「A-3 系统定义变量一览」(P.A-22)。

关于变量分类的详情，请参考□□「2-3-1 变量的种类」(P.2-10)。

### 6-3-3 POU 对应的用户定义变量的种类

用户定义变量根据在 POU 上的功能大致可分为以下 6 种。

○：可定义

用户定义变量的种类		POU 类型		
		程序	FB	FUN
本地变量	内部变量	○	○	○
	输入变量	无	○	○
	输出变量	无	○	○
	输入输出变量	无	○	○
全局变量		○ (*)	○ (*)	○ (*)
外部变量		○	○	○

\* 将全局变量定义为外部变量后，可通过外部变量，参照全局变量。

#### 本地变量

只能通过 POU（程序、FB、FUN）读写的变量。

POU 为程序时，本地变量等于内部变量。POU 为 FB 或 FUN 时，为内部变量、输入变量、输出变量、输入输出变量、外部变量的总称。

##### ● 内部变量

只在 POU 内部使用的变量。

在 POU 的本地变量表中定义（声明）。

不可从 POU 外部参照内部变量的值。

即使在不同的 POU 中声明相同名称的内部变量，各变量也会分配到不同的分配目标中。

##### ● 输入变量

调用 POU 时，通过调用源代入输入参数值的变量。在 POU 的本地变量表中定义（声明）。

##### ● 输出变量

POU 的处理结束前，在输出参数中代入该值，并返回到调用源的变量。在 POU 的本地变量表中定义（声明）。

##### ● 输入输出变量

调用 POU 时，通过调用源代入输入输出参数本身（变量指示本身）的变量。若在 POU 内替换，输入输出参数的值将在替换时发生变化。

在 POU 的本地变量表中定义（声明）。

##### ● 外部变量

通过 POU 进行外部参照的变量。

通过 POU 参照登录的全局变量。

#### 全局变量

在全局变量表中定义（声明）。

根据从站 / 单元构成自动生成的设备变量以及根据全轴设定表自动生成的“轴 / 轴组变量”将自动登录到全局变量中。

### 6-3-4 变量的属性

变量中可设定或附加以下属性。

#### 变量的属性一览和各变量种类的属性有无

##### ● 变量的属性一览

属性	说明	指定	初始设定
变量名称	用于识别变量的名称。	—	—
数据类型	对变量中保存的数据定义格式。	—	BOOL
AT (分配目标) 指定	将 NX 单元或 EtherCAT 从站的 I/O 端口作为变量时, 指定该变量的分配位置。	<ul style="list-style-type: none"> <li>不指定</li> <li>指定</li> </ul>	不指定
保持	指定在以下任意情况下是否保持值。 <ul style="list-style-type: none"> <li>断电后接通电源时</li> <li>切换到运行模式时</li> <li>发生全部停止故障等级的控制器异常时</li> </ul>	<ul style="list-style-type: none"> <li>保持: 保持左侧情况时的值<sup>*1</sup></li> <li>不保存: 变为初始值</li> </ul>	不保存: 变为初始值
初始值	可选择初始值设定“无”和“有”。 初始值设定为“有”时: 无保持指定, 指定以下任意情况下变量的值。 <ul style="list-style-type: none"> <li>电源接通时</li> <li>动作模式变更时</li> <li>发生全部停止故障等级的控制器异常时</li> </ul> 初始值设定为“无”时: 不保持值	初始值设定 <ul style="list-style-type: none"> <li>有</li> <li>无</li> </ul>	取决于数据类型。 (请参考□□「初始值」(P.6-58)。)
常量	在下载变量时设定初始值, 之后无法替换值。	设为 / 不设为常量	—
网络公开	可使用 CIP 通信或数据链接功能, 从控制器外部读写变量。	<ul style="list-style-type: none"> <li>不公开</li> <li>仅公开</li> <li>输入</li> <li>输出</li> </ul>	不公开
边缘	检测 FB 输入参数的上升沿、下降沿。 仅可设定为 BOOL 型输入变量。	<ul style="list-style-type: none"> <li>无</li> <li>上升沿</li> <li>下降沿</li> </ul>	无

\*1. 将工业电脑连接到 UPS 上, 并正常关机后, 将保存在非易失性存储器中。



##### 参考

##### 任务间排他控制功能

设为只能通过一个任务读写全局变量, 防止在处理过程中全局变量的值发生变化。不是作为变量的属性, 而是在任务设定中指定。

### ● 各变量种类的属性有无

变量种类		变量名称	数据类型	AT (分配目标) 指定	保持	初始值	常量	网络 公开	边缘
全局变量		可设定	可设定	可设定	可保持	可设定	可指定	可公开	不支持
程序	内部变量	可设定	可设定	可设定	可保持	可设定	可指定	不可公开	不支持
	外部变量	不可设定	不可设定	不可设定	不可保持	不可设定	可指定	不可公开	不支持
功能块	内部变量	可设定	可设定	可设定	可保持	可设定	可指定	不可公开	不支持
	输入变量	可设定	可设定	不可设定	可保持	可设定	可指定	不可公开	可以
	输出变量	可设定	可设定	不可设定	可保持	不可设定	不可指定	不可公开	不支持
	输入输出变量	可设定	可设定	不可设定	不可保持	不可设定	可指定	不可公开	不支持
	外部变量	不可设定	不可设定	不可设定	不可保持	不可设定	可指定	不可公开	不支持
功能	内部变量	可设定	可设定	不可设定	不可保持	可设定	可指定	不可公开	不支持
	输入变量	可设定	可设定	不可设定	不可保持	可设定	可指定	不可公开	不支持
	输出变量	可设定	可设定	不可设定	不可保持	不可设定	不可指定	不可公开	不支持
	输入输出变量	可设定	可设定	不可设定	不可保持	不可设定	可指定	不可公开	不支持
	外部变量	不可设定	不可设定	不可设定	不可保持	不可设定	可指定	不可公开	不支持

### 6-3-5 数据类型

数据类型是指规定了表现变量值的形式和范围的属性。

根据数据类型不同，变量声明时确保的存储器大小也不同。存储器大小越大，可表现的值的范围越大。

此外，指令的输入 / 输出 / 输入输出变量的数据类型因各指令而不同。指令的参数——输入 / 输出 / 输入输出参数的数据类型请根据对应的输入 / 输出 / 输入输出变量的数据类型设定。

### 基本数据类型和派生数据类型

数据类型包括已事先规定规格的基本数据类型和由用户定义规格的派生数据类型。

#### ● 基本数据型的种类

基本数据类型可进行以下分类。

分类	定义
布尔型	值为 0 或 1 中任意一个的数据类型。
位列型	以位列表现值的数据类型。
整数型	值为整数值的数据类型。
实型数据	值为实数值的数据类型。
持续时间型	值为时间（日·时·分·秒·毫秒）的数据类型。
时刻型	值为时刻（时·分·秒）的数据类型。
日期型	值为日期（年·月·日）的数据类型。
日期时刻型	值为日期时刻（年·月·日·时·分·秒·毫秒）的数据类型。
字符串型	值为字符串的数据类型。

基本数据类型有 20 种。各规格如下所示。

下表的“数据大小（容量值）”及“校准（临界值）”的含义为

- 数据大小（容量值）：实际值的容量。
- 校准（临界值）：确保持存器时的单位。

分类	数据类型	数据大小 (容量值)	校准 (临界值)	值的范围	记述方法
布尔型	BOOL	16 位	2 字节	FALSE、TRUE	BOOL#0、BOOL#1 FALSE、TRUE
位列型	BYTE	8 位	1 字节	BYTE#16#00 ~ FF	BYTE#2#01011010
	WORD	16 位	2 字节	WORD#16#0000 ~ FFFF	BYTE#2#0101_1010
	DWORD	32 位	4 字节	DWORD#16#00000000 ~ FFFFFFFF	BYTE#16#5A (还可使用分隔符“_”。)
	LWORD	64 位	8 字节	LWORD#16#0000000000000000 ~ FFFFFFFFFFFFFFFF	
整数型 <sup>*1</sup>	SINT	8 位	1 字节	SINT#-128 ~ +127	100
	INT	16 位	2 字节	INT#-32768 ~ +32767	INT#2#00000000_01100100
	DINT	32 位	4 字节	DINT#-2147483648 ~ +2147483647	INT#8#144 INT#10#100 INT#16#64
	LINT	64 位	8 字节	LINT#-9223372036854775808 ~ +9223372036854775807	-100
	USINT	8 位	1 字节	USINT#0 ~ +255	
	UINT	16 位	2 字节	UINT#0 ~ +65535	
	UDINT	32 位	4 字节	UDINT#0 ~ +4294967295	
	ULINT	64 位	8 字节	ULINT#0 ~ +18446744073709551615	
实型数据	REAL	32 位	4 字节	REAL#-3.402823e+38 ~ -1.175495e-38 0 1.175495e-38 ~ 3.402823e+38 + ∞ / - ∞	REAL#3.14 LREAL#3.14 3.14 -3.14 1.0E+6 1.234e4
	LREAL	64 位	8 字节	LREAL#-1.79769313486231e +308 ~ -2.22507385850721e-308 0 2.22507385850721e-308 ~ 1.79769313486231e+308 + ∞ / - ∞	
持续时间 型 <sup>*1*2</sup>	TIME	64 位	8 字节	T#-9223372036854.775808ms (T#-106751d_23h_47m_16s_854. 775808ms) ~ T#+9223372036854.775807ms (T#+106751d_23h_47m_16s_854 .775807ms)	T#12d3h3s T#3s56ms TIME#6d_10m TIME#16d_5h_3m_4s T#12d3.5h T#10.12s T#61m5s (与 T#1h1m5s 相同) TIME#25h_3m
日期型	DATE	64 位	8 字节	D#1970-01-01 ~ D#2106-02-06 (1970 年 1 月 1 日 ~ 2106 年 2 月 6 日)	开头加 'DATE#' 或 'date#'、或 'D#' 或 'd#', 以 '年-月-日' 记述。  (例) d#1994-09-23
时刻型 <sup>*2</sup>	TIME_OF_DAY	64 位	8 字节	TOD#00:00:00.000000000 ~ TOD#23:59:59.999999999 (0 点 0 分 0.000000000 秒 ~ 23 点 59 分 59.999999999 秒)	开头加 'TIME_OF_DAY#' 或 'time_of_day#'、或 'TOD#' 或 'tod #', 以 '时:分:秒' 记述。  (例) tod#12:16:28.12

分类	数据类型	数据大小 (容量值)	校准 (临界值)	值的范围	记述方法
日期时刻 型*2	DATE_ AND_ TIME	64 位	8 字节	DT#1970-01-01-00:00:00.0000000 00 ~ DT#2106-02-06-23:59:59.9999999 99 (1970 年 1 月 1 日 0 点 0 分 0.000000000 秒~ 2106 年 2 月 6 日 23 点 59 分 59.999999999 秒)	开头加 'DT#"dt#', 以 '年 - 月 - 日 - 时 : 分 : 秒' 记述。  (例) dt#1994-09-23-12:16:28.12
字符串型	STRING	(半角字符 数 +1) ×8 位*3	1 字节	字符编码: UTF-8。  0 ~ 1986 字节 (半角英文数字, 0 ~ 1985 字符、日文约 0 ~ 661 字 符)。*4  默认大小为 256 字节。	用 "" (半角单引号) 括起来记述。  (例) 'OMRON' 'PLC'

- \*1. 持续时间型和整数型的转换请使用 NanoSecToTime 指令或 TimeToNanoSec 指令。指令规格的详情请参考 □□《NY 系列 指令基准手册 基本篇 (SBCA-CN5-437)》。
- \*2. 比较指令中变量的比较以 ns 为单位进行。变更比较的单位时, 请使用 TruncTime 指令 / TruncDt 指令 / TruncTod 指令。指令规格的详情请参考 □□《NY 系列 指令基准手册 基本篇 (SBCA-CN5-437)》。
- \*3. 字符串末尾添加 NULL (1 字节) 字符。因此, 请确保要处理的最大字符数 +1 字符的存储空间。例如, 半角英文和数字的最大字符数为 10 时, 请定义为 11 字符 (11 字节) 的 STRING 型变量。STRING[11]
- \*4. 记载制表符或换行代码等时, 记载以 "\$" (半角美元符号) 为前缀的转义字符。关于转义字符, 请参考 □□「字符串型」(P.6-78)。



### 使用注意事项

所有变量需要的存储容量与各变量的数据大小 (容量值) 总和不一致, 敬请注意。  
因为数据配置到存储器的开头位置将自动配置到各数据类型的“校准 (临界值)”倍数的位置, 因此各数据类型间可能出现空白。  
例如, 即使数据类型相同, 因数据类型的顺序不同, 总占用大小也可能不同。

通过结构体型变量等与其他设备进行数据交换时, 各数据类型的“校准 (临界值)”必须与数据位置一致, 需要注意。

详情请参考 □□「A-6 变量的存储器确保方法」(P.A-82)。



### 参考

- 位列数据类型（BYTE、WORD、DWORD、LWORD）不可进行大小比较。进行大小比较时，请使用 WORD\_TO\_UINT 指令等数据类型转换指令，将变量的数据类型转换为整数。

例)

```
BCD_data : WORD
```

```
IF WORD_BCD_TO_UINT (BCD_data) > UINT#1234 THEN
```

- 整数的数据类型（SINT、INT、DINT、LINT、USINT、UINT、UDINT、ULINT）不可进行逻辑运算。进行逻辑运算时，请使用 INT\_TO\_WORD 指令等数据类型转换指令，将变量的数据类型转换为位列。

例)

以下示例程序中，对 INT 型变量 a 的值为奇数时，在 a 上加 1。

```
IF (INT_TO_WORD (a) AND WORD#16#0001) = WORD#16#0001 THEN
```

```
  a = a+1;
```

```
END_IF;
```

## ● 派生数据型的种类

派生数据类型是指由用户定义规格的数据类型。将登录到 Sysmac Studio 的“数据类型”视图中。派生数据类型中包括以下内容。

种类	内容
结构体型	将多个数据类型按层级整合为一个的数据类型。
联合体型	同一数据可根据不同情况作为不同的数据类型处理的数据类型。
列举型	将事先准备的名称列表中的一个作为值的数据类型。

详情请参考后述的□「6-3-6 派生数据类型」（P.6-40）。

## ● 指定数据类型

可对各数据类型进行以下排列指定、范围指定。

种类	内容
排列指定	汇总相同数据类型的要素，从开头起使用编号（后缀）指定。 可指定为基本数据类型及派生数据类型。
范围指定	明确表示只能提取预先设定的范围内的数值。 可指定为基本数据型的整型。

详情请参考后述的□「6-3-7 数据类型的排列 / 范围指定」（P.6-50）。



### 参考

数据类型除了基本数据类型和派生数据类型外，还有 POU 实例型。POU 实例型为 FB 实例的数据类型。要生成 FB 的实例，需要将实例名称作为变量、将 FB 定义名称作为数据类型登录到变量表中。

## 数据类型的使用限制

各 POU 中不可使用的数据类型如下所示。

POU 种类	变量种类	不可使用的数据类型	
		基本数据类型	派生数据类型
程序	内部变量	无	
	全局变量	无	
FUN	输入变量、输出变量、输入输出变量	无	联合体型
	内部变量	无	
	返回值	无	结构体型、联合体型
FB	输入变量、输出变量、输入输出变量	无	联合体型
	内部变量	无	

## 位列型、实型数据、字符串型的数据格式

下面介绍位列型、实型数据、字符串型的数据格式。

### ● 位列型的数据格式

位列型变量的位位置中，最低位为“位 0”。

此外，位值以 1 或 0 表述，但用 BOOL 型变量表述 1 位的值时，1 表示 TRUE，0 表示 FALSE。



### ● 实型数据（REAL 型、LREAL 型）

实型数据（REAL 型、LREAL 型）是指拥有实数型数据格式的数据类型。

下面介绍其表述方法和运算处理。

#### 数据大小

REAL 型的数据大小为 32 位，LREAL 型的数据大小为 64 位。

#### 数据格式

浮动小数点格式是指用符号、指数、尾数表示实数的格式。

用以下公式表示某实数时，s 的值为符号，e 的值为指数，f 的值为尾数。

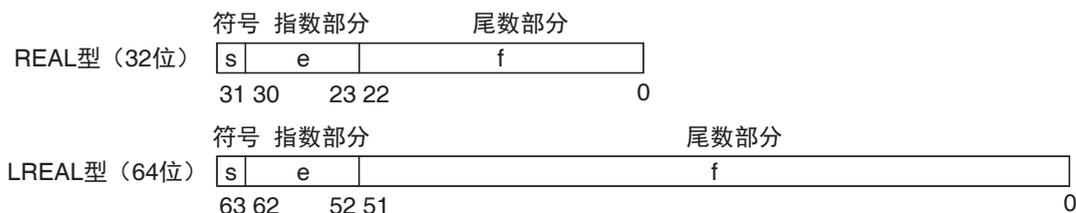
##### • REAL 型

$$\text{数值} = (-1)^s 2^{e-127} (1+f \times 2^{-23})$$

##### • LREAL 型

$$\text{数值} = (-1)^s 2^{e-1023} (1+f \times 2^{-52})$$

浮动小数点格式的数据格式符合 IEEE754 标准。分别如下所示。



（例）REAL 型，表示 -86.625 时

- 1 由于为负数，所以  $s=1$ 。
- 2 若用 2 进制数表示 86.625，则为 1010110.101。
- 3 若将上述值进行标准化，则变为  $1.010110101 \times 2^6$ 。
- 4 根据上述公式， $e-127=6$ ， $e=133$ （2 进制数为 1000101）。
- 5 1.010110101 的小数点以下为 010110101。在此基础上将尾数部 23 位的不足部分用 0 填充，则为 f。因此， $f=01011010100000000000000$ 。

所以，-86.625 可以如下图表示。

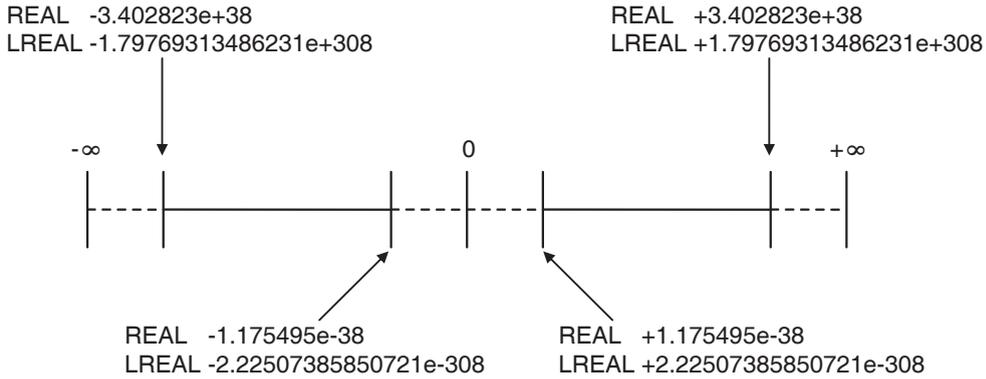


## 有效范围

REAL 型、LREAL 型的有效范围如下表所示。

0 附近存在无法表示的值的范围。

数据类型	$-\infty$	负数	0	正数	$+\infty$
REAL	$-\infty$	$-3.402823\text{e}+38 \sim -1.175495\text{e}-38$	0	$+1.175495\text{e}-38 \sim +3.402823\text{e}+38$	$+\infty$
LREAL	$-\infty$	$-1.79769313486231\text{e}+308 \sim -2.22507385850721\text{e}-308$	0	$+2.22507385850721\text{e}-308 \sim +1.79769313486231\text{e}+308$	$+\infty$



## 特殊值

$+\infty$ 、 $-\infty$ 、 $+0$ 、 $-0$ 、非数值称为特殊值。

非数值是指无法视为浮动小数点数的数据，不作为数值处理。

$+0$  和  $-0$  在数学上都是 0 的意思，但运算处理内容不同。详情信息后面再介绍。

特殊值的符号 s、指数 e、尾数 f 的值如下表所示。

数据类型	特殊值	符号 s	指数 e	尾数 f
REAL	$+\infty$	0	255	0
	$-\infty$	1	255	0
	$+0$	0	0	0
	$-0$	1	0	0
	非数值	-	255	0 以外

数据类型	特殊值	符号 s	指数 e	尾数 f
LREAL	$+\infty$	0	2047	0
	$-\infty$	1	2047	0
	$+0$	0	0	0
	$-0$	1	0	0
	非数值	-	2047	0 以外

## 非标准数

浮动小数点格式无法表示 0 附近（绝对值非常小）的值。为了扩大 0 附近的有效范围而设计了非标准数。与正常数据格式表示的值（称为标准数）相比，非标准数可以表示绝对值较小的值。

指数  $e = 0$  且尾数  $f \neq 0$  的数值将视为非标准数，值的表示如下。

- REAL 型

$$\text{数值} = (-1)^s 2^{-126} (f \times 2^{-23})$$

- LREAL 型

$$\text{数值} = (-1)^s 2^{-1022} (f \times 2^{-52})$$

（例）REAL 型，表示  $0.75 \times 2^{-127}$  时

**1** 由于为正数，所以  $s=0$ 。

**2** 若用 2 进制数表示 0.75，则为 0.11。

**3**  $(0.11)_2 \times 2^{-127} = 2^{-126} (f \times 2^{-23})$ ，所以  $f = (0.11)_2 \times 2^{22}$ 。

**4** 根据以上公式， $f=0110000000000000000000$ 。

所以， $0.75 \times 2^{-127}$  可以如下图表示。



非标准数比标准数的有效位数少。因此，标准数的运算结果为非标准数时，或在运算过程中结果变为非标准数时，运算结果的有效位数可能小于标准数的有效位数。

## 运算处理

浮动小数点格式为近似值表现，因此与真值存在误差。此外，值的有效范围也有限制。因此，运算时，需要进行取整、溢出、下溢的处理。

### 取整

真值超出尾数的有效位数时，按照以下规格取整数。

- 在浮动小数点格式可表现的数值中，将最接近真值的值作为运算结果。
- 在浮动小数点格式可表现的数值中，最接近真值的值有 2 个时（真值为 2 个近似值的中间值时），将尾数最低位的位值有 0 的值作为运算结果。



### 使用注意事项

#### 验证 2 个值是否相等时，对与真值的误差的考虑

实数以浮动小数点格式表现，因此与真值存在误差。验证 2 个值是否相等时，受这个误差影响，可能无法得到预期的结果。

例如，执行  $0.1+0.2$  和  $0.3$  的比较 `boolv := (0.1 + 0.2 = 0.3);` 时，BOOL 型变量 `boolv` 的值不变为 TRUE，而是 FALSE。

为了避免这样的情况，验证 2 个实数的值是否相等时，请勿使用 EQ、=、NE、<>。请使用大小比较指令，验证两者差的绝对值是否在足够小的容许范围内。

例如，验证 REAL 型变量 `real_a` 和 `real_b` 的和是否等于  $0.3$  时，请如下记述。若 `boolv` 的值为 TRUE，判断为两者相等。

```
将 boolv := (ABS((real_a + real_b) - 0.3) < 0.000001); // 0.000001 以内的误差
// 视为容许范围时
```

#### 溢出、下溢

真值的绝对值大于浮动小数点格式可表现的值时，称为溢出。

反之，小于浮动小数点格式可表现的值时，称为下溢。

- 溢出时，如果真值的符号为正，则运算结果为  $+\infty$ 。如果真值的符号为负，则运算结果为  $-\infty$ 。
- 下溢时，如果真值的符号为正，则运算结果为  $+0$ 。如果真值的符号为负，则运算结果为  $-0$ 。

#### 特殊值的运算

特殊值 ( $+\infty$ 、 $-\infty$ 、 $+0$ 、 $-0$ 、非数值) 的运算按以下规则进行。

- $+\infty$  和  $-\infty$  的相加结果为非数值。
- 符号相同的无限大值的相减结果为非数值。
- $+0$  和无限大、 $-0$  和无限大的相乘结果为非数值。
- $+0$  的相除结果、 $-0$  的相除结果、无限大的相除结果为非数值。
- $+0$  和  $-0$  的相加结果为  $+0$ 。
- $+0$  的相减结果、 $-0$  的相减结果为  $+0$ 。
- 包含非数值的四则运算结果为非数值。
- 比较指令 (Cmp 指令等) 中， $+0$  和  $-0$  判断为相等。
- 比较指令中，如果比较对象中包含非数值，将始终判断为不相等。

#### ● 字符串型的数据格式

字符串型变量的末尾附加 NULL 字符 (字符编码 BYTE#16#00)，以表示结尾。

## 数据类型的转换

使用了不同数据类型的变量时，可能自动发生数据类型的转换。此外，用户还可使用类型转换指令来转换。

### 什么是数据类型的转换

所有的变量都有“数据类型”，因此在程序中需要按照数据类型进行操作。

例如，作为代入式，左边和邮编的数据类型要相同。但是，实际情况下，为了实现功能，可能需要代入不同数据类型的变量。

例)

var3 := var1; ————— 代入到不同的类型中

var1为INT型变量

var3为REAL型变量

此时，需要将 var1 的数据转换为适合 var3 的数据类型。这样的转换成为“数据类型转换”，或简称为“类型转换”。

### ● 发生数据类型转换时

数据类型的转换在以下 2 种情况下进行。

- (1) 使用数据类型转换指令，由用户明示转换
- (2) 通过代入 / 指令自动转换
  - ST 语言的代入
  - 梯形图语言中的连接



#### 参考

整数型和 TIME 型的转换请使用时间 / 时刻指令的 NanoSecToTime 指令和 TimeToNanoSec 指令。详情请参考□《NY 系列 指令基准手册 基本篇（SBCA-CN5-437）》。

### 6-3-6 派生数据类型

派生数据类型为根据基本数据类型对构成进行定义的数据类型。

派生数据类型中包括以下内容。

- 结构体型
- 联合体型
- 枚举型

创建派生数据类型时，数据类型名称的字符数等限制请参考□□「6-3-12 变量名称等程序相关名称的限制」(P.6-73)。



#### 参考

NY 系列控制器中，备有以下 3 种类型作为系统定义的派生数据类型。

- 系统定义变量的结构体型
- 指令的输入 / 输出 / 输入输出变量使用的结构体型等派生数据类型
- 高性能单元的扩展存储用结构体型（但在单元编辑器上登录后可使用）

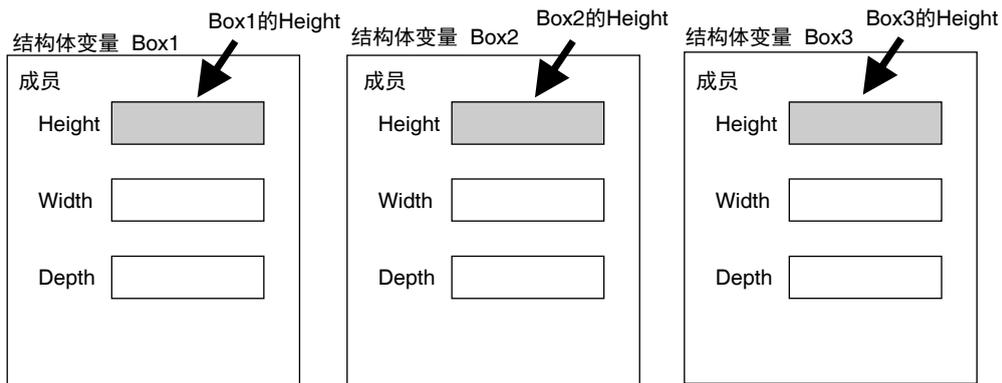
### 结构体型

结构体型是指将数据类型相同或不同的多个数据汇总为 1 个进行处理的派生数据类型。

将数据设为结构体型，可方便地进行数据的登录、变更等管理。

例如，定义名为“Box”的结构体型时，将“Width”、“Height”、“Depth”这 3 个数据作为成员组合并定义。使用该结构体型数据类型，登录名为“Box1”的变量后，可在程序中如“Box1 . Width”、“Box1 . Height”所示，用“.”（句点）按层级指定数据。

需要其他方框中的新数据时，按照与“Box1”相同的要领，将“Box2”登录到变量表中。



在所用指令的变量中使用结构体型时，需要在输入、输出或输入输出参数中选择该结构体型，然后登录变量。

例) 通信指令

## ● 结构体型变量的表现

### 各成员的指定

构成结构体型的数据称为“成员”。

结构体型的成员在表示整个结构体型的变量名称后面加半角句点作为成员名称。

此外，结构体型的成员本身为结构体型亦可。

例)      abc.x                  结构体型变量 abc 的成员 x  
          abc.Order.z        结构体型变量 abc 的成员 (结构体型) Order 的成员 z

### 所有成员（整个结构体）的指定

指构成结构体的所有成员，称为“整个结构体”。

整个结构体仅用结构体的变量名称表示。

上述示例中，以“abc”表示。

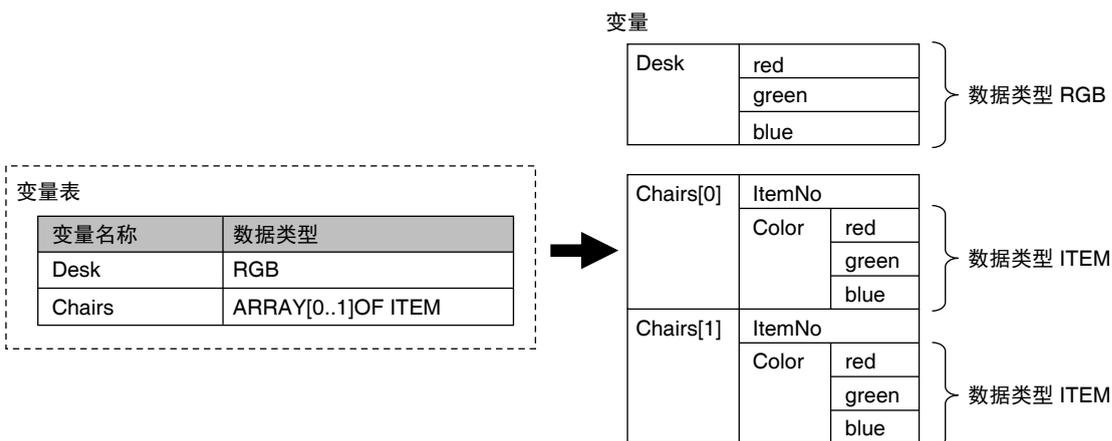
## ● 结构体型的创建方法

### 1 在数据类型表中，创建结构体型的数据类型。

指定数据类型名称、成员及数据类型。

名称	成员	数据类型
RGB	red	INT
	green	INT
	blue	INT
ITEM	ItemNo	INT
	Color	RGB

### 2 在变量表中，指定上述结构体型数据类型、成员名称作为数据类型，然后登录变量。



## ● 结构体型的规格

结构体型的规格如下所示。

项目	规格
结构体型名称	字母不区分大小写。 禁用字符和字符数限制与变量名称相同。
成员的数据类型	请参考后述“结构体成员的数据类型”中的表。
成员的属性	成员名称 注释
成员数	1 ~ 2048
结构体型的层级深度	最多 8 层 (但是, 包括成员名称在内变量名称应在 511 字节以内)
1 个结构体变量的最大大小	8MB

## 结构体型成员的数据类型

分类	数据类型	使用
基本数据类型	布尔型、位列型、整数型、实数型、持续时间型、日期型、时刻型、日期时刻型、字符串型	可以
	布尔型、位列型、整数型、实数型、持续时间型、日期型、时刻型、日期时刻型、字符串型的排列指定	可以
派生数据类型	结构体型 (*)、联合体型、列举型 * 递归或循环的数据不可 (程序检查时将发生错误)。	可以
	结构体型、联合体型、列举型的排列指定	可以
POU 实例型		不支持

## ● 排列指定和结构体型

还可设定为以结构体型为要素的排列指定。此外, 还可设定为以排列指定为成员的结构体型。

## ● 结构体成员的偏置指定

可通过成员的偏置指定, 对各结构体型任意设定成员的存储器配置。这样与 CJ 系列 CPU 单元或外部设备进行标签数据链接时, 可以将两者的结构体型成员的存储器配置统一起来。

结构体型成员的偏置种类可从“NJ”、“CJ”、“任意”中选择。选择了“NJ”时, 将自动向 NY 系列配置最佳的存储器。关于 NY 系列的存储器配置, 请参考□□「A-6 变量的存储器确保方法」(P.A-82)。关于 CJ 系列 CPU 单元和标签数据链接的示例, 请参考□□「A-6-2 应注意的案例」(P.A-91)。

偏置种类的含义如下所示。

偏置种类	含义
NJ	自动向 NY 系列配置最佳的存储器, 以最高速度动作。
CJ	自动变为与 CJ 系列相同的存储器配置。 可与 CJ 系列 CPU 单元的存储器配置统一。
任意	可任意设定各成员的存储器配置。 与 CJ 系列以外的外部设备统一存储器配置时使用。

## 偏置的设置

偏置种类设定为“任意”时，可对各结构体型的成员设定存储器配置的偏置。偏置包括“字节偏置”、“位偏置”。偏置种类为“NJ”、“CJ”时，将自动决定存储器配置，因此无需设定偏置。

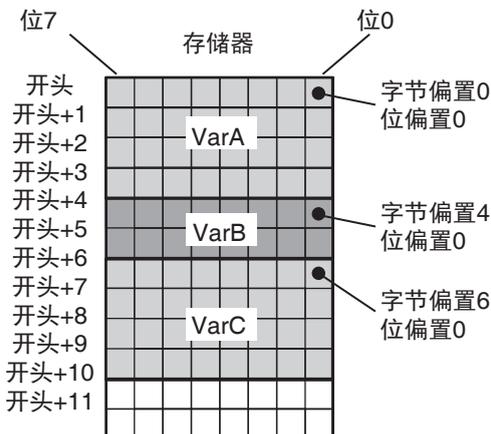
偏置的含义如下所示。

偏置	含义	单位	值的范围
字节偏置	设定各成员从结构体型开头起的偏置。 可使用基本数据类型和派生数据类型的所有数据类型。	字节	0 ~ 1023
位偏置	设定各成员从字节偏置中设定的字节位置开头起的偏置。	位	0 ~ 63

例)

表示结构体型编辑器的设定为以下情况时的存储器配置。

名称	数据类型	偏置种类	字节偏置	位偏置
StrA	STRUCT	任意		
VarA	DINT		0	0
VarB	INT		4	0
VarC	DINT		6	0



## 偏置的设定可否

偏置种类设定为“任意”时，有些数据类型中偏置也可能是固定值。偏置的设定可否如下所示。

分类	数据类型	字节偏置	位偏置
布尔型	BOOL	可设定	可设定
位列型	BYTE,WORD,DWORD,LWORD	可设定	固定
整数型	SINT,INT,DINT,LINT,USINT,UINT,U DINT,ULINT	可设定	固定
实型数据	REAL,LREAL	可设定	固定
持续时间型	TIME	可设定	固定
日期型	DATE	可设定	固定
时刻型	TIME_OF_DAY	可设定	固定
日期时刻型	DATE_AND_TIME	可设定	固定
字符串型	STRING	可设定	固定
排列指定		可设定	仅当要素的数据类型为 BOOL 时可设定
结构体型		可设定	固定
联合体型		可设定	固定
列举型		可设定	固定
POU 实例 型		固定	固定

## 成员的偏置指定限制

进行成员的偏置指定时，有以下限制。结构体型的成员中包括有成员偏置指定的结构体型时，结构体型也有相同的限制。

- 偏置种类设定为“任意”时，需要对该结构体型的所有成员设定偏置。
- 指定了偏置的结构体型成员不可将初始值设定为任意值。初始值为各数据类型中规定的默认值。详情请参考□□「初始值指定为空白时」(P.6-59)。
- 结构体型的存储大小由各成员的大小、数据类型的校准（临界值）和存储器配置决定。

## 成员的偏置指定相关的错误

成员的偏置指定相关的错误如下。

错误名称	含义	偏置种类	处理方法
超出偏置范围错误	偏置中设定的值超出范围。	任意	请变更为合适的偏置值。
偏置未设定错误	有未设定偏置的成员。	任意	请为所有成员设定偏置。
存储器配置重复错误	成员的存储器配置重复。	任意	请变更为合适的偏置值。
初始值设定错误	创建变量表时，欲对指定了偏置的结构体型成员设定初始值。	CJ、任意	请勿设定初始值。

## ● 以整个结构体为参数的指令

指令中包括将结构体变量作为参数传递的指令。此时，应只将结构体整体指定为输入参数。

例) 向 MOVE 指令只传递 1 个结构体型成员时和传递整个结构体变量时

名称	成员	数据类型
RGB	red	UINT
	green	UINT
	blue	UINT

变量名称	数据类型
Color1	RGB

### ■ 仅指定结构体型变量的1个成员



### ■ 指定整个结构体型变量



## 向特定指令的系统定义结构体型输入变量传递值的方法

有些特定指令的输入变量会使用预先准备的结构体型变量。

例) 串行通信指令的输入变量“Port”（目标端口指定）为数据类型名称“\_sPORT”的结构体。

使用特定指令时，请按以下步骤创建结构体型用户定义变量，指定输入参数。

- 1 Sysmac Studio 中，已事先登录指令用的系统定义数据类型。**  
在 Sysmac Studio 上选择该系统定义数据类型，将用户定义的结构体型变量登录到变量表中。
- 2 通过用户程序或利用初始值设定该结构体型变量中各成员的值。**
- 3 将该结构体型变量指定为指令的输入参数。**

## 联合体型

联合体型是指可通过多个不同数据类型对同一数据进行访问的派生数据类型。

指定可访问的多个不同数据类型，如要素数 16 的 BOOL 型排列指定或 16 个 BOOL 型、WORD 型。

## ● 联合体型的表现

### 各成员的指定

定义联合体型时，需要对可访问的各数据类型进行命名。这种名称称为“成员”。

联合体型的成员在表示整个联合体型的变量名称后面加半角句点作为成员名称。

例)

如下定义 “My Union” 作为联合体型的数据类型

数据类型定义：

名称	成员	数据类型
My Union	data	WORD
	bit	ARRAY [0..15] OF BOOL

变量表：

变量名称	数据类型
Output	My Union

Output.bit[0]: 将联合体型 Output 视为 16 位 BOOL 型排列指定的变量时, 要素编号 0, 即第 0 位的值

Output.data: 将联合体型 Output 视为 1 个 WORD 型变量时的值

### 所有成员（整个联合体）的指定

指构成联合体的所有成员, 称为“整个联合体”。

整个联合体仅用联合体的变量名称表示。

上述示例中, 以 “Output” 表示。

## ● 联合体型的创建方法

**1** 在联合体型表中, 创建联合体型的数据类型。  
将数据类型名称、多个不同的数据类型指定为其成员。

**2** 在变量表中, 指定上述联合体型作为数据类型, 然后登录变量。

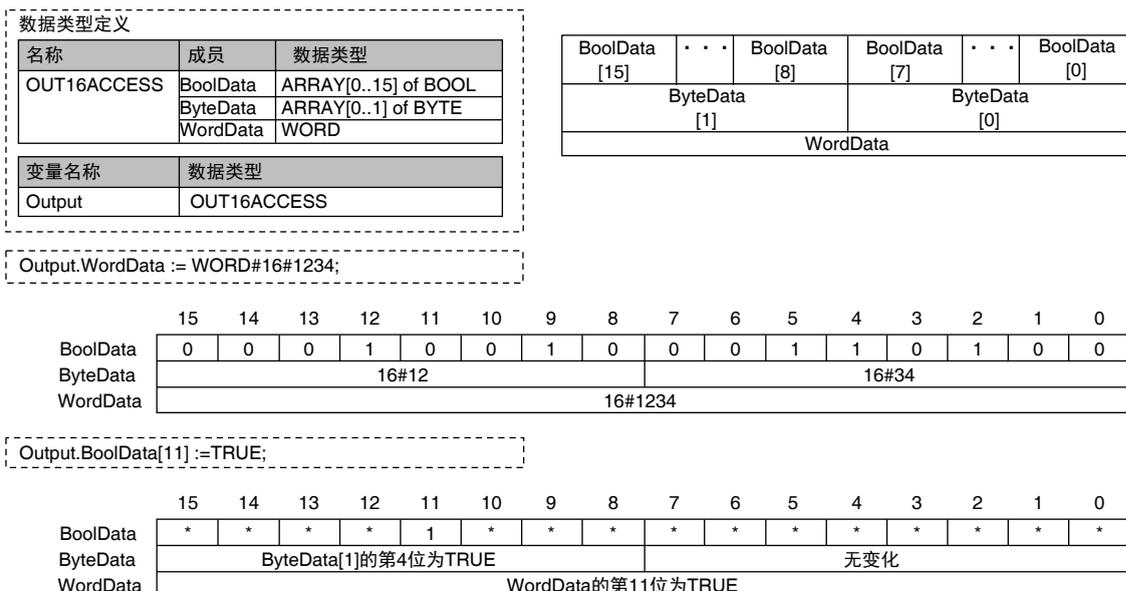
例)

将 OUT16\_ACCESS 定义为联合体型的数据类型。

将该联合体型的成员作为要素数 16 的 BOOL 型排列指定及 WORD 型。

登录使用了数据类型 OUT16\_ACCESS 的变量 Output。

变量 Output 可对 16 位中的任意 1 位进行 BOOL 型值的读写, 也可对 WORD 型的值进行读写。



## ● 联合体型的规格

项目	规格
可指定为成员的数据类型	请参考后述“可指定为联合体成员的数据类型”中的表。
成员数	最大 4
初始值设定	不可。必定变为 0。

### 可指定为联合体成员的数据类型

分类	数据类型	使用
基本数据类型	BOOL 型、位列型	可以
	BOOL 型、位列型的排列指定	可以
	其他基本数据类型	不支持
派生数据类型	结构体型、联合体、枚举型的排列指定	不支持
POU 实例型	—	不支持

## ● 限制事项

- 联合体的初始值固定为 0。
- 不可传送整个联合体。
- POU 的参数不可指定为整个联合体。

## 枚举型 (ENUM)

枚举型是指以名为“枚举值”的字符表示变量值的派生数据类型。

需要预先将该变量可获得的价值设定为枚举值（字符）。

若使用枚举型，变量值的含义会更容易理解。

### ● 枚举型的表现

定义枚举型时，将变量可获得的价值命名为枚举值。

### ● 枚举型的创建方法

#### 1 在枚举型表中，创建枚举型的数据类型。

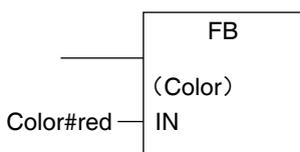
设定与可获得的枚举值对应的值。

#### 2 在变量表中，指定上述枚举型作为数据类型，然后登录变量。

例)

创建 Color 作为枚举型的数据类型。可获得的枚举值为“red”、“yellow”、“green”三个。枚举值对应的数值为 red=0、yellow=1、green=2。

登录使用了数据类型 Color 的变量 DiscColor。变量 DiscColor 可获得的值为 red(0)、yellow(1)、green(2) 中的一个。



枚举型列表	
数据类型	
Color	ENUM
枚举值	值
red	0
yellow	1
green	2

变量表	
变量名称	数据类型
DiscColor	Color

### ● 枚举型的规格

项目	规格
枚举值	只含半角英文和数字，字母不区分大小写。禁用字符与变量名称相同。 若重复指定了相同的枚举值，链接时将发生异常。 指定为与用户程序中的变量相同的枚举值时，或指定为与其他枚举型中存在的枚举值相同的名称时，链接时将发生异常。
值	有效范围为 -2147483648 ~ +2147483647 的整数。 不是连续值亦可。 若重复指定了相同的值，链接时将发生异常。 注：枚举型变量不可进行大小比较。只能比较一致 / 不一致。
枚举值的数量	1 ~ 2048 个

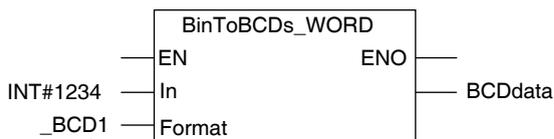
### ● 将枚举值作为 FB、FUN 参数时的记述

将枚举值作为 FB、FUN 的参数传递时，有以下 2 种记述方法。

#### 仅枚举值的记述

如果是参数指定为枚举型的 FB、FUN，只能记述枚举值。

例) 向 BCDToBins\_WORD 指令传递枚举值时

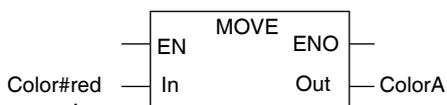


BinToBCDs\_WORD指令的输入变量Format有枚举型指定。  
因此，仅以枚举型的形式记述。

### 枚举型 # 枚举值的记述

如果是参数的数据类型没有特别指定的 FB、FUN，只有枚举值的记述为无效。链接时将发生异常。  
为了明确表示参数为枚举值，以枚举型 # 枚举值的形式记述。

例) 向 MOVE 指令传递枚举值时



MOVE指令的输入变量In没有数据类型的指定。  
传递枚举值时，以枚举型#枚举值的形式记述。



#### 参考

如果是参数指定为枚举型的 FB、FUN，以枚举型 # 枚举值的形式记述参数亦可。  
因此，如果是上述BinToBCDs\_WORD指令，传递到Format变量的参数记述为\_eBCD\_FORMAT#  
\_BCD1 亦可。

#### ● 值的检查

通过指令执行，将值写入枚举型变量时，即使该值不是定义为枚举型的枚举值，也不会发生错误。因此，需要检查值是否是定义为枚举型的枚举值时，请在用户程序中进行值的检查处理。

### 6-3-7 数据类型的排列 / 范围指定

各数据类型的变量中可指定以下属性。

- 排列指定
- 范围指定

#### 排列指定（ARRAY[OF]）

排列指定是针对将相同属性的数据汇总起来，作为一个块处理的数据类型的指定。可对基本数据类型、派生数据类型进行指定。

如运动控制时的坐标值等，处理多个相同概念的数据时非常方便。

#### ● 排列指定的表现

##### 各要素的指定

构成排列指定的各数据称为“要素”。

排列指定的要素通过在表示整个排列指定的排列指定变量名称中，从前往后添加后缀（要素编号）来表示。

后缀用半角方括号 [] 括起来。后缀为常数或变量均可。此外，ST 语言时，后缀还可用公式表示。

例)

变量表	
变量名称	数据类型
Mem	ARRAY[0..99] OF INT

x:=10;

Mem[x] 排列指定的变量 Mem 的要素编号 x（变量 x 的值为 10，所以要素编号为 10）

变量表	
变量名称	数据类型
Data	ARRAY[0..99] OF INT

x:=10;

y:=20;

Data[x+y] 排列指定的变量 Data 的要素编号 (x+y)（变量 x 的值为 10、变量 y 的值为 20，所以要素编号为 30）

##### 所有要素（排列整体）的指定

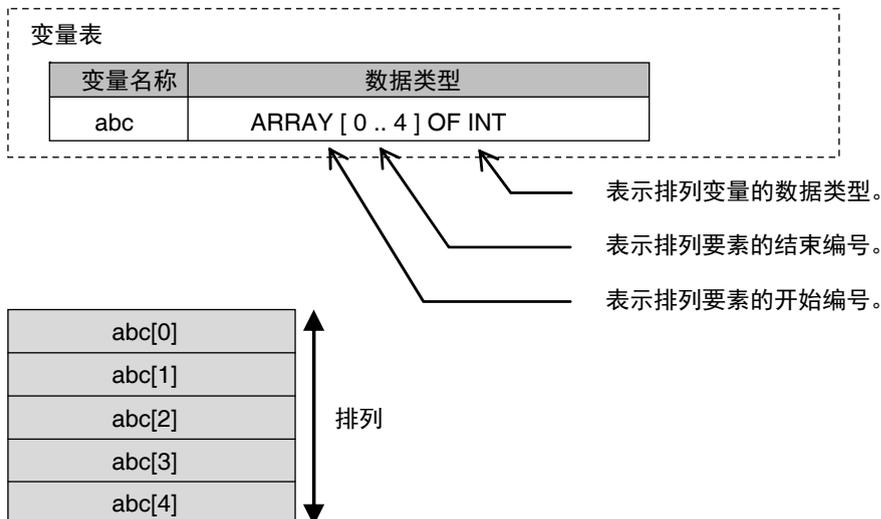
指构成一个排列指定的所有要素，称为“整个排列”。

整个排列仅用排列指定的变量名称表示。

上述示例中，以“Mem”、“Data”表示。

## ● 排列指定的创建方法

- 1 在变量表的数据类型栏中输入“A”，从候选的数据类型名称中选择“ARRAY[?..?] OF ?”。
- 2 在[?..?]左侧的?处输入要素的开始编号，右侧的?处输入结束编号，OF ?的?处输入数据类型，然后登录变量。



## ● 排列指定的规格

项目	规格																	
排列指定的变量最大要素数	65,535																	
要素编号	0 ~ 65535 要素的开始编号不是 0 亦可。																	
后缀	常数：0 ~ 65535 的整数 变量： <table border="1" data-bbox="571 1406 1407 1691"> <thead> <tr> <th>分类</th> <th>数据类型</th> <th>使用</th> </tr> </thead> <tbody> <tr> <td rowspan="2">基本数据类型</td> <td>整数型 SINT、INT、DINT、USINT、 UINT、UDINT</td> <td>支持</td> </tr> <tr> <td>LINT、ULINT</td> <td>不支持</td> </tr> <tr> <td></td> <td>布尔型、位列型、实数型、持续时间型、 日期型、时刻型、日期时刻型、字符串型</td> <td>不支持</td> </tr> <tr> <td>派生数据类型</td> <td>结构体型、联合体型、列举型</td> <td>不支持</td> </tr> <tr> <td>POU 实例型</td> <td></td> <td>不支持</td> </tr> </tbody> </table> 运算式：仅在 ST 语言时可指定运算式 例) $y := x[a+b]$ ；	分类	数据类型	使用	基本数据类型	整数型 SINT、INT、DINT、USINT、 UINT、UDINT	支持	LINT、ULINT	不支持		布尔型、位列型、实数型、持续时间型、 日期型、时刻型、日期时刻型、字符串型	不支持	派生数据类型	结构体型、联合体型、列举型	不支持	POU 实例型		不支持
分类	数据类型	使用																
基本数据类型	整数型 SINT、INT、DINT、USINT、 UINT、UDINT	支持																
	LINT、ULINT	不支持																
	布尔型、位列型、实数型、持续时间型、 日期型、时刻型、日期时刻型、字符串型	不支持																
派生数据类型	结构体型、联合体型、列举型	不支持																
POU 实例型		不支持																
1 个排列变量的最大大小	8MB																	

● 排列指定的维数

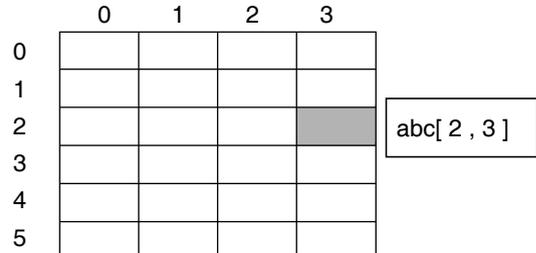
1 维排列指定可以视为将要素排列在 1 列的 1 维数据。

同样，也可设定 2 维或 3 维排列指定。

此时，排列指定要素中将标示与排列指定的变量名称相同维数的后缀表示。维数的最大值为 3。

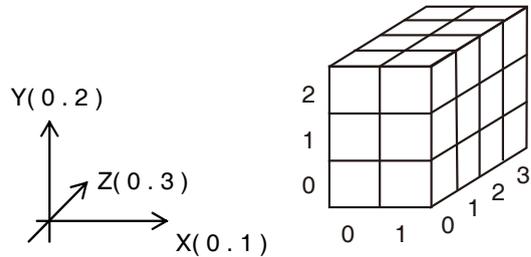
2 维排列指定

变量表	
变量名称	数据类型
abc	ARRAY [0..5, 0..3] OF INT



3 维排列指定

变量表	
变量名称	数据类型
ITEM	ARRAY [0..1, 0..2, 0..3] OF INT



● 排列指定和结构体型

还可设定为以结构体型为要素的排列指定。

此外，还可设定为以排列指定为成员的结构体型。

以结构体型为要素的排列指定

数据类型表		
数据类型	成员	成员
Str	x	INT
	y	DINT

变量表	
变量名称	数据类型
abc	ARRAY [1..4] OF Str

变量
abc[1].x
abc[1].y
:
abc[4].x
abc[4].y

以排列指定为成员的结构体型

数据类型表		
数据类型	成员	数据类型
Str	x	ARRAY [0..1] OF INT
	y	DINT

变量表	
变量名称	数据类型
abc	ARRAY [1..4] OF Str

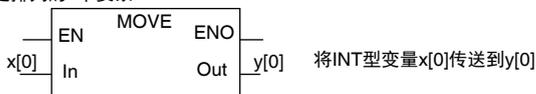
变量
abc[1].x[0]
abc[1].x[1]
abc[1].y
:
abc[4].x[0]
abc[4].x[1]
abc[4].y

## ● 以排列指定为参数的指令

指令中包括将整个排列作为参数传递的指令。此时，应只将排列指定的变量名称指定为输入参数。

例) 向 MOVE 指令只传递 1 个排列指定要素时，传递整个排列

### ■ 指定排列的1个要素



变量表	
变量名称	数据类型
x	ARRAY [ 0 .. 199 ]OF INT
y	ARRAY [ 0 .. 199 ]OF INT

### ■ 指定整个排列



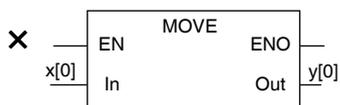
#### · 限制事项:

指定整个排列时，包括要素编号的范围在内，均需为同一数据类型。



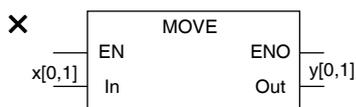
## 参考

不可将多维排列的部分排列指定为参数。



#### 变量表

变量名称	数据类型
x	ARRAY [ 0..9 , 0..9 , 0..9 ]OF INT
y	ARRAY [ 0..9 , 0..9 , 0..9 ]OF INT

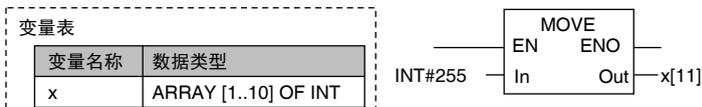


## ● 排列指定的保护功能

若要超出排列的要素数进行访问，将发生以下错误。

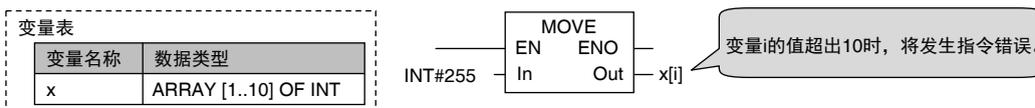
### 后缀为常数时

在 Sysmac Studio 中输入时，以及程序检查时将显示错误。



### 后缀为变量时

在执行指令后，将输出参数代入输出变量时，控制器进行要素数超出检查。指定的后缀变量超出排列变量的要素数时，将视为指令错误。此时，指令的内部处理已正常结束，因此 ENO 的值变为 TRUE。敬请注意。



## 范围指定 ((..))

范围指定是指针对以下整数型变量，限定其值的范围。

分类	数据类型
整数型	SINT、INT、DINT、LINT、USINT、UINT、UDINT、ULINT

以下情况下，可对输入值判断是否在容许范围内。

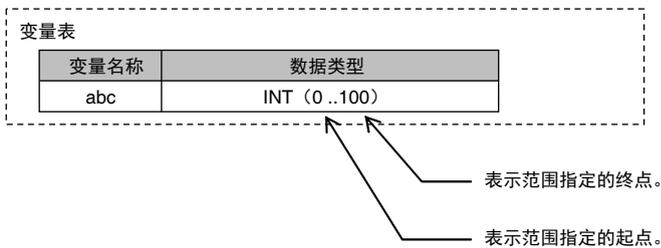
- 指定变量的初始值时
- 通过 CIP 信息通信写入了值时

### ● 范围指定的创建方法

在变量表的数据类型栏中，在数据类型名称后面输入（起点 .. 终点）。

- 起点： 变量中可保存的最小值
- 终点： 变量中可保存的最大值

例)



abc 范围指定：最小0~最大100

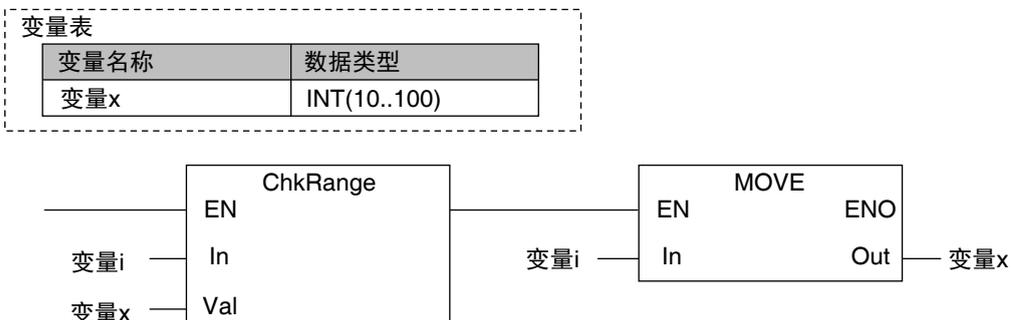
### ● 范围指定的规格

项目	规格																					
可指定的数据类型	变量型																					
尝试写入超出范围的值时的动作	<table border="1"> <thead> <tr> <th>情况</th> <th>动作</th> </tr> </thead> <tbody> <tr> <td>用户程序</td> <td>不发生错误，替换值。 执行指令后变量的值发生变化时，控制器不进行范围检查。</td> </tr> <tr> <td rowspan="4">通信</td> <td>通过 Sysmac Studio 写入、CIP 信息</td> <td> <table border="1"> <tr> <td>整数型时</td> <td rowspan="2">发生指令错误。</td> </tr> <tr> <td>整数型排列变量的要素时</td> </tr> <tr> <td>整数型的结构体型成员时</td> <td rowspan="2">不发生指令错误，替换值。</td> </tr> <tr> <td>整数型的整个结构体时</td> </tr> <tr> <td>整个整数型排列时</td> <td></td> </tr> </table> </td> </tr> <tr> <td>标签数据链接（内置 EtherNet/IP、EtherNet/IP 单元亦同）</td> <td>对指定了范围的成员单独进行写入时将发生错误。 对整个结构体（包含指定了范围的成员）进行写入时，不会发生错误。</td> </tr> <tr> <td>从站 / 单元的 IN 刷新</td> <td>不发生错误，替换值。</td> </tr> <tr> <td>强制值刷新功能的强制值</td> <td>不发生错误，替换值。</td> </tr> </tbody> </table>	情况	动作	用户程序	不发生错误，替换值。 执行指令后变量的值发生变化时，控制器不进行范围检查。	通信	通过 Sysmac Studio 写入、CIP 信息	<table border="1"> <tr> <td>整数型时</td> <td rowspan="2">发生指令错误。</td> </tr> <tr> <td>整数型排列变量的要素时</td> </tr> <tr> <td>整数型的结构体型成员时</td> <td rowspan="2">不发生指令错误，替换值。</td> </tr> <tr> <td>整数型的整个结构体时</td> </tr> <tr> <td>整个整数型排列时</td> <td></td> </tr> </table>	整数型时	发生指令错误。	整数型排列变量的要素时	整数型的结构体型成员时	不发生指令错误，替换值。	整数型的整个结构体时	整个整数型排列时		标签数据链接（内置 EtherNet/IP、EtherNet/IP 单元亦同）	对指定了范围的成员单独进行写入时将发生错误。 对整个结构体（包含指定了范围的成员）进行写入时，不会发生错误。	从站 / 单元的 IN 刷新	不发生错误，替换值。	强制值刷新功能的强制值	不发生错误，替换值。
	情况	动作																				
	用户程序	不发生错误，替换值。 执行指令后变量的值发生变化时，控制器不进行范围检查。																				
	通信	通过 Sysmac Studio 写入、CIP 信息	<table border="1"> <tr> <td>整数型时</td> <td rowspan="2">发生指令错误。</td> </tr> <tr> <td>整数型排列变量的要素时</td> </tr> <tr> <td>整数型的结构体型成员时</td> <td rowspan="2">不发生指令错误，替换值。</td> </tr> <tr> <td>整数型的整个结构体时</td> </tr> <tr> <td>整个整数型排列时</td> <td></td> </tr> </table>	整数型时	发生指令错误。		整数型排列变量的要素时	整数型的结构体型成员时	不发生指令错误，替换值。	整数型的整个结构体时	整个整数型排列时											
		整数型时	发生指令错误。																			
		整数型排列变量的要素时																				
整数型的结构体型成员时		不发生指令错误，替换值。																				
整数型的整个结构体时																						
整个整数型排列时																						
标签数据链接（内置 EtherNet/IP、EtherNet/IP 单元亦同）	对指定了范围的成员单独进行写入时将发生错误。 对整个结构体（包含指定了范围的成员）进行写入时，不会发生错误。																					
从站 / 单元的 IN 刷新	不发生错误，替换值。																					
强制值刷新功能的强制值	不发生错误，替换值。																					

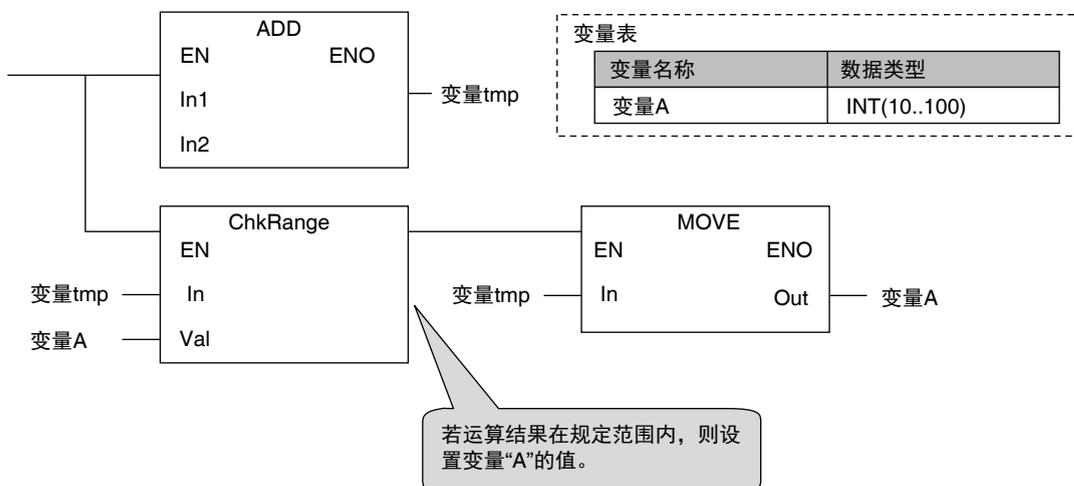


### 使用注意事项

变量的范围指定在通过用户程序执行指令而导致变量的值发生变化前，不会检查。要检查因用户程序执行而发生变化的变量值范围时，请如下所示，使用范围检查指令。



此外，设置四则运算结果等时，因无法进行事先检查，所以请如下所示，对四则运算（例：ADD）后值的范围进行检查。



使用范围指定时，请注意初始值。

Sysmac Studio 的初始值列为空白时，初始值为 0。将不含 0 的范围设为有效范围时也为 0。

### 6-3-8 变量的各属性说明

下面介绍数据类型以外的变量各属性。

#### 变量名称

变量名称是指用于识别变量的名称。

在一个 POU 中，不可重复声明相同变量名称的变量。但是，在不同的 POU 中，可以声明相同变量名称的本地变量，并作为不同的变量处理。

变量名称的限制请参考□□「6-3-12 变量名称等程序相关名称的限制」（P.6-73）。

#### AT（分配目标）指定

该属性用于将设备变量指定到 NX 单元或 EtherCAT 从站的 I/O 端口。将变量分配到 I/O 端口，自动在内部进行 AT（分配目标）指定。

##### ● AT（分配目标）指定的显示

AT（分配目标）指定在变量表及 I/O 映射的“分配目标”列中显示如下。

变量种类	“分配目标”列中的显示	例
EtherCAT 从站设备变量	EtherCAT 从站终端上的 NX 单元时： ECAT://node#[ 节点地址 .NX 单元编号 ]/[I/O 端口名称]	ECAT://node#[10.15]/Input1
	上述以外时： ECAT://node#[ 节点地址 ]/[I/O 端口名称]	ECAT://node#1/Input1
轴变量	MC://_MC_AX[]	MC://_MC_AX[1]
轴组变量	MC://_MC_GRP[]	MC://_MC_GRP[1]

##### ● 可进行 AT 指定的变量种类

以变量为单位指定。排列变量、结构体变量、联合体变量将对整体进行指定。

##### ● AT（分配目标）指定的变量属性

	指定	备注
名称	支持	
数据类型	支持	
保持	支持	取决于 AT（分配目标）指定目标的存储器属性。 NX 单元及 EtherCAT 从站的 I/O 端口全部为不保持。
初始值	支持	直接使用存储器的值时，设定为“无初始值指定（None）”。
常量	支持	不可通过指令写入值。
网络公开	支持	
边缘	不支持	（仅可对功能块的输入变量）



#### 使用注意事项

同一地址中可分配多个变量，但考虑到程序的可读性和调试的难度，并不推荐。此时，设定初始值的变量请只设定为 1 个。若将各变量设定为不同的初始值，初始值将变得不确定。

## 保持

本属性用于指定在以下情况时是否保持变量的值。

- 断电后接通电源时
- 动作模式切换时
- 发生全部停止故障等级的控制器异常时

未设定本属性时，变量的值在上述情况下为初始值。

用于保存生产数量等（NY 系列控制器的电源断开后仍需保持的数据）的变量可设定为有保持，这样比较方便。

### ● 将保持设为有效的必要条件

将工业电脑连接到 UPS 上，并正常关机后，将保存在非易失性存储器中。

关于关机的详情，请参考□□「2-8 关机处理」（P.2-29）。

### ● 要将初始值反映到保持变量时

传送用户程序时，勾选“初始化保持变量的当前值”。

### ● 有 / 无保持的动作区别

与保持有关的情况以及此时变量中设定的值之间的关系如下所示。

情况	变量的值	
	有保持	无保持
断电后接通电源时	保持	不保持
动作模式切换时		
发生全部停止故障等级的控制器异常时		
用户程序传送时	“不初始化保持变量”设定	不保持
	“初始化保持变量”设定	

### ● 可进行保持指定的变量种类

以变量为单位指定。排列变量、结构体变量、联合体变量将对整体进行指定。

## 初始值

初始值是指以下情况下设定的变量值。

- 电源接通时
- 切换为运行模式 ↔ 程序模式时
- 设定为传送用户程序时“初始化保持变量”时
- 发生全部停止故障等级的控制器异常时

设定初始值后，可省略对变量进行初始化的程序。例如，可预先将配方等数据设定为初始值。此外，初始值不设定亦可。

### ● 可设定初始值的变量种类

根据变量的种类，有可设定和不可设定初始值的变量。以下表示其列表。

变量的种类	初始值设定
全局变量	支持
内部变量	
输入变量	
输出变量	
FUN 的返回值	不支持
输入输出变量	
外部变量	

### ● 初始值的有 / 无选择

创建变量时可选择初始值的有 / 无。

有初始值指定

<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><th style="background-color: #cccccc;">初始值</th></tr> <tr><td style="height: 20px;"> </td></tr> </table>	初始值		或	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><th style="background-color: #cccccc;">初始值</th></tr> <tr><td style="text-align: center;">3.14</td></tr> </table>	初始值	3.14
初始值						
初始值						
3.14						

无初始值指定

初始值
None

各类型的选择可否如下所示。

种类	例	初始值的有 / 无选择	
基本数据类型变量	aaa	可以	
排列指定的变量	排列整体	bbb	可以
	要素	bbb[2]	不支持
结构体型变量	结构体整体	ddd	可以
	成员	ddd.xxx	不支持
联合体型变量	联合体整体	eee	不可（初始值固定为 0）
	成员	eee.word	初始值必定为 0
列举型变量	ccc	可以	
POU 实例型	instance	不支持	



## 参考

部分基本 I/O 单元中，一个 I/O 端口可以有多种访问方法，如位列型、布尔型等。对本 I/O 端口使用初始值时，请将一方设定为“None（无初始值指定）”。

### ● 设定初始值的时间

在设定了初始值的变量中实际代入初始值的时间如下所示。

- 电源接通时
- 动作模式从程序切换为运行、从运行切换为程序时
- 指定“初始化变量”，传送了用户程序时
- 发生全部停止故障等级的控制器异常时

### ● 初始值指定为空白时

初始值指定为空白时，变量的初始值如下所示。

数据类型	初始值	
布尔型 / 位列型	0	
整数型	0	
实型数据	0.0	
持续时间型、日期型、时刻型、日期时刻型	TIME	T#0S
	DATE	D#1970-01-01
	TIME_OF_DAY	TOD#00:00:00
	DATE_AND_TIME	DT#1970-01-01-00:00:00
字符串型	"（空字符）	

### ● 排列指定的变量初始值

数据类型	初始值的规格
排列指定	<ul style="list-style-type: none"> <li>• 可对各要素设定初始值。</li> <li>• 指定初始值时，需要对所有要素设置值或设为空白。</li> </ul>

### ● 派生数据型的初始值

不是对数据类型本身指定初始值，而是指定各变量的初始值。

数据类型	初始值的规格
结构体型	<ul style="list-style-type: none"> <li>• 可对各成员设定初始值。</li> <li>• 指定初始值时，需要对所有要素设置值或设为空白。</li> </ul>
联合体型	<ul style="list-style-type: none"> <li>• 不可指定。始终为 0。</li> </ul>
列举型	<ul style="list-style-type: none"> <li>• 可指定。</li> </ul>

### ● 不应用初始值的变量

如果是以下变量，接通电源时不会应用初始值，而是保持断电前的值。

- 保持属性的变量

**使用注意事项**

工业电脑上未连接 UPS 时或未正常关机时，上述变量的值也将变为初始值。

**常量**

常量是指无法通过指令、ST 语言的运算符或 CIP 信息通信进行值写入的属性。因此，可以防止程序对变量进行值的写入。

如果是这种属性的变量，设置初始值后，可通过指令禁止写入。

若 POU 中存在向有常量属性的变量写入值的指令，链接时将发生异常。

**● 动作**

POU 中存在向有常量属性的变量写入值的指令或运算符时，动作如下所示。

写入源		尝试写入值时的动作
用户程序		程序检查时将发生错误。Sysmac Studio 会在链接时进行检查，发生链接异常。
通信	通过 Sysmac Studio 写入	不支持
	CIP 信息	发生指令错误。
	标签数据链接	开始标签数据链接时会发生错误，但标签数据链接继续。但是，带常量属性的变量值不会替换。
从站 / 单元的 IN 刷新		不发生错误，替换值。
强制值刷新		

**● 可进行常量指定的单位**

以变量为单位指定。排列变量、结构体变量、联合体变量将对整体进行指定。

**参考**

“常量”属性也禁止通过用户程序写入。

## 网络公开

网络公开属性是指利用 CIP 信息通信或标签数据链接功能，可在外部（其他控制器或上位计算机）读写变量（这称为“网络公开”）的属性。

未设定本属性时，可通过声明了该变量的控制器来读写变量，不可在外部（其他控制器或上位计算机）读写。

网络公开的变量称为“网络变量”。

### ● 网络公开的规格

网络公开包括“仅公开”、“输入”、“输出”指定。各规格如下所示。

网络公开		规格
不公开		无法从外部访问。 无论是否公开，都可通过支持软件访问。
公开	仅公开	可通过 CIP 通信从外部访问的变量。 不可进行标签数据链接。
	输入	可通过 CIP 通信或标签数据链接从外部访问的变量。 标签数据链接时，为数据的输入用（其他控制器→本控制器）变量。
	输出	可通过 CIP 通信或标签数据链接从外部访问的变量。 标签数据链接时，为数据的输出用（本控制器→其他控制器）变量。

### ● 可进行网络公开指定的单位

以变量为单位指定。排列变量、结构体变量、联合体变量将对整体进行指定。

## 边缘

边缘属性是指 BOOL 型变量的值从 FALSE 上升为 TRUE 时，或从 TRUE 下降为 FALSE 时，向 FB 传递 TRUE 值的属性。

仅可设定为 FB 的 BOOL 型输入变量。

### ● 用途

想要用 FB 仅接收输入参数的上升沿 / 下降沿时使用。

例如，用于仅在输入参数发生变化时启动 FB 等时。

### ● 动作

- 若指定为“上升沿”，则仅在与该输入变量连接的输入参数的值从 FALSE 上升为 TRUE 时，输入变量的值变为 TRUE。
- 若指定为“下降沿”，则仅在输入参数从 TRUE 下降为 FALSE 时，输入变量的值变为 TRUE。

指定	输入参数的值	变量的值
上升沿	FALSE → TRUE	TRUE
	其他	FALSE
下降沿	TRUE → FALSE	TRUE
	其他	FALSE
无	—	每次根据输入参数的值变化

### 6-3-9 与状态变化对应的变量值的变化

接通电源、变更动作模式、下载变量表时、发生全部停止故障等级的控制器异常时、在线编辑时，NY 系列控制器内部的变量值将如下表所示发生变化。

变量的保持属性	变量种类	电源接通时	动作模式变更时	下载后		
				“初始化保持变量” 设定时	“不初始化保持变量” 设定时	
非保持	用户定义变量	<ul style="list-style-type: none"> <li>有初始值设定：指定的初始值。</li> <li>无初始值设定 (None)：16#00<sup>*1</sup></li> </ul>				
	EtherCAT 从站设备变量	<ul style="list-style-type: none"> <li>有初始值设定：指定的初始值。</li> <li>无初始值设定 (None)：16#00<sup>*1</sup></li> </ul>				
保持	用户定义变量	满足保持条件 <sup>*2</sup>	无变化 (保持断电前的值)	无变化 (保持动作模式变更前的值)	<ul style="list-style-type: none"> <li>有初始值设定：指定的初始值。</li> <li>无初始值设定 (None)：16#00<sup>*1</sup></li> </ul>	保持下载前的值
		不满足保持条件 <sup>*2</sup>				<ul style="list-style-type: none"> <li>有初始值设定：指定的初始值。</li> <li>无初始值设定 (None)：16#00<sup>*1</sup></li> </ul>

\*1. 因数据类型不同，可能变为 16#00 以外的值。详情请参考 [□□「初始值指定为空白时」](#) (P.6-59)。

\*2. 保持条件请参考 [□□「保持条件」](#) (P.6-62)。

变量的保持属性	变量种类	发生全部停止故障等级的控制器异常时	在线编辑时	
			追加到在线编辑对象内的 POU 的变量	在线编辑对象内的 POU 中已存在的变量
非保持	用户定义变量 / 设备变量	<ul style="list-style-type: none"> <li>有初始值设定：指定的初始值</li> <li>无初始值设定时：16#00</li> </ul>	<ul style="list-style-type: none"> <li>有初始值设定：指定的初始值</li> <li>无初始值设定时：16#00</li> </ul>	无变化
保持	用户定义变量 / 设备变量	无变化 (保持异常发生前的值)	<ul style="list-style-type: none"> <li>有初始值设定：指定的初始值</li> <li>无初始值设定时：16#00</li> </ul>	
其它	强制值刷新状态	解除	无变化	

## 保持条件

保持条件是指下载前后满足以下所有条件。

- 变量名称相同。
- 数据类型名称相同。
- 保持属性为“有保持”。

关于保持属性为“有保持”的变量，变更了 POU 名称变量名称时变量值的详情请参考 [□□「变更 POU 名称或新建 POU 等时保持变量的值」](#) (P.6-63)，变更了数据类型时变量值的详情请参考 [□□「变更了保持变量的数据类型时变量的值」](#) (P.6-65)。

## 变更 POU 名称或新建 POU 等时保持变量的值

将 Sysmac Studio 中创建的项目下载到控制器后，控制器将以与 Sysmac Studio 中显示的 POU 名称不同的 POU 名称处理。因此，即使 Sysmac Studio 上显示相同的 POU 名称，控制器可能会视为不同的 POU 进行处理。

在 Sysmac Studio 上进行 POU 名称变更或 POU 新建等操作后，执行了下载时，对控制器处理的 POU 名称和保持属性为“有保持”的本地变量值进行说明。

### ● 用 Sysmac Studio 操作前的状态

Sysmac Studio 中创建的项目“Project\_A”将下载到控制器中。“Project\_A”中包含“POU\_A”，“POU\_A”中包含本地变量 VarA。

Sysmac Studio 中显示为“POU\_A”的 POU，在控制器上作为“POU\_XXX”处理。同时，假设控制器上的本地变量 VarA 的当前值已变为“20”。

Sysmac Studio 侧		下载	控制器侧	
项目名称	: Project_A	➔	项目名称	: Project_A
POU 名称	: POU_A		POU 名称	: POU_XXX
变量名称	: VarA		变量名称	: VarA
VarA 的初始值	: 0		VarA 的当前值	: 20

### ● 变更了 POU 名称时

即使在 Sysmac Studio 上将“POU\_A”的名称变更为“POU\_B”并下载，控制器侧的 POU 名称及本地变量 VarA 的当前值不会发生变更。

Sysmac Studio 侧		下载	控制器侧	
项目名称	: Project_A	➔	项目名称	: Project_A
POU 名称	: <b>POU_B</b>		POU 名称	: <b>POU_XXX</b>
变量名称	: VarA		变量名称	: VarA
VarA 的初始值	: 0		VarA 的当前值	: <b>20</b>

### ● 新建了 POU 时

在 Sysmac Studio 上新建包含本地变量 VarA 的“POU\_C”并下载时，控制器上将作为新的“POU\_YYY”及新的本地变量 VarA 处理。

因此，新的本地变量 VarA 的值为初始值“0”。

Sysmac Studio 侧		下载	控制器侧	
项目名称	: Project_A	➔	项目名称	: Project_A
POU 名称	: POU_A		POU 名称	: POU_XXX
变量名称	: VarA		变量名称	: VarA
VarA 的初始值	: 0		VarA 的当前值	: 20
POU 名称	: <b>POU_C</b>		POU 名称	: <b>POU_YYY</b>
变量名称	: VarA		变量名称	: <b>VarA</b>
VarA 的初始值	: 0	VarA 的当前值	: <b>0</b>	

### ● 复制 POU 并创建了 POU 时

在 Sysmac Studio 上复制“POU\_A”并创建“POU\_A\_copy”然后下载时，控制器上将作为新的“POU\_ZZZ”及新的本地变量 VarA 处理。

因此，新的本地变量 VarA 的值为初始值“0”。

Sysmac Studio 侧		下载	控制器侧	
项目名称	: Project_A	➔	项目名称	: Project_A
POU 名称	: POU_A		POU 名称	: POU_XXX
变量名称	: VarA		变量名称	: VarA
VarA 的初始值	: 0		VarA 的当前值	: 20
POU 名称	: <b>POU_A_copy</b>		POU 名称	: <b>POU_ZZZ</b>
变量名称	: VarA		变量名称	: <b>VarA</b>
VarA 的初始值	: 0		VarA 的当前值	: <b>0</b>

### ● 复制了导出的项目文件时

即使在 Sysmac Studio 上复制“Project\_A”并创建“Project\_A\_copy”然后下载，控制器侧的 POU 名称及本地变量 VarA 的当前值不会发生变更。

Sysmac Studio 侧		下载	控制器侧	
项目名称	: <b>Project_A_copy</b>	➔	项目名称	: <b>Project_A_copy</b>
POU 名称	: POU_A		POU 名称	: <b>POU_XXX</b>
变量名称	: VarA		变量名称	: VarA
VarA 的初始值	: 0		VarA 的当前值	: <b>20</b>

将复制“Project\_A”时未存在的本地变量 VarB 追加到“Project\_A”中并下载，然后将相同定义的本地变量 VarB 追加到“Project\_A\_copy”中并下载时，控制器上的本地变量 VarB 的当前值不会发生变更。从“Project\_A”复制本地变量 VarB，或合并变量表时，控制器上的本地变量 VarB 的当前值同样不会发生变更。

Sysmac Studio 侧		下载	控制器侧	
项目名称	: Project_A_copy	➔	项目名称	: Project_A_copy
POU 名称	: POU_A		POU 名称	: POU_XXX
变量名称	: VarA		变量名称	: VarA
VarA 的初始值	: 0		VarA 的当前值	: 20
变量名称	: <b>VarB</b>		变量名称	: <b>VarB</b>
VarB 的初始值	: <b>0</b>		VarB 的当前值	: <b>50</b>

### ● 在另外创建的项目中创建同名的 POU

在 Sysmac Studio 上创建“Project\_B”，其中包含与“Project\_A”的 POU 名称同名的“POU\_A”，然后下载时，控制器上将作为新的“POU\_YYY”及新的本地变量 VarA 处理。因此，新的本地变量 VarA 的值为初始值“0”。

Sysmac Studio 侧		下载	控制器侧	
项目名称	: <b>Project_B</b>	➔	项目名称	: <b>Project_B</b>
POU 名称	: <b>POU_A</b>		POU 名称	: <b>POU_YYY</b>
变量名称	: <b>VarA</b>		变量名称	: <b>VarA</b>
VarA 的初始值	: <b>0</b>		VarA 的当前值	: <b>0</b>

### ● 将原项目的 POU 复制到其他项目时

在 Sysmac Studio 上创建“Project\_B”，从“Project\_A”复制“POU\_A”并粘贴，然后下载时，控制器上将作为新的“POU\_YYY”及新的本地变量 VarA 处理。因此，新的本地变量 VarA 的值为初始值“0”。

Sysmac Studio 侧		下载	控制器侧	
项目名称	: Project_B	➔	项目名称	: Project_B
POU 名称	: POU_A		POU 名称	: POU_YYY
变量名称	: VarA		变量名称	: VarA
VarA 的初始值	: 0		VarA 的当前值	: 0

## 变更了保持变量的数据类型时变量的值

在 Sysmac Studio 上变更保持属性为“有保持”的变量数据类型后，对包含“有保持”的变量当前值执行恢复、传送、下载等操作时，控制器上变量值的情况。

### ● 操作和变量的值

保持属性为“有保持”的变量动作有以下两种模式。操作和动作模式的关系如下表所示。

操作	变量值的动作模式
<ul style="list-style-type: none"> <li>利用 Sysmac Studio 控制器备份功能恢复</li> <li>利用 SD 存储卡备份功能恢复</li> </ul>	模式 1
<ul style="list-style-type: none"> <li>利用 Sysmac Studio 变量 / 存储器的备份功能恢复变量</li> <li>从 Sysmac Studio 下载</li> </ul>	模式 2

下面根据不同的数据类型介绍模式 1 和模式 2 的动作。

## ● 基本数据型的变更

用 Sysmac Studio 变更前的变量定义如下。

变量名称	数据类型	分配目标
ABC	INT	—
DEF	INT	%D100

假设各变量的保持属性为“有保持”，当前值为以下值。

变量 / 存储器	当前值
ABC	10
DEF	20

假设 CJ 单元用存储器的 DM100 和 DM101 的当前值为以下值。

变量 / 存储器	当前值
D100	20
D101	50

数据类型的变更内容和控制器上变量动作的关系如下表所示。

变更内容	模式 1	模式 2
无变更	保持变量的值。 ABC = 10 DEF = 20	
变量名称从“ABC”变更为“GHI”	变量的值为初始值。 GHI = 初始值	
变量“ABC”的数据类型从 INT 变更为 DINT	变量的值为初始值。 ABC = 初始值	
变量“ABC”的保持属性变更为“不保持”	变量的值为初始值。 <sup>*1</sup> ABC = 初始值	
变量“DEF”的分配目标从“D100”变更为“D101” <sup>*2</sup>	变量的值为变更后的分配目标值。 DEF = 50	
删除变量“DEF”的分配目标	变量的值为初始值。 DEF = 初始值	

\*1. 保持属性为“无保持”的变量变更为“有保持”时，变量的值也变温初始值。

\*2. 将分配目标指定为 CJ 单元用存储器的示例。NY 系列控制器中无法使用 CJ 单元用存储器，因此请将其视为变更基本数据型时的参考示例。

## ● 结构体型的变更

用 Sysmac Studio 变更前的结构体型定义如下。

数据类型名称	成员	成员的数据类型
STR_1	x	INT
	y	INT

假设变量 ABC 的数据类型为“STR\_1”、保持属性为“有保持”，各成员的当前值为以下值。

成员	当前值
ABC.x	100
ABC.y	-200

数据类型的变更内容和控制器上变量动作的关系如下表所示。

同时符合保持的条件和变为初始值的条件时，变量的值变为初始值。

变更内容	模式 1	模式 2
无变更	保持成员的值。 ABC.x = 100 ABC.y = -200	
变量名称从“ABC”变更为“DEF”	成员的值变为初始值。 DEF.x = 初始值 DEF.y = 初始值	
成员“x”变更为“x1”	保持成员的值。 ABC.x1 = 100 ABC.y = -200	变更后的成员值变为初始值。 ABC.x1 = 初始值 ABC.y = -200
成员“x”的数据类型从 INT 变更为 WORD <sup>*1</sup> (数据大小相同)	保持成员的值。 ABC.x = WORD#16#0064 (= 100) ABC.y = -200	
成员“y”的数据类型从 INT 变更为 UINT <sup>*1</sup> (数据大小相同, 从有符号变更为无符号)	保持成员的值。变更后的成员将直接代入, 因此绝对值和符号会发生变化。 ABC.x = 100 ABC.y = 65336	
成员“y”的数据类型从 INT 变更为 DINT <sup>*1</sup> (数据大小扩大)	成员的值变为初始值。 ABC.x = 初始值 ABC.y = 初始值	保持成员的值。变更后的成员发生符号扩展, 绝对值和符号无变化。 ABC.x = 100 ABC.y = -200
成员“y”的数据类型从 INT 变更为 UDINT <sup>*1</sup> (数据大小扩大, 从有符号变更为无符号)	成员的值变为初始值。 ABC.x = 初始值 ABC.y = 初始值	保持成员的值。变更后的成员发生符号扩展, 绝对值和符号发生变化。 ABC.x = 100 ABC.y = 4294967096
成员“y”的数据类型从 INT 变更为 SINT <sup>*1</sup> (数据大小缩小)	成员的值变为初始值。 ABC.x = 初始值 ABC.y = 初始值	数据大小缩小后, 成员的值变为初始值。 ABC.x = 100 ABC.y = 初始值
成员“x”变更为“y”, 成员“y”变更为“x”	成员的值保持不正确。 ABC.y = 100 ABC.x = -200	保持成员的值。 ABC.y = -200 ABC.x = 100

变更内容	模式 1	模式 2
在成员“y”后面追加 INT 的成员“z”	成员的值初始化为初始值。 <sup>*2</sup> ABC.x = 初始值 ABC.y = 初始值 ABC.z = 初始值	保持现有成员的值。追加成员的值变为初始值。 ABC.x = 100 ABC.y = -200 ABC.z = 初始值
在成员“x”和成员“y”之间追加 INT 的成员“z”	成员的值初始化为初始值。 <sup>*2</sup> ABC.x = 初始值 ABC.z = 初始值 ABC.y = 初始值	保持现有成员的值。追加成员的值变为初始值。 ABC.x = 100 ABC.z = 初始值 ABC.y = -200
成员“x”的排列要素数从“1”变更为“3”	成员的值初始化为初始值。 <sup>*2</sup> ABC.x[0] = 初始值 ABC.x[1] = 初始值 ABC.x[2] = 初始值 ABC.y = 初始值	保持现有成员的值。变更了排列要素数的成员将从头开始保持，扩展成员的值变为初始值。 ABC.x[0] = 100 ABC.x[1] = 初始值 ABC.x[2] = 初始值 ABC.y = -200

\*1. 变更数据类型时，遵照默认的 CAST 执行。关于默认的 CAST，请参考 [□□「默认的 CAST」](#) (P.6-117)。

\*2. 下载前请先上传变量的值，下载后再下载变量的值，可保持现有的成员值。

## ● 联合体型的变更

用 Sysmac Studio 变更前的联合体定义如下。

数据类型名称	成员	成员的数据类型
UNI_1	x	WORD
	y	DWORD

假设变量 ABC 的数据类型为“UNI\_1”、保持属性为“有保持”，各成员的当前值为以下值。

成员	当前值
ABC	WORD#16#0100

数据类型的变更内容和控制器上变量动作的关系如下表所示。  
同时符合保持的条件和变为初始值的条件时，变量的值变为初始值。

变更内容	模式 1	模式 2
无变更	保持成员的值。 ABC = WORD#16#0100	
变量名称从“ABC”变更为“DEF”	成员的值变为初始值。 DEF = 初始值	
成员“x”的数据类型从 WORD 变更为 BYTE* <sup>1</sup> (作为联合体型的数据大小相同)	保持成员的值。 ABC = WORD#16#0100	
成员“x”的数据类型从 WORD 变更为 LWORD* <sup>1</sup> (作为联合体型的数据大小扩大)	成员的值变为初始值。 ABC = 初始值	保持成员的值。 ABC = WORD#16#0100
成员“y”的数据类型从 DWORD 变更为 BYTE* <sup>1</sup> (作为联合体型的数据大小缩小)	成员的值变为初始值。 ABC = 初始值	
成员“x”变更为“y”，成员“y”变更为“x”	保持成员的值。 ABC = WORD#16#0100	
在成员“y”后面追加 BYTE 的成员“z” (作为联合体型的数据大小相同)	保持成员的值。 ABC = WORD#16#0100	
在成员“y”后面追加 LWORD 的成员“z” (作为联合体型的数据大小扩大)	成员的值变为初始值。 ABC = 初始值	保持成员的值。 ABC = WORD#16#0100
在成员“y”后面追加 BYTE 的成员“z”，删除成员“y” (作为联合体型的数据大小缩小)	成员的值变为初始值。 ABC = 初始值	

\*1. 变更数据类型时，遵照默认的 CAST 执行。关于默认的 CAST，请参考□□「默认的 CAST」(P.6-117)。

## ● 排列变量的变更

用 Sysmac Studio 变更前的排列变量定义如下。

变量名称	数据类型
ABC	ARRAY[1..3] OF INT

假设变量 ABC 的保持属性为“有保持”，各要素的当前值为以下值。

成员	当前值
ABC[1]	100
ABC[2]	200
ABC[3]	300

数据类型的变更内容和控制器上变量动作的关系如下表所示。

同时符合保持的条件和变为初始值的条件时，变量的值变为初始值。

变更内容	模式 1	模式 2
无变更	保持要素的值。 ABC[1] = 100 ABC[2] = 200 ABC[3] = 300	
变量名称从“ABC”变更为“DEF”	要素的值为初始值。 DEF[1] = 初始值 DEF[2] = 初始值 DEF[3] = 初始值	
要素的数据类型从 INT 变更为 UINT* <sup>1</sup> (数据大小相同)	要素的值为初始值。 ABC[1] = 初始值 ABC[2] = 初始值 ABC[3] = 初始值	
要素的数据类型从 INT 变更为 DINT* <sup>1</sup> (数据大小扩大)	要素的值为初始值。 ABC[1] = 初始值 ABC[2] = 初始值 ABC[3] = 初始值	
要素的数据类型从 INT 变更为 SINT* <sup>1</sup> (数据大小缩小)	要素的值为初始值。 ABC[1] = 初始值 ABC[2] = 初始值 ABC[3] = 初始值	
排列要素的结束编号从“3”变更为“4”* <sup>2</sup> (要素数从“3”扩展至“4”)	要素的值为初始值。 ABC[1] = 初始值 ABC[2] = 初始值 ABC[3] = 初始值 ABC[4] = 初始值	保持现有要素编号的要素值。新要素编号的要素值为初始值。 ABC[1] = 100 ABC[2] = 200 ABC[3] = 300 ABC[4] = 初始值
排列要素的开始编号从“1”变更为“0”* <sup>2</sup> (要素数从“3”扩展至“4”)	要素的值为初始值。 ABC[0] = 初始值 ABC[1] = 初始值 ABC[2] = 初始值 ABC[3] = 初始值	保持现有要素编号的要素值。新要素编号的要素值为初始值。 ABC[0] = 初始值 ABC[1] = 100 ABC[2] = 200 ABC[3] = 300
排列要素的结束编号从“3”变更为“2”* <sup>2</sup> (要素数从“3”缩小为“2”)	要素的值为初始值。 ABC[1] = 初始值 ABC[2] = 初始值	保持现有要素编号的要素值。 ABC[1] = 100 ABC[2] = 200

变更内容	模式 1	模式 2
排列要素的开始编号从“1”变更为“2” <sup>2</sup> (要素数从“3”缩小为“2”)	要素的值为初始值。 ABC[2] = 初始值 ABC[3] = 初始值	保持现有要素编号的要素值。 ABC[2] = 200 ABC[3] = 300
排列要素的开始编号从“1”变更为“3”，结束编号从“3”变更为“5” <sup>2</sup> (要素数相同)	要素数相同时，要素值从头开始保持。 <sup>3</sup> ABC[3] = 100 ABC[4] = 200 ABC[5] = 300	保持现有要素编号的要素值。新要素编号的要素值为初始值。 ABC[3] = 300 ABC[4] = 初始值 ABC[5] = 初始值

\*1. 变更数据类型时，遵照默认的 CAST 执行。关于默认的 CAST，请参考□□「默认的 CAST」(P.6-117)。

\*2. 无论变量数据为基本数据类型还是派生数据类型，结果都一样。

\*3. 要素数发生扩展或缩小时，所有要素值均变为初始值。

## ● 枚举型的变更

用 Sysmac Studio 变更前的枚举型定义如下。

数据类型名称	枚举值	值
ENU_1	x	1
	y	2

假设变量 ABC 的数据类型为“ENU\_1”、保持属性为“有保持”，变量的当前值为以下值。

成员	当前值
ABC	1 (= x)

数据类型的变更内容和控制器上变量动作的关系如下表所示。

变更内容	模式 1	模式 2
无变更	保持变量的值。 ABC = 1 (= x)	
变量名称从“ABC”变更为“DEF”	变量的值为初始值。 ABC = 初始值	
在枚举值“y”后面追加值为“3”的枚举值“z”	保持变量的值。 ABC = 1 (= x)	
删除枚举值“x”	枚举值不存在时，保持变量的值。 ABC = 1	
枚举值“X”的值从“1”变更为“5”	即使变更了枚举值，仍保持变量的值。 ABC = 1	

### 6-3-10 功能块 (FB) 的实例

实例化的 FB 为数据类型的一种，登录到本地变量表中并显示。



#### 参考

FB 的实例视为记述了实例的程序中的本地变量，登录到该程序的本地变量表中并显示。不会变成全局变量，敬请注意。

### 6-3-11 变量值的监视

变量的值可通过 Sysmac Studio 的监视窗口进行监视。

- 1** 选择 [显示] - [监视窗口]。  
显示监视窗口。
- 2** 与控制器在线连接，通过以下任意方法登录变量。
  - (1) 在监视窗口的名称单元格中输入变量名称。
  - (2) 从编辑器或变量表拖放到监视窗口中登录。  
显示该变量的当前值。

## 6-3-12 变量名称等程序相关名称的限制

程序相关名称的限制如以下列表所示。

### 字符限制

程序相关的名称	可使用的字符	保留字	多字节字符使用可否	大小写的区分	最大大小 (不含 NULL)	字符编码
变量名称 (含 POU 实例名称)	<ul style="list-style-type: none"> <li>● 可使用的字符               <ul style="list-style-type: none"> <li>• 0 ~ 9、A ~ Z、a ~ z</li> <li>• 半角假名</li> <li>• _ (下划线)</li> <li>• 多字节字符 (例: 日语) 保留字请参考后述的“保留字”。</li> </ul> </li> <li>● 不可组合使用的字符               <ul style="list-style-type: none"> <li>• 以数字 (0 ~ 9) 开始的字符串</li> <li>• 以 P_ 开始的字符串</li> <li>• 以下划线开始的字符串</li> <li>• 有 2 个以上连续下划线 ( ) 的字符串</li> <li>• 以下划线 ( ) 结束的字符串</li> <li>• 构成识别符的字符串的前缀或后缀中包含 1 字节以上的扩展空白符 (全半角空格等, 扩展至 Unicode 的空白字符) 的字符串</li> </ul> </li> </ul>	参考后述的“保留字”	支持	否	127 字节	UTF-8 <sup>*1</sup>
POU 定义名称						
数据类型名称						
结构体成员名称						
联合体成员名称						
列举值						
任务名称						
名称空间名						
变量名称的完整路径						
设备名称						
段落名称						
轴名称						
轴组名称						
凸轮表名称						
				是	63 字节	
				否	93 字节	
					网络变量: 255 字节 其他: 511 字节	
					127 字节	

\*1. UTF-8 中, 1 个字符的半角英文和数字使用 1 字节, 多字节字符使用多个字节。日文字符一般为 3 字节。

### 保留字

以下名称在程序检查时将发生错误。

- 与《NY 系列 指令基准手册 基本篇 (SBCE-CN5-437)》的指令相同的名称
- 与《NY 系列 指令基准手册 运动篇 (SBCE-CN5-380)》的指令相同的名称
- 系统内部使用的保留字

### 名称不可重复

---

以下名称不可重复。程序检查时将发生错误。

- 在同一控制器中，全局变量名称重复
- 在同一 POU 中，变量名称重复
- 在同一 POU 中，段落名称重复
- 在同一结构体或联合体中，成员名称重复
- 在同一枚举型中，列举值重复
- 本地变量名称和全局变量名称重复
- POU 名称和数据类型名称重复
- 数据类型名称和变量名称重复
- 列举值和其他枚举型的列举值重复
- 列举值和变量名称重复

## 6-4 常数（文本）

下面详细介绍常数。

### 6-4-1 常数定义

变量的值因变量中代入的数据不同而变化。与之相对的是值不会变化的常数。

常数与变量不同，不是存储器中保存的信息。常数不需要声明，也可在 POU 的算法部分中记述。

NY 系列控制器中，常数与变量一样，都有“数据类型”的概念。

### 6-4-2 各数据类型的记述形式

各数据类型的常数记述形式如下所示。若用规定以外的形式记述常数，链接时将发生异常。

#### 布尔型

布尔型的数据类型名称为“BOOL”。此外，值除了 1 或 0 以外，还可使用“TRUE”和“FALSE”。记述形式及含义如下所示。

记述形式	含义
TRUE 或 FALSE	TRUE、BOOL#1、BOOL#TRUE、1 为等价。 FALSE、BOOL#0、BOOL#FALSE、0 为等价。
BOOL#1 或 BOOL#0	
BOOL#TRUE 或 BOOL#FALSE	
1 或 0	

#### 位列型

位列型使用“BYTE”、“WORD”、“DWORD”、“LWORD”中的任意一个作为数据类型名称。此外，使用“2”、“8”、“10”、“16”中的任意一个作为基数。记述形式和记述示例如下所示。

记述形式	记述示例
{数据类型名称}#{基数}#{数值}	WORD#16#0064
{基数}#{数值}	16#0064
{数值} <sup>*1</sup>	100

\*1. 基数为 10，即解释为 10 进制数。

## 整数型

整数型使用“SINT”、“USINT”、“INT”、“UINT”、“DINT”、“UDINT”、“LINT”、“ULINT”中的任意一个作为数据类型名称。此外，使用“2”、“8”、“10”、“16”中的任意一个作为基数。

记述形式和记述示例如下所示。

记述形式	记述示例
{数据类型名称}#{基数}#{数值}	INT#10#-1
{基数}#{数值}	10#-1
{数据类型名称}#{数值} <sup>*1</sup>	INT#-1
{数值} <sup>*1</sup>	-1

\*1. 基数为 10，即解释为 10 进制数。

## 实型数据

实型数据使用“REAL”和“LREAL”中的一个作为数据类型名称。此外，实型数据中可使用的基数仅限“10”。记述形式和记述示例如下所示。

记述形式	记述示例
{数据类型名称}#{基数}#{数值}	LREAL#10#-3.14
{基数}#{数值}	10#-3.14
{数据类型名称}#{数值} <sup>*1</sup>	LREAL#-3.14
{数值} <sup>*1</sup>	-3.14

\*1. 基数为 10，即解释为 10 进制数。

## 持续时间型

持续时间型使用“TIME”和“T”中的一个作为数据类型名称。记述形式和记述示例如下所示。

记述形式	记述示例
TIME#{日}d{时}h{分}m{秒}s{毫秒}ms	TIME#61m5s
T#{日}d{时}h{分}m{秒}s{毫秒}ms	T#61m5s

持续时间型常数的记述有以下规则。

- {日}、{时}、{分}、{秒}、{毫秒}不需要全部记述。记述其中的一个或几个即可。
- 如“TIME#12d3.5h”，可用小数记述。
- 记述内容可超出时间的有效范围。例如，“T#-61m5s”与“T#-1h1m5s”的含义相同。
- 数值解释为 10 进制数。若记述了 10 进制数以外的数值，链接时将发生异常。
- 记述时可改变时间的顺序。例如，“T#1h2d”与“T#2d1h”的含义相同。

## 日期型

日期型使用“DATE”和“D”中的一个作为数据类型名称。记述形式和记述示例如下所示。

记述形式	记述示例
DATE#{年}-{月}-{日}	DATE#2010-1-10
D#{年}-{月}-{日}	D#2010-1-10

日期型常数的记述有以下规则。

- {年}、{月}、{日}的开头分别可添加1个以上的0。例如，“DATE#2010-01-10”与“D#2010-1-10”的含义相同。
- 若记述内容超出日期的有效范围，链接时将发生异常。例如，“D#2010-01-35”时将发生异常。
- 数值解释为10进制数。若记述了10进制数以外的数值，链接时将发生异常。

## 时刻型

时刻型使用“TIME\_OF\_DAY”和“TOD”中的一个作为数据类型名称。记述形式和记述示例如下所示。

记述形式	记述示例
TIME_OF_DAY#{时}:{分}:{秒}	TIME_OF_DAY#23:59:59.999999999
TOD#{时}:{分}:{秒}	TOD#23:59:59.999999999

时刻型常数的记述有以下规则。

- {时}、{分}、{秒}的开头分别可添加1个以上的0。例如，“TOD#23:01:01”与“TOD#23:1:1”的含义相同。
- 若记述的数值超出时分秒的有效范围，链接时将发生异常。例如，“TOD#24:00:00”时将发生异常。
- 数值解释为10进制数。若记述了10进制数以外的数值，链接时将发生异常。

## 日期时刻型

日期时刻型使用“DATE\_AND\_TIME”和“DT”中的一个作为数据类型名称。记述形式和记述示例如下所示。

记述形式	记述示例
DATE_AND_TIME#{年}-{月}-{日}-{时}:{分}:{秒}	DATE_AND_TIME#2010-10-10-23:59:59.123
DT#{年}-{月}-{日}-{时}:{分}:{秒}	DT#2010-10-10-23:59:59.123

日期时刻型常数的记述有以下规则。

- {年}、{月}、{日}、{时}、{分}、{秒}的开头分别可添加1个以上的0。例如，“DT#2010-01-10-23:01:01”和“DT#2010-1-10-23:1:1”的含义相同。
- 若记述的数值超出年月日时分秒的有效范围，链接时将发生异常。例如，“DT#2010-01-35-00:00:00”或“DT#2010-01-30-24:00:00”时将发生异常。
- 数值解释为10进制数。若记述了10进制数以外的数值，链接时将发生异常。

## 字符串型

字符串型用两个半角单引号将字符串括起来记述。同时，可使用“STRING”作为数据类型名称。记述形式和记述示例如下所示。

记述形式	记述示例
{字符串}	'这是字符串'
STRING#{字符串}	STRING#'这是字符串'

字符串的记述有以下规则。

- 还可记述字符数为 0 的字符串。此时记述为“”。
- 如下所示，若跨多行记述字符串，链接时将发生异常。  
strVar := 'ABC  
DEF'
- 记述制表符或换行代码等时，使用开头带“\$”的转义字符。转义字符的名称和含义如下所示。

转义字符	名称	含义
\$\$	半角美元符号	半角美元符号字符（\$：字符编码 0x24）
'	半角单引号	半角单引号符号字符（'：字符编码 0x27）
"	半角双引号	半角双引号符号字符（"：字符编码 0x22）
\$L 或 \$l	换行	将光标移动到下一行。 LF 控制字符（Line Feed：字符编码 0x0A）
\$N 或 \$n	换行	将光标移动到下一行。 NL 控制字符（New Line：字符编码 0x0A）
\$P 或 \$p	翻页	将光标移动到下一页。 FF 控制字符（Form Feed：字符编码 0x0C）
\$R 或 \$r	复位	将光标移动到行的开头。 CR 控制字符（Carriage Return：字符编码 0x0D）
\$T 或 \$t	制表符字符	显示制表符。 制表符字符（字符编码 0x09）
\$（字符编码）	直接指定字符编码	用 2 位的 16 进制数直接指定字符编码。字符编码的范围为 00 ~ FF。 例如，'\$L' 与 '\$0A' 的含义相同。 此外，2 字节以上的字符每个字节都会带 \$。 例如，字符编码为 0xE38182 平假名“あ”，记述为“\$E3\$81\$82”。

- 还可直接指定字符编码后记述字符。此时，在开头加“\$”。用 2 位的 16 进制数记述字符编码。例如，换行“\$L”的字符编码为 0x0A，因此可记述为“\$0A”。
- 直接指定字符编码，记述 2 字节以上的字符时，每个字节加“\$”。例如，字符编码为 0xE38182 平假名“あ”，记述为“\$E3\$81\$82”。

## 枚举型

枚举型中记述枚举型的数据类型名称和枚举值。记述形式和记述示例如下所示。

记述形式	记述示例
{枚举型的数据类型名称} # {枚举值}	_eDAYOFWEEK#_WED



### 参考

向参数的数据类型指定为枚举型的 FB 或 FUN 传递枚举值时，可省略枚举型的数据类型名称，只记述枚举值。

例如，BinToBCDs 指令的输入变量“Format”指定为枚举型 \_eBCD\_FORMAT。因此，除了记述枚举型的数据类型名称和枚举值的“\_eBCD\_FORMAT#\_BCD0”之外，还可省略枚举型的数据类型名称，记述为“\_BCD0”。

## 6-5 编程语言

下面详细介绍各编程语言。

用 Sysmac Studio 输入编程语言的方法请参考 □□《Sysmac Studio Version 1 操作手册 (SBCA-CN5-362)》。

### 6-5-1 编程语言定义

表示 POU（程序、FB、FUN）中算法的语言称为编程语言。

NY 系列控制器中可使用的编程语言分为梯形图语言（LD）、ST 语言（Structured Text：结构化文本）两种。

### 6-5-2 梯形图语言（LD）

梯形图语言是以电气回路示意图记述的图形编程语言。包括 FB 或 FUN 在内，各部分的处理全部以图形表示，然后用直线将它们连接起来，构建一系列的算法。用梯形图语言撰写的算法称为梯形图。

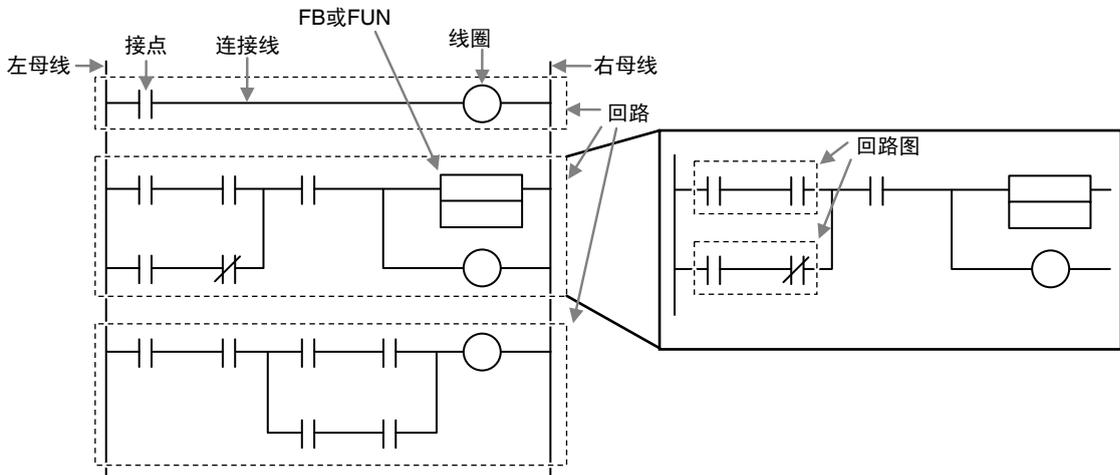
#### 梯形图语言的构成要素

梯形图语言由左右母线、连接线、梯形图语法（接点、线圈）、FB 及 FUN 构成（\*）。

\* 只有 JMP 指令及 Label 指令时，以表示 JMP 和 Label 的符号记述。

算法由连接以上要素的多个回路构成。

回路是指将左母线到右母线之间的构成要素连接起来。此外，回路由以（表示逻辑启动的）LD/LDN 指令开头的回路图构成。



#### ● 母线

梯形图左端和右端的竖线称为母线。

各母线处于 TRUE 或 FALSE 中的任意状态。这表示将梯形图视为电气回路时的通电状态和非通电状态。

启动写有梯形图的 POU 后，左母线的值变为 TRUE。同时，针对与左母线连接的接点等构成要素，也会传达 TRUE 的状态。按这些构成要素的动作，TRUE 的状态依次向右传达。

这种 TRUE 状态的流动称为“功率流”。母线可以说是功率流的源泉。

## ● 连接线

水平方向连接母线和各种构成要素的直线称为连接线。

连接线处于 TRUE 或 FALSE 中的任意状态，起到向右传达功率流的作用。

## ● 接点

接点配置在连接线上，通过接收功率流来动作。

接点有多个种类，将根据各自的规格将左侧传来的功率流向右传递，或将其阻断。

接点向右传达功率流后，接点右侧的连接线状态将变为 TRUE，若阻断，则变为 FALSE。

各接点的详细规格请参考□□《NY 系列 指令基准手册 基本篇（SBCA-CN5-437）》。

## ● 线圈

线圈配置在连接线上，通过接收功率流来动作。

线圈的功能是向变量写入 TRUE 或 FALSE 的值。

线圈也有多个种类。各线圈的详细规格请参考□□《NY 系列 指令基准手册 基本篇（SBCA-CN5-437）》。

## ● FB、FUN

FB、FUN 配置在连接线上，通过接收功率流来动作。

各指令的详细规格请参考□□《NY 系列 指令基准手册 基本篇（SBCA-CN5-437）》。

## 梯形图的执行顺序

接点、线圈、FB、FUN 通过接收功率流来执行。

执行顺序为梯形图从上往下。此外，如果时高度相同的要素，则从左往右执行。

## 梯形图的结束

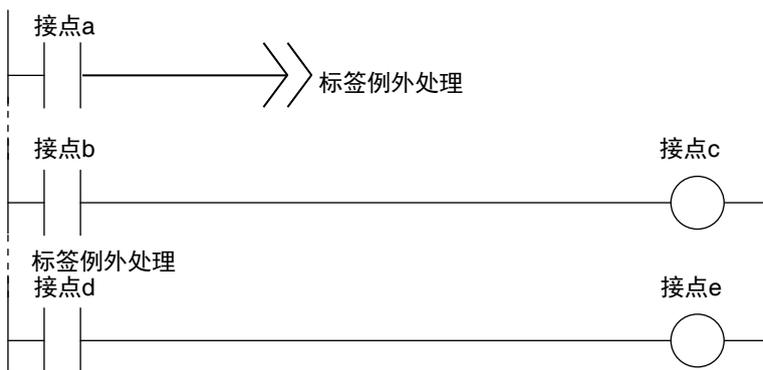
梯形图从上往下执行，执行到最底下，则结束处理。

但是，若中途有 END 指令或 RETURN 指令，将在该时间点结束处理。上述指令后面的处理将不执行。

## 梯形图的执行控制

梯形图基本上从上往下执行，但如果使用执行控制指令，可以更改执行顺序。

例如，以下情况时，若接点 a 变为 TRUE，则转移到有“例外处理”标签的位置并执行。



## 梯形图中 FB、FUN 的连接

### ● 连接形式

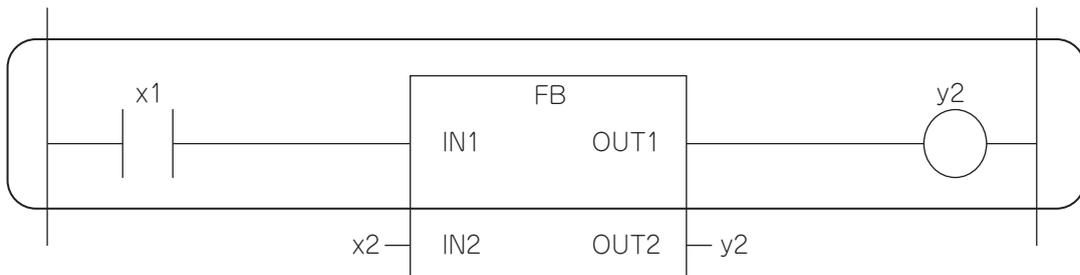
FB 及 FUN 中，有以下两种连接形式。

#### (1) 功率流输入、输出

在梯形图语言中，是指 FB 及 FUN 的输入变量与左母线连接、输出变量与右母线连接时 BOOL 型的输入、BOOL 型的输出。

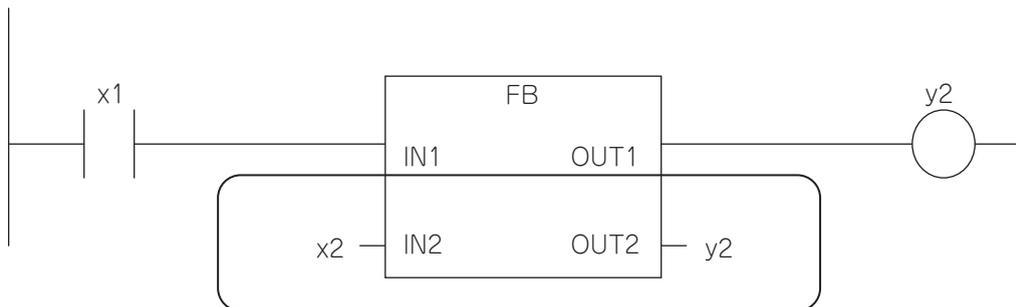
例)

左母线上连接的功率流与接点 (CONTACT) 连接，右母线与线圈 (COIL) 连接。



#### (2) 参数输入、参数输出

在梯形图语言中，是指 FB 及 FUN 的输入变量、输出变量不左母线和右母线连接的情况。



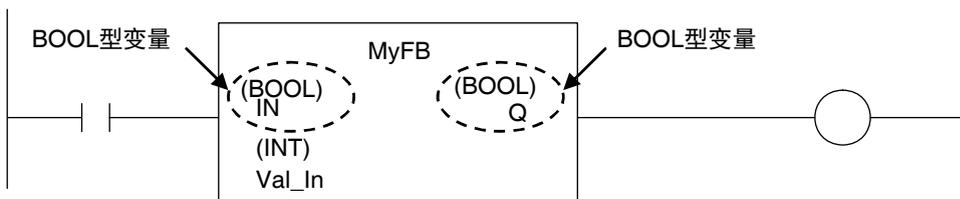
输入及输出参数中可如下指定变量或常数。

FB/FUN 侧变量	输入参数	输出参数
输入变量	可指定变量或常数	—
输出变量	—	只能指定变量
输入输出变量	只能指定变量	只能指定变量

### ● BOOL 型变量的数量

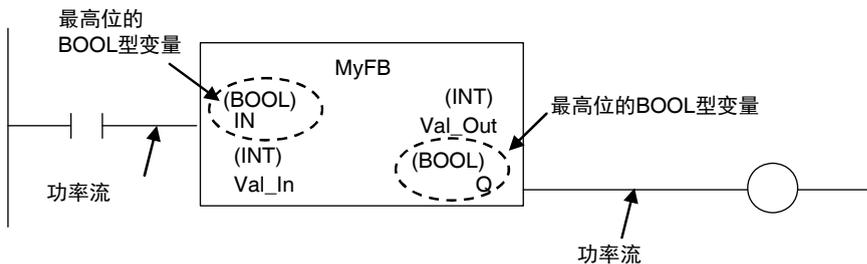
FB 及 FUN 的输入及输出必须有一个以上的 BOOL 型变量 (含 EN、ENO)。

例)



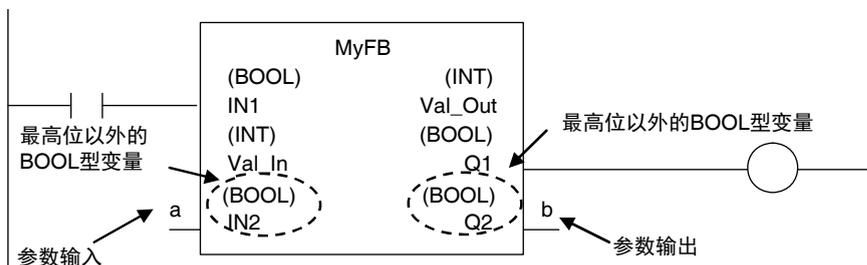
### ● BOOL 型变量不同位置的连接状态

最高位的 BOOL 型变量与左母线及右母线连接。即为功率流输入及功率流输出。

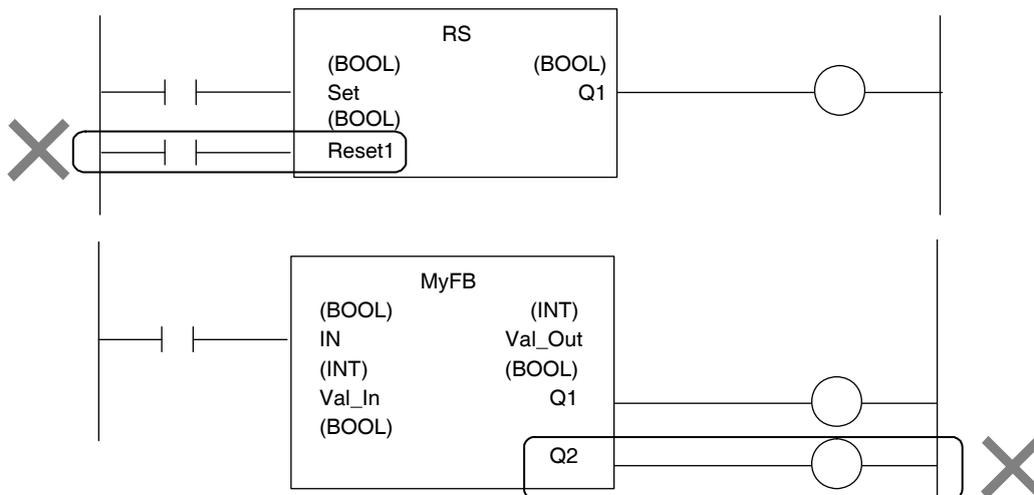


与 FB 及 FUN 连接的功率流输入、功率流输出分别为 1 个。

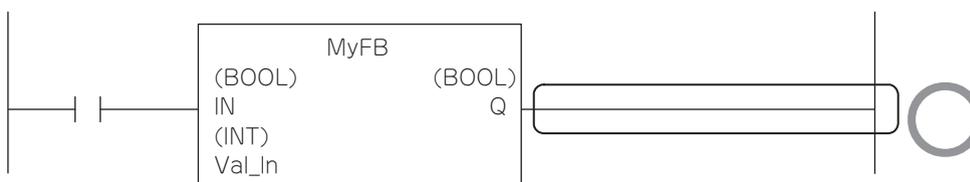
最高位以外的 BOOL 型变量为参数输入及参数输出。



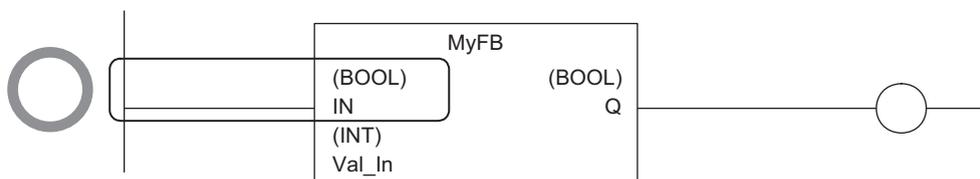
因此，不能像下面这样，将多个 BOOL 型变量连接到左母线或右母线上。



没有 OUT 指令等，与右母线直接连接亦可。

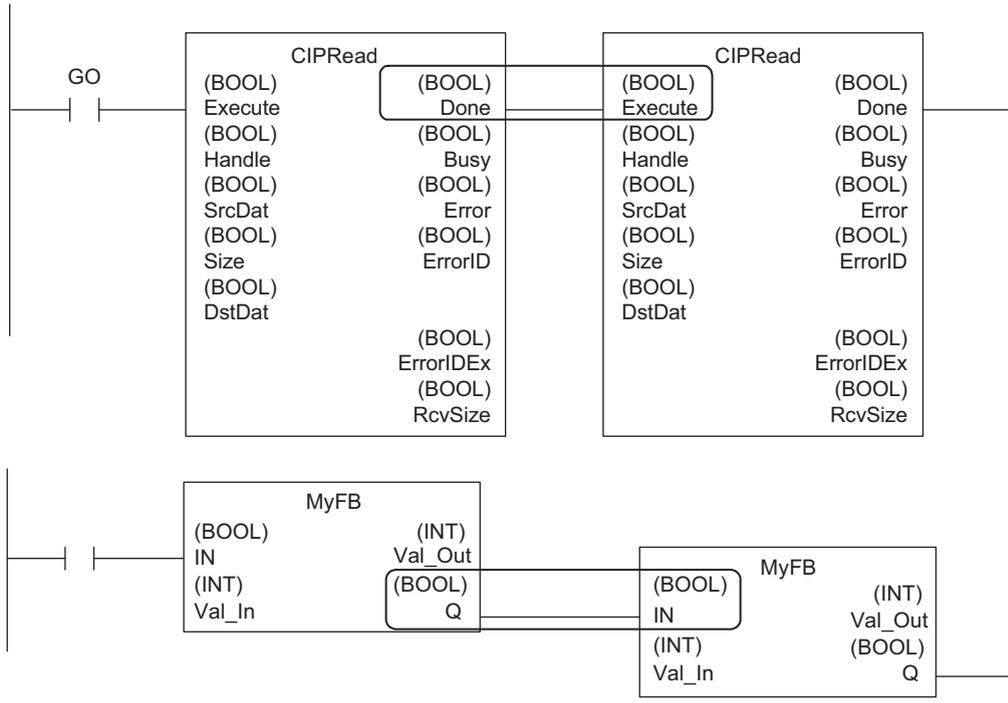


没有 LD 指令等，与左母线直接连接亦可。



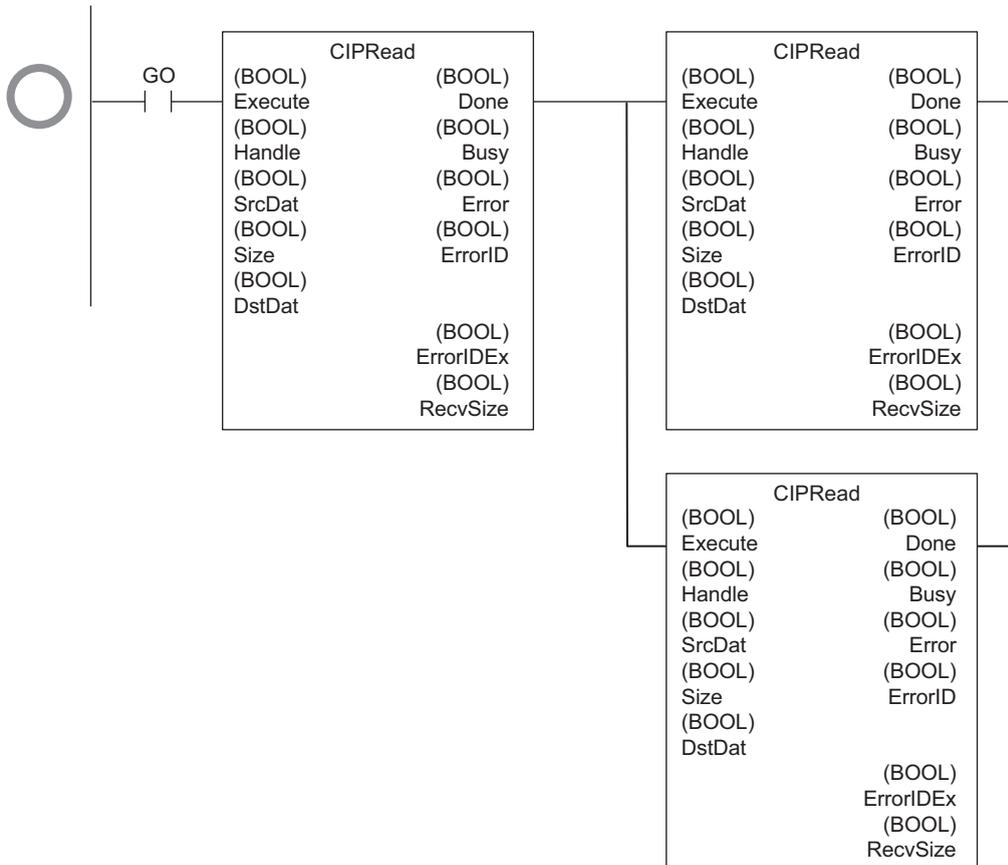
● 级联连接

将 FB/FUN 的输出作为其他 FB/FUN 输入的级联连接只能在功率流输出和功率流输入之间进行。



- 功率流输出可以分支。

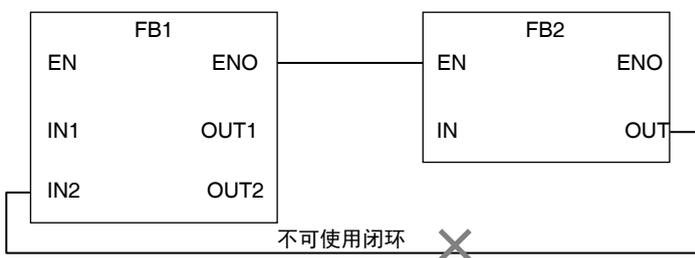
例)



限制

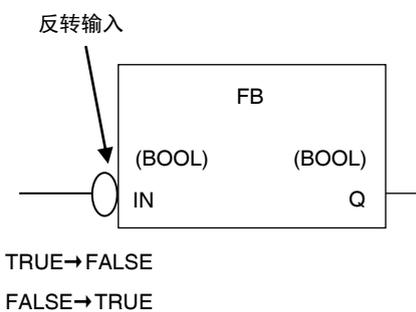
- 不可形成闭环或连接线交叉。

例)



## ● 输入的反转

对于 BOOL 型输入变量，可以将值反转后输入。

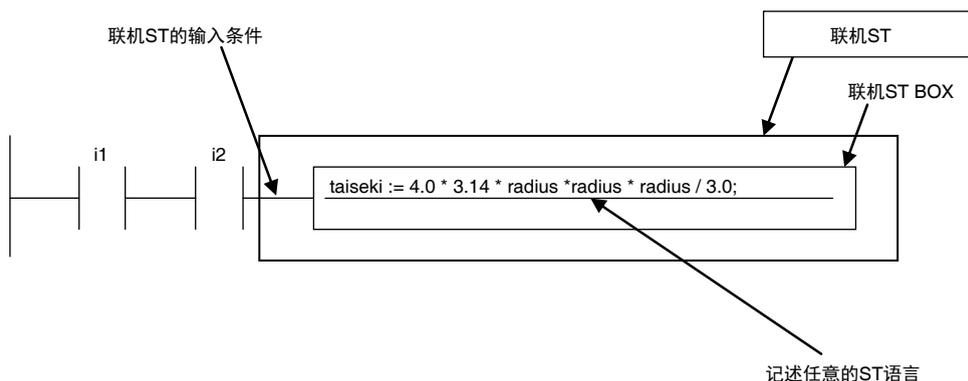


## 联机 ST

### ● 概要

联机 ST 是指在名为“联机 ST BOX”的方框（空白的文本输入区域）中记述 ST 语言的梯形图语言要素。这样，可以方便地在梯形图语言中记述数值运算处理或字符串操作处理。

“联机 ST BOX”的连接线变为输入条件，ST BOX 内的 ST 语言按该输入条件执行。具体如下所示。



此外，联机 ST 将作为以梯形图语言记载的回路要素的一种。所以，与 FB 或 FUN 不同，没有输入变量、输出变量、输入输出变量。

### ● 联机 ST 中的 ST 语言限制

联机 ST BOX 中，可记述任意的 ST 语言程序。

### ● 联机 ST 的执行条件

联机 ST 的执行条件如下所示。

状态	动作
输入条件 TRUE 时	按输入条件动作。 输入条件可直接与功率流或左母线连接。 上升沿或下降沿指定在输入条件的接点中指定。
输入条件 FALSE 时	无操作
通过主控执行复位时	无操作

### ● 联机 ST 中变量的范围

联机 ST 可访问的变量范围与记述有联机 ST 的梯形图程序的 POU 相同。

### ● 联机 ST 的限制规格

项目	内容
每条回路的联机 ST 数	1 个

## 6-5-3 ST 语言

ST (Structured Text) 语言是指 IEC 61131-3 规定的工业控制 (主要为可编程控制器) 的高级语言。使用预备的控制语句、运算符、函数, 适用于对梯形图语言中难以记述的数值运算处理进行部分处理。ST 语言有以下特点。

- 备有循环语句、IF THEN ELSE 等控制语句。
- 利用与 C 语言等高级语言相同的记述或包括汉字在内的文字注释, 可创建简单易懂的程序。

```

1
2 // TableNo決定
3 FOR i:=0 TO ItemNum DO
4
5     IF (MinNo[i] <= ItemBox[i]) AND (ItemBox[i] <= MaxNo[i]) THEN // 範囲内
6         TableNo[i] := ItemBox[i];
7         RangeOK[i] := TRUE;
8
9     ELSIF (ItemBox[i] > MaxNo[i]) THEN // 上限超え
10        TableNo[i] := MaxNo[i];
11        RangeOK[i] := FALSE;
12
13    ELSE // 下限超え
14        TableNo[i] := MinNo[i];
15        RangeOK[i] := FALSE;
16
17    END_IF;
18
19 END_FOR;

```

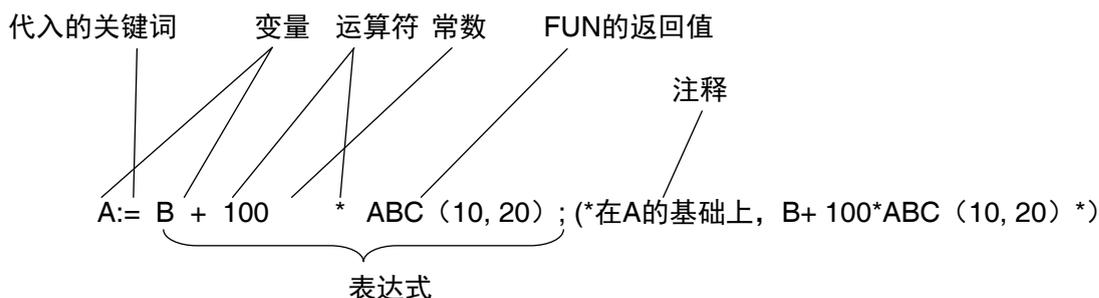
## ST 语言的构成

ST 语言的程序由 1 个以上的语句构成。1 个语句表示 1 个处理。

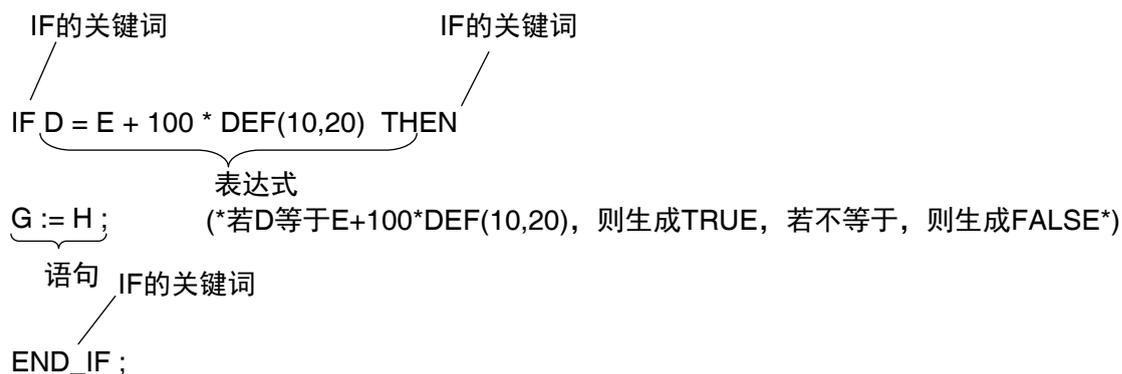
语句从上往下逐行执行，ST 语言程序（ST 程序）的处理逐步推进。

语句由关键词和表达式构成。关键词是指表示代入或执行控制的符号或字符串。表达式是指用变量、常数、FUN 的返回值或将它们与运算符组合，来生成数值的记述。即，语句表示单独结束的处理，而表达式可将生成的数值和关键词组合起来，构成语句。

代入语句的示例：



IF 语句的示例：



## ST 语言的表现

### ● 语句的分隔

- 语句必须以半角分号 (;) 结束。只是换行将不视为语句的结束。因此，可以多行记述较长的语句。
- 1 个语句中带 1 个半角分号 (;)。以下示例中，1 个 IF 语句中有 1 个代入语句，因此要在各语句的最后记述半角分号 (;)。

```
IF A=B THEN
  C := D; } 代入语句 } IF语句
END_IF;
```

## ● 注释

- 为了便于阅读者更容易理解程序内容，可在程序中记述任意注释。
- 记述为注释的语句不会执行。
- 注释的记述有以下两种。

注释的记述	例	备注
用半角括号和半角星号 (* 和 *) 括起来。	(* 多行注释 IF ErrCode = 3 THEN Value := 1000; END_IF; 为止 *)	可记述多行注释。 注释不可嵌套。
以两个半角斜杠 // 开始，以换行结束。	// 注释 // A := SIN (X) ^2 ;	只能逐行注释掉代码。

## ● 空格（空白）、换行、制表符

- 可将空格（空白）、换行、制表符或它们的任意组合以任意的数量记述在任意的位罝中。  
因此，考虑到可读性，可在运算符 / 关键词和表达式之间换行，或在语句前面输入空格或制表符。
- 运算符 / 关键词和变量名称之间请务必输入空格（空白）、换行、制表符。

例) ■表示应该输入空格（空白）、换行、制表符的地方。

```
IF ■ A>0 ■ THEN ■ X:=10;
ELSE
  X:=0;
END_IF;
```

## ● 小写 / 大写字母、半角 / 全角

- 运算符 / 关键词及变量名称不区分大小写（哪种都可以）。
- 请输入半角的运算符 / 关键词。若以全角输入，将发生语法错误。

## ● 变量名称的禁用字符

变量名称的限制请参考□□「6-3-12 变量名称等程序相关名称的限制」(P.6-73)。

## ● 字符串

字符串的限制请参考□□「6-3-12 变量名称等程序相关名称的限制」(P.6-73)。

## ST 语言的关键词和运算符

### ● 语句的关键词

关键词	含义	例
:=	代入	d := 10;
	FB/FUN 的调用	FBname (para1 := 10, para2 := 20) ; 详情请参考□□「FB 调用」(P.6-111)
RETURN	返回	
IF	选择	IF d < e THEN f := 1; ELSIF d = e THEN f :=2; ELSE f := 3; END_IF;
CASE	多方向分支	CASE f OF 1: g :=11; 2: g :=12; ELSE g :=0; END_CASE;
FOR	重复 (1)	FOR i := 100 TO 1 BY -1 DO Val[ i ] := i; END_FOR;
WHILE	重复 (2)	WHILE Val < MaxVal DO Val := Val + 1; END_WHILE;
REPEAT	重复 (3)	REPEAT Val := Val + 1; UNTIL ( Val > 4 ) END_REPEAT;
EXIT	循环的中断	FOR i := 1 TO 100 DO FOR j := 1 TO 10 DO IF Val[ i, j ]>100 THEN EXIT; END_IF; END_FOR; END_FOR;
;	空语句	Val[ i ] := i ; (* 空语句 *)  WHILE (Var <>0) DO ; (* 空语句 *) END_WHILE;
(* 文本 *)	注释	(* 注释掉多行 IF MyFun (ErrorCode) = 3 THEN ReturnValue := GetDetail ( ) ; END_IF; 为止 *)
// 文本	注释	A := SIN ( X ) ^ 2 + COS ( Y ) ^ 2 + 10; // A := SIN ( X ) ^ 2 + COS ( Y ) ^ 2 + 5;

## ● 运算符

下表表示运算符的种类和优先顺序。

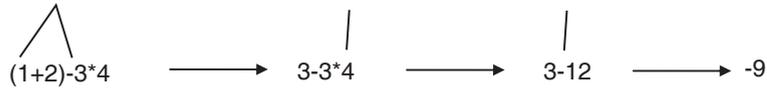
一个表达式中有不同优先顺序的运算符时，先执行优先顺序高的运算。一个表达式中最多可记述 64 个运算符。

例)  $X := (1+2) - 3*4$ ; 时，变量 X 中代入 -9。

1. 进行优先顺序为1的括号内的运算。

2. 进行优先顺序为5的乘法运算。

3. 进行优先顺序为6的减法运算。



运算	运算符	记述示例和评估值	优先顺序
括号	()	$(1+2) * (3+4)$ 值 21	1
FB、FUN 的调用		FUN1( FUN2( Var2A, Var2B), Var1B) 分层调用了 FB、FUN 时，将先执行下面的层级。上述示例中，先执行 FUN2，再执行 FUN1。	2
符号	+, -	+100 -100	3
否定	NOT	NOT TRUE 值 FALSE	
乘幂	**	-2**2 值 4 负号的优先顺序高于乘幂运算符。因此，上述示例中，-2**2=(-2)**2，值为 4。  2**3**2 值 64 乘幂运算符有多个重叠时，从左开始运算。上述示例中，2**3**2=(2**3)**2，值为 64。	4
乘法	*	100*200 值 20000	5
除法	/	100/200 值 0.5	
余数	MOD	10 MOD 7 值 3  -17 MOD 6 值 -5  -17 MOD (-6) 值 -5  17 MOD 6 值 5  17 MOD (-6) 值 5	
加法	+	100+200 值 300	6
减法	-	100-200 值 -100	

运算	运算符	记述示例和评估值	优先顺序
比较	<, >, <=, >=	100<200 若比较结果正确, 则值为 TRUE, 若不正确, 则值为 FALSE。 上述示例中, 100<200 是正确的, 所以值为 TRUE	7
等于	=	100=200 若两个值相等, 则值为 TRUE, 若不相等, 则值为 FALSE。 上述示例中, 100=200 是错误的, 所以值为 FALSE	8
不等于	<>	100<>200 若两个值不相等, 则值为 TRUE, 若相等, 则值为 FALSE。 上述示例中, 100<>200 是不相等的, 所以值为 TRUE	
逻辑与	AND,&	所有位执行 1 个位的逻辑与。 1 个位的逻辑与结果如下所示。 0 AND 0 = 0 0 AND 1 = 0 1 AND 0 = 0 1 AND 1 = 1  0101 AND 1100 值 0100	9
异或	XOR	所有位执行 1 个位的异或。 1 个位的异或结果如下所示。 0 XOR 0 = 0 0 XOR 1 = 1 1 XOR 0 = 1 1 XOR 1 = 0  0101 XOR 1100 值 1001	10
逻辑或	OR	所有位执行 1 个位的逻辑或。 1 个位的逻辑或结果如下所示。 0 OR 0 = 0 0 OR 1 = 1 1 OR 0 = 1 1 OR 1 = 1  0101 OR 1100 值 1101	11



### 使用注意事项

若分层记述同一个函数, 可能不按预期动作。请务必如以下示例所示, 分开记述语句。

例) ×     out     := MyFunc( In1:=x1, In2:=MyFunc( In1:=x2, In2:=x3 ) );

○     tmp     := MyFunc( In1:=x2, In2:=x3 );  
      out     := MyFunc( In1:=x1, In2:=tmp );



### 使用注意事项

运算符的优先顺序可能因规格或制造商不同而异。特别是乘幂运算符的优先顺序，需要注意。为了确保按计划顺序进行运算，建议使用括号标明各运算的优先顺序。

例)  $X := -2^{**3^{**4}}$ ; 记述为  $X := ((-2)^{**3})^{**4}$ 。



### 参考

根据数据类型进行运算。

例如，整数型之间的运算以整数型进行。因此，如果以 INT 型变量  $A = 3$ 、 $B = 2$  进行  $A/B$  运算，运算结果不是 1.5，将舍去小数点后的数位，变为 1。所以， $(A/B) * 2$  的运算结果不是 3，而是 2。

## ● 各运算符的操作对象中可使用的数据类型

1 个运算符中使用的操作对象全部为同一数据类型时，可将以下数据类型设定为操作对象。

但是，若运算符中设定了不同数据类型的操作对象，将发生默认的 CAST。

关于默认的 CAST，请参考□□「默认的 CAST」(P.6-117)。

数据类型	代入运算符	参数设定运算符	数值运算符	余数运算符	乘幂运算符	比较运算符	等价运算符	逻辑运算	正负符号
	:=	:= =>	+ - * /	MOD	**	< <= => >	= <>	NOT AND & OR XOR	+ -
BOOL 型	○	○	×	×	×	×	○	○	×
位列型	○	○	×	×	×	×	○	○	×
整数型	○	○	○	○	○	○	○	×	○
实型数据	○	○	○	×	○	○	○	×	○
持续时间型	○	○	×	×	×	×	×	×	○
日期型	○	○	×	×	×	×	×	×	×
时刻型	○	○	×	×	×	×	×	×	×
日期时刻	○	○	×	×	×	×	×	×	×
字符串型	○	○	×	×	×	×	×	×	×
列举型	○	○	×	×	×	×	○	×	×
母结构体	○	○	×	×	×	×	×	×	×
母排列	○	○	×	×	×	×	×	×	×

○：可以

×：链接时将发生异常。

\* 字符串型变量的比较请使用指令 (EQascii 等) 执行，而不是运算符。

## ST 语言的语句

### ● 代入

#### 概要:

将右边（表达式的值）代入左边（变量）。

#### 保留字:

:=

冒号（:）和等号（=）的组合

#### 语句的结构:

```
< 变量 >:=< 表达式、变量或常数 >;
```

#### 用途:

将值代入变量时使用。

例如，用于初始值的设定、计算结果的保存等。

#### 说明:

将 < 表达式的值 > 代入（保存到） < 变量 > 中。

#### 例:

例 1) 将表达式 X+1 的结果代入变量 A 中

```
A:=X+1;
```

例 2) 将变量 B 的值代入变量 A 中

```
A:=B;
```

例 3) 将常数 10 代入变量 A 中

```
A:=10;
```

#### 注意事项:

- 代入源的数据类型需要与代入目标的数据类型一致，或使用默认 CAST 中允许的数据类型组合。如果不是，链接时将发生异常。
- 代入源为字符串时，代入目标 STRING 型变量的大小应大于代入源字符串的大小。若非如此，将发生异常。
- 变量为 STRING 型时，如果（左边的变量大小）>（右边变量中保存的字符串大小），则可以代入。  
例）  
以下情况下可以代入。
- 变量表:

变量名称	数据类型	大小
Var1	STRING	10
Var2	STRING	20

- 用户程序：  
Var2 := 'ABC';  
Var1 := Var2;

不可代入到整个联合体变量中。请以成员为单位代入。

### ● RETURN

#### 概要：

根据使用 ST 的程序种类，如下所示。

< ST 程序 >

中途结束 ST 程序，执行下一个程序。

< 在 FB 的实例内 FUN 中使用的 ST >

中途强制结束 FB、FUN，执行 FB、FUN 调用源的下一个指令。

< 联机 ST >

结束包括 RETURN 语句的联机 ST 的 POU。

#### 保留字：

RETURN

#### 语句的结构：

```
RETURN;
```

#### 用途：

强制结束正在执行的程序、FB 或 FUN 时使用。

## ● IF (1 个条件)

### 概要:

某个条件成立时, 执行指定的语句。未成立时, 执行其他语句。

条件成立、未成立时, 表现如下。

真: 条件成立时

假: 条件未成立时

### 保留字:

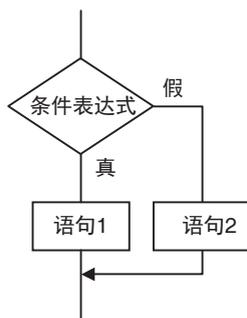
IF、THEN、(ELSE)、END\_IF

注: ELSE 可省略。

### 语句的结构:

```
IF < 条件式 > THEN
  < 语句 1>;
ELSE
  < 语句 2>;
END_IF;
```

### 处理流程图:



### 用途:

根据某个条件 (条件式) 是否成立来区分时使用。

### 说明:

< 条件式 > = 真时, 执行 < 语句 1>。

< 条件式 > = 假时, 执行 < 语句 2>。

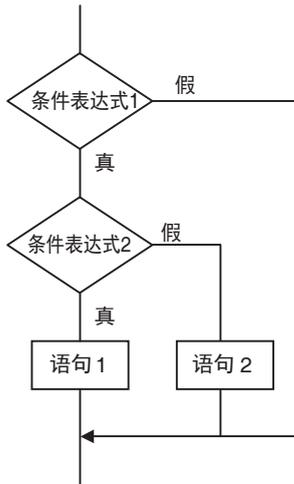
### 注意事项:

- IF 必须与 END\_IF 成对使用。
- < 条件式 > 中可记述评估结果为真或假的表达式 (例如 IF A>10)、BOOL 型变量 (例如 IF A)。
- < 语句 1>、< 语句 2> 中可跨多行记述程序。此时, 语句和语句之间用 ; 分隔。

例) 在 < 语句 1> 中添加了 IF 语句时

```
IF < 条件式 1> THEN
  IF < 条件式 2> THEN
    < 语句 1>;
  ELSE
    < 语句 2>;
  END_IF;
END_IF;
```

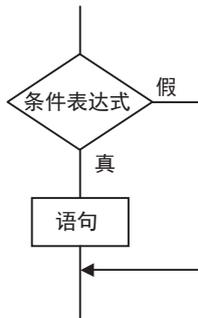
处理流程图：



如上所示， ELSE 与之前的 THEN 对应。

- < 语句 1>、< 语句 2> 中，均可执行 1 个以上的语句。此时，语句和语句之间用 ; 分隔。
- ELSE 语句可省略。省略时，如果 < 条件式 >= 假，将不执行任何内容。

处理流程图：



例：

例 1) 变量 A>0 为真时，在变量 X 中代入数值 10。变量 A>0 为假时，在变量 X 中代入数值 0。

```

IF A>0 THEN
  X:=10;
ELSE
  X:=0;
END_IF;
  
```

例 2) 变量 A>0 且变量 B>1 为真时，在变量 X 中代入数值 10，在变量 Y 中代入数值 20。变量 A>0 且变量 B>1 为假时，在变量 X 中代入数值 0，在变量 Y 中代入数值 0。

```

IF A>0 AND B>1 THEN
  X:=10;Y:=20;
ELSE
  X:=0;Y:=0;
END_IF;
  
```

例 3) BOOL 型变量 A = TRUE 时, 在变量 X 中代入数值 10。变量 A=FALSE 为假时, 在变量 X 中代入数值 0。

```
IF A THEN X:=10;
ELSE X:=0;
END_IF;
```

## ● IF (多个条件)

### 概要:

某个条件成立时, 执行指定的语句。未成立时且其他条件成立时, 执行其他语句。均未成立时, 执行其他语句。

条件成立、未成立时, 表现如下。

真: 条件成立时

假: 条件未成立时

### 保留字:

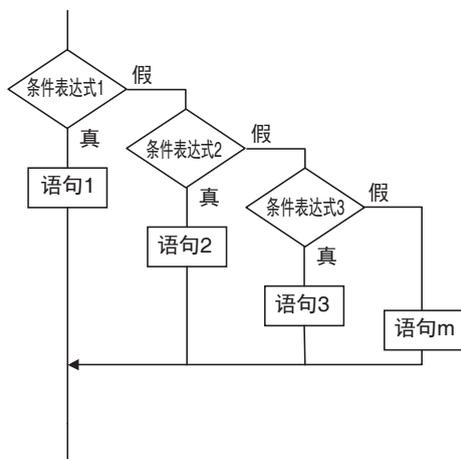
IF、THEN、ELSIF、(ELSE)、END\_IF

注: ELSE 可省略。

### 语句的结构:

```
IF <条件式 1> THEN <语句 1>;
  ELSIF <条件式 2> THEN <语句 2>;
  ELSIF <条件式 3> THEN <语句 3>;
  .
  .
  .
  ELSIF <条件式 n> THEN <语句 n>;
ELSE <语句 m>;
END_IF;
```

### 处理流程图:



### 用途:

根据多个条件 (条件式) 是否成立来区分时使用。

**说明：**

< 条件式 1> = 真时，执行 < 语句 1>。  
 < 条件式 1> = 假时，  
 如果 < 条件式 2> = 真，执行 < 语句 2>。  
 < 条件式 2> = 假时，  
 如果 < 条件式 3> = 真，执行 < 语句 3>。  
 .  
 .  
 .  
 如果 < 条件式 n> = 真，执行 < 语句 n>。  
 不符合任一条件时，执行 < 语句 m>。

**注意事项：**

- IF 必须与 END\_IF 成对使用。
- < 条件式□> 中记述评估结果为真或假的表达式。例) IF (A>10)  
 此外，< 条件式□> 中除了表达式外，还可只指定 BOOL 型变量（包括 FUN 的 BOOL 型返回值）。  
 此时，变量的值为 TRUE = 评估结果真，FALSE = 评估结果假。
- < 语句□> 中可跨多行记述程序。此时，语句和语句之间用 ; 分隔。
- ELSE 语句可省略。省略时，若不符合任一条件，则不执行任何内容。

**例：**

变量 A>0 为真时，在变量 X 中代入数值 10。  
 变量 A>0 为假时，如果变量 B = 1 为真，则在变量 X 中代入数值 1。  
 变量 A>0 为假时，如果变量 B = 2 为真，则在变量 X 中代入数值 2。  
 不符合任一条件时，在变量 X 中代入数值 0。

```
IF A>0 THEN X:=10;
  ELSIF B=1 THEN X:=1;
  ELSIF B=2 THEN X:=2;
ELSE X:=0;
END_IF;
```

## ● CASE

### 概要:

执行拥有与整数式的值一致的整数选择值的语句。

### 保留:

CASE

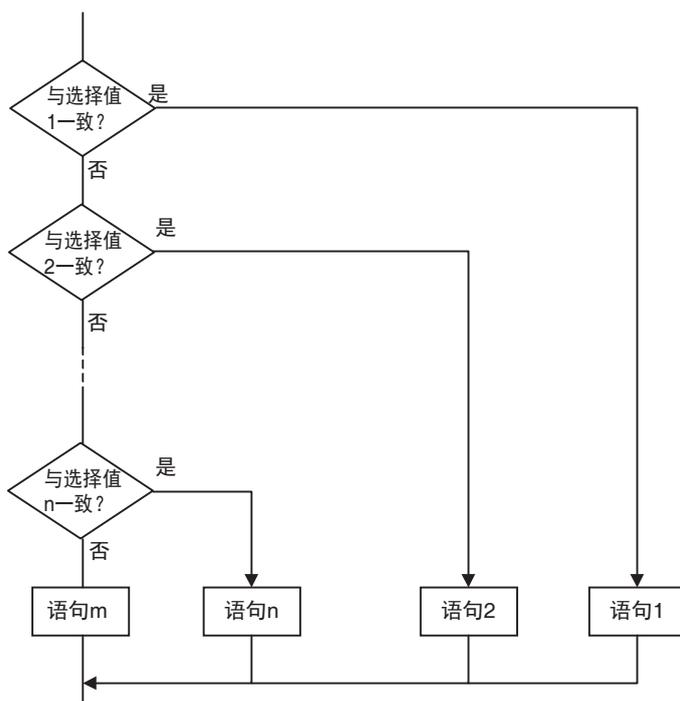
### 语句的结构:

```

CASE < 整数式 > OF
  < 整数式的值 1>:< 语句 1>;
  < 整数式的值 2>:< 语句 2>;
  .
  .
  .
  < 整数式的值 n>:< 语句 n>;
ELSE< 语句 m>;
END_CASE;

```

### 处理流程图:



### 用途:

根据某个整数的值进行区分时使用。

### 说明:

< 整数式 > 与 < 整数式的值 n > 一致时, 执行 < 语句 n >。

< 整数式 > 与任一 < 整数式的值 n > 都不一致时, 执行 < 语句 m >。

**注意事项:**

- CASE 必须与 END\_CASE 成对使用。
- 在 < 整数式 > 中记述
  - 整数型或枚举型变量 (例: abc)
  - 整数式 (例: abc+def)
  - 将整数值作为返回值的 FUN (例: xyz ())
- < 语句□ > 中可跨多行记述程序。此时, 语句和语句之间用 ; 分隔。
- 在 < 整数式的值 n > 中指定多个整数的 OR 逻辑时, 数值用 , (逗号) 分隔。指定连续的整数时, 开始整数和结束整数之间以 .. (2 个句点) 分隔。

例 1) 可为特定的整数值指定条件, 或为多个整数值指定相同的条件。

CASE A OF		
1: X:=1;	-----	变量A为1时, 在变量X中代入数值1
2: X:=2;	-----	变量A为2时, 在变量X中代入数值2
3: X:=3;	-----	变量A为3时, 在变量X中代入数值3
ELSE		
X:=0;	-----	不符合上述任一情况时, 在变量X中代入0
END_CASE;		

CASE A OF		
1: X:=1;	-----	变量A为1时, 在变量X中代入数值1
2,5: X:=2;	-----	变量A为2或5时, 在变量X中代入数值2
6..10: X:=3;	-----	变量A为6~10中的任意一个时, 在变量X中代入数值3
11,12,15..20: X:=4;	-----	变量A为11、12或15~20时, 在变量X中代入数值4
ELSE		
X:=0;	-----	不符合上述任一情况时, 在变量X中代入0
END_CASE;		

例 2) < 整数式 > 中可记述整数型变量 · 表达式 · FUN 的返回值, 以及枚举型变量 · FUN 的返回值。  
示例如下。

- 整数 · 枚举型变量的示例

```
CASE ColorVar OF
  RED:
    X := 0;
  BLUE:
    X := 1;
  ELSE
    X := 2;
END_CASE;
```

- 整数式示例

```
CASE (a1 + a2) OF
  0:
    X := 0;
  1:
    X := 1;
  ELSE
    X := 2;
END_CASE;
```

- 整数·列举型 FUN 的返回值示例

```

CASE FUN() OF
  0: ----- 根据FUN()的返回值分支
      X := 10;
  1:
      X := 11;
  ELSE
      X := 12;
END_CASE;

```

### CASE 中可使用的数据类型

分类	数据类型		< 整数式 >
基本数据类型	整数型		○
	布尔型、位列型、实数型、持续时间型、日期型、时刻型、日期时刻型、字符串型		×
数据类型的指定	排列指定	排列整体	×
		要素	○ 仅可指定整数型、列举型
派生数据类型	结构体型	结构体整体	×
		成员	○ 仅可指定整数型、列举型
	联合体型	联合体整体	×
		成员	○ 仅可指定整数型、列举型
	列举型		○

## ● FOR

### 概要:

变量（以下称“反复变量”）从某个值变为某个值前，重复执行某语句。

条件成立、未成立时，表现如下。

真：条件成立时

假：条件未成立时

### 保留字:

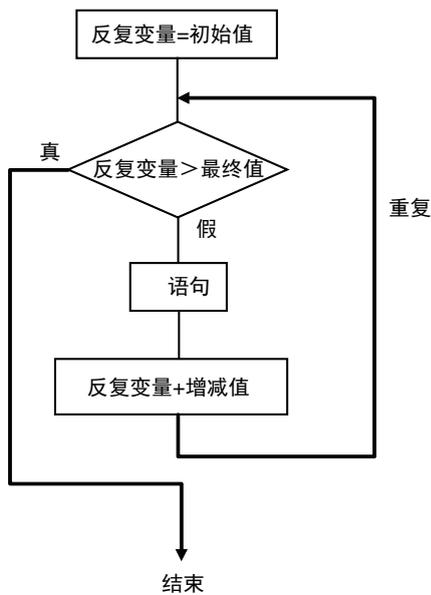
FOR、TO、(BY)、DO、END\_FOR

注：BY 可省略。

### 语句的结构:

```
FOR <反复变量> := <初始值> TO <最终值> BY <增减值> DO
  <语句>;
END_FOR;
```

### 处理流程图:



### 用途:

已事先知道重复次数时使用。

特别是根据某个反复变量的值切换排列指定的变量各要素并指定时，为有效。

**说明：**

评估 < 初始值 >、< 最终值 >、< 增减值 > 并决定。

< 反复变量 > 为 < 初始值 > 时，执行 < 语句 >。

执行后，在 < 反复变量 > 中加上 < 增减值 > 中的值，如果 < 反复变量 > < 最终值 > 的值，则执行 < 语句 >。

执行后，在 < 反复变量 > 中加上 < 增减值 > 中的值，如果 < 反复变量 > < 最终值 > 的值，则执行 < 语句 >。

重复上述内容。

如果 < 反复变量 > > < 最终值 >，则结束。

此外，< 增减值 > 为负数时，大小比较符号的方向相反。

**注意事项：**

- 反复变量为带符号整数型时，< 增减值 > 可指定为负值。
- FOR 必须与 END\_FOR 成对使用。
- 反复变量中，以最终值执行处理后，变为（最终值 + 增减值），结束反复处理。

例) 在以下 ST 语句中，FOR 语句结束时，i 的值为 101。

```
FOR i:=0 TO 100 DO
  X[i]:=0;
END_FOR;
// 此时 i 为 101
```

- 请勿记述要在 FOR 语句内直接变更反复变量的处理。可能发生意外动作。  
例)

```
FOR i:=0 TO 100 BY 1 DO
  X[i]:=0;
  i:=i+INT#5;
END_FOR;
```

- < 语句 > 中可跨多行记述程序。此时，语句和语句之间用 ; 分隔。
- BY< 增减值 > 可省略。省略时，BY 作为 1 执行。
- < 初始值 >、< 最终值 >、< 增减值 > 中可指定整数型（SINT、INT、DINT、LINT、USINT、UINT、UDINT 或 ULINT）的变量或整数值。此外，也可指定 FUN 的整数型返回值。

例 1) 反复变量从 n = 0 开始，以 5 为单位直至 50，不断重复，在排列指定变量 SP[n] 中代入 100。

```
FOR n := 0 TO 50 BY 5 DO
  SP[n] := 100;
END_FOR;
```

例 2) 计算排列指定的变量 DATA[n] 的要素 DATA[1] 到 DATA[50] 的合计值，代入变量 SUM 中。

```
IF a THEN
  FOR n := 0 TO 50 BY 1 DO
    DATA[n]:= 1 ;
  END_FOR;

  FOR n := 0 TO 50 BY 1 DO
    SUM:= SUM + DATA[n] ;
  END_FOR;

  a:=FALSE;
END_IF;
```

例 3) 计算排列指定的变量 DATA[n] 的要素 DATA[1] 到 DATA[50] 的合计值，代入变量 SUM 中。最大值代入变量 MAX 中，最小值代入变量 MIN 中。此外，DATA[n] 的值在 0~1000 的范围内。

```
MAX :=0;
MIN :=1000;
FOR n :=1 TO 50 BY 1 DO
  IF DATA[n] > MAX THEN
    MAX :=DATA[n];
  END_IF;
  IF DATA[n] < MIN THEN
    MIN :=DATA[n];
  END_IF;
END_FOR;
```

- 在<反复变量>的<初始值>上加<增减值>，超过<最终值>前，若包括FOR语句在内的程序执行时间超出任务周期设定时间，将发生任务周期超限。

- 逻辑上 <反复变量> 不会达到 <最终值> 时

例)

```
FOR i := 0 TO 100 BY 1 DO
  intArray[i] := i;
  i := INT#50;
END_FOR;
```

----- 变为无限循环，发生任务周期超限

例)

```
FOR i := 0 TO 100 BY 0 DO
;
END_FOR;
```

----- 变为无限循环，发生任务周期超限

- < 反复变量 > 超过 < 最终值 > 时，发生溢出 / 下溢的情况  
例)  
变量 i 为 USINT 型时，可表现最终值 256。将视为 0，变为无限循环。

```
FOR i := 0 TO 254 BY 2 DO
  INTArray[i] := i;
END_FOR;
```



### 参考

< 最终值 > 和 < 增减值 > 中可记述运算式。

但是，< 最终值 > 和 < 增减值 > 的评估在 FOR 循环内的运算前执行。开始 FOR 循环内的运算后，< 最终值 > 和 < 增减值 > 不会发生变化。

下图情况下，< 最终值 > 为 10、< 增减值 > 为 3。即使开始 FOR 循环内的运算、变量 A 或变量 C 的值发生变化，< 最终值 > 保持 10，< 增减值 > 保持 3。

```
A := INT#1;
B := INT#2;
C := INT#10;

FOR i := 0 TO C BY A+B DO
  INTArray[i] := i;
  A := B + i;
  C := C + i;
END_FOR;
```

## FOR 中可使用的数据类型

分类	数据类型		< 反复变量 >、< 初始值 >、< 最终值 >、< 增减值 > <sup>*1</sup>
基本数据类型	布尔型、位列型、实数型、持续时间型、日期型、时刻型、日期时刻型、字符串型		×
	整数型		○
数据类型的指定	排列指定	排列整体	×
		要素	○ 仅可指定整数型、列举型 <sup>*2</sup>
派生数据类型	结构体型	结构体整体	×
		成员	○ 仅可指定整数型、列举型 <sup>*2</sup>
	联合体型	联合体整体	×
		成员	×
列举型		○ <sup>*2</sup>	

\*1. 请统一< 反复变量 >、< 最终值 >、< 增减值 > 三者的数据类型。若数据类型不一致，Sysmac Studio 的程序链接时将发生异常。

\*2. < 反复变量 >、< 最终值 >、< 增减值 > 中不可使用列举型。

## ● WHILE

### 概要:

某条件式为真期间，重复执行指定的语句。

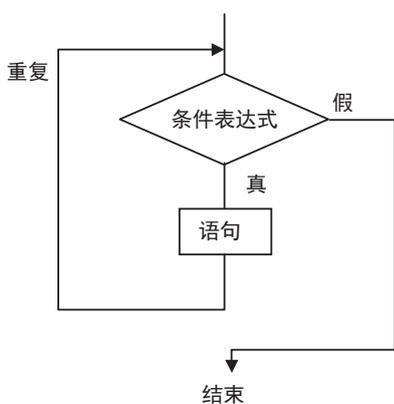
### 保留字:

WHILE、DO、END\_WHILE

### 语句的结构:

```
WHILE < 条件式 > DO  
  < 语句 >;  
END_WHILE;
```

### 处理流程图:



### 用途:

不知道重复次数（根据条件，不知道几次）时，某条件式成立期间，重复执行某处理时使用。仅在条件式成立时执行某处理时使用（前判定的重复）。

### 说明:

执行 < 语句 > 前，对 < 条件式 > 进行评估。

< 条件式 > 为真时，执行 < 语句 >。然后，重新评估 < 条件式 >。重复执行上述内容。

< 条件式 > 为假时，不执行 < 语句 >，结束 < 条件式 > 的评估。

**注意事项:**

- WHILE 必须与 END\_WHILE 成对使用。
- 执行 < 语句 > 前, 若 < 条件式 > 为假, 将不执行 < 语句 > 并结束。
- < 语句 1>、< 语句 2> 中可跨多行记述程序。此时, 语句和语句之间用 ; 分隔。
- < 语句 > 中, 可执行 1 个以上的语句。此时, 语句和语句之间用 ; 分隔。
- < 条件式 > 中除了表达式外, 还可只指定 BOOL 型变量 (包括 FUN 的 BOOL 型返回值)。

例 1) 计算 7 的倍数且大于 1000 的数字, 代入变量 A 中。

```
A := 0;
WHILE A <= 1000 DO
  A := A+INT#7;
END_WHILE;
```

例 2) 变量 X < 3000 期间, 将 X 的值乘以 2, 然后将该值代入排列指定的变量 DATA[1] 中。然后, 将 X 的值再乘以 2, 将该值代入排列指定的变量 DATA[2] 中。重复上述内容。

```
n := 1;
X := 1;
WHILE X < 3000 DO
  X:= X*INT#10#2;
  DATA[n]:= X;
  n := n+INT#1;
END_WHILE;
```

- < 条件式 > 的撰写不正确时, 程序执行时间将延长, 可能发生任务周期超限。

例)

```
boolVar := TRUE;
WHILE boolVar DO
  intVar := intVar + INT#1;
END_WHILE;
```

## ● REPEAT

条件成立、未成立时，表现如下。

真：条件成立时

假：条件未成立时

### 概要：

执行某语句，在某条件式变为真前，重复该语句。

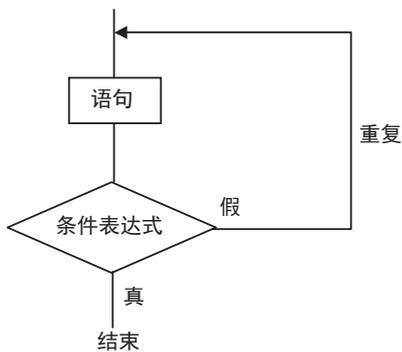
### 保留字：

REPEAT、UNTIL、END\_REPEAT

### 语句的结构：

```
REPEAT
  < 语句 >;
UNTIL < 条件式 >
END_REPEAT;
```

### 处理流程图：



### 用途：

不知道重复次数（根据条件，不知道几次）时，某处理后的条件式成立期间，重复执行该处理时使用。根据某处理的执行结果，判断是否重复时使用（后判定的重复）。

### 说明：

< 语句 > 无条件最先执行。然后，评估 < 条件式 >。

< 条件式 > 为假时，重新执行 < 语句 >。

< 条件式 > 为真时，不执行 < 语句 >，直接结束。

### 注意事项：

- REPEAT 必须与 END\_REPEAT 成对使用。
- 执行 < 语句 > 前，即使 < 条件式 > 为真，也会执行 < 语句 >。  
即，< 语句 > 至少会执行 1 次。
- < 语句 > 中，可跨行记述定为语句关键词的语句。此时，语句和语句之间用 ; 分隔。
- < 条件式 > 中除了表达式外，还可只指定 BOOL 型变量（包括 FUN 的 BOOL 型返回值）。

例) 1 到 10 的数字相加, 然后代入变量 TOTAL 中。

```
A := 1;
TOTAL := 0;
REPEAT
    TOTAL := TOTAL + A;
    A := A+INT#1;
UNTIL A>10
END_REPEAT;
```

- < 条件式 > 的撰写不正确时, 程序执行时间将延长, 可能发生任务周期超限。

例)

```
intVar := INT#1;
REPEAT
    intVar := intVar + INT#1;
UNTIL intVar = INT#0
END_REPEAT;
```

## ● EXIT

### 概要:

仅在重复语句（FOR 语句、WHILE 语句、REPEAT 语句）中使用，结束重复语句。  
若在 IF 语句中使用，某条件成立时，可在中途跳过重复语句。

### 保留字:

EXIT

### 语句的结构（例：在 IF 语句中使用）:

```
FOR (WHILE、REPEAT) 语句
.
.
.
IF < 条件式 > THEN EXIT;
END_IF;
.
.
.
END_FOR (WHILE、REPEAT) ;
```

### 用途:

结束条件成立前，要强制结束重复处理时使用。

### 说明（例：在 IF 语句中使用）:

< 条件式 > 为真时，强制结束重复语句（FOR 语句、WHILE 语句、REPEAT 语句），EXIT 后面记述的语句将不执行。

(注) 1 < 条件式 > 中除了表达式外，还可只指定 BOOL 型变量。

2 执行 < 语句 > 前，即使 < 条件式 > 为真，也会执行 < 语句 >。

### 例:

例 1) 变量从  $n = 1$  开始，以 1 为单位直至 50，不断重复，在排列指定变量 DATA[n] 中加 n。但是，DATA[n] 超过 100 后，将结束。

```
IF A THEN
  DATA[3] := 98;
  FOR n := 1 TO 50 BY 1 DO
    DATA[n] := DATA[n] + n;
    IF DATA[n] > 100 THEN EXIT;
  END_IF;
END_FOR;
A := FALSE;
END_IF;
```

## ● FB 调用

### 概要:

调用 FB。

**保留字:** 无

### 语句的结构:

FB 的实例名称后面，在 ( ) 内记述参数指定 (将指定变量的值传递到被调用方 FB 的输出变量) 和返回值指定 (将被调用方 FB 的输出变量值代入到指定变量)。

以下①或②的任意记述方式均可。

出于用户程序的可读性考虑，推荐①。

### 记述方式①:

所调用 FB 的变量名称和调用方的参数名称都要记述时

ABC (A:=x1, B:=x2, C=>y1) ;

ABC : FB 的实例名称

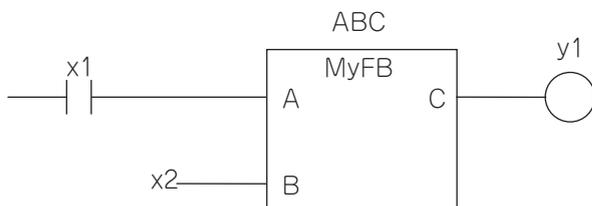
A、B : 所调用 FB 的输入变量名称或输入输出变量名称

x1、x2 : 调用方的输入参数或输入输出参数 (包括常数)

C : 所调用 FB 的输出变量名称

y1 : 调用方的输出参数

### • 梯形图语言中的表现



- 各参数或返回值的记述顺序为任意。
- 输入变量名称和输入参数名称可省略。省略时，上次调用时代入到输入变量的值将代入到输入变量中。第一次调用时，将在输入变量中代入初始值。
- 输出变量和输出参数可省略。省略时，输出变量的值将不会代入到任何地方。

**记述方式②:**

省略所调用 FB 的变量名称，仅记述调用方的参数名称时

ABC (x1, x2, y1) ;

ABC: FB 的实例名称

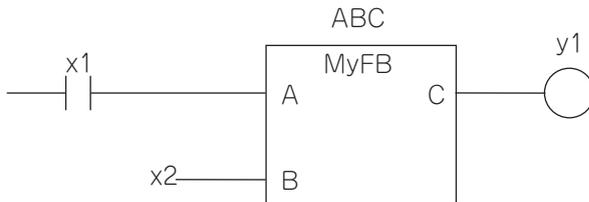
A、B: 省略。(所调用 FB 的输入变量名称或输入输出变量名称)

x1、x2: 调用方的输入参数或输入输出参数 (包括常数)

C: 省略。(所调用 FB 的输出变量名称或常数)。

y1: 调用方的输出参数

- 梯形图语言中的表现



- 所记述参数的顺序根据 FB 的定义决定。按 FB 的本地变量定义从上往下的顺序。

**用途:**

调用 FB。

**例**

- 程序

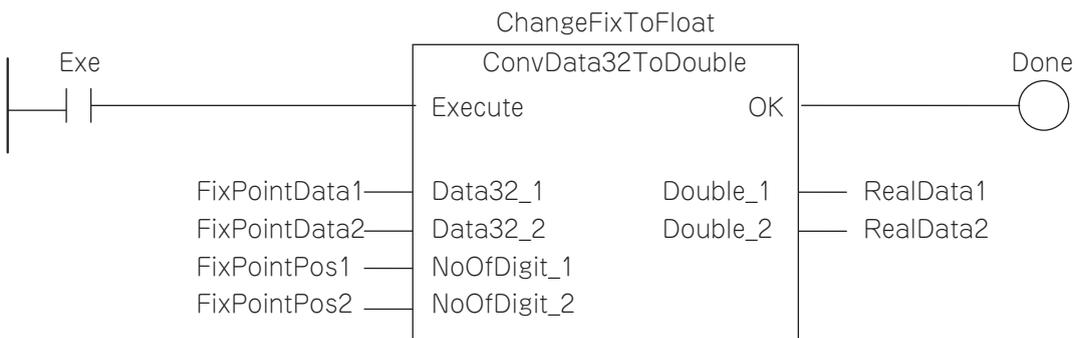
- 记述①

```
ChangeFixToFloat (Execute:=Exe,Data32_1:=FixPointData1,
  Data32_2:=FixPointData2,
  NoOfDigit_1:=FixPointPos1,
  NoOfDigit_2:=FixPointPos2,OK=>Done,Double_1=>RealData1,
  Double_2=>RealData2) ;
```

- 记述②

```
ChangeFixToFloat (Exe, FixPointData1, FixPointData2, FixPointPos1, FixPointPos2,
  Done, RealData1, RealData2) ;
```

- 梯形图语言中的表现



- 功能块的定义

功能块名称：ConvData32ToDouble

功能块的变量

输入 / 输出	变量名称	数据类型
输入变量	Execute	BOOL
	Data32_1	DINT
	Data32_2	DINT
	NoOfDigit_1	INT
	NoOfDigit_2	INT
输出变量	OK	BOOL
	Double_1	LREAL
	Double_2	LREAL

- 程序的变量

变量名称	数据类型	注释
ChangeFixToFloat	ConvData32ToDouble	固定小数点→浮动小数点转换
Exe	BOOL	启动触发
FixPointData1	DINT	小数点位置指定数据 #1
FixPointPos1	INT	小数点以下位数 #1
FixPointData2	DINT	小数点位置指定数据 #2
FixPointPos2	INT	小数点以下位数 #2
Done	BOOL	正常结束
RealData1	LREAL	浮动小数点数据 #1
RealData2	LREAL	浮动小数点数据 #2

### 参数的省略可否

调用 FB 时，如果不需要设定参数的值，可省略所记述的参数。

参数的省略可否如下所示。

○：可记述。省略时，将代入初始值，以代替参数。

×：链接时将发生异常。

POU 的种类	所调用 POU 的变量	记述模式		记述的可否
		记述的参数	例	
FB	有记述 (记述方式①)	记述所有参数	instance (x:=a,y:=b,z:=c) ;	○
		记述多个参数	instance (x:=a,y:=b) ;	
		记述 1 个参数	instance (y:=b) ;	
		不记述参数	instance (x:=) ;	
	不记述 (记述方式②)	记述所有参数	instance (a,b,c) ;	○
		不记述所有参数	instance () ;	
		只记述开头的参数	instance (a) ;	×
		记述 1 个参数	instance (a, ) ;	
	记述多个参数	instance (a,b) ;		

## ● FUN 调用

### 概要:

调用 FUN。

### 保留字: 无

### 语句的结构:

记述在代入关键词 (:=) 的左边代入返回值的输出参数。在右边, 在 FUN 名称后面的 ( ) 内记述参数指定 (将指定变量的值传递到被调用方 FUN 的输入变量)。

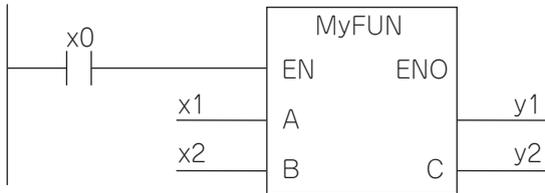
以下①或②的任意记述方式均可。

出于用户程序的可读性考虑, 推荐①。

### 记述方式①:

```
IF (x0=TRUE) THEN
  y1 := MyFUN (A:=x1, B:=x2, C=>y2) ;
END_IF;
```

### • 梯形图语言中的表现



MyFUN : FUN 名称  
 x0 : FUN 的调用有无  
 A, B : 所调用 FUN 的输入变量名称  
 x1, x2 : 所调用 FUN 的输入参数  
 C : 所调用 FUN 的输出变量名称  
 y1 : 所调用 FUN 的返回值保存位置  
 y2 : 所调用 FUN 的输出参数

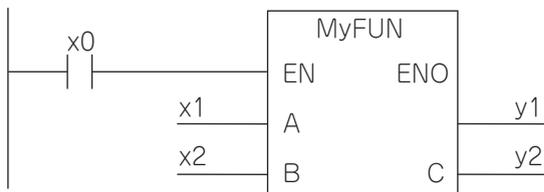
- 各参数的记述顺序为任意。
- 输入变量名称和输入参数名称可省略。省略时, 将在输入变量中代入初始值。
- EN 也可省略。省略时, 在 EN 中代入 TRUE。

**记述方式②:**

省略所调用 FUN 的变量名称, 仅记述调用方的参数名称时

```
IF (x0=TRUE) THEN
  y1 := MyFUN (x1, x2, y2) ;
END_IF;
```

- 梯形图语言中的表现



MyFUN : FUN 名称  
 x0 : FUN 的调用有无  
 A, B : 所调用 FUN 的输入变量名称  
 x1, x2 : 所调用 FUN 的输入参数  
 C : 所调用 FUN 的输出变量名称  
 y1 : 所调用 FUN 的返回值保存位置  
 y2 : 所调用 FUN 的输出参数

- 所记述参数的顺序根据 FUN 的定义决定。按 FUN 的本地变量定义从上往下的顺序。

**例**

- 程序

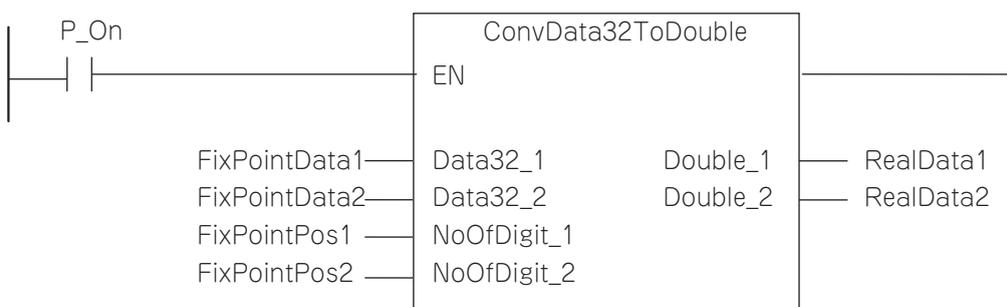
- 记述①

```
ConvData32ToDouble (Data32_1:=FixPointData1,Data32_2:=FixPointData2,
  NoOfDigit_1:=FixPointPos1, NoOfDigit_2:=FixPointPos2,
  Double_1=>RealData1, Double_2=>RealData2) ;
```

- 记述②

```
ConvData32ToDouble (FixPointData1, FixPointData2, FixPointPos1, FixPointPos2,
  RealData1, RealData2) ;
```

- 梯形图语言中的表现



- 函数的定义

函数名称：ConvData32ToDouble

函数的变量

In/Out	变量名称	数据类型
输入变量	Execute	BOOL
	Data32_1	DINT
	Data32_2	DINT
	NoOfDigit_1	INT
	NoOfDigit_2	INT
输出变量	Double_1	LREAL
	Double_2	LREAL
返回值	-	BOOL

- 程序的变量

变量名称	数据类型	注释
ChangeFixToFloat	ConvData32ToDouble	固定小数点→浮动小数点转换
Exe	BOOL	启动触发
FixPointData1	DINT	小数点位置指定数据 #1
FixPointPos1	INT	小数点以下位数 #1
FixPointData2	DINT	小数点位置指定数据 #2
FixPointPos2	INT	小数点以下位数 #2
Done	BOOL	正常结束
RealData1	LREAL	浮动小数点数据 #1
RealData2	LREAL	浮动小数点数据 #2

### 用途：

调用 FUN。

### 参数的省略可否

调用 FUN 时，如果不需要设定参数的值，可省略所记述的参数。

参数的省略可否如下所示。

○：可记述。省略时，将代入初始值，以代替参数。

×：链接时将发生异常。

POU 的种类	所调用 POU 的变量	记述模式		记述的可否
		记述的参数	例	
FUN	有记述 (记述方式①)	记述所有参数	FUN (x:=a,y:=b,z:=c) ;	○
		记述多个参数	FUN (x:=a,y:=b) ;	
		记述 1 个参数	FUN (y:=b) ;	
		不记述参数	FUN (x:=) ;	
	不记述 (记述方式②)	记述所有参数	FUN (a,b,c)	○
		不记述参数	FUN () ;	
		只记述开头的参数	FUN (a) ;	×
		记述 1 个参数	FUN (a, ,) ;	
	记述多个参数	FUN (a,b) ;		

## ST 语言相关的注意事项

用 ST 语言创建用户程序，应注意以下事项。

### ● 默认的 CAST

如下所示，操作对象的数据类型不一致时，将根据默认的 CAST，自动转换数据类型后处理。此外，若不满足默认的 CAST 规则，链接时将发生异常。

- (1) 代入语句右边的表达式内，操作对象的数据类型不一致时  
例)

```
A: = INT#10 + SINT#2;
```

- (2) 代入语句右边和左边的操作对象数据类型不一致时  
例)

```
A: = B + C;
    /   \
  REAL  INT
```

- (3) 语法的表达式内，操作对象的数据类型不一致时  
例)

```
      INT  LINT
       \   /
CASE A+B OF
  INT#1:
    def:=INT#10;
```

下面介绍以下 3 种情况时的 CAST 规则。

### 在代入语句的右边进行运算时的 CAST 规则

- 如果右边的操作对象为运算符的操作对象中可使用的数据类型且为允许的数据类型，则可以同时存在。
- 在右边的操作对象中，拥有最高等级的数据类型将变为右边的数据类型。  
(数据类型的等级请参考以下 [数据类型等级表])。

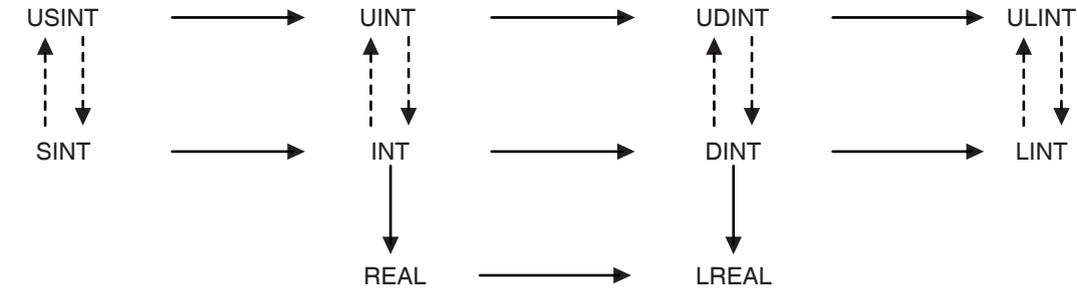
[数据类型等级表]:

等级越高，表示该数据类型可表现的数值范围越大。

等级	数据类型
1	SINT
2	USINT
3	INT
4	UINT
5	DINT
6	UDINT
7	LINT
8	ULINT
9	REAL
10	LREAL
11	BYTE
12	WORD
13	DWORD
14	LWORD

## 将代入语句右边的值代入到左边时的 CAST 规则

在以下图中，如果代入源的类型和代入目标的类型用箭头连接，将进行 CAST 处理。  
若未连接，链接时将发生异常。



—————> 代入时，数值的符号和绝对值无变化。

- - - - -> 代入时，数值的符号或绝对值可能发生变化。

例) `intVar := -1;` (\* `intVar := 16#FFFF` \*)

`uintVar := 1;`

`uintVar := intVar;` (\* `uintVar := 16#FFFF`，即代入了-1，结果为65535\*)

箭头即使没有直接连接，也可以代入。

例如，`SINT → USINT → UINT → UDINT → ULINT` 这样连接时，`ULINT := SINT` 这样的记述也可以。



### 使用注意事项

`UDINT → ULINT`、`DINT → LINT`、`DINT → LREAL` 的 CAST 时，请注意以下事项。

以上均为从数据大小 32 位的数据类型到 64 位的数据类型的 CAST。此时，代入语句右边的运算结果如果超出 32 位可表现值的范围，可能无法代入正确的数值。

例) 在以下用户程序中，第 3 行的相加结果超出 32 位可表现值的范围。将发生溢出，在 `LintVar` 中代入 0。

```
UdintVar := UDINT#16#FFFF_FFFF;           // 32 位的上限值
DintVar  := DINT#1;                       // 1
LintVar  := (UdintVar + DintVar) / DINT#2; // (32 位的上限值 + 1) / 2
```

此时，请先将各数值转换为 64 位的数据类型后再进行运算。上述示例将变更如下。

```
LintTmp1 := UDINT_TO_LINT(UDINT#16#FFFF_FFFF); // UDINT → LINT 转换
LintTmp2 := DINT_TO_LINT(DINT#1);              // DINT → LINT 转换
LintVar  := (LintTmp1 + LintTmp2) / DINT#2;
```

## 语法表达式内的 CAST 规则

在语法表达式中，代入语句右边的运算式或右边代入左边时的默认 CAST 规则同样适用。

例)

```
CASE (A+B+C) OF
  Result1:
    ~;
  ResultN:
    ~;
END_CASE;
```

## ● FUN 的执行顺序

在包含 FUN 的表达式中，FUN 的执行顺序无法完全确定。根据 NY 系列控制器或 Sysmac Studio 的版本以及记述的方法不同，FUN 的执行顺序也不同。因此，如下所示，结果会因 FUN 的执行顺序不同而变化的表达式请按后述方法记述。

- 包含多个访问同一全局变量的 FUN 之表达式
- 包含因 FUN 及 FUN 的执行其值会变化的变量之表达式

### 包含多个访问同一全局变量的 FUN 之表达式

以下表达式时，3 个 FUN 的执行顺序与按照运算符的优先顺序决定的运算顺序无关。因此，3 个 FUN 不一定按 FUN2、FUN3、FUN1 的顺序执行。

```
result := FUN1() + FUN2() * FUN3();
```

上述表达式中的 3 个 FUN 如果是对同一全局变量进行访问（包括写入）的 FUN，因 FUN 的执行顺序不同，变量“result”的值也可能不同。

此时，为了确定 3 个 FUN 的执行顺序，应分开记述表达式。要以 FUN2、FUN3、FUN1 的顺序执行时，请如下记述。

```
tmp2 := FUN2();
tmp3 := FUN3();
result := FUN1() + tmp2 * tmp3;
```

### 包含因 FUN 及 FUN 的执行其值会变化的变量之表达式

以下表达式中包含因 FUN 及 FUN 的执行其值会变化的变量。

```
result := varA + FUN4(out => varA);
```

上述表达式时，左右第 1 项变量“varA”的评估不一定会在 FUN4 之前执行。因此，根据“varA”的评估和 FUN4 的执行顺序不同，变量“result”的值可能会不同。

此时，为了确定“var4”的评估和 FUN4 的执行顺序，应分开记述表达式。

要在 FUN4 之前执行“varA”的评估时，请如下记述。

```
tmp := varA;
result := tmp + FUN4(out => varA);
```

要在 FUN4 之后执行“varA”的评估时，请如下记述。

```
tmp := FUN4(out => varA);
result := varA + tmp;
```

### ● 表达式中包含不指定数据类型的常数时，表达式的运算精度

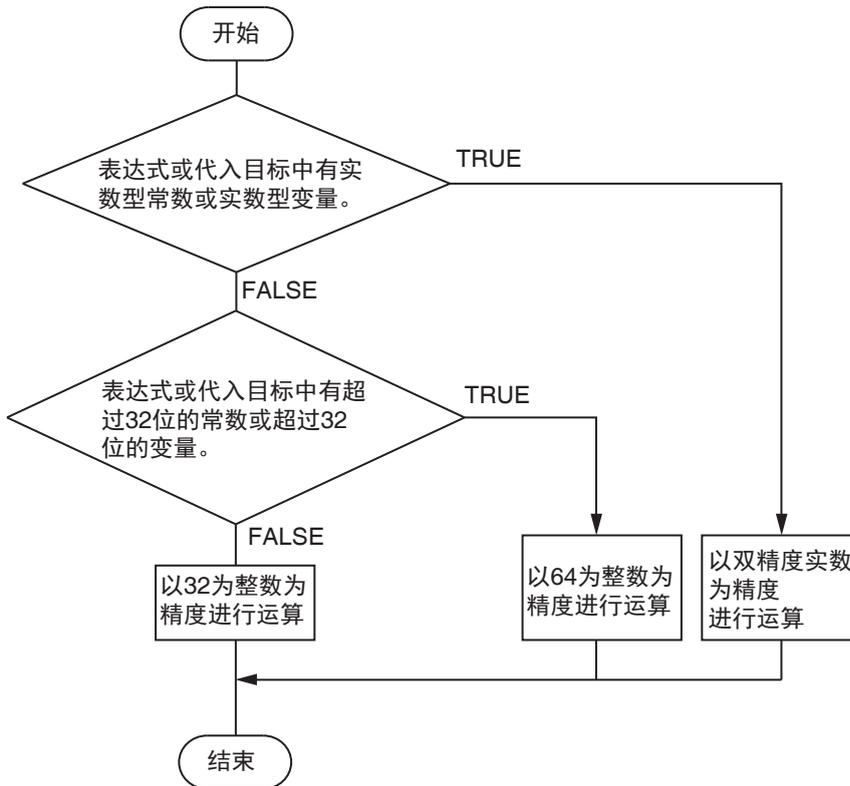
表达式中包含不指定数据类型的常数时，运算精度根据其常数的表述、表达式及代入目标中写入的变量以及常数的数据类型自动决定。

#### 常数的表述

如果常数为“100”等没有小数点的表述，作为整数处理。如果为“100.0”等带小数点的表述，作为实数处理。

#### 表达式的运算精度

表达式的运算精度根据表达式及代入目标中写入的变量以及常数的数据类型不同，为 32 位整数、64 位整数、双精度实数中的一个。表达式的运算精度按以下规则决定。



例) `realv := 2 + 3 * 4;` // 变量 `realv` 的数据类型为 `REAL`

变量 `realv` 为实数型, 因此以双精度实数的精度进行运算。运算结果为 14.0。

`realv := 2 + 3.0 * 4;` // 变量 `realv` 的数据类型为 `REAL`

常数 3.0 为实数型表述, 变量 `realv` 也是实数型, 因此以双精度实数的精度进行运算。运算结果为 14.0。

`lintv := 2 + 3 * 4;` // 变量 `lintv` 的数据类型为 `LINT`

无实数型常数或变量, 变量 `lintv` 为超过 32 位的常数, 因此以 64 位整数的精度进行运算。运算结果为 14。

`intv := 2 + 3 * 4;` // 变量 `intv` 的数据类型为 `INT`

无实数型常数或变量, 也没有超过 32 位的常数, 因此以 32 位整数的精度进行运算。运算结果为 14。

但是, 除法的运算精度只根据除数和被除数决定。决定运算精度的规则与前面的流程图相同。

例) `realv := 2 / 3 * 4;` // 变量 `realv` 的数据类型为 `REAL`

除法  $2/3$  由于被除数和除数都是 32 位以内的整数, 因此以 32 位整数的精度进行运算。相除结果为整数 0。

然后, 由于变量 `realv` 为实数型, 因此  $0*4$  以双精度实数的精度进行运算。运算结果为 0.0。

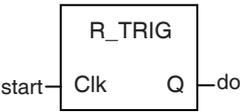
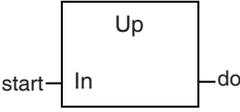
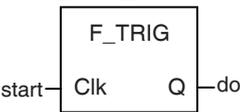
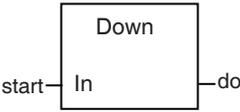
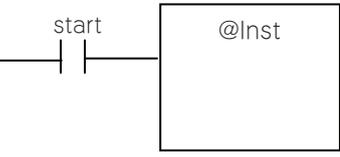


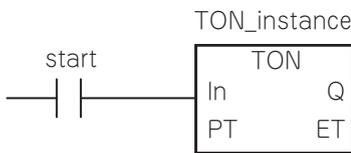
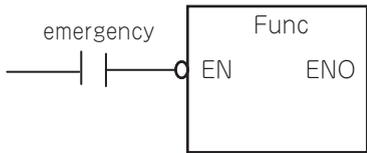
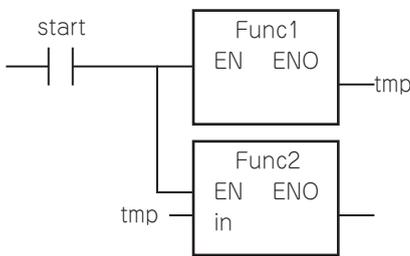
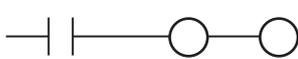
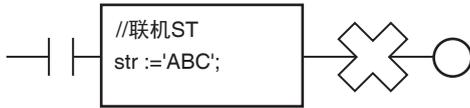
#### 使用注意事项

表达式中包含不指定数据类型的常数时, 根据其常数的表述、表达式中的变量及常数的数据类型自动决定。因此, 可能以用户预期以外的精度进行运算。常数如 `REAL#1.0`, 建议以指定数据类型的方式表述。

## ST 和梯形图的差异

ST 和梯形图之间有差异时的示例如下所示。

项目	梯形图	ST (包含联机 ST)
接点的微分	<ul style="list-style-type: none"> <li>● 上升沿           <ul style="list-style-type: none"> <li>• 方法1 start do --- ↑ -----○</li> <li>• 方法2 R_TRIG_instance </li> <li>• 方法3 </li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>● 上升沿</li> <li>• 方法 1 R_TRIG_instance (Clk:=start, Q=&gt;do) ; ※ R_TRIG_instance 为 R_TRIG 指令的实例</li> </ul>
	<ul style="list-style-type: none"> <li>● 下降沿           <ul style="list-style-type: none"> <li>• 方法1 start do --- ↓ -----○</li> <li>• 方法2 F_TRIG_instance </li> <li>• 方法3 </li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>● 下降沿</li> <li>• 方法 1 F_TRIG_instance (Clk:=start, Q=&gt;do) ; ※ F_TRIG_instance 为 F_TRIG 指令的实例</li> </ul>
指令的微分	<ul style="list-style-type: none"> <li>● 上升沿微分 </li> </ul>	<ul style="list-style-type: none"> <li>● 上升沿微分 无符合的项, 用逻辑来实现。例如,</li> <li>• 方法 1 R_TRIG_instance (Clk:=start, Q=&gt;do) ; IF (do = TRUE) THEN Inst ( ) ; END_IF;</li> <li>• 方法 2 IF (start = TRUE) THEN IF (pre_start = FALSE) THEN Inst ( ) ; END_IF; END_IF; pre_start:=start;// 上次值的更新</li> </ul>

项目	梯形图	ST (包含联机 ST)
多个任务周期指令的记述		<p>如下所示，声明计时用和复位用的 2 个实例</p> <pre> IF (start = TRUE) THEN     TON_instance (In:=TRUE, 后面省略);     // 定时器启动 ELSE     TON_instance (In:=FALSE, 后面省略);     // 定时器复位 END_IF; </pre>
FB / 函数参数反转指定		<p>在参数上添加 NOT 运算符 ※ 除了参数之外，NOT 运算符还可添加到所有 BOOL 型变量中</p> <pre> IF (NOT emergency) THEN     Func (); END_IF; </pre>
多段连接		<pre> IF (start=TRUE) THEN     Func2 ( in := Func1 ()); END_IF; </pre>
梯形图的后续	<p>Out 指令后面只能连接 OUT 指令</p> 	<p>联机 ST 的后面不可连接梯形图</p> 
程序的分割单位	有段落功能	无段落功能

## 6-6 指令

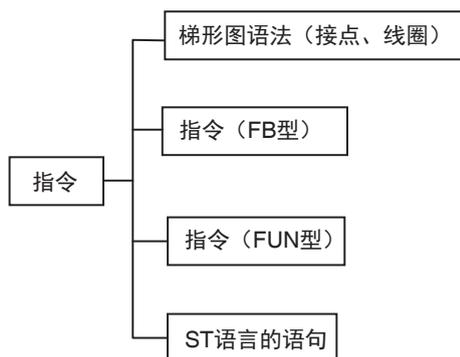
下面介绍预先在 NY 系列控制器中定义的指令。

指令的详情请参考□□《NY 系列 指令基准手册 基本篇（SBCA-CN5-437）》及□□《NY 系列 指令基准手册 运动篇（SBCE-CN5-380）》。

### 6-6-1 指令定义

指令是指 POU 的算法中记载的、欧姆龙指定的最小单位的处理要素。

指令分为以下种类。



程序、用户定义的 FB、用户定义的 FUN 由本指令的组合构成。

### 6-6-2 指令的基本知识

指令的基本规格遵照 FB、FUN 的规格。

下面介绍指令的规格。

#### 梯形图语法（接点、线圈）

##### ● 可配置位置

以梯形图语言的接点和线圈形态表现的指令如下所示，有可配置的位置。

	分类	可配置位置	图
输入类	逻辑启动型 (LD 型)	与左母线直接连接或回路图的起点	
	下一段连接型	逻辑启动型和输出型的中间位置	
输出类		与右母线直接连接	

### ● 动作选项

梯形图语言的指令（接点）中，也有通过标向上（↑）、向下（↓）后，可进行上升沿微分、下降沿微分的动作。

上升沿 (↑)	<p>变量A</p> <p>输出</p> <p>1个任务周期</p> <p>1个任务周期</p> <p>时序图</p>	<p>该指令为每个任务周期执行接点的读取 / 比较 / 位测试等处理，若结果从 FALSE 变为 TRUE，则输出功率流。</p> <p>输出的功率流在（1 个任务周期 TRUE 后）下一任务周期变为 FALSE。</p>
下降沿 (↓)	<p>变量A</p> <p>输出</p> <p>1个任务周期</p> <p>1个任务周期</p> <p>时序图</p>	<p>该指令为每个任务周期执行接点的读取等处理，若结果从 TRUE 变为 FALSE，则输出功率流。</p> <p>输出的功率流在（1 个任务周期 TRUE 后）下一任务周期变为 FALSE。</p>

## 功能块 (FB) 型的指令

### ● 执行条件

FB 型指令的执行条件因指令不同而异。

执行条件用的特定输入变量根据各指令定义。

例：Execute：上升沿指定、下降沿指定

Enable：每个任务周期执行

FB 型指令在执行该指令的调用源 POU 期间，将持续执行。

### ● 动作选项

不可指定。

## 函数 (FUN) 型指令

### ● 执行条件

FUN 型指令的执行条件全部为 EN 输入。EN 为 TRUE 期间，每个任务周期执行。

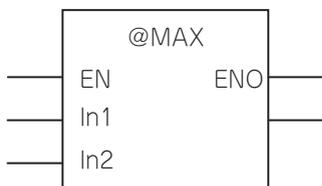
### ● 动作选项

梯形图语言时，可添加以下动作选项，将该执行条件作为上升沿指定。ST 语言中无动作选项。

动作选项		选项符号	
微分选项	上升沿	@	指令为输入上升沿微分型指令。 EN 的上升沿时，执行 1 次指令。

要添加动作选项，需要在指令名称前面添加上述选项符号。

例)



## 功能块 (FB) 型 / 函数 (FUN) 型的指令共通

### ● 条件标志

将值作为指令处理的结果代入的系统定义变量称为“条件标志”。NY 系列控制器的条件标志只有进位标志 (P\_CY)。

进位标志的含义为以下之一。

- 判定指令的处理结果是否超过输出变量的数据类型可表现的范围
- 位列数据的位移指令等，判定是否发生位数溢出的详情，请参考□□《NY 系列 指令基准手册 基本篇 (SBCA-CN5-437)》中各指令的说明。

### 6-6-3 指令错误

在执行指令时发生的异常称为指令错误。下面介绍指令错误的发生时序、指令错误中检测到的异常、发生指令错误时的动作等。

#### 指令错误的发生时序

指令错误的发生时序有以下 3 种。因指令错误的发生时序不同，检测到的异常或发生异常时的动作有所不同。

- 执行指令前，检查输入参数或输入输出参数的值时发生
- 执行指令时，在指令的内部处理时发生
- 执行指令后，检查输出参数的值时发生

#### 指令错误中检测到的异常

指令错误中检测到的异常如下所示。因指令错误的发生时序不同，检测到的异常有所不同。

##### ● 执行指令前、执行指令后检测到的异常

执行指令前、执行指令后检测到的异常如下所示。

- 读写超出排列指定变量的排列区域的要素
- 在 STRING 型变量中，代入比定义字节数更长的字符串
- 在 STRING 型变量中，代入不是以 NULL 字符结束的字符串
- 整数型变量除以 0

##### ● 执行指令时检测到的异常

执行指令时检测到的异常因指令不同而异。各指令中检测到的异常详情请参考□□《NY 系列 指令基准手册 基本篇（SBCA-CN5-437）》。

## 发生指令错误时的动作

因是否发生指令错误而进行不同动作的要素包括输出变量“ENO”、输出变量“Error”、输出变量“ErrorID”、系统定义变量“P\_PRGER”、事件。下面分别介绍。

### ● 输出变量“ENO”、输出变量“Error”、输出变量“ErrorID”

表示是否发生异常的输出变量包括“ENO”（允许启用）、“Error”（错误）、“ErrorID”（错误代码）。具体是哪个输出变量因指令不同而异。各变量的含义和发生指令错误时的值如下所示。因指令错误的发生时序不同，动作有所不同。

输出变量	数据类型	含义	发生指令错误时的值		
			执行指令前 <sup>*1</sup>	指令执行中	执行指令后 <sup>*2</sup>
“ENO”	BOOL	TRUE: 正常结束 FALSE: 异常结束、执行中、未执行	FALSE	FALSE	TRUE
“Error”	BOOL	TRUE: 异常结束 FALSE: 正常结束、执行中、未执行	FALSE	TRUE	FALSE
“ErrorID”	WORD	异常结束时为错误代码， 正常结束时为 WORD#16#0	WORD#16#0	错误代码	WORD#16#0

\*1. 如果在执行指令前发生指令错误，则不执行指令。因此，各输出参数的值保持执行指令前的值。

\*2. 如果在执行指令后发生指令错误，则判断为指令本身已正常结束。因此，各输出参数中将代入指令的输出变量值。但是，发生异常的输出参数的值保持执行指令前的值。

### ● 系统定义变量“P\_PRGER”

系统定义变量“P\_PRGER”为指令错误发生标志。若发生指令错误，无论指令错误发生时序如何，值都变为 TRUE。正常结束时，保持值。“P\_PRGER”的详情请参考□□「指令错误发生标志」(P.6-131)。

### ● 事件

若发生指令错误，将发生针对异常内容的事件。事件的详情请参考□□「因发生指令错误而发生的事件」(P.6-132)。

## 梯形图语言中的输出参数

用梯形图语言记述的指令、用户定义的函数、用户定义的功能块正常结束时，以及发生指令错误时，输出参数的值如下所示。

条件	输出参数的种类	输出参数的值
正常结束时	功率流输出	根据内部算法更新值。
	参数输出 BOOL 型	
	参数输出 BOOL 型以外	
发生指令错误时	功率流输出	变为 FALSE。
	参数输出 BOOL 型	保持之前的值。
	参数输出 BOOL 型以外	

## 用 ST 语言记述的 POU 中发生语法错误时的动作

### ● 代入语句中的错误

用 ST 语言记载的代入语句发生错误时，不执行该行。

```
5 a = b / (c + d) + e * f + ABS(g);
6 x := 1;
```

例如，第5行的b/(c+d)中发生除以0的情况时，第5行的执行将中断（a的值无变化），执行第6行。

用户创建函数的输出 ENO 为 OFF 时亦同。

```
5 a = 用户创建函数(b) + c;
6 x := 1;
```

用户创建函数的输出 ENO 为 OFF 时，第5行的执行将中断（a的值无变化），执行第6行。

### ● IF 语法中的错误

ST 语言的语法发生错误时，根据语法执行错误处理。

以下示例中（c+d）的值为 0 时，不执行 IF ~ END\_IF 之间的内容。

POU“○○”

```
5 IF a = b / (c + d) THEN
6   x := 1;
7 ELSE
8   x := 2;
9 END_IF;
10 y := 10;
   :
   IF P_PRGER = TRUE THEN
     x := 初始值; (* 发生异常时的处理 *)
     y := 初始值;
   END_IF;
```

记载发生错误时用户应采取的安全措施。

### ● ST 语言中的语法错误

ST 语言中的语法错误包括以下情况。

- 超出排列 [i] 的要素数
- 输入输出变量中未设置参数
- 字符串型的代入：左边的字符串型大小（Byte）<右边字符串长度（Byte）时
- 除以 0（浮动小数点运算除外）

※ 浮动小数点数的值为非数值时，运算结果为非数值，但不视为错误。

### ● 发生语法错误时的动作

P\_PRGER 标志变为 TRUE，具体如下。

语法	错误发生位置	动作
代入语句		不执行该行
控制语句	IF 条件式内	不执行 IF ~ END_IF 之间
	CASE 条件式	不执行 CASE ~ END_CASE 之间
	FOR 条件式内	不执行 FOR ~ END_FOR 之间
	WHILE 条件式内	不执行 WHILE ~ END_WHILE 之间
	REPEAT 条件式内	不执行 REPEAT ~ END_REPEAT 之间

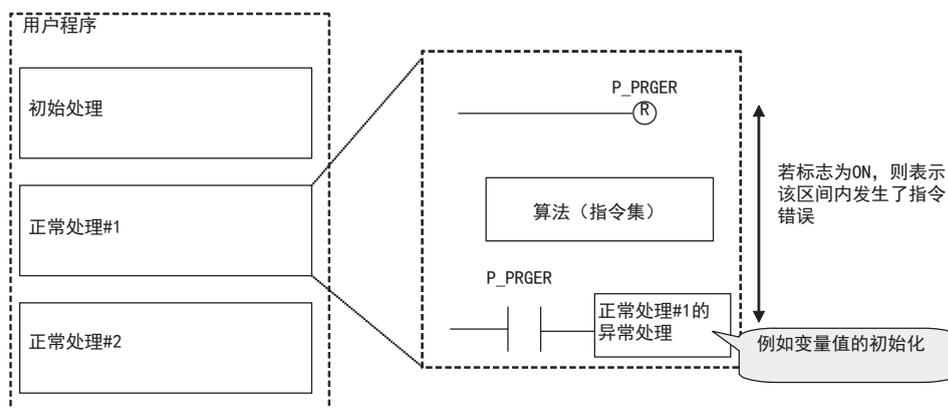
## 指令错误发生标志

在梯形图算法内发生指令错误或在 ST 算法内发生语法 / 函数错误时，系统定义变量“P\_PRGER”（指令错误发生标志）将变为 TRUE。

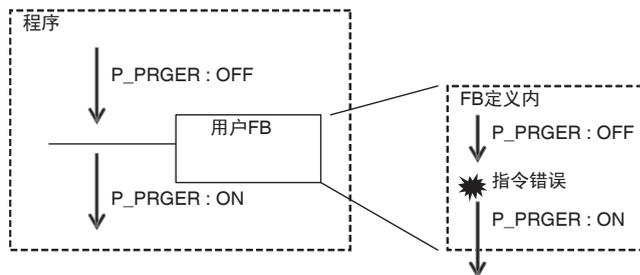
P\_PRGER 标志为程序的本地变量。程序中发生指令错误时变为 TRUE，在下一任务周期内仍保持值。

变量名称	名称	功能	数据类型	值的范围	初始值	R/W
P_PRGER	发生指令错误标志	发生指令错误时，持续变为 TRUE 的标志。一旦变为 TRUE，在通过用户程序变为 FALSE 前，将保持 TRUE。	BOOL	FALSE/TRUE	FALSE	RW

P\_PRGER 标志还可由用户覆盖。因此，可如下所示，当通过用户操作设为 FALSE 时，可在特定范围内判断是否发生指令错误。一旦变为 TRUE，将变更动作模式，或保持值，直至通过用户程序等覆盖。例)



程序中使用的用户定义 FB 发生指令错误时，P\_PRGER 标志也会变为 TRUE。例)



## 因发生指令错误而发生的事件

---

因发生指令错误，发生与之对应的事件。

事件的详细确认方法请参考□□「8-5 事件日志功能」(P.8-45)。此外，事件的内容请参考□□《NY 系列指令基准手册 基本篇 (SBCA-CN5-437)》。



### 使用注意事项

---

- 发生指令错误时发生事件的情况为在 Sysmac Studio 的控制器设定中将[事件日志设定]-[指令错误输出] 设定为 [是] 时。关于 Sysmac Studio 的控制器设定详情，请参考□□「4-2-2 控制器设定」(P.4-4) 及□□《Sysmac Studio Version 1 操作手册 (SBCA-CN5-362D 以后)》。
  - 发生指令错误后若变更用户程序，可能无法在事件日志中正确记录信息。
  - 若重复执行有异常的指令，每次都会发生指令错误。因此，事件日志的记录条数可能超出上限。此时，将从旧到新依次覆盖，敬请注意。
- 



### 参考

---

- 若在执行运动控制指令时发生异常，将同时发生指令错误对应的事件和运动控制指令固有的事件。运动控制指令的事件详情请参考 □□《NY 系列指令基准手册 运动篇 (SBCE-CN5-380)》。
  - 在 Sysmac Studio 的控制器设定中将[事件日志设定]-[指令错误输出] 设定为[否]时，运动控制指令固有的事件仍会发生。
-

## 6-7 名称空间

下面介绍名称空间的规格和使用方法。关于操作方法，请参考□□《Sysmac Studio Version 1 操作手册 (SBCA-CN5-362C 以后)》。

### 6-7-1 名称空间定义

名称空间是指将 FB 定义等分组，以层级结构进行管理的机制。这相当于将文件归类到文件夹，以目录的层级结构进行管理。不使用名称空间时，FB 定义等不可重复使用相同的名称，但如果使用名称空间，设定为不同的名称空间后，可重复使用。

名称空间可使用，也可不用。

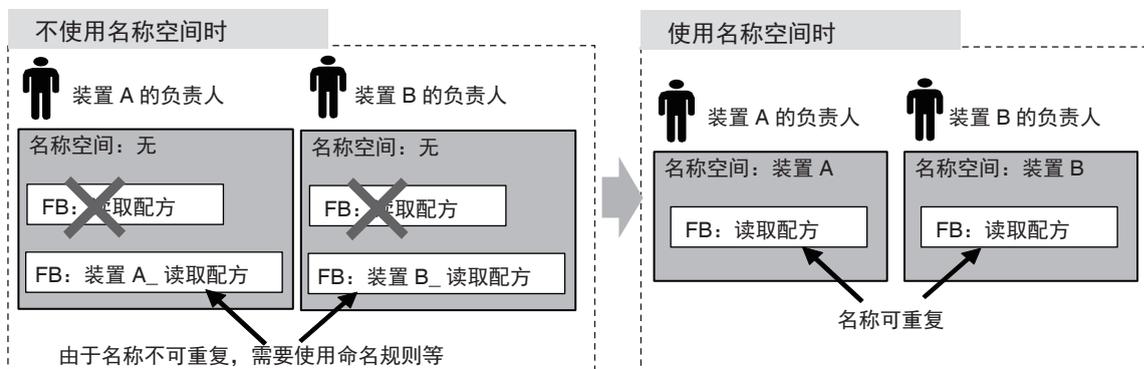
### 名称空间的特点

有以下特点。

#### ● 避免名称的重复

如果是不同的名称空间，FB 定义名称等可以使用相同的名称。

例如，对多个装置进行编程时，如果不同的装置由不同的人员负责编程，FB 定义名称等可能会重复。不使用名称空间时，应利用命名规则等，避免名称重复。如果使用名称空间，为不同的装置设定名称空间，编程时无需顾忌装置的名称会重复。



## 6-7-2 名称空间的规格

下面介绍名称空间的对象、表现及使用声明。

### 名称空间的对象

名称空间的对象如下所示。

对象	详细
POU 定义	FB 定义名称、FUN 定义名称
数据类型	结构体型名称、联合体型名称、列举型名称

### 名称空间的表现

名称空间以“\（反斜杠）”分隔表现。在 POU 的算法中使用，在名称空间的开头加“\\（2 个反斜杠）”。“\（反斜杠）”可能因 Windows 的本地设置或字体的关系，显示为“¥”。

例)

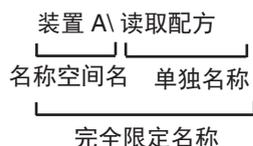
使用场合	表现
在算法以外中使用	装置 A\ 读取配方
在算法内使用	\\ 装置 A\ 读取配方

#### ● 完全限定名称和单纯名称

不省略名称空间名，表现名称空间对象的名称称为完全限定名称。省略名称空间名的名称称为单纯名称。

在 POU 定义的算法中，属于相同名称空间的 POU 定义可用单纯名称记述。

例)



#### ● 名称空间的表现限制

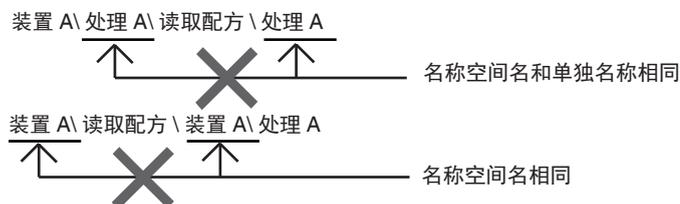
- 可使用的字符与变量名称相同。详情请参考□□「6-3-12 变量名称等程序相关名称的限制」(P.6-73)。
- 名称空间的字符数限制如下所示。

名称	最大大小	字符编码
名称空间名	93 字节	UTF-8
单独名称	127 字节	

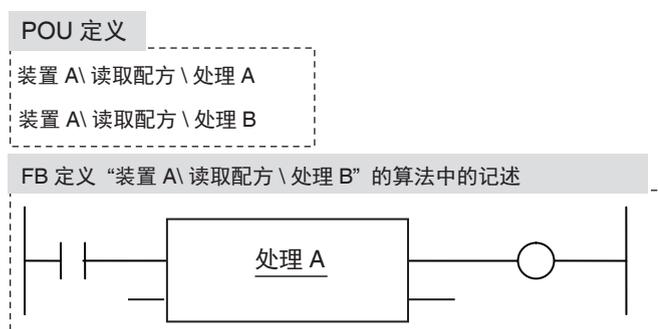


### 使用注意事项

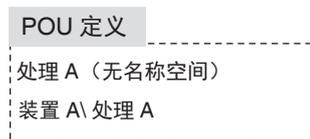
- 如果名称空间名称和单纯名称相同，链接时将发生异常。



- 在 POU 定义的算法中，属于相同名称空间的 POU 定义可用单纯名称记述，但如果该名称空间的上位层级中存在相同单纯名称的 POU 定义或数据类型等，链接时将发生异常。如下图所示，存在 POU 定义时，在名称空间“装置 A \ 读取配方”上的 FB 定义“处理 B”算法的记述中，可用单纯名称调用属于相同名称空间的“装置 A \ 读取配方 \ 处理 A”。



但是，如下所示，在名称空间“装置 A \ 读取配方”的上位层级中存在 POU 定义“装置 A”时，由于存在多个单纯名称“处理 A”，因此链接时将发生异常。此时，请用完全限定名称记述，或变更单纯名称。



- 与系统内部使用的保留字相同时，程序检查时将发生错误。

## 名称空间的使用声明

编程时，事先对 POU 定义的算法中使用的名称空间进行声明称为使用声明。

在 POU 定义中进行使用声明后，属于已进行使用声明的名称空间的 POU 定义可用单纯名称表示。进行使用声明后，仍可使用完全限定名称。

在 POU 定义的算法中，使用属于与 POU 定义相同名称空间的 FB 定义或 FUN 定义时，无需进行使用声明，也可使用单纯名称。

1 个 POU 定义可进行多次使用声明。

### ● 记述示例

下面介绍在创建 FB 定义时，对使用对象 FB 定义的名称空间进行使用声明和不进行使用声明时的记述示例。

例)

在 FB 定义“升降机”的算法内，记述 FB 定义“读取配方”及 FB 定义“上限计算”的示例。

各 FB 定义属于不同的名称空间。FB 定义“升降机”中，仅对名称空间“装置 C”进行使用声明。

不属于名称空间“装置 C”的 FB 定义“读取配方”以完全限定名称记述。属于名称空间“装置 C”的 FB 定义“上限计算”可用单纯名称记述。

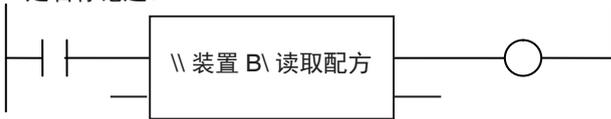
名称空间	对象种类	单独名称
装置A	FB定义	升降机
装置B	FB定义	读取配方
装置C	FB定义	上限计算

FB 定义“升降机”的 [ 名称空间 - 使用声明 ] 的记述

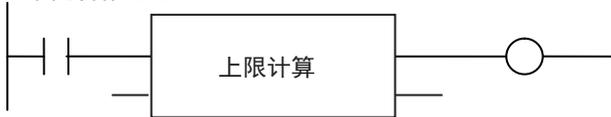
装置 C

FB 定义“升降机”的算法内的记述

- 不对名称空间进行使用声明时  
未对 FB 定义“读取配方”所属的名称空间“装置 B”进行使用声明，因此以完全限定名称记述。



- 对名称空间进行了使用声明时  
已对 FB 定义“上限计算”所属的名称空间“装置 C”进行了使用声明，因此以单独名称记述。



## ● 使用声明的限制

单纯名称仅在 POU 定义的算法内可以使用。



### 使用注意事项

- 以下情况下，在程序检查时将发生错误。
  - 对不存在的名称空间进行了使用声明时
  - 对1个POU定义声明了多个名称空间，各名称空间中存在相同名称的POU定义或数据类型时
- 以下情况下，如果将POU定义或数据类型命名为相同的名称，链接时将发生异常。
  - POU定义所属的名称空间及其上位层级的名称空间
  - 进行了使用声明的名称空间
  - 无名称空间

名称空间	对象种类	单独名称	
装置A\升降机	FB定义	处理A	✗ POU定义所属的名称空间
装置A	FB定义	处理A	✗ POU定义所属的名称空间的上位层级
装置B	FB定义	处理B	
		处理A	✗ 进行了使用声明的名称空间
无	FB定义	处理A	✗ 无名称空间

FB定义“处理A”的[名称空间-使用声明]的记述

装置B



### 参考

不可对程序名称设定名称空间，但对程序的算法中使用的部件名称空间可以进行使用声明。

## 6-7-3 名称空间的使用步骤

使用 Sysmac Studio，进行名称空间的设定和使用声明。

创建数据类型或创建 FB 定义、FUN 定义等部件时，进行步骤 1 和步骤 2 的设定。

使用设定了名称空间的部件时，进行步骤 3 的使用声明。

- 1** 用数据类型编辑器，为对象数据类型设定名称空间。
- 2** 为对象 FB 定义或 FUN 定义的属性设定名称空间。
- 3** 在梯形图编辑器或 ST 编辑器中，针对对象 FB 定义或 FUN 定义的属性中设定的名称空间进行使用声明。
- 4** 在用户程序中记述对象数据类型或 FB 定义等。

操作详情请参考 □ 《Sysmac Studio Version 1 操作手册（SBCA-CN5-362C 以后）》。

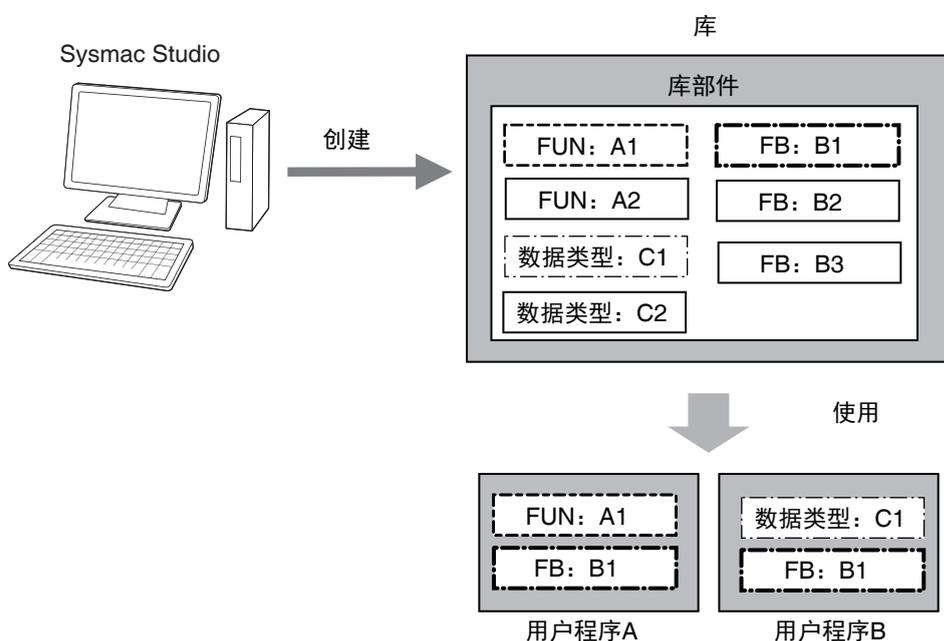
## 6-8 库

下面介绍库的规格。详细的操作方法请参考□□《Sysmac Studio Version 1 操作手册（SBCA-CN5-362C 以后）》。

### 6-8-1 库的概要

将 POU 定义或数据类型汇总为可作为流用所需的部件使用的形式，得到库。库的部件称为库部件。  
NY 系列控制器中，可创建和使用库。

下图中，在库的库部件中，用户程序A使用FUN: A1和FB: B1，用户程序B使用数据类型: C1和FB: B1。



## 6-8-2 库的规格

下面介绍库的设定和同步。

### 库的设定

库中可设定以下项目。

设定项目	说明
名称	库的名称。
版本	库的版本。
创建者	库创建者的姓名。(任意)
创建日期	新建库时的日期。
更新日期	更新库时的日期。
注释	库的注释。(任意)
公司名称	库的创建公司名称。(任意)
ID	库核对用的唯一 ID。自动生成, 无法变更。
不可显示源代码	可选择源代码的显示或隐藏。 <sup>*1</sup>
附件	可附加多个任意文件。

\*1. 库部件设定为数据保护时, 若要显示源代码, 需要密码。

库部件可参照其他库后创建。此时, 选择是否包含参照处的库数据。



#### 参考

如果包含参照处的库, 创建参照源的库时, 将包含参照处库的副本。因此, 库的文件只需 1 个, 但如果参照源的库有多个, 变更时需要逐个进行。

如果不包含, 创建参照源的库时, 将不包含参照处的库。因此, 库的文件会分割为参照源和参照处多个, 如果参照源的库有多个, 变更时只需变更参照处的库即可。

#### ● 库部件的选择

选择要包含到库中的库部件。

### 库的同步

能够以库为单位, 从 Sysmac Studio 下载、用控制器上传、与控制器进行核对。



#### 参考

- 向正在传送“显示源代码的库”的控制器, 通过 Sysmac Studio 以“不传送程序的源代码”的设定传送了项目时, 将不传送库的源代码。
- 存储器全部清除后, 控制器中的库将消失。

### 6-8-3 库部件的规格

下面介绍库部件的对象和库部件的设定。

#### 库部件的对象

以下数据可视为库部件。

库部件	详细
POU 定义	FB、FUN
数据类型 <sup>*1</sup>	结构体型、联合体型、列举型

\*1. 在 Sysmac Studio 的部件选择画面中，数据类型始终包含在库中。

#### 库部件的设定

各库部件中可设定以下项目。

设定项目	定义
名称	库部件的名称。
名称空间	库部件的名称空间。
版本 <sup>*1</sup>	库部件的版本。
创建者 <sup>*1</sup>	库部件创建者的姓名。(任意)
创建日期 <sup>*1</sup>	新建库部件时的日期。
更新日期 <sup>*1</sup>	更新库部件时的日期。
注释	库部件的注释。(任意)

\*1. 在 Sysmac Studio 的各 POU 定义的属性中，只能对 FB 和 FUN 进行设定。

## 6-8-4 库的使用步骤

库的创建步骤和使用步骤如下所示。

### 库创建步骤

库作为项目创建。按以下步骤创建库并保存。

- 1 新建库项目。**  
在项目窗口的项目种类中选择“库项目”，创建项目。
- 2 新建库部件。**  
在库项目中创建 POU 定义或数据类型，确认动作。
- 3 库的设定。**  
在库项目中，进行库部件的选择、源代码的显示、附件等的设定。
- 4 保存为库文件。**  
在 [ 创建库 ] 窗口中，保存为库文件。



#### 参考

- 即使是已创建的项目，如果项目中登录的设备只有控制器，可在项目的属性中将种类变更为“库项目”。
- 在库项目中，还可创建库部件对象外的数据，但可选择作为库部件的仅限库部件的对象。
- 作为库部件创建的函数、功能块定义、数据类型名称建议使用名称空间，防止多个库之间的名称重复。名称空间请参考□□「6-7 名称空间」(P.6-133)。

### 库的使用步骤

将创建的库导入到项目后，可在用户程序中使用。按以下步骤使用库。

- 1 指定库。**  
在使用库的项目的 [ 参照库 ] 窗口中，指定要使用的库文件。
- 2 使用库部件进行编程。**  
在项目中使用所导入库的库部件。  
使用方法与普通的函数、功能块定义、数据类型相同。

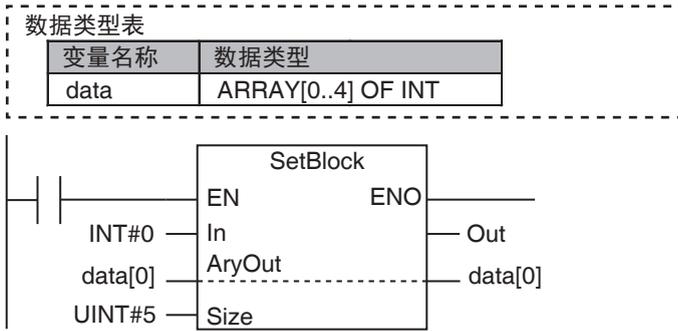
## 6-9 创建用户程序时的注意事项

下面介绍创建用户程序时的注意事项。

### 6-9-1 输入变量 / 输出变量 / 输入输出变量为排列指定的变量时

部分指令中存在处理排列变量的指令。

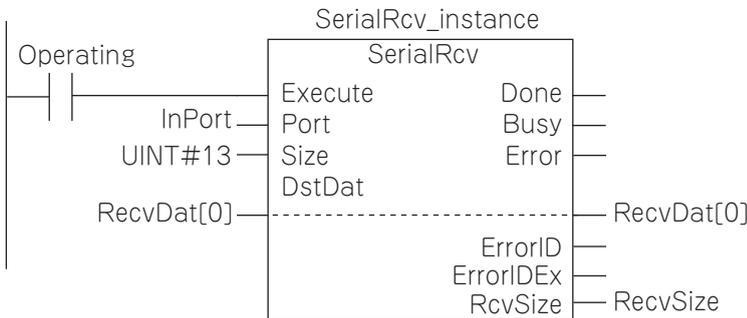
例)



### 6-9-2 输入变量 / 输出变量 / 输入输出变量为结构体型变量时

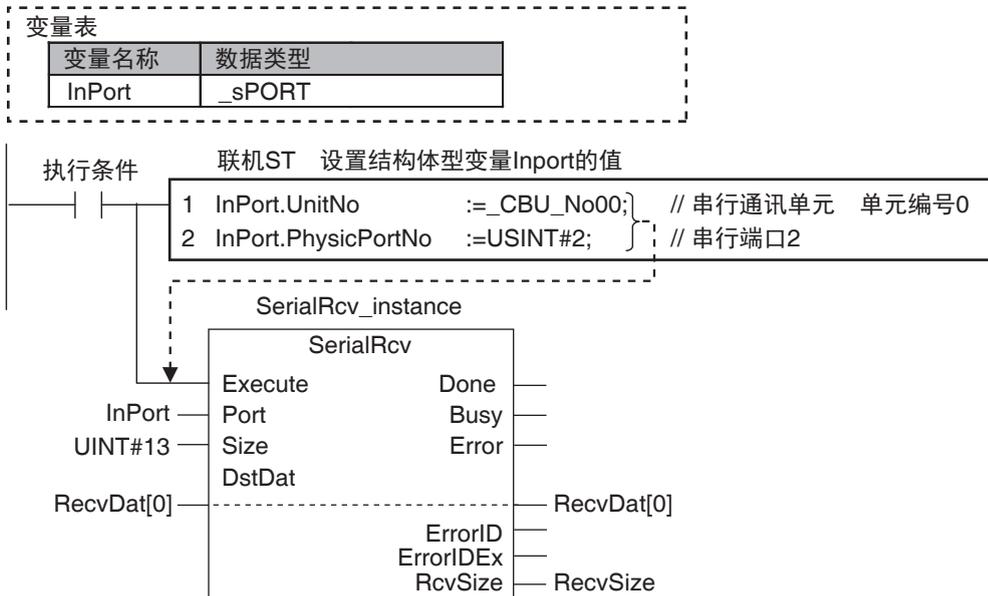
部分指令的输入变量、输出变量、输入输出变量的数据类型为结构体型。

例)



此时，需要事先创建用于输入参数、输出参数、输入输出参数的结构体型变量，并用MOVE指令等设置值。

例)



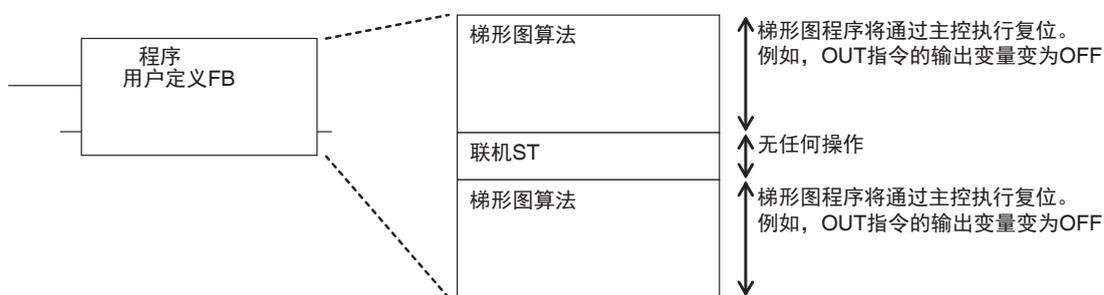
### 6-9-3 主控

#### 概要

主控是指对 MC（主控开始）指令和 MCR（主控结束）指令之间存在的处理进行 OFF 输出的功能。使用主控功能，可方便地控制较长的处理执行条件。  
详情请参考□□《NY 系列 指令基准手册 基本篇（SBCA-CN5-437）》的 MC/MCR 指令。

#### 可使用主控的程序语言

主控可在梯形图语言中使用。  
ST 语言中无法使用。梯形图程序中的联机 ST 也无法使用。  
例)  
主控区域内：



#### 主控复位时区域内指令的动作

主控复位时区域内指令的动作请参考□□《NY 系列 指令基准手册 基本篇（SBCA-CN5-437）》的“MC/MCR 指令”。



# 7

## 动作确认和运行

本章介绍包括离线调试在内的动作确认内容和动作确认到运行之间的步骤。

<b>7-1</b>	<b>动作确认和运行的步骤概要</b>	<b>7-2</b>
<b>7-2</b>	<b>离线调试</b>	<b>7-3</b>
7-2-1	模拟功能的特点	7-3
7-2-2	模拟功能的执行	7-3
7-2-3	模拟功能的设定	7-6
<b>7-3</b>	<b>上机动作确认和运行</b>	<b>7-8</b>
7-3-1	从上机动作确认到运行开始之间的步骤	7-8
7-3-2	项目的传送	7-9
7-3-3	I/O 的配线确认	7-9
7-3-4	MC 调试	7-9
7-3-5	用户程序的动作确认	7-10
7-3-6	正式运行的开始	7-10

## 7-1 动作确认和运行的步骤概要

动作确认和运行在整体步骤中，与下图阴影标示的位置对应。STEP2-4 离线调试中，不与控制器在线连接，利用模拟功能进行动作确认。STEP5. 上机进行的动作确认和运行时，与控制器在线连接，上机确认动作，动作确认完成后开始运行。

整体步骤请参考□□「1-4 NY 系列的整体使用步骤」（P.1-10）。



## 7-2 离线调试

下面介绍离线调试中使用的模拟功能。所谓模拟功能，就是通过 PC 上模拟 NY 系列控制器的动作，执行用户程序动作确认的功能。模拟功能中含有实际控制器上无法实现的调试功能，因此可以提高用户程序开发和调试的效率。

### 7-2-1 模拟功能的特点

与控制器在线连接后进行动作确认的在线调试相比，模拟功能在以下情况下有效。

- 使用断点、逐步执行、暂停等，确认程序的逻辑。
- 选择模拟对象，确认部分程序的动作。
- 通过变更模拟执行速度，以慢于实际情况的速度进行动作确认。
- 用调试程序进行控制器外部输入等。

### 7-2-2 模拟功能的执行

模拟功能中可执行以下内容。

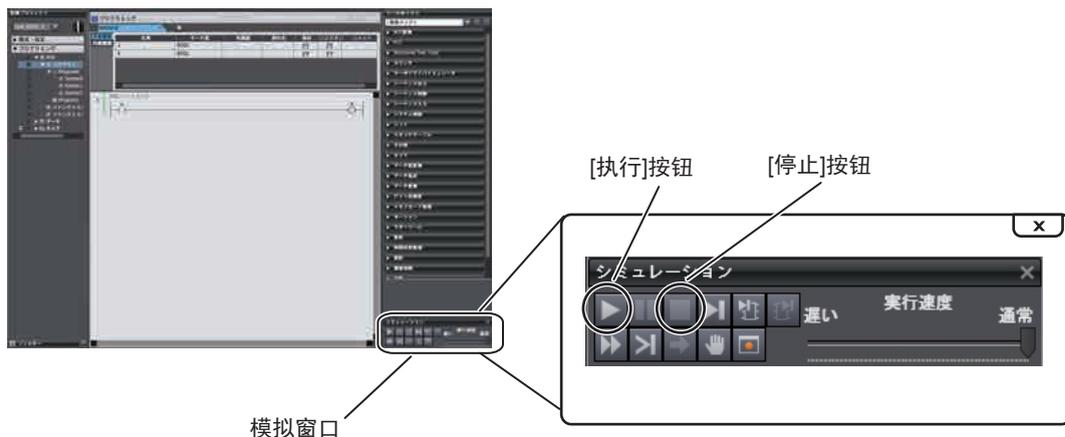
- 模拟器的启动和停止
- 确认程序的逻辑
- 在线连接时的调试功能

#### 模拟器的启动和停止

模拟功能通过 Sysmac Studio 的模拟窗口启动模拟器后执行。利用模拟功能完成动作确认后，停止模拟器。

模拟执行和停止操作的概要如下所示。

- 1 在 Sysmac Studio 上，从主菜单中选择 [显示] - [模拟窗口]。  
工具画面右下方将显示模拟窗口。



### 2 按下模拟窗口的 [ 执行 ] 按钮。

程序将传送到模拟器中，开始模拟。开始模拟后，编辑器等 Sysmac Studio 的画面将变为与控制器在线连接时相同的状态。

### 3 完成动作确认后，按下模拟窗口的 [ 停止 ] 按钮，停止模拟器。

## 确认程序的逻辑

为了确认程序逻辑的正确性，可暂停模拟器的动作，或逐步执行。利用模拟窗口的按钮，可进行以下操作。

操作名称	操作内容
断点	在程序中指定的位置，暂停执行程序。
逐步执行	若在 ST 语言的程序中执行 1 行、在梯形图语言的程序中执行 1 个指令，将暂停模拟器。
连续逐步执行	以一定间隔连续逐步执行。
暂停	暂停模拟的执行。
Step-In 执行	在功能块或函数的资源内逐步执行。
Step-Out 执行	Step-In 执行中的功能块或函数结束后执行。
1 周期执行	执行 1 个周期当前正在执行的任务。在下一周期的开头暂停。
条件断点	设定断点的停止条件，仅在条件成立时暂停执行程序。

## 在线连接时的调试功能

在模拟器上，只能执行与控制器在线连接时使用的调试功能中的一部分。控制器上的在线调试和模拟器的离线调试的区别如下表所示。各调试功能中控制器和模拟器的动作区别请参考□□《Sysmac Studio Version 1 操作手册（SBCA-CN5-362）》。

(○：可使用、×：不可使用)

调试功能	控制器	模拟器
监控	○	○
通过监视窗口监视	○	○
通过 I/O 映射监视	○	○
微分监控	○	○
接点的设置 / 复位	○	○
接点的强制值刷新 TRUE/FALSE/ 解除	○	○
数据的当前值变更	○	○
存储器全部清除	○	×
交叉参考弹出窗口	○	○
在线编辑	○	○
控制器状态监视	○	×
任务执行状态监视	○	○
轴状态监视 (MC 监视表)	○	○
动作模式的变更	○	×
控制器的重置	○	×
数据跟踪	○	○
触发的设定	○	○
采样变量的设定	○	○
追踪的开始和停止	○	○
追踪结果的显示	○	○
追踪结果的导出	○	○
3D 装置模型的创建	○	○
以数字、模拟图表显示	○	○
3D 形式的轨迹显示	○	○
任务执行时间的监视	○	×
程序的模拟调试	×	○
模拟程序的设定	×	○
模拟执行速度的变更	×	○
断点的设定	×	○
逐步执行	×	○
故障排除	○	○
异常信息监视	○	○
异常记录显示	○	○
事件表设定	○	○
用户存储器的使用量监视	○	○
时钟信息的设定	○	×
访问权的强制释放	○	×

### 7-2-3 模拟功能的设定

模拟功能中可设定以下内容。

- 模拟程序的设定
- 调试程序的设定
- 模拟执行速度的设定

#### 模拟程序的设定

模拟功能中，能以任务或程序为单位设定模拟对象并执行。在整个程序中，可用模拟器运行模拟对象的部分程序。

设定模拟程序的操作如下所示。

##### 1 显示模拟窗口。

多视图浏览器的 [编程] - [任务] 下方的程序左侧，将显示模拟对象设定复选框。



##### 2 以任务或程序为单位勾选模拟对象设定用复选框。

#### 调试程序的设定

调试程序是指通过离线调试进行动作确认时使用的程序。调试程序中，记述控制器外部的虚拟输入处理和用户异常的强制发生等。调试程序仅在模拟器上可以执行。

创建调试程序的操作概要如下所示。

##### 1 在多视图浏览器上右击 [编程] - [POU] - [程序]，从菜单中选择 [调试用追加] - [ST] 或 [梯形图]。

将新建调试程序。

##### 2 在创建的调试程序中，记述测试用的程序代码。

##### 3 将调试程序分配到任务中。

已创建的普通程序也可变更为调试程序。

##### 1 在多视图浏览器上右击 [编程] - [POU] - [程序] 中的各程序，设定为 [调试用设定] - [有效]。

## 模拟速度的设定

用模拟窗口的“模拟速度变更滑片”，可在 0.1 ~ 1 倍的范围内，变更计算机上的执行速度。模拟执行中或停止中均可变更。

要以比实际慢的速度显示模拟器执行时使用。



## 7-3 上机动作确认和运行

下面介绍从上机动作确认到运行开始之间的步骤概要和操作概要。

### 7-3-1 从上机动作确认到运行开始之间的步骤

从上机动作确认到运行开始之间的步骤概要如下所示。

STEP1 与 Sysmac Studio 的在线连接和项目的下载	参考
1. 接通控制器的电源。 2. 与 Sysmac Studio 在线连接。 3. 通过 Sysmac Studio 传送项目（用户程序或单元构成等各种设定）。	□ □ 《Sysmac Studio Version 1 操作手册（SBCA-CN5-362）》



STEP2 上机确认动作	参考
1. 在程序模式下，通过 I/O 映射或梯形图编辑器对实际 I/O 进行强制值刷新，确认 I/O 配线。 2. 进行运动控制时，在 MC 调试画面，进行配线检查，通过点动进给确认电机的方向，通过相对定位确认移动量（电子齿轮的设定），确认原点复位动作。 3. 设为运行模式，确认用户程序的动作。	□ □ 「7-3-3 I/O 的配线确认」（P.7-9） □ □ 《NY 系列工业平板电脑 / 工业 BOX PC 用户手册 运动控制篇（SBCE-CN5-379）》 □ □ 「7-3-5 用户程序的动作确认」（P.7-10）



STEP3 控制器的正式运行	参考
1. 确认按设计动作后，开始正式运行。	—



#### 参考

要从 Sysmac Studio 下载项目，需要利用“同步功能”，从计算机传送到控制器上。“同步”的详情请参考 □ □ 《Sysmac Studio Version 1 操作手册（SBCA-CN5-362）》。

## 7-3-2 项目的传送

从 Sysmac Studio 向实际控制器传送项目。

- 1** 与控制器在线连接后，从 [ 控制器 ] 菜单选择 [ 同步 ]。  
计算机上的数据和实际控制器的数据将自动核对。
- 2** 单击 [ 传送 [ 计算机→控制器 ] ] 按钮。

## 7-3-3 I/O 的配线确认

通过 Sysmac Studio 的监视窗口等执行强制值刷新，确认 I/O 的配线。I/O 的配线可向进行配线确认的从站或单元的 I/O 写入值，并对结果进行监视。

强制值刷新的详情请参考 □□「8-4-1 强制值刷新」(P.8-27)。

## 7-3-4 MC 调试

MC 调试是指不使用用户程序，主要通过 Sysmac Studio 进行以下操作。

- 配线的确认：伺服驱动器连接器的输入输出信号监视、  
伺服驱动器的状态监视
- 电机的动作及方向确认：伺服 ON、点动进给
- 电子齿轮的设定确认：进行相对值定位，确认 / 变更移动量
- 原点复位确认：确认原点复位动作

通过 Sysmac Studio 与控制器进行在线连接，在 MC 调试画面进行。

详情请参考 □□《NY 系列 工业平板电脑 / 工业 BOX PC 用户手册 运动控制篇 (SBCE-CN5-379)》。

进行以下操作。

- 1** 进行必要的配线后，通过 Sysmac Studio 与控制器在线连接。
- 2** 进行“轴”的创建、轴的分配、以下轴参数的设定。  
MC 调试所需的轴参数设定：  
显示单位选择、电机每转 1 圈的脉冲数及移动量  
最高速度、点动最高速度、最大加速度和减速度、  
软件限位功能选择、软件限位、计数模式
- 3** 打开 MC 调试画面，进行以下操作。  
例)
  - 监视器、配线确认
  - 通过点动进给确认电机的方向
  - 通过相对定位确认移动量 (电子齿轮的设定)
  - 原点复位动作的确认

### 7-3-5 用户程序的动作确认

上机进行用户程序的动作确认时，应将控制器的动作模式设为运行模式。动作确认包括以下项目。

- 确认用户程序的动作
- 通过在线编辑修改用户程序
- 通过数据追踪确认用户程序的动作

#### 确认用户程序的动作

用户程序的动作确认按以下步骤进行。

- 1 监视用户程序的执行状态**
- 2 更新接点或线圈的状态、变量的值，进行动作确认**

##### ● 监视用户程序的执行状态

对控制器的接点、线圈的 ON/OFF 状态、变量的当前值进行监视。监视可通过 Sysmac Studio 的梯形图编辑器、监视窗口、I/O 映射进行。

##### ● 更新接点或线圈的状态、变量的值，进行动作确认

更新用户程序的接点、线圈的 ON/OFF 状态、变量的值，通过监视确认是否设计动作。若要变更接点或线圈的状态，则使用“强制值刷新”，若要变更变量的值，则使用“变更当前值”。

详情请参考□□「8-4-1 强制值刷新」(P.8-27)、□□「8-4-2 当前值变更」(P.8-31)。

#### 通过在线编辑修改用户程序

通过动作确认，如果有判断为需要修改的用户程序，则通过“在线编辑”进行修改。使用“在线编辑”，无需停止控制器的运行，即可变更用户程序。

详情请参考□□「8-4-3 在线编辑」(P.8-33)。

#### 通过数据追踪确认动作

使用“数据追踪”，确认接点或线圈的 ON/OFF 时间和变量值的变动等。

详情请参考□□「8-4-4 数据追踪」(P.8-35)。

### 7-3-6 正式运行的开始

将动作模式设为运行模式，开始正式运行。请对编写完成的用户程序及各种数据、设定值进行充分的动作确认后，再转移到正式运行。

# 8

## 控制器内置功能

本章介绍 NY 系类控制器中可使用的功能。

<b>8-1</b>	<b>数据管理、时钟、运行相关功能</b>	<b>8-2</b>
8-1-1	存储器全部清除	8-2
8-1-2	时钟功能	8-2
<b>8-2</b>	<b>SD 存储卡功能</b>	<b>8-5</b>
8-2-1	SD 存储卡功能一览	8-5
8-2-2	共享文件夹和虚拟 SD 存储卡设定	8-6
8-2-3	虚拟 SD 存储卡的识别和识别解除	8-6
8-2-4	Windows 的共享文件夹及文件规格	8-8
8-2-5	SD 存储卡指令	8-9
8-2-6	FTP 客户端通信指令	8-10
8-2-7	FTP 服务器功能	8-10
8-2-8	通过 Sysmac Studio 进行的文件操作	8-10
8-2-9	SD 存储卡相关的系统定义变量一览	8-11
8-2-10	虚拟 SD 存储卡中文件访问的排他控制	8-12
<b>8-3</b>	<b>安全功能</b>	<b>8-13</b>
8-3-1	用户程序执行用 ID 认证功能	8-14
8-3-2	无用户程序复原信息传送功能	8-17
8-3-3	项目文件整体的保护功能	8-18
8-3-4	数据保护功能	8-19
8-3-5	操作权限的认证功能	8-21
8-3-6	CPU 单元的写入保护功能	8-23
8-3-7	CPU 单元名称功能及序列 ID 功能	8-24
<b>8-4</b>	<b>调试功能</b>	<b>8-27</b>
8-4-1	强制值刷新	8-27
8-4-2	当前值变更	8-31
8-4-3	在线编辑	8-33
8-4-4	数据追踪	8-35
8-4-5	微分监控	8-40
<b>8-5</b>	<b>事件日志功能</b>	<b>8-45</b>
8-5-1	概要	8-45
8-5-2	事件日志功能的详情	8-46
8-5-3	控制器事件（控制器异常 / 信息）	8-51
8-5-4	用户事件（用户异常 / 信息）	8-52
<b>8-6</b>	<b>事件重要程度变更功能</b>	<b>8-58</b>
8-6-1	事件重要程度变更的用途	8-58
8-6-2	可变更事件重要程度的事件	8-58
8-6-3	事件重要程度变更的操作方法	8-59

## 8-1 数据管理、时钟、运行相关功能

下面介绍数据管理功能、时钟和运行相关功能。

### 8-1-1 存储器全部清除

可通过 Sysmac Studio，将 NY 系列控制器内的用户程序、控制器配置和设定、变量、绝对值编码器原点位置偏置初始化为出厂状态。这种操作称为“存储器全部清除”。



#### 使用注意事项

- 存储器全部清除仅可在程序模式下执行。
- 设定了安全功能的“CPU 单元的写入保护”时，无法执行。
- 请勿在存储器全部清除过程中断开 NY 系列控制器的电源。

存储器全部清除后，与用出厂状态的 NY 系列控制器创建系统构成后的动作相同。

#### ● 通过 Sysmac Studio 操作

通过 Sysmac Studio 与 NY 系列控制器在线连接，从 [ 控制器 ] 菜单中选择 [ 存储器全部清除 ]。

操作详情请参考  《Sysmac Studio Version 1 操作手册（SBCA-CN5-362）》。

### 8-1-2 时钟功能

下面就 NY 系列工业电脑内置时钟中控制器的时钟进行说明。

#### 概要

NY 系列工业电脑中，Windows 以及安装有实时 OS 的控制器分别拥有时钟和时区。

关于时钟，Windows 侧的时钟时刻为主时钟，控制器侧的时钟时刻与 Windows 的时钟同步。因此，Windows 侧的时钟时刻可以变更，但控制器侧的时钟时刻无法变更。

无法从 Windows 获取时刻（如只关闭 Windows 等）时，控制器侧的时钟时刻不与 Windows 的时钟同步。在不同步期间，控制器侧的时钟仍会继续运行。

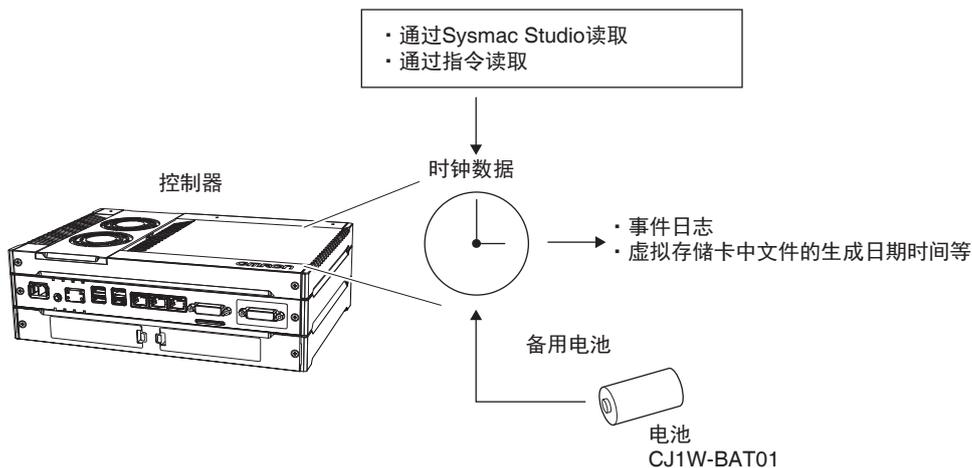
可以从 Windows 获取时刻（如重新启动 Windows 等）时，控制器侧的时钟时刻将再次与 Windows 的时钟同步。

控制器侧的时钟时刻数据将反映到事件日志、虚拟 SD 存储卡的文件生成日期时间中。

此外，控制器侧的时区也与 Windows 的时区相同。  
Windows 的时区和控制器的时区不可单独设定。

控制器的时钟有以下功能。

- 通过 Sysmac Studio 读取（不可写入）
- 通过指令读取（不可写入）
- 通过系统定义变量读取（不可写入）



### ● 控制器侧的时刻数据范围

- 2000/1/1 ~ 2099/12/31

### ● 时区和本地时间的设定

控制器侧的时区和本地时间（时刻数据）以 Windows 的时区设定为准。因此，首次使用时，请在 Windows 上登录时区和本地时间（时刻数据）。

EtherCAT 从站从控制器读取的时刻数据及设定的时刻数据为与时区对应的本地时间。



#### 参考

未安装电池或电池用完时，时区设定会保持，但时刻数据不会保持，将变为不正确的值。

## 时刻数据的读取方法

时刻数据不正确时，将直接读取不正确的值。

- **通过指令读取（不可写入）**

在用户程序中，可使用 `GetTime` 指令读取时刻数据。

- **通过系统定义变量读取（不可写入）**

使用以下系统定义变量，可读取时刻数据。

“\_CurrentTime”（系统时刻）

- **通过 Sysmac Studio 读取（不可写入）**

可使用 Sysmac Studio，通过 [控制器] 菜单的 [控制器时钟]，读取时刻数据并显示。

## 相关系统定义变量

变量名称	名称	功能	数据类型	R/W
_CurrentTime	系统时刻	保存控制器的内部时钟数据。	DATE_AND_TIME	R

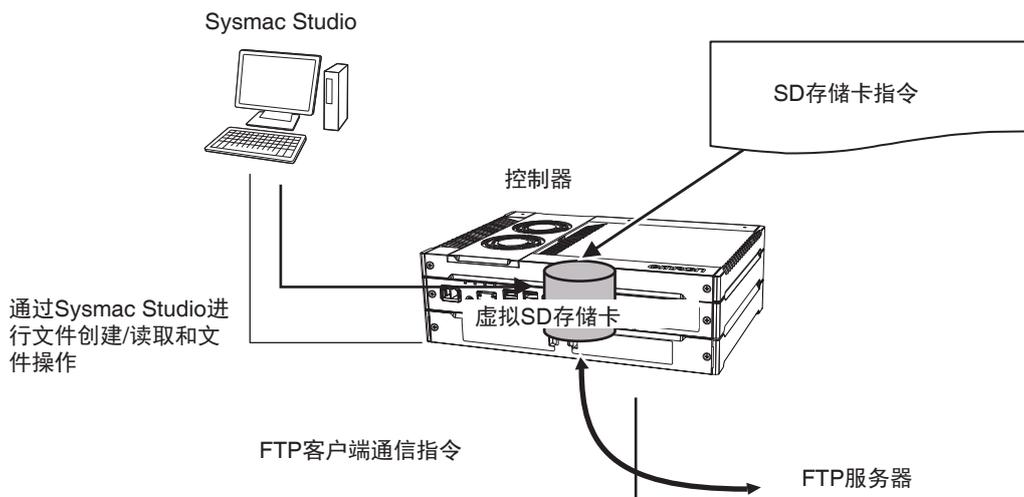
## 8-2 SD 存储卡功能

下面介绍 SD 存储卡功能。

NY 系列控制器中，将 Windows 的共享文件夹作为虚拟 SD 存储卡使用。本手册中可能会省略“虚拟”，记载为“SD 存储卡”。

### 8-2-1 SD 存储卡功能一览

NY 系列控制器支持的 SD 存储卡功能如下。



功能名称	概要
SD 存储卡指令	通过用户程序中的指令访问虚拟 SD 存储卡的功能。
FTP 客户端通信指令	通过控制器使用 FTP 向 Ethernet 上的计算机及控制器传输文件所需的指令。
FTP 服务器功能	通过内网的 FTP 客户端，使用 FTP 指令，经由 EtherNet/IP，读写虚拟 SD 存储卡中的大容量文件的功能。
通过 Sysmac Studio 进行的文件操作功能 <sup>*1</sup>	通过 Sysmac Studio 对控制器中的虚拟 SD 存储卡进行文件操作的功能。用户不仅可以向虚拟 SD 存储卡保存控制器用文件，还可以保存通用文档文件等。
SD 存储卡的备份功能	使用控制器中的 SD 存储卡，对控制器的用户程序和数据进行备份、恢复、核对的功能。详情请参考 [9-2 SD 存储卡的备份功能] (P.9-10)。

\*1. NY 系列控制器中，不可进行 SD 存储卡的初始化或变更文件夹、文件的属性。

此外，虚拟 SD 存储卡使用共享文件夹，因此 Sysmac Studio 的 SD 存储卡窗口的状态栏中不会显示卷标。

### 8-2-2 共享文件夹和虚拟 SD 存储卡设定

下面介绍在 NY 系列控制器中使用 SD 存储卡功能所需的设定。

#### 共享文件夹的设定

设定虚拟 SD 存储卡前，需要在 Windows 上设定共享文件夹。

NY 系列工业电脑的 Windows 在初始状态下，设定的虚拟 SD 存储卡共享文件夹为“D:\OMRON-NY\VirtualSDCard”。

关于共享文件夹的详细设定，请参考 [□□](#) 《NY 系列 工业平板电脑 / 工业 BOX PC 用户手册 安装篇 (SBCA-CN5-441)》。

#### 虚拟 SD 存储卡的设定

虚拟 SD 存储卡的设定在 Symac Studio 的控制器设定中进行。

关于虚拟存储卡设定的设定项目，请参考 [□□](#) 「虚拟 SD 存储卡设定 (P.4-5)」。



#### 参考

可以不通过 Symac Studio，使用 Industrial PC Support Utility 进行虚拟 SD 存储卡设定。

关于使用 Industrial PC Support Utility 设定虚拟 SD 存储卡的方法，请参考 [□□](#) 《NY 系列 工业平板电脑 / 工业 BOX PC 用户手册 安装篇 (SBCA-CN5-441)》。

### 8-2-3 虚拟 SD 存储卡的识别和识别解除

下面介绍虚拟 SD 存储卡的识别和识别解除。

#### 虚拟 SD 存储卡的识别

执行以下操作后，将识别虚拟 SD 存储卡。

识别到虚拟 SD 存储卡后，系统定义变量 `_Card1Ready` (SD 存储卡可使用标志) 变为 TRUE。

- NY 系列控制器的启动
- 执行下载
- 执行恢复
- 通过 Industrial PC Support Utility 变更虚拟 SD 存储卡的设定
- 通过 Industrial PC Support Utility 确认虚拟 SD 存储卡的识别

以下情况下，将无法识别虚拟 SD 存储卡。

- Windows 停止
- Windows 的文件共享服务停止
- 虚拟 SD 存储卡的设定不正确

## 虚拟 SD 存储卡的识别解除

以下情况下，将解除虚拟 SD 存储卡的识别。

原因
Windows 停止（含重新启动）
Windows 的共享服务停止·无效
Windows 的电脑名称变更
Windows 内部端口的 IP 地址变更
共享文件夹的共享解除
共享文件夹删除
共享文件夹的访问权变更
控制器侧内部通信端口的 IP 地址变更为与 Windows 侧内部通信端口不同的 IP 地址

解除虚拟 SD 存储卡的识别后，系统定义变量 `_Card1Ready`（SD 存储卡可使用标志）变为 `FALSE`，发生“共享文件夹识别解除完成”事件。

正在访问虚拟 SD 存储卡时，如果解除识别，正在执行的文件访问将失败。

解除后的动作因解除原因不同而异。

### ● 原因为“共享文件夹的共享解除”或“共享文件夹的删除”时

解除虚拟 SD 存储卡的原因为“共享文件夹的共享解除”或“共享文件夹的删除”时，如果只在 Windows 侧再次打开文件夹的共享，不会变为识别状态。

可通过以下操作再次变为识别状态。

- 再次打开文件夹的共享，且通过 Industrial PC Support Utility 执行 [识别确认]。但是，如果在解除共享前通过 SD 存储卡指令打开了文件，需要关闭对象文件。
- 再次打开文件夹的共享，且解除共享后已经过 Windows 的段落·最大空闲时间（初始值 15 分钟）。

### ● 原因为“共享文件夹的共享解除”或“共享文件夹的删除”以外时

排除原因并恢复后，自动变为识别状态。

识别到虚拟 SD 存储卡时，如果进行下载、恢复、存储器全部清除或通过 Industrial PC Support Utility 传送设定，将变为以下状态。

- 虚拟 SD 存储卡的识别解除
- 系统定义变量“`_Card1Ready`（SD 存储卡可使用标志）”为 `FALSE`
- 发生“共享文件夹识别解除完成”事件

但是，正在访问虚拟 SD 存储卡时，不会解除虚拟 SD 存储卡的识别，会发生“共享文件夹识别解除失败”事件。



### 使用注意事项

正在访问虚拟存储卡时，请勿关闭工业电脑的电源。

如果关闭电源，文件可能损坏。

此外，再次启动时，系统定义变量“`_Card1PowerFail`（SD 存储卡访问中断电标志）”变为 `TRUE`。

## 8-2-4 Windows 的共享文件夹及文件规格

下面介绍作为虚拟 SD 存储卡使用的共享文件夹和文件的规格。

### 共享文件夹的规格

下面介绍在 Windows 上设定的共享文件夹的规格。

- 可设定的存储设备

工业电脑中内置的 HDD 或 SSD 中设定的共享文件夹可作为虚拟 SD 存储卡使用。

工业电脑中内置的 HDD 或 SSD 以外中设定的共享文件夹不能保证虚拟 SD 存储卡的动作。

- 共享文件夹的容量限制

作为虚拟 SD 存储卡使用的共享文件夹应在 4GB 以内。

共享文件夹中的文件容量超过 4GB 时，可能会无法正确运行。

- 文件系统

可作为虚拟 SD 存储卡使用的文件系统仅限 NTFS。

### 文件夹 / 文件的规格

- 字符限制

用户命名的内容	可使用的字符	保留字	多字节字符使用可否	大小写的区分	最大大小（不含末尾的 NULL 字符）
目录名	0 ~ 9、A ~ Z、a ~ z	CON, PRN, AUX, CLOCK\$, NUL, COM0, COM1, COM2, COM3, COM4, COM5, COM6, COM7, COM8, COM9, LPT0, LPT1, LPT2, LPT3, LPT4, LPT5, LPT6, LPT7, LPT8, LPT9	不支持*1	否	65 字节
文件名	\$ % ' - _ @ ! ` ( ) ~ = # & + ^ [ ] { } , . ;				65 字节

\*1. 多字节字符（日文等）在计算机上可以使用，但在控制器上无法使用。



#### 使用注意事项

目录名或文件名中包含可使用字符以外的字符时，控制器将无法正确识别。可能无法通过指令、Sysmac Studio 的 SD 存储卡窗口及 FTP 服务器功能找到文件。

### ● 子目录层级

最多 5 层

例) 共享文件夹的共享名为“VirtualSDCard”时

VirtualSDCard\f1\f2\f3\f4\f5\abc.txt

### ● 保存文件的数量限制

保存到虚拟 SD 存储卡的文件数应在下表的值以内。文件数超过下表的值时，可能会无法正常运行。

目录层级	保存的文件数
根目录	65,535
子目录	65,534

### ● 1 个文件的最大大小

1 个文件的最大大小为 2,147,483,647 字节 (2GB-1 字节)。

## 8-2-5 SD 存储卡指令

可利用以下指令，对虚拟 SD 存储卡进行各种操作。

指令名称	指令	功能
读取变量文件	<b>FileReadVar</b>	读取虚拟 SD 存储卡中的文件（二进制格式），保存到指定的变量（排列变量或结构体变量亦可）中。
写入变量文件	<b>FileWriteVar</b>	将指定变量（排列变量或结构体变量亦可）的值以二进制格式写入到虚拟 SD 存储卡的文件中。 文件名中指定的目录不存在时，将同时创建该目录。
打开文件	<b>FileOpen</b>	打开指定的文件。
关闭文件	<b>FileClose</b>	关闭指定的文件。
查找文件	<b>FileSeek</b>	设定指定文件的文件位置指示符。
读取文件	<b>FileRead</b>	从指定的文件中读取数据。
写入文件	<b>FileWrite</b>	将数据写入指定的文件中。
读取字符串	<b>FileGets</b>	从指定的文件中读取 1 行字符串。
写入字符串	<b>FilePuts</b>	将 1 行字符串写入指定的文件中。
删除文件	<b>FileRemove</b>	删除虚拟 SD 存储卡中的指定文件。
重命名文件	<b>FileRename</b>	变更指定文件的文件名或目录名。
复制文件	<b>FileCopy</b>	将指定的文件复制为其他文件名。
创建目录	<b>DirCreate</b>	在虚拟 SD 存储卡中创建目录。
删除目录	<b>DirRemove</b>	删除虚拟 SD 存储卡中的指定目录。

### 8-2-6 FTP 客户端通信指令

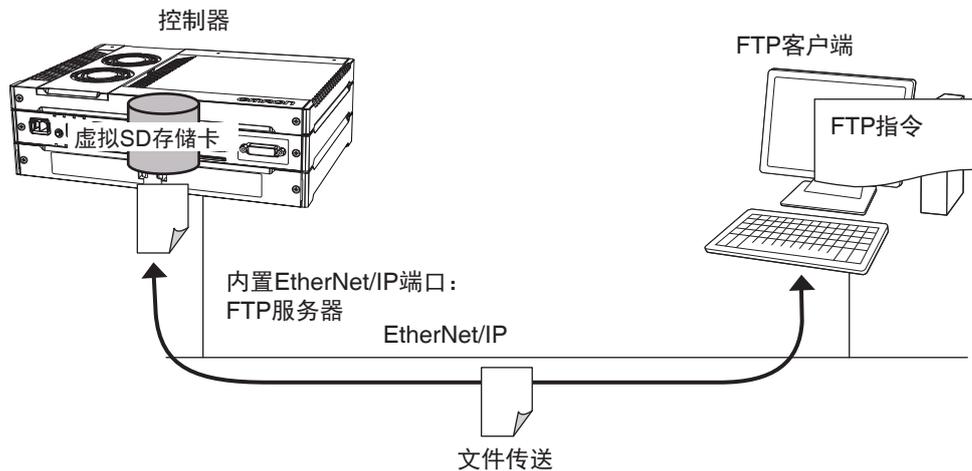
FTP 客户端通信指令是指，通过 NY 系列控制器使用 FTP 向 Ethernet 上的计算机及控制器传输文件所需的指令。以下指令中，在执行指令时，将对虚拟 SD 存储卡中的文件进行读写。

指令名称	指令	功能
向 FTP 服务器上传文件	FTPPutFile	将 FTP 客户端的虚拟 SD 存储卡中的文件上传到 FTP 服务器上。
从 FTP 服务器下载文件	FTPGetFile	将 FTP 服务器上的文件下载到 FTP 客户端的虚拟 SD 存储卡中。

### 8-2-7 FTP 服务器功能

可通过 FTP 客户端，对内置 EtherNet/IP 端口发行 FTP 指令，经由 EtherNet/IP，读写虚拟 SD 存储卡中的文件。

详情请参考《NY 系列 工业平板电脑 / 工业 BOX PC 用户手册 内置 EtherNet/IP 端口篇 (SBCD-CN5-369)》。



### 8-2-8 通过 Sysmac Studio 进行的文件操作

可通过 Sysmac Studio 对 NY 系列控制器中的虚拟 SD 存储卡进行文件操作。不仅可以向虚拟 SD 存储卡保存控制器用文件，还可以保存文档文件等。

### 8-2-9 SD 存储卡相关的系统定义变量一览

作为 SD 存储卡功能的状态进行通知的系统定义变量如下所示。

变量名称	名称	功能	数据类型
_Card1Ready	SD 存储卡可使用标志	NY 系列识别到虚拟 SD 存储卡时，变为 TRUE。 未识别到时，为 FALSE。 TRUE：可使用 FALSE：不可使用	BOOL
_Card1Protect	SD 存储卡写保护标志	NY 系列中不使用。 始终为 FALSE。	BOOL
_Card1Err	SD 存储卡错误标志	NY 系列中不使用。 始终为 FALSE。	BOOL
_Card1Access	SD 存储卡访问中标志	NY 系列中不使用。 始终为 FALSE。	BOOL
_Card1Deteriorated	SD 存储卡寿命警告标志	NY 系列中不使用。 始终为 FALSE。	BOOL
_Card1PowerFail	SD 存储卡访问中断电标志 <sup>*1</sup>	正在访问虚拟 SD 存储卡的过程中，若控制器发生断电，则变为 TRUE。 TRUE：正在访问虚拟 SD 存储卡时发生断电 FALSE：正常	BOOL

\*1. 使用 SD 存储卡访问中断电标志 (“\_Card1PowerFail”) 时的注意事项

SD 存储卡访问中断电标志为 TRUE 时，请确认虚拟 SD 存储卡中是否存在正确的文件，或装置是否正确动作。不存在正确的文件或装置无法正确动作时，请将正确的文件下载到虚拟 SD 存储卡中。请重新接通控制器的电源或重置控制器，确认装置正确动作

处理完成后，请将 SD 存储卡访问中断电标志设为 FALSE。 (本标志不会自动变为 FALSE)。

(注) SD 存储卡备份功能相关的系统定义变量请参考 [□「9-2 SD 存储卡的备份功能」](#) (P.9-10)。

### 8-2-10 虚拟 SD 存储卡中文件访问的排他控制

虚拟 SD 存储卡中的文件访问方法有以下 4 种。

- (1) FTP 服务器功能
- (2) SD 存储卡指令
- (3) FTP 客户端通信指令
- (4) 通过 Sysmac Studio 进行的文件操作

若同时对虚拟 SD 存储卡中的同一文件进行访问，可能发生意外处理，如读取正在写入的文件、覆盖正在读取的文件等。因此，需要进行排他控制，防止对同一文件同时执行上述处理。以下表示需要排他控制的组合。执行需要排他控制的组合处理时，请在确认一个处理结束后，再执行另一个处理。

			前面正在执行的访问					
			指令*1		通过 Sysmac Studio 进行的文件操作		FTP 服务器功能	
			读取	写入	读取	写入	读取	写入
之后执行的访问	指令	读取	自动进行排他控制，之后执行的指令将发生错误。		无需排他控制。	请执行排他控制。	无需排他控制。	请执行排他控制。
		写入			请执行排他控制。		请执行排他控制。	
	通过 Sysmac Studio 进行的文件操作	读取	无需排他控制。	请执行排他控制。	—	—	无需排他控制。	
		写入	请执行排他控制。		—	—	请执行排他控制。	
	FTP 服务器功能	读取	无需排他控制。	请执行排他控制。	无需排他控制。	请执行排他控制。	—	—
		写入	请执行排他控制。				—	—

\*1. 指令包括虚拟 SD 存储卡指令和 FTP 客户端通信指令。

NY 系列控制器中，可从 Windows 端访问共享文件夹的文件。此时的排他控制遵照 Windows 的文件共享规格。

## 8-3 安全功能

下面介绍 NY 系列控制器中可使用的安全功能。

使用安全功能后，可保护控制器中的用户程序和各种数据，以保护资产。此外，为了防止误操作，还可限制 Sysmac Studio 上的操作。

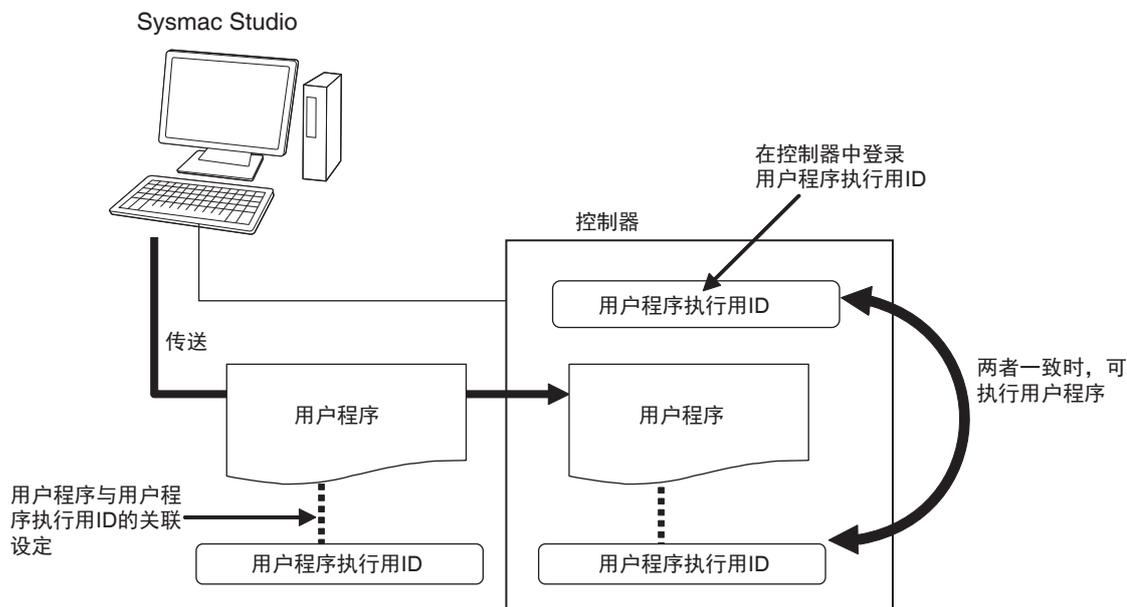
NY 系列控制器拥有以下安全功能。

目的	安全功能	功能概要	参考
防止资产盗用	用户程序执行用 ID 认证功能	即使复制了用户程序，也无法在其他控制器上运行。	☐「8-3-1 用户程序执行用 ID 认证功能」(P.8-14)
	无用户程序复原信息传送功能	不向控制器传送程序的复原信息（源代码），用其他计算机上传时，将不会显示程序。	☐「8-3-2 无用户程序复原信息传送功能」(P.8-17)
	项目文件整体的保护功能	对项目文件设置密码保护，保护用户资产信息。	☐「8-3-3 项目文件整体的保护功能」(P.8-18)
	数据保护功能	对项目中需要保护的数据进行保护，以保护用户资产信息。	☐「8-3-4 数据保护功能」(P.8-19)
防止误操作	操作权限的认证功能	在 Sysmac Studio 上对控制器设定操作权限，限制可操作的功能。	☐「8-3-5 操作权限的认证功能」(P.8-21)
	CPU 单元的写入保护功能	将无法通过 Sysmac Studio 替换控制器中的数据。	☐「8-3-6 CPU 单元的写入保护功能」(P.8-23)
防止误连接	CPU 单元名称功能	确认计算机和控制器的控制器名称或序列 ID 是否一致，防止在线连接错误。	☐「8-3-7 CPU 单元名称功能及序列 ID 功能」(P.8-24)

## 8-3-1 用户程序执行用 ID 认证功能

## 概要

事先在控制器上设定特定的 ID（称为“用户程序执行用 ID”），可只执行与该 ID 关联的用户程序。

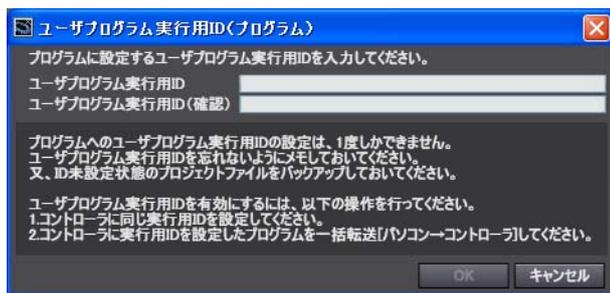


因此，可防止在不同的控制器（硬件）上使用某个用户程序。

此外，本功能与保护功能不同，程序仍可显示和编辑。

## 操作方法

- 1 尚未进行用户程序执行用 ID 关联设定的项目文件必须事先备份。
- 2 在 Sysmac Studio 上，以离线的状态，在用户程序上进行用户程序执行用 ID 的关联设定。

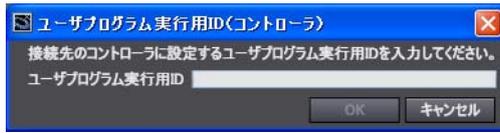


## 使用注意事项

进行了用户程序执行用 ID 关联设定的用户程序将不可进行 ID 的变更及删除。

若要变更，请读取步骤 1 中事先备份的项目文件，再关联为不同的用户程序执行用 ID。若要删除，请使用步骤 1 中事先备份的项目文件。

### 3 将 Sysmac Studio 在线连接，在控制器中登录 2 中关联的用户程序执行用 ID。



向控制器登录用户程序执行用 ID 的操作将登录到事件日志中。

此时，即使已登录用户程序执行用 ID，也会覆盖登录。

### 4 将关联设定为相同用户程序执行用 ID 的用户程序传送到控制器。

如果用户程序侧和控制器侧的用户程序执行用 ID 不一致，或者一方不存在时，变更为运行模式后，将发生 ID 核对错误（全部停止故障等级的控制器异常），无法运行。



#### 使用注意事项

进行了用户程序执行用 ID 关联设定的控制器将不可进行 ID 的删除及读取。如果要删除，请对控制器进行存储器全部清除。

#### ● 发生“ID 核对错误”时的操作

控制器侧的用户程序执行用 ID 不正确或未登录时：

通过 Sysmac Studio 与控制器在线连接，

- 1 将正确的用户程序执行用 ID 登录到控制器中。
- 2 请将控制器电源从 OFF 设为 ON，或通过 Sysmac Studio 进行复位操作。

用户程序侧的用户程序执行用 ID 不正确或未进行关联设定时：

- 1 请用 Sysmac Studio 读取已备份的项目文件，再关联为正确的用户程序执行用 ID。
- 2 请将 Sysmac Studio 与控制器在线连接，传送用户程序。
- 3 请将控制器电源从 OFF 设为 ON，或通过 Sysmac Studio 进行控制器复位操作。

- 其他情况时

**要删除用户程序侧的用户程序执行用 ID 时：**

请用 Sysmac Studio 读取已备份的项目文件。

**要删除控制器侧的用户程序执行用 ID 时：**

请将 Sysmac Studio 与控制器在线连接，进行存储器全部清除。

**要确认用户程序侧的用户程序执行用 ID 时：**

出于安全考虑，不可通过 Sysmac Studio 等确认与用户程序关联的用户程序执行用 ID。请根据备份的项目文件，重新进行关联设定。

**要确认控制器侧的用户程序执行用 ID 时：**

出于安全考虑，不可通过 Sysmac Studio 等确认控制器中登录的用户程序执行用 ID。请在进行存储器全部清除后，重新登录。

## 规格

- 用户程序执行用 ID 的核对规格

### 核对时序

控制器将在开始运行时核对控制器中登录的用户程序执行用 ID 以及与用户程序关联的用户程序执行用 ID。

### 核对条件

核对条件如下所示。

A、B 表示 ID。

控制器中登录的用户程序执行用 ID	与用户程序关联的用户程序执行用 ID	异常	运行
A	A	无	可以
无	无		
无	A	ID 核对错误	不支持
A	无		
A	B		

### 核对不一致时的动作

核对不一致时，将发生“ID 核对错误”（全部停止故障等级的控制器异常），控制器无法运行。

要解除异常，需要将控制器的电源从 OFF 设为 ON，或通过 Sysmac Studio 重置控制器。

- 用户程序执行用 ID 中可使用的文字规格

可使用的字符	大小写的区分	最大大小（不含 NULL）
0~9、A~Z、a~z	是	8~32 个字符

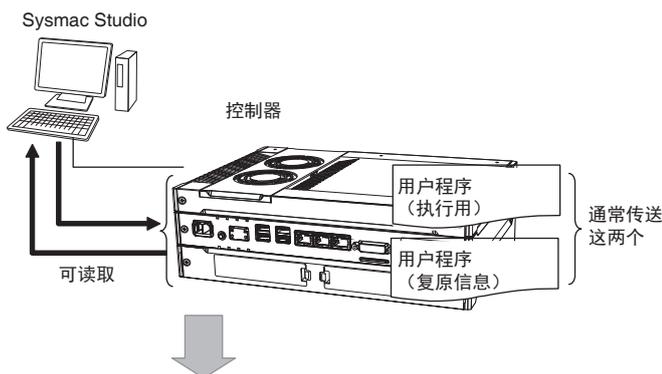
### 8-3-2 无用户程序复原信息传送功能

不向控制器传送程序的复原信息（源代码），用其他计算机上传时，将不会显示程序。

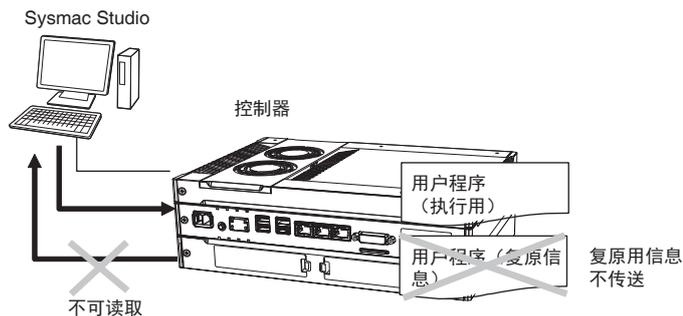
#### 概要

通常，通过 Sysmac Studio 向控制器传送用户程序时，会同时传送复原所需的信息。利用本功能，可以不传送复原所需的信息，禁止读取用户程序。

##### ●正常传送时



##### ●无用户程序复原信息传送时



用户无需在现场修改用户程序且需要防止用户程序数据被盗用的情况。

#### 操作方法

向控制器传送用户程序时，在 Sysmac Studio 的同步窗口，勾选 [不传送程序的源代码] 设定，单击 [传送 [计算机 | 控制器]] 按钮。

- 保持変数の現在値を初期化する (有効: パソコン→コントローラ)。
- プログラムのソースを転送しない (有効: パソコン→コントローラ)。本オプションを変更すると全データを再転送します。
- 高機能ユニットのパラメータとEtherCATスレーブのバックアップパラメータ(同期の対象外)は転送しない。

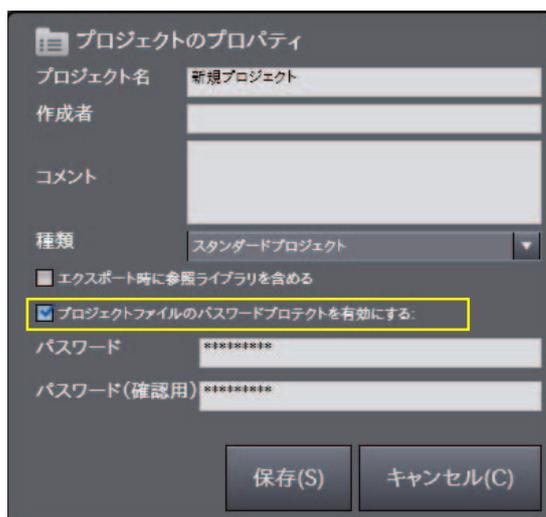
### 8-3-3 项目文件整体的保护功能

对项目文件设置密码保护，保护用户资产信息。

#### 操作方法

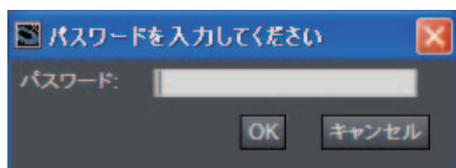
下面介绍为项目设定密码时的操作。

另存项目文件时，如果勾选 [开启项目文件的密码保护]，则可以设定密码。



下面介绍打开了设置了密码的项目时的操作。

打开或导入设置了密码的项目文件时，将显示 [输入密码] 对话框。



本功能的操作详情请参考 [□](#) 《Sysmac Studio Version 1 操作手册（SBCA-CN5-362）》。

### 8-3-4 数据保护功能

对项目数据中需要保护的数据进行保护，以保护用户资产信息。

#### 概要

以项目数据中需要保护的数据为单位设定保护，以限制对数据的访问。

设定保护时，可从 3 种强度的访问限制中选择并设定。

访问设有访问限制的数据时，需要临时解除保护。临时解除的有效期限因执行的操作不同而异。

#### 保护对象数据

可进行保护设定的数据如下所示。

- 梯形图程序（程序 / 函数 / 功能块共通）。
- ST 程序（程序 / 函数 / 功能块共通）。
- 凸轮轮廓

#### 访问限制的种类

访问限制的强度可从以下 3 种中选择。但是，凸轮轮廓只能选择禁止变更。

- 复制、显示、禁止变更
- 显示、禁止变更
- 禁止变更

对数据的访问操作以及不同访问限制种类的限制和临时解除保护时的有效期限关系如下。

（○：可限制操作、—：不可限制操作）

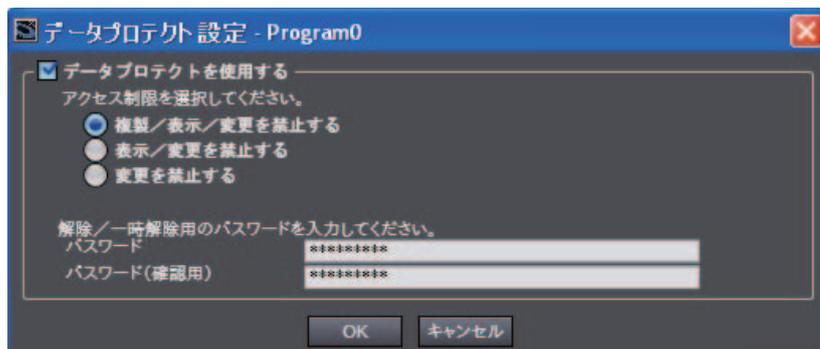
数据访问的操作	访问限制的种类			临时解除有效期限
	复制、显示、禁止变更	显示、禁止变更	禁止变更	
对象物的显示	○	○	—	已打开项目时
对象物的打印	○	○	—	
对象物的变更	○	○	○	
对象物的复制	○	—	—	每次操作都需要临时解除
核对结果（概要）的显示	—	—	—	不限制本操作
核对结果（详情）的显示	○	○	—	已打开项目时
从事件日志或交叉参考跳转到内容	○	○	—	
登录到库	○	—	—	不能临时解除。

## 操作方法

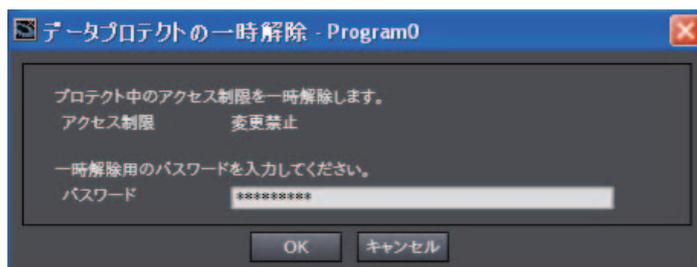
本功能可离线操作 Sysmac Studio。

设定内容将保存到项目中。如果用 Sysmac Studio 的同步功能传送项目，传送方（计算机 / 控制器）对象数据的数据保护设定将传送到接收方（控制器 / 计算机）。

保护设定请在 Sysmac Studio 的 [控制器] - [安全] - [数据保护设定 / 解除] 中设定。



要临时解除保护，请选择 [控制器] - [安全] - [禁止变更的临时解除]。



要关闭临时解除保护，请选择 [控制器] - [安全] - [关闭禁止变更的临时解除]。

本功能的操作详情请参考 [□](#) 《Sysmac Studio Version 1 操作手册（SBCA-CN5-362）》。

### 8-3-5 操作权限的认证功能

#### 概要

因操作失误可能对装置或人施加危害，根据操作权限限制在线操作功能的功能。

例如以下情况。

- I/O 监视器 : 写入、强制值刷新等
- 控制器操作 : 动作模式变更、在线编辑、轴调试等

通过 **Sysmac Studio** 在控制器中登录操作权限的密码，在线连接时输入操作权限的密码后，与操作权限类别对应的在线操作功能即变为可用。

管理员应设定不同操作权限的密码，根据技能水平将操作权限名称和密码告知使用者。

操作权限相关设定的操作详情请参考 [□](#) 《Sysmac Studio Version 1 操作手册（SBCA-CN5-362）》。

#### 操作方法

操作权限的认证功能可选择 **Sysmac Studio** 的 [控制器] - [安全] - [操作权限的设定] 后执行。

操作方法的详情请参考 [□](#) 《Sysmac Studio Version 1 操作手册（SBCA-CN5-362）》。

## 规格

### ● 操作权限的种类

按权限从大到小的顺序， Sysmac Studio 的操作权限种类有以下 5 种。

中文名称	英文名称	密码设定
管理员	Administrator	必须
设计者	Planning Engineer	任意*1 设定省略 [操作权限认证设定] 对话框的密码输入时，是否需要密码。
维护者	Maintenance	
操作者	Operator	
观察者	Observer	不需要

\*1. 设定省略 [操作权限认证设定] 对话框的密码输入时，是否需要密码。如果是比省略时的操作权限更高级的权限，则需要密码。相同或更低级的权限则不需要密码。

### ● 对象在线操作功能示例

不同在线操作种类的操作权限示例如下。

详情请参考 □□ 《Sysmac Studio Version 1 操作手册 (SBCA-CN5-362)》。

(○ : 可操作、△ : 每次操作时需要认证、× : 不可操作)

状态监视功能 (示例)	管理员	设计者	维护者	操作者	观察者
异常监视 (故障排除)	○	○	○	○	○

I/O 监视功能 (示例)	管理员	设计者	维护者	操作者	观察者
I/O 监视 (读取)	○	○	○	○	×
I/O 监视 (写入)	○	○	○	△	×
设置 / 复位	○	○	○	△	×
强制值刷新	○	○	○	×	×

控制器操作功能 (示例)	管理员	设计者	维护者	操作者	观察者
运行模式 / 程序模式	○	○	△	×	×
在线编辑	○	○	△	×	×
控制器重置	○	○	×	×	×
异常解除 (故障排除)	○	○	○	△	×
轴调试、轴调试重新开始	○	○	△	×	×
用户程序执行用 ID (控制器)	○	×	×	×	×
CPU 单元的写入保护	○	○	○	×	×

### ● 密码规格

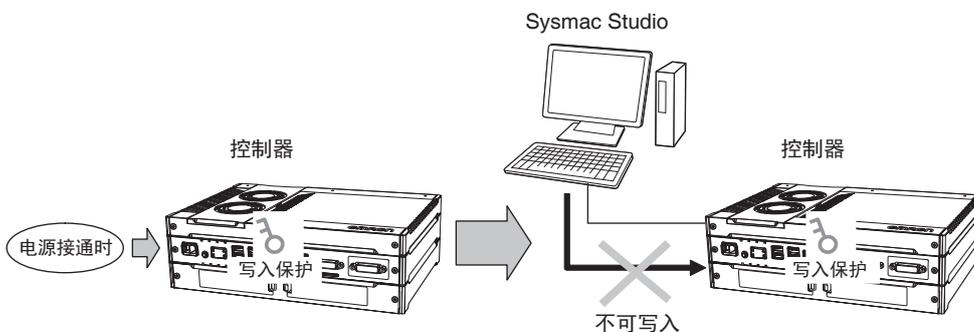
项目	内容
有效字符数	8 个字符以上且 32 个字符以内
可使用的字符	半角英文和数字 (区分大小写)

### 8-3-6 CPU 单元的写入保护功能

为保护用户程序资产和防止误操作，使控制器内各种数据的写入变为无效的功能。包括以下 2 种。

#### ● 控制器电源接通时的写入保护设定

接通控制器的电源时，自动启用写入保护的功能。

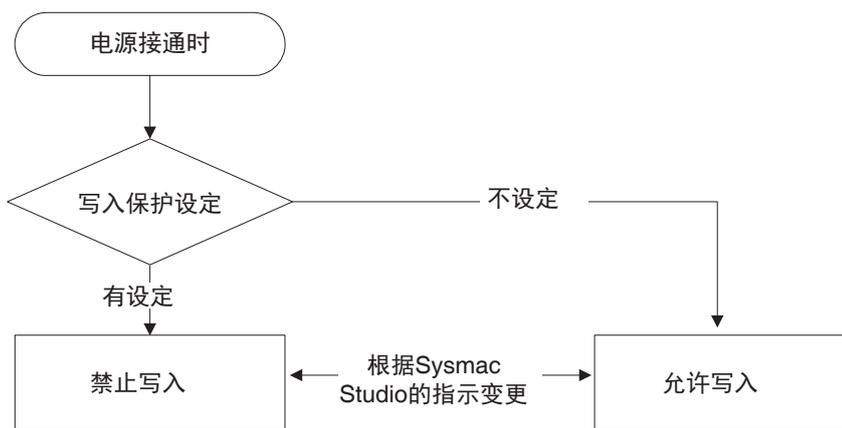
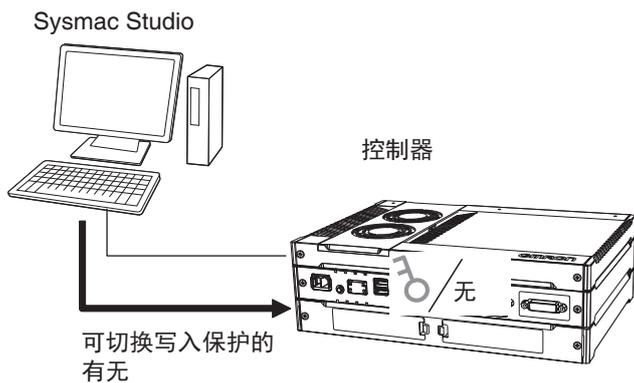


Sysmac Studio 上的操作方法请在 Sysmac Studio 的 [配置和设定]-[控制器设定]-[动作设定] 中，设定在接通电源时是否自动进行写入保护。

设定大项目	设定中项目	设定小项目	功能说明	设定值
动作设定 - 动作设定标签 - 整体设定	安全设定	电源接通时的写入保护	设定启用 / 禁用写入保护。	不保护 保护

#### ● 通过 Sysmac Studio 进行写入保护的“设置” / “解除”切换

通过 Sysmac Studio 进行在线连接，在 [控制器] 菜单的 [安全] - [CPU 单元的写入保护] 中，切换“设置” / “解除”。



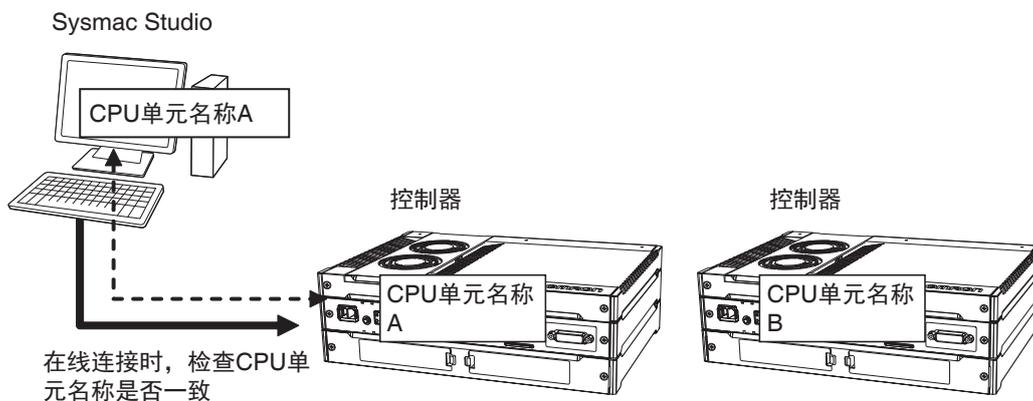
### 8-3-7 CPU 单元名称功能及序列 ID 功能

#### 概要

请在控制器中登录 CPU 单元名称。

通过 Sysmac Studio 进行在线连接时，可确认项目中的 CPU 单元名称与连接对象控制器的 CPU 名称是否一致。

由此，可防止 Sysmac Studio 与错误的控制器连接。特别是通过 EtherNet/IP 操作时非常有效。



此外，除了 CPU 单元名称之外，还可识别与控制器的生产信息对应的序列 ID（选项）。

## 设定方法

- 1 在 **Sysmac Studio** 上创建项目时，设定 **CPU** 单元名称。  
CPU 单元名称显示如下。



要变更名称时，可右击控制器图标，在“重命名”中编辑。

- 2 第一次与控制器在线连接时，**Sysmac Studio** 会提醒是否在控制器中保存 **CPU** 单元名称。
- 3 之后，与控制器在线连接时，**Sysmac Studio** 会对项目上的 **CPU** 单元名称与连接对象控制器的 **CPU** 单元名称是否一致进行核对。  
如果不一致，将显示警告对话框，并提醒是否继续连接。

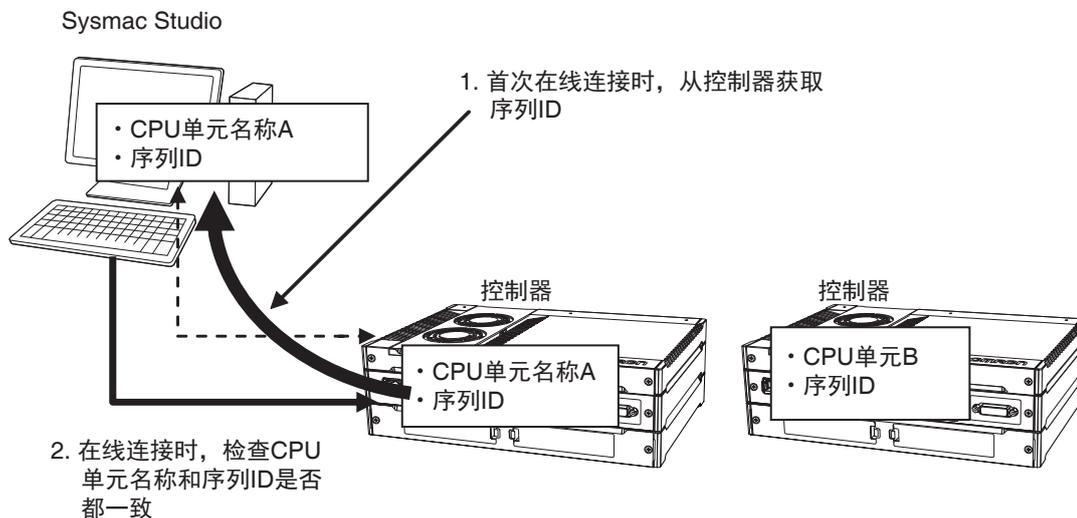


### 参考

NetworkConfigurator 中，可对 EtherNet/IP 端口进行命名。

## 序列 ID 功能

Sysmac Studio 第一次在线连接时，可获取与控制器的生产信息对应的序列 ID，并保存到项目中。然后，Sysmac Studio 进行在线连接时，将核对 CPU 单元名称和序列 ID。由此，可严密地进行控制器核对。



## 8-4 调试功能

下面介绍调试功能。

NY 系列控制器中有以下调试功能。

- 强制值刷新
- 当前值变更
- 在线编辑
- 数据追踪
- 微分监控

### 8-4-1 强制值刷新

#### 概要

强制值刷新是指用户进行调试等时，以 Sysmac Studio 中指定的强制值刷新来自外部的输入或向外部的输出。

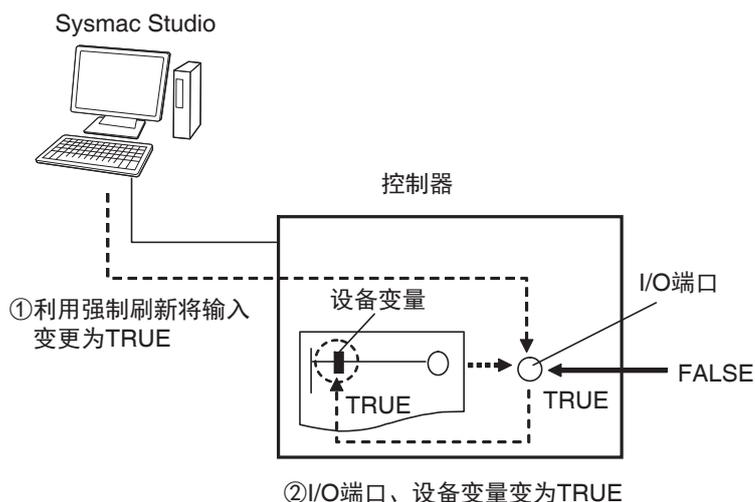
强制值刷新将针对设备变量中分配的 I/O 端口执行，而不是指定的设备变量。

强制值刷新中指定的状态在通过 Sysmac Studio 发出强制状态解除指示前，将一直保持。（关于对控制器状态变化的保持 / 解除，请参考后述的“强制值刷新的保持 / 解除”。）

此外，异常停止时、存储器全部清除时、模式变更时、断电时、下载时将不保存，直接清除。

#### ● 输入

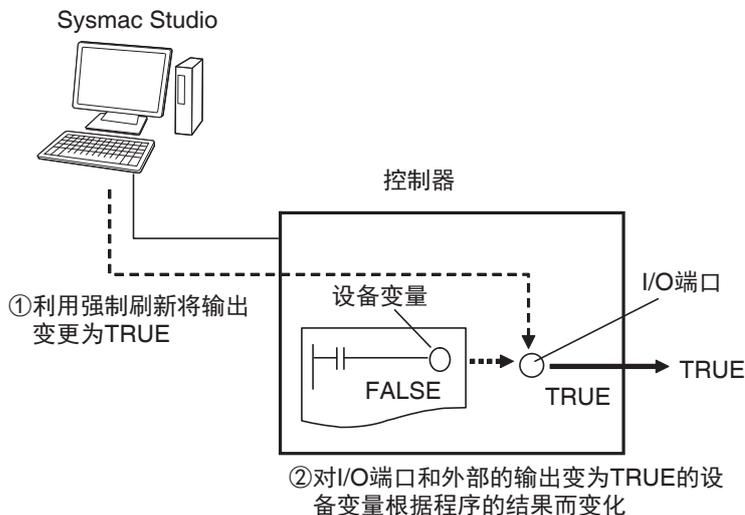
无论来自外部的输入如何，I/O 端口和设备变量将变为强制值刷新中指定的状态。



### ● 输出

I/O 端口和向外部的输出将变为强制值刷新中指定的状态。

用户程序中的、I/O 端口上分配的设备变量的状态将根据用户程序的结果而变化，而不是强制值刷新中指定的状态。



## 对象区域

强制值刷新可对以下 I/O 端口执行。

- EtherCAT 从站的 I/O 端口

## 可同时进行强制值刷新的个数

可进行强制值刷新的个数如下。

- EtherCAT 从站：共 64 点

上述个数为外部输入输出接点的数量。例如，多个变量的分配目标指定为 1 个外部输入输出接点时，计为 1 个。

## 用途

### ● 输入

- 发出虚拟输入信号，进行用户程序的调试
- 创造出只有故障时才可能出现的状态（同时控制排他 2 接点的 ON/OFF 等）

### ● 输出

- 为检查配线，进行输出的 ON/OFF
- 无论用户程序的结果如何，都将强制关闭不需要的输出。

## 操作方法

可在以下画面中操作。

- 程序画面（梯形图语言）
- I/O 映射
- 监视窗口

### ● 梯形编辑器中强制值刷新的操作步骤

- 1** 在 [ 控制器 ] 的 [ 监视器 ] 中打开监视器。
- 2** 在多视图浏览器的 [ 编程 ] 中，双击对象梯形图程序或梯形图功能块 / 函数。  
在梯形图上，回路将以监视状态显示。
- 3** 选择强制值刷新时打开的接点和线圈，右击选择 [ 强制值刷新 ] - [ TRUE ]。或者，选择强制值刷新时要关闭的接点和线圈，右击选择 [ 强制值刷新 ] - [ FALSE ]。
- 4** 梯形编辑器上的对象接点和线圈将强制变为 On 或 Off 状态，On 时为导通状态，Off 时为非导通状态。  
接点、线圈中将显示如下强制状态的标志。



梯形图

强制状态的“TRUE”“FALSE”显示为强制值刷新中指定的状态，并不是显示接点或线圈的当前值。

强制状态标志	操作
	指定了强制值刷新（TRUE）时
	指定了强制值刷新（FALSE）时



### 参考

如果有变量指定了与强制值刷新变量的 AT（分配目标）指定位置相同的存储器地址，该变量中也会显示强制状态标志。

## 动作模式与电源断开的关系

### ● 可进行强制值刷新的动作模式

可在程序模式、运行模式的任意模式下执行。

但是，发生全部停止故障等级的控制器异常时，将无法强制。

### ● 动作模式变更时或电源断开时的强制值刷新状态

从运行模式变更为程序模式时或电源断开时，将解除强制值刷新状态。

## 强制值刷新的保持 / 解除

与控制器的状态变化对应的强制值刷新的保持 / 解除如下。

状态变化		强制值刷新状态
电源接通时		解除
动作模式变更时	运行模式→程序模式 程序模式→运行模式	解除
下载后		解除
发生全部停止故障等级的控制器异常时		解除
在线编辑时		保持

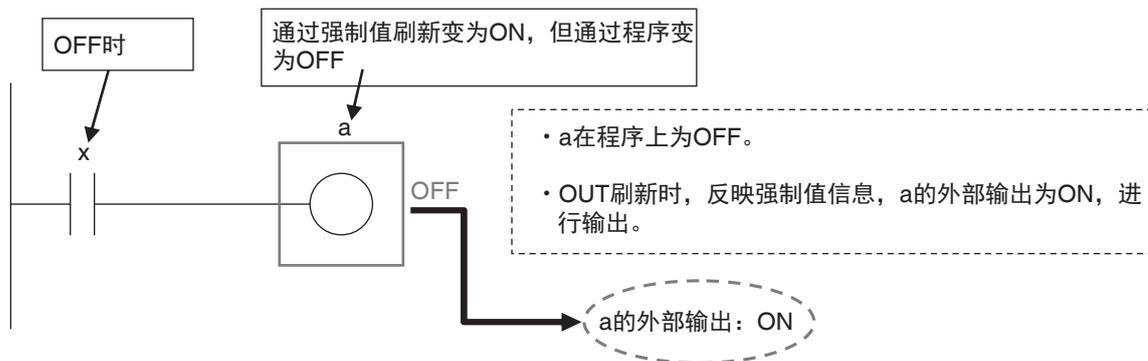
## 在程序中使用强制值时的注意事项

在用户程序上执行强制值刷新时，设定为强制值刷新对象的变量将被用户程序覆盖。因此，不保持强制值刷新中指定的状态。

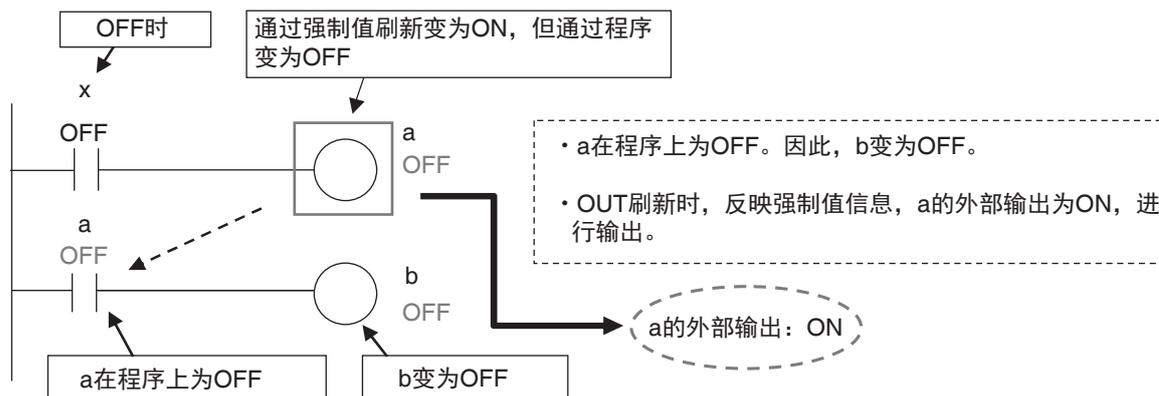
但是，对外部的刷新与用户程序上的变量状态无关，继续以强制值刷新中指定的值输出。

在程序上使用强制值时，程序上的变量和外部输出的状态有差异，敬请注意。

例) 通过强制值刷新将 a 设为 ON 时



然后，接收该接点时：



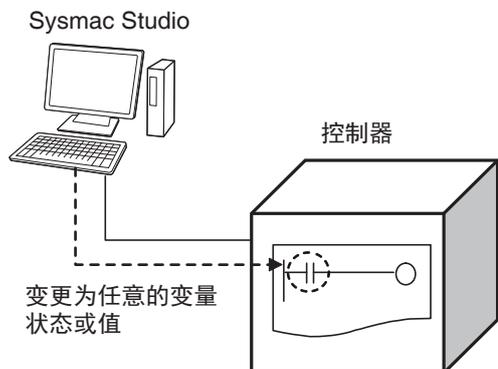
### 使用注意事项

- 进行强制值刷新操作前，请确认其不会对设备造成不良影响。
- 强制值刷新是指与用户程序执行结果无关，始终以设定的值覆盖并执行 I/O 刷新的功能。因此，I/O 刷新对象以外的接点会因本功能的影响先变为设定的值，再根据用户程序替换。因强制状态不同，控制对象可能发生意外动作。

## 8-4-2 当前值变更

### 概要

当前值变更是指为了确认程序或设定是否正确动作，可通过 Sysmac Studio 变更程序或设定中使用的变量值或控制接点、线圈的 ON/OFF（TRUE/FALSE）。



### 使用注意事项

请在确认不会对设备造成影响后，再变更当前值。

### 用途

#### ● 任意变更接点或线圈的 ON/OFF（TRUE/FALSE）

控制任意 BOOL 型变量状态的 ON/OFF。该状态在由用户程序覆盖后，变为执行状态，通过模式变更和电源重启，恢复为初始值状态。

可通过梯形编辑器、监视窗口、I/O 映射进行。

#### ● 变更为任意的变量值

将用户定义变量、系统变量、设备变量的当前值变更为任意值并反映。可通过监视窗口进行。



### 使用注意事项

请在确认不会对设备造成影响后，再变更当前值。

### 操作方法

当前值变更可在以下画面中操作。各画面的操作方法详情请参考□□《Sysmac Studio Version 1 操作手册（SBCA-CN5-362）》。

- 程序画面（梯形图语言、ST 语言）
- I/O 映射
- 监视窗口

### 通过当前值变更，变更向外部的输出状态时的注意事项

通过当前值变更，变更输出到 EtherCAT 输出从站的 I/O 端口的状态时，请注意以下事项。

#### ● 运动模式时，通过 I/O 映射变更当前值

即使通过 I/O 映射变更 I/O 端口的值，也会被用户程序的执行结果覆盖。

因此，不会将当前值变更中变更的值输出到外部。

要将 I/O 端口的值变更为任意值，然后输出到外部时，请通过强制值刷新执行。

#### ● 程序模式时，通过监视窗口变更当前值

即使通过监视窗口变更设备变量（\*）（外部变量、本地变量）的值，也不会输出到外部。

要将任意值输出到外部时，请进行以下操作。

- 进行强制值刷新
- 通过监视窗口，对设备变量（\*）（本地变量）进行当前值变更
  - \* EtherCAT 输出从站的 I/O 端口上的设备变量。

### 8-4-3 在线编辑

下面介绍在线编辑的概要。详情请参考□□《Sysmac Studio Version 1 操作手册（SBCA-CN5-362）》。

#### 概要

在线编辑是指通过 Sysmac Studio，直接变更和追加控制器中部分用户程序的功能。

可选择以下内容，进行在线编辑。

- POU（程序、FB、FUN）  
如果是以梯形图语言记述的程序，则选择段落。
- 全局变量

#### 用途

想要不停止控制器的运行，直接变更用户程序时使用。

#### Sysmac Studio 上的操作

##### ● 在线编辑的操作

- 1 选择在线编辑对象。
- 2 从 [项目] 菜单中选择 [在线编辑] - [开始]。
- 3 进行变更。
- 4 从 [项目] 菜单中选择 [在线编辑] - [传送变更]。
- 5 进行反映确认。
- 6 以在线编辑后的用户程序开始动作。

#### 注意

请确认输入输出时间打乱后也不会受到影响，再执行在线编辑。执行在线编辑后，任务执行时间可能超过任务周期，与外部之间的输入输出无法更新，导致输入信号无法读取、输出时间紊乱。





### 使用注意事项

---

- 在线编辑中编辑的回路所含的微分指令的微分状态将初始化。
  - 反映在线编辑内容时，任务的执行时间会延长。设定任务周期时，请确保不会因在线编辑而发生“任务周期超限”。
  - 正在反映在线编辑(\*)时，如果控制器发生断电，断电恢复时，将发生全部停止故障等级的控制器异常（“用户程序 / 控制器配置和设定时传送错误”、“用户程序 / 控制器配置和设定不正确”、“有非易失性存储器修复”中的一个）。此时，请重新下载用户程序。
  - 正在反映在线编辑(\*)时，请勿执行 `MC_SaveCamTable`（凸轮表保存）指令。若执行，可能无法正确反映在线编辑的变更。
- \* 是指在反映确认窗口中选择 [ 是 ] 后，关闭“反映确认窗口”之后显示的“内置非易失性存储器备份中窗口”之前的时间。
-

### 8-4-4 数据追踪

无需程序，对指定变量进行采样的功能。可通过 Sysmac Studio 读取，然后确认并保存文件。可用于装置启动、运用 / 维护。

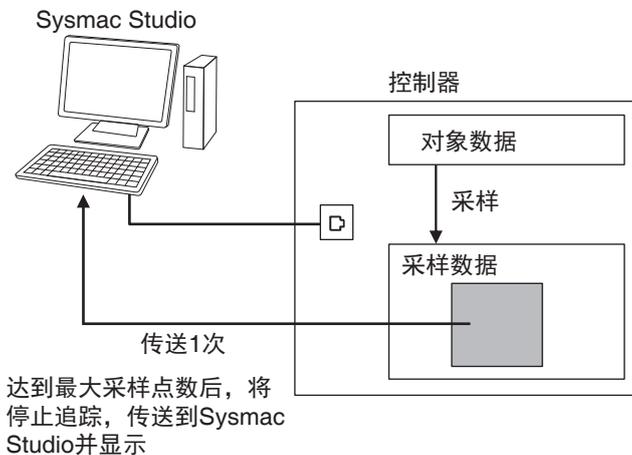
下面介绍数据追踪的概要。操作方法请参考 □□《Sysmac Studio Version 1 操作手册 (SBCA-CN5-362)》。

追踪方法分为以下 2 种。

#### ● 触发跟踪

通过设定触发条件，记录问题发生前后的数据。达到最大采样点数后，自动停止采样。未与 Sysmac Studio 连接时，也可在触发条件成立时进行数据追踪，与 Sysmac Studio 连接后再上传。

- 可根据变量的当前值变更情况确认程序的流程。
- 用于调查变量发生意外变更的原因。

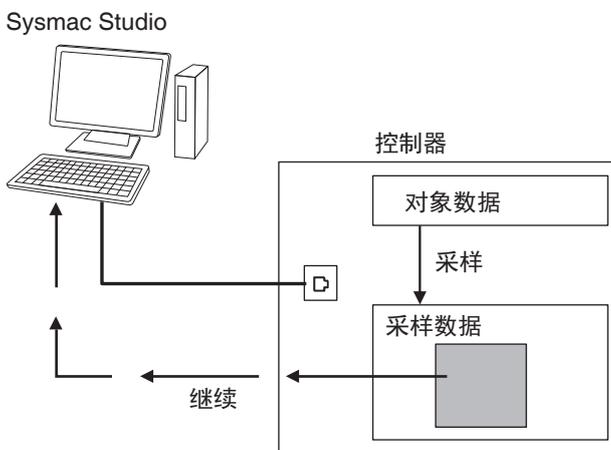


#### ● 连续跟踪

在没有触发的情况下开始采样，超出 10,000 点后继续采样。采样数据将依次传送到计算机上，然后保存到文件中。

达到显示缓存大小后，将自动保存为 CSV 文件。

因此，从开始追踪到停止为止，可将长期的追踪结果数据保存为多个 CSV 文件。



## 数据追踪的规格

数据追踪的规格如下所示。

项目		内容
数据追踪的种类	触发跟踪	设定开始采样的触发条件，记录条件成立前后的数据。
	连续跟踪	采样数据将依次传送到计算机上，然后保存到文件中。
采样时间	指定任务的周期	指定任务，将该任务的周期作为采样周期。
	固定时间间隔	将输入的时间作为采样周期。但是，时间将取整至主周期任务的整数倍。
	追踪采样指令	在用户程序中执行 <b>TraceSamp</b> 指令时，执行采样。
采样对象设定*1	最大对象数	192 变量
	数据类型	基本数据类型（字符串除外） 排列（指定要素） 枚举型 结构体 / 联合体型（指定成员）
记录最大数		每个变量 10,000 点
触发位置设定		设定所有追踪时间（数量）中的触发位置。
触发设定	条件的数据类型	基本数据类型（时刻、时间长度、日期、字符串除外） 排列（指定要素）、结构体 / 联合体（指定成员）
	条件表达式*2	当以下条件表达式成立时开始追踪。 BOOL: 上升沿、下降沿 BOOL 以外: =、>、≥、<、≤、≠*3
	来自 Sysmac Studio 的指示	单击触发 ON 按钮后，开始追踪。
	数据追踪触发条件成立指令	执行 <b>TraceTrig</b> 指令后，开始追踪。
	判定时序	<ul style="list-style-type: none"> <li>采样时间设定为追踪采样指令以外时，在指定的任务周期内执行 1 次触发判定。</li> <li>采样时间设定为追踪采样指令时，在执行指令时执行触发判定。</li> </ul>
	延时值	用滑块设定触发成立前 / 成立后的采样数比例。（例：20%/80%）
追踪开始	来自 Sysmac Studio 的指示	按下 Sysmac Studio 的“执行按钮”后开始。
	运行时开始设定	开始运行（程序模式→运行模式）时自动开始追踪。
追踪停止	触发跟踪	<ul style="list-style-type: none"> <li>达到采样上限值 10,000 点后停止。</li> <li>按下 Sysmac Studio 的“停止按钮”后停止。</li> </ul>
	连续跟踪	<ul style="list-style-type: none"> <li>达到上限时的设定为“追踪停止”时，达到上限大小或上限期间后，则停止。</li> <li>按下 Sysmac Studio 的“停止按钮”后停止。</li> </ul>

项目	内容	
连续跟踪专用设定	数据保存期限上限	设定连续跟踪数据的保存期限上限。
	数据保存大小上限	设定连续跟踪中保存的文件总大小上限。
	每个文件的数据数	设定在连续跟踪时，一个文件中保存的数据数。
	文件保存位置	指定在连续跟踪时，创建数据保存文件的位置。
	文件名前缀	指定自动命名时文件名开头的名称。
	达到上限时的设定	指定在达到保存期限上限或保存大小上限时的动作。 <ul style="list-style-type: none"> <li>追踪停止</li> <li>删除旧文件并继续</li> </ul>
追踪结果显示	图表显示	显示的图表为 X 轴：时间、Y 轴：变量值。可在一个图表中显示 <b>BOOL</b> 型和其他变量。
	表显示	以表格的形式显示各变量的最大值、最小值、平均值及指定时间的值。
	3D 运动监视显示	在 3D 空间内设置虚拟结构模型，根据运动轴的指令位置及当前位置虚拟显示结构的动作。
追踪数据导出	CSV 文件导出	将追踪结果及追踪编号以外的设定保存为 <b>CSV</b> 格式文件。
可同时启动的数据追踪 <sup>*4</sup>		4 个
追踪数据导入		在当前显示的图标中，重叠显示保存为 <b>CSV</b> 格式的追踪结果。
保存		将追踪结果与追踪设定一起，保存到项目中。
打印		打印图表。使用 <b>Sysmac Studio</b> 的打印功能。

\*1. 系统定义变量 EN、ENO、P\_off、P\_on、P\_CY、P\_First\_RunMode、P\_First\_Run、P\_PRGER 及功能块的实例中使用的输入输出变量、函数中的变量不在采样对象范围内。

\*2. 开始数据追踪时，即使触发的条件表达式为成立状态，也不会开始数据追踪。

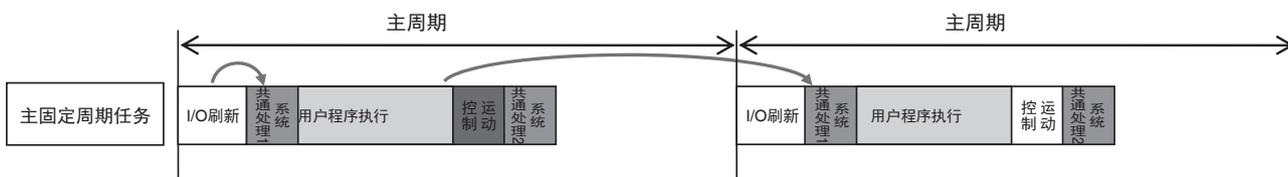
\*3. 不可组合多个条件表达式。此外，比较常数的有效范围遵照条件式左边的变量类型文字表现的有效范围。

\*4. NY 系列中，设定了 0 ~ 3 的追踪编号。使用该编号，执行指令或访问系统定义变量。

## 数据追踪的动作

数据追踪处理（采样及触发判定）在 I/O 刷新处理后、执行用户程序前的“系统共通处理 1”中执行。

例）采样时间指定为“主固定周期任务”时，按下图“系统共通处理 1”的时序执行数据追踪。





## 参考

任务处理中，“I/O 刷新、用户程序、运动控制”为同一任务周期，但在数据追踪的显示中，“用户程序、运动控制、下一主周期的 I/O 刷新”作为同一任务周期显示。☞《NY 系列 指令基准手册 运动篇（SBCE-CN5-380）》中记载的时序图以任务周期为基准记载，因此与数据追踪的显示不同。

以下为按指定任务周期采样时，数据追踪的动作及执行结果的显示示例。

例 1)

创建了以下程序时：传感器的输入信号中分配的变量“Sensor1”变为 TRUE 后，将变量“SysRun”变为 TRUE 的程序



数据追踪的动作及执行结果的显示如下所示。

- (1) 在系统共通处理 1 的数据追踪处理中获取“Sensor1”的 TRUE
- (2) 在用户程序中，将“SysRun”设为 TRUE
- (3) 在下一主周期的系统共通处理 1 的数据追踪处理中获取“SysRun”的 TRUE

因此，在数据追踪的显示中，“SysRun”会比“Sensor1”晚一个任务周期变为 TRUE。

数据追踪的显示



## 参考

在用户程序中变更了变量的值时，在数据追踪的显示中，值的变化和 I/O 刷新的输出处理在同一任务周期内变化。

例 2)

创建了以下程序时：以速度控制动作时，按钮的输入信号中分配的变量“Button2”变为 TRUE 后，使轴 0 (“MC\_Axis000”) 减速停止的程序

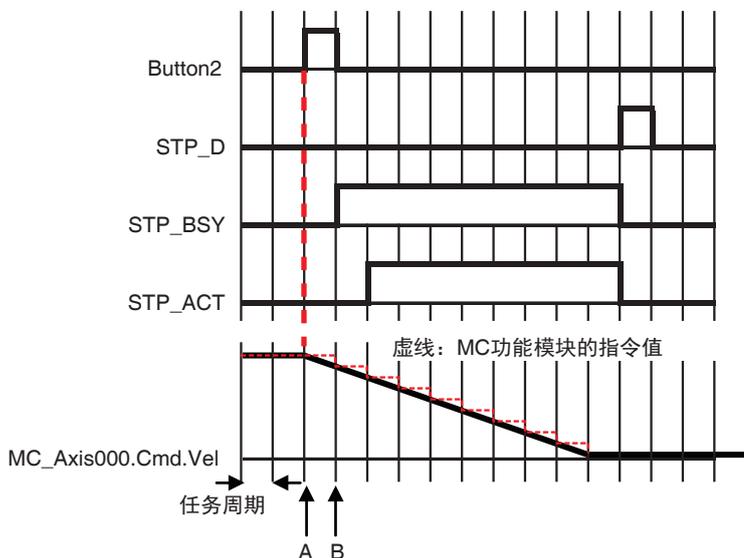


数据追踪的动作及执行结果的显示如下所示。

- (1) 在系统共通处理 1 的数据追踪处理中获取“Button2”的 TRUE
- (2) 在用户程序中将“STP\_BSY”设为 TRUE，运动控制功能模块进行减速处理
- (3) 在下一主周期的系统共通处理 1 的数据追踪处理中获取 STP\_BSY 的 TRUE 和运动变量
- (4) 在用户程序中，将“STP\_ACT”设为 TRUE
- (5) 在下一主周期的系统共通处理 1 的数据追踪处理中获取 STP\_ACT 的 TRUE

运动控制功能模块的指令值在用户程序中的“STP\_BSY”变为 TRUE，运动控制功能模块开始减速处理时，开始变化（下图 B 的时间点）。指令值与主固定周期任务同步进行阶段性变化。

数据追踪将对前周期的值和下一周期的值进行直线插补。因此，运动变量的指令速度“MC\_Axis000.Cmd.Vel”显示为在指令值发生变化前的一个周期，即从“Button2”变为 TRUE 开始（下图 A 的时间点）进行减速。开始减速后一个周期，“STP\_BSY”变为 TRUE，再一个周期后，“STP\_ACT”变为 TRUE。



#### 参考

在包含运动控制指令的功能块中，开始执行功能块时，输入参数的值将设置到输入变量中，功能块执行结束时，输出变量的值设置到输出参数中（参考□「功能块（FB）内变量的规格」（P.6-12））。数据追踪的显示与输入参数和输入变量、输出参数和输出变量在同一任务周期内变化。

## 相关系统定义变量

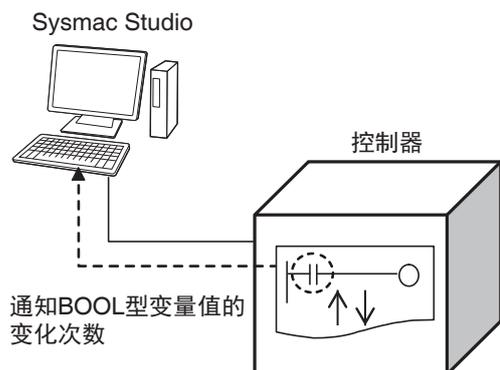
变量名称	成员名称	名称	功能	数据类型	R/W
_PLC_TraceSta[0..3] <sup>*1</sup>				_sTRACE_STA	R
	.IsStart	追踪执行中 ON 标志	开始追踪时，变为 TRUE。	BOOL	R
	.IsComplete	追踪完成时 ON 标志	追踪执行完成时，变为 TRUE。 下次开始追踪时，变为 FALSE。	BOOL	R
	.IsTrigger	追踪触发监视标志	追踪触发条件成立时，变为 TRUE。 下次开始追踪时，变为 FALSE。	BOOL	R
	.ParamErr	追踪参数异常标志	开始追踪时，若追踪设定异常，则变为 TRUE。 正常时，为 FALSE。	BOOL	R

\*1. 对应数据追踪 0 ~ 3。

(注) 本系统定义变量在用户程序中不可使用。通过用户程序参照数据追踪的状态时，请使用 GetTraceStatus 指令。

### 8-4-5 微分监控

微分监控是指对指定 BOOL 型变量的值与指定条件一致的次数进行通知的功能。按主固定周期任务的任务周期（称为主周期）执行条件一致判定，并通知目前为止的条件一致次数。



### 用途

可确认来自外部的输入信号或用户网络中接点的 ON/OFF 有无和次数。在设备启动时以及调查运行过程中发生动作不良的原因时非常有用。

## 微分监控的规格

微分监控的规格如下所示。

项目		规格
微分监控条件	设定数量	最大 8
	指定变量	<ul style="list-style-type: none"> <li>• BOOL 型</li> <li>• BOOL 型排列的 1 个要素</li> <li>• 结构体、联合体的 1 个 BOOL 型成员</li> </ul>
	条件表达式	<ul style="list-style-type: none"> <li>• 上升沿 (FALSE → TRUE)</li> <li>• 下降沿 (TRUE → FALSE)</li> </ul>
条件一致判定	时序	主周期内一次
	次数计算	对各个指定变量分别计算条件一致次数
开始和停止	开始条件	来自 Sysmac Studio 的指示
	停止条件	<ul style="list-style-type: none"> <li>• 来自 Sysmac Studio 的指示</li> <li>• 发生全部停止故障等级的控制器异常</li> <li>• 下载用户程序</li> <li>• 存储器全部清除</li> <li>• 解除 Sysmac Studio 的在线连接</li> </ul>
动作模式		<ul style="list-style-type: none"> <li>• 运行模式</li> <li>• 程序模式</li> </ul>

## 微分监控条件

微分监控的设定对象变量或变化种类称为微分监控条件。下面介绍它们的规格。

### ● 设定数量

最大 8。即，可同时检测 8 个变量值的条件一致次数。

### ● 指定变量

可指定为微分监控对象的变量数据类型如下所示。

- BOOL 型
- BOOL 型排列的 1 个要素
- 结构体、联合体的 1 个 BOOL 型成员

不可指定为整个排列、整个结构体、整个联合体。

此外，作为变量的种类，可指定以下内容。

变量的种类		可否指定
系统定义变量		支持 <sup>*1</sup>
准用户定义变量		支持
用户定义变量	全局变量	支持
	程序中使用的变量	支持
	功能块中使用的变量	支持 <sup>*2</sup>
	函数中使用的变量	不支持

\*1. 以下变量除外。

EN, ENO, P\_Off, P\_CY, P\_First\_RunMode, P\_First\_Run, P\_PRGER

\*2. 输入输出变量除外。

### ● 条件表达式

所检测变量的变化条件称为条件表达式。条件表达式有以下 2 种，选择其中之一。条件表达式可按变量指定。

- 上升沿 (FALSE → TRUE)
- 下降沿 (TRUE → FALSE)



#### 使用注意事项

例如，条件表达式设定为“上升沿”时，开始微分监控功能时，即使指定变量的值为 TRUE，也不会判定为上升沿。变量的值先变为 FALSE，再变为 TRUE 时，判定为上升沿。

## 条件一致判定时间和次数计算

指定变量的条件一致判定按主周期执行。对上次判定时的变量值和本次判定时的变量值进行比较。所以，如果与指定条件表达式一致的变量值发生变化，将计算其次数。

- 条件一致计算次数按变量执行。
- 计数值会在开始微分监控时重置为 0。
- 不可单独将变量的计数值重置为 0。



#### 使用注意事项

- 在主周期内，即使与条件表达式一致的值发生多次变化，一个主周期将只计算为一次。
- 如果上次判定时和本次判定时变量的值相同，即使在此期间值发生过变化，也不会视为与条件表达式一致。

## 开始条件和停止条件

要开始微分监控，需要通过 Sysmac Studio 发出指示。

要停止微分监控，通常由 Sysmac Studio 发出指示。但是，以下情况下，也会停止微分监控。

- 发生全部停止故障等级的控制器异常
- 下载用户程序
- 存储器全部清除
- 解除 Sysmac Studio 的在线连接

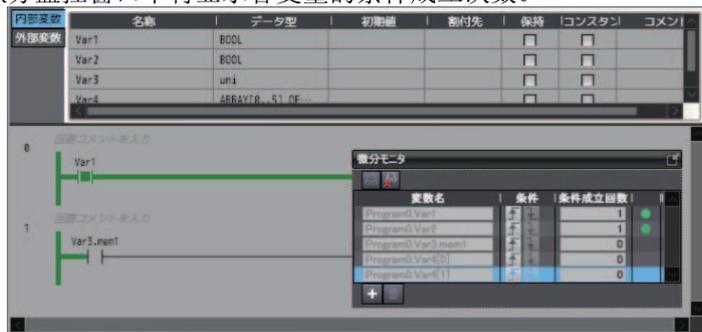
## 操作方法

微分监控的操作方法如下所示。详情请参考□《Sysmac Studio Version 1 操作手册（SBCA-CN5-362E 之后）》。

- 1 选择 Sysmac Studio 的 [ 显示 ]-[ 微分监控 ]。
- 2 选择可指定为微分监控对象的变量并右击，选择 [ 追加到微分监控 ]。
- 3 在微分监控窗口中，对各变量设定微分监控的条件表达式。



- 4 执行用户程序。  
微分监控窗口中将显示各变量的条件成立次数。



## 微分监控功能的注意事项

使用微分监控功能时，请注意以下事项。

- 执行微分监控的过程中与 Sysmac Studio 断开通信  
执行微分监控的过程中，由于通信电缆断线、Sysmac Studio 异常结束等，与 Sysmac Studio 的通信中断。此时，控制器将继续执行微分监控。重新执行微分监控时，请在重新开始与 Sysmac Studio 的通信后，先停止微分监控。
- 微分监控不可多个执行  
微分监控不可通过多 Sysmac Studio 同时执行。

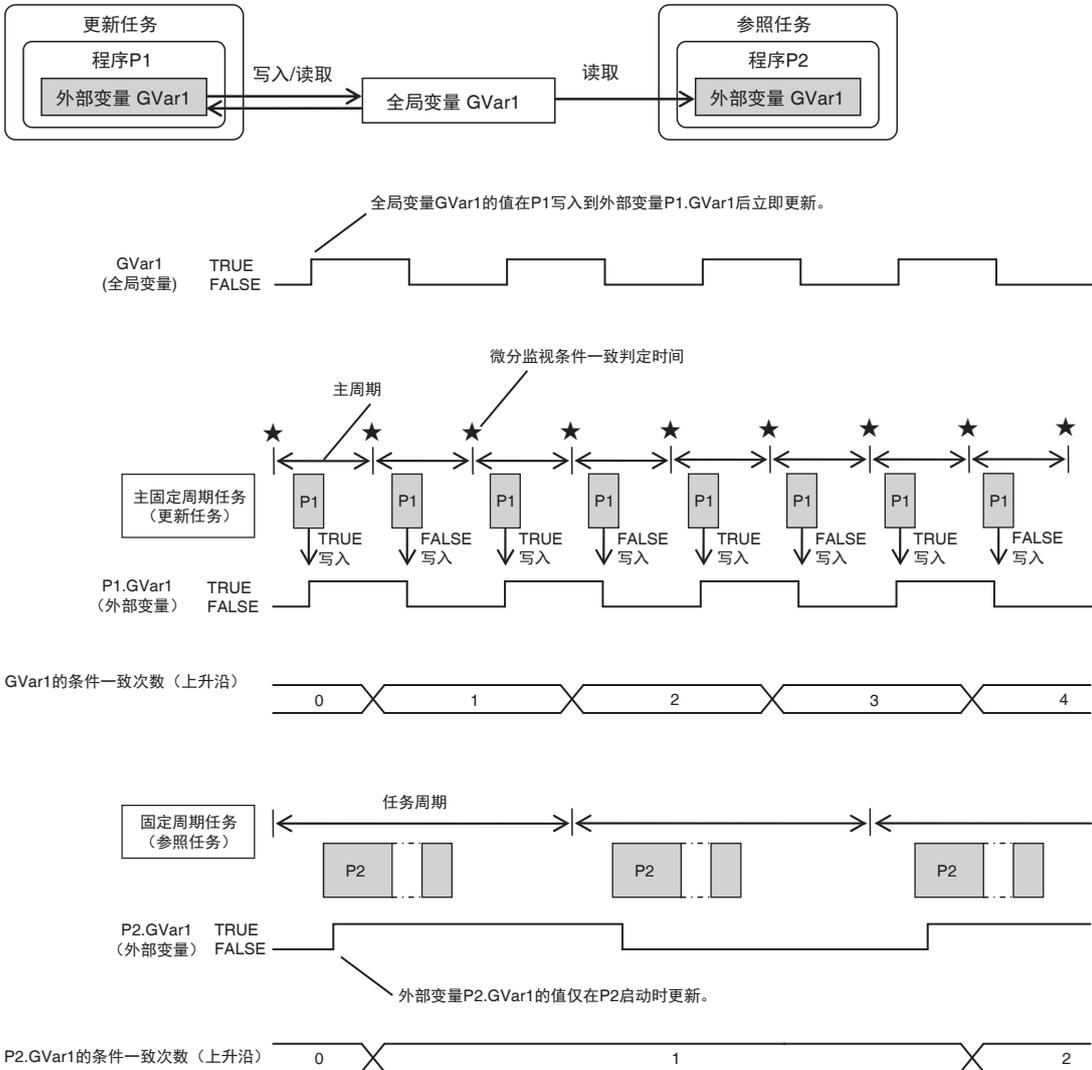
● 全局变量、外部变量的指定

全局变量以及在 POU 内对其进行指定的外部变量均可设为微分监控的指定变量。但是，全局变量和外部变量的值更新时间不同，敬请注意。

全局变量会在写入值后立即更新。但外部变量只会对该外部变量进行声明的 POU 启动时更新。说明如下图。假设监视以下两个变量。

变量名称	变量的种类	要读写的 POU
GVar1	全局变量	分配到主固定周期任务中的程序 P1
P2.GVar1	程序 P2, 指定为 GVar1 并进行声明的外部变量	分配到固定周期任务中的程序 P2

全局变量 GVar1 通过主固定周期任务中分配的程序 P1 读写，因此如果每个周期发生写入，将按主周期更新。而外部变量 P2.GVar1 只在分配到固定周期任务中的程序 P2 启动时更新。即只会按固定周期任务的任务周期更新。固定周期任务的任务周期长于主周期，因此 P2.GVar1 的更新次数比 GVar1 少。

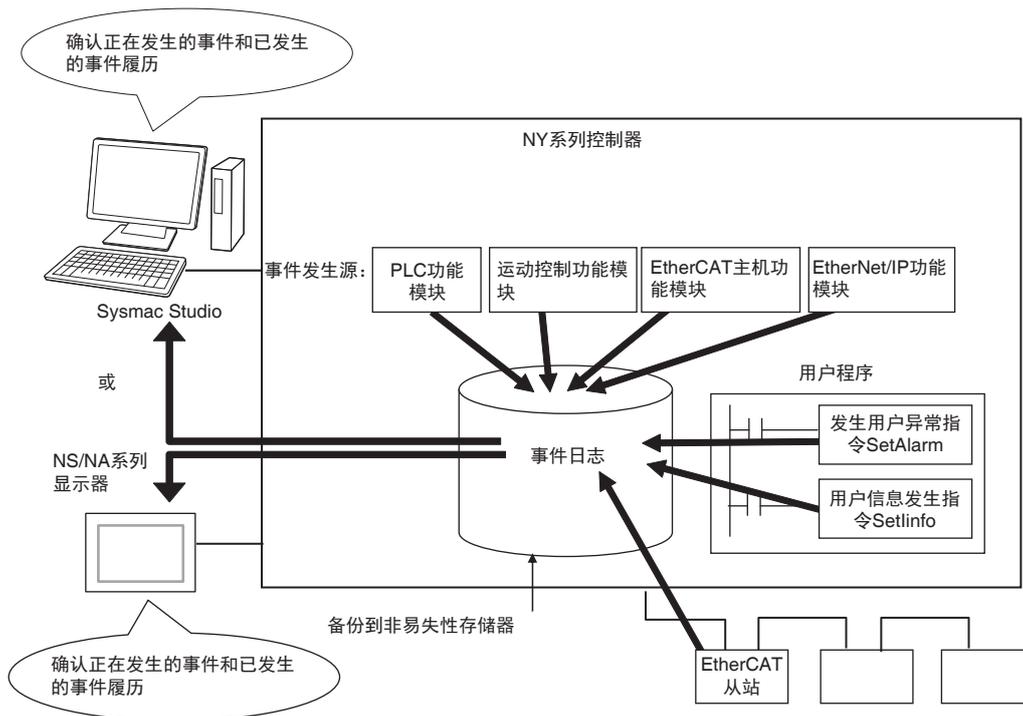


## 8-5 事件日志功能

下面介绍事件日志功能。

### 8-5-1 概要

事件日志功能是指对 NY 系列控制器中发生的异常、状态变化及用户定义现象等事件 (\*) 进行记录的功能。



\* “事件”是指控制器上意外发生的、包括异常在内的现象。异常是指根据控制器或用户定义，包括要向用户通知的非异常信息在内的总称。

包括以下两种四类。

- 控制器事件
  - 控制器异常
  - 控制器信息
- 用户事件
  - 用户异常
  - 用户信息

通过 NS/NA 系列触摸屏确认事件时，请通过控制器的内置 EtherNet/IP 端口连接 NS/NA 系列触摸屏。



#### 使用注意事项

关于触摸屏的故障排除应对范围，请参考□□《NY 系列 故障排除手册（SBCA-CN5-438）》。

## 特点

事件日志功能有以下特点。

- 除了异常记录外，还会记录电源接通 / 电源断开或开始运行等各种记录。
- 这些记录可按时间顺序确认，因此在发生故障时，有助于推断异常原因。

## 可记录内容

事件种类如下所示。

### ● 系统定义的事件（控制器事件）

控制器自动检测的事件。NY 系列控制器的各功能模块、EtherCAT 从站的事件。包括以下事件。

- 控制器异常
- 控制器信息

### ● 用户定义的事件（用户事件）

用户设计的应用程序事件。通过指令执行，可发生以下事件。

- 用户异常
- 用户信息

可通过 Sysmac Studio 或触摸屏读取事件日志。

## 8-5-2 事件日志功能的详情

### 事件发生源

用于识别在控制器的哪个位置发生了事件。

控制器事件和用户事件各自的发生位置如下。

#### ● 控制器事件发生位置

控制器事件中，发生位置为存在于控制器中的功能模块。

可进一步通过功能模块获得更加详细的事件发生源。将其称为“事件发生源详情”。

包括以下内容。

事件发生源	事件发生源详情
PLC 功能模块	指令、Windows
运动控制功能模块	通用、轴、轴组
EtherCAT 主机功能模块	通信端口、EtherCAT 主站、EtherCAT 耦合器单元、NX 单元、EtherCAT 从站
EtherNet/IP 功能模块	通信端口 / 通信端口 1 / 内部通信端口 1、CIP、FTP、NTP、SNMP

#### ● 用户事件发生位置

用户事件中，事件发生位置为 PLC 功能模块。

## 类别

表示事件日志的分类信息。

通过 Sysmac Studio 或触摸屏参照时的单位。

事件的种类	事件日志的类别	内容
控制器事件	系统日志	控制器自身检测到的事件记录。
	访问日志	因用户操作，对控制器的动作产生影响的事件记录。
用户事件	用户事件日志	用户定义的事件记录。

## 记录条数

各种事件日志中可记录的条数分别如下。

控制器发生超出以下可记录条数的事件时，最早的信息将被覆盖。

事件的种类	事件日志的类别	每 1 类别的可记录条数
控制器事件	系统日志	2,048
	访问日志	
用户事件	用户事件日志	

## 断电保持

NY 系列控制器即使发生断电，也可通过非易失性存储器来保持事件日志。



### 使用注意事项

事件日志由非易失性存储器保持。工业电脑上未连接 UPS 时或未正常关机时，将无法保持，敬请注意。

必要时，请定期导出事件日志。

## 事件代码

根据事件种类，由系统事先分配到控制器事件的代码或由用户分配到用户事件的代码。控制器事件的事件代码以 16 进制 8 位表示。

发生中的异常其事件代码可通过异常状态获取指令获取。

用户事件中可以标注 1 ~ 60000 的 10 进制数事件代码。

## 事件重要程度

各事件有表示重要程度的“事件重要程度”。

事件重要程度的分类因事件种类不同而异。控制器事件和用户事件分别独立定义。

### ● 控制器事件时

根据对控制的影响大小，控制器事件的重要程度分为以下 5 个阶段。

编号	重要程度	分类	分类
1	高	控制器异常	全部停止故障等级
2			部分停止故障等级
3			轻度故障等级
4			监视信息等级
5	低	控制器信息	一般信息等级

异常的重要程度越高，控制器提供的功能会受损，修复的紧急程度也越高。

该分配会在发生异常时显示在 **System Studio** 或触摸屏中。

### ● 用户事件时

用户事件可按 NY 系列系统提供的以下重要程度分类。

对用户事件的重要程度进行定义。

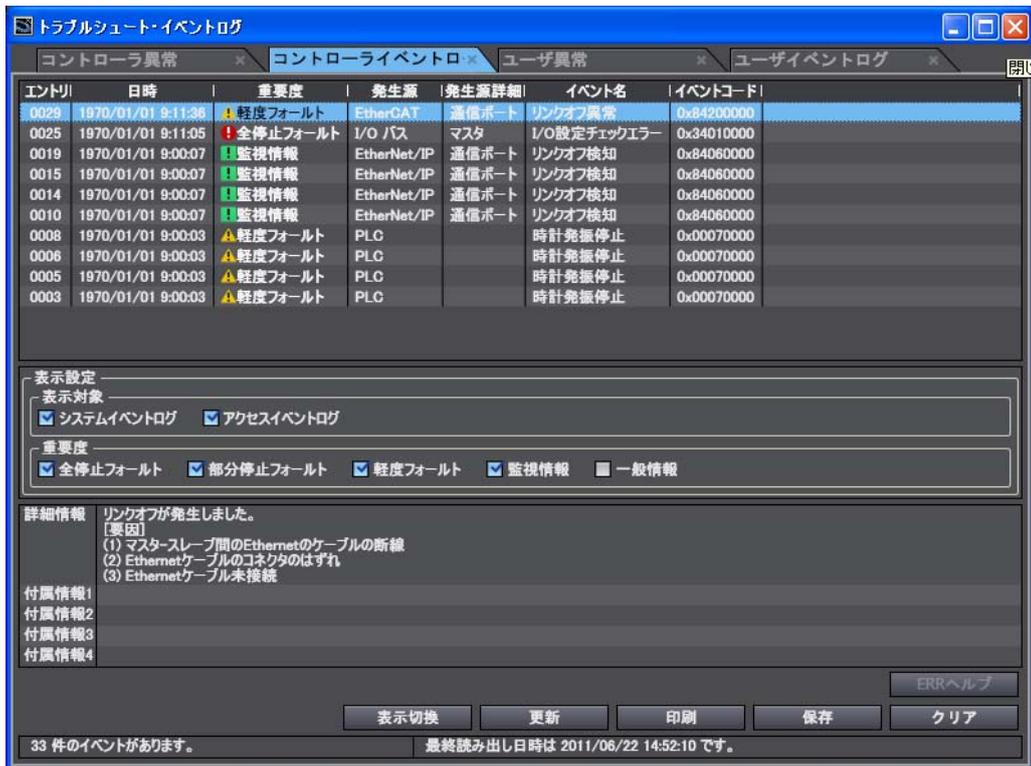
编号	重要程度	分类	含义
1		User fault Level1	该事件表示应用上用户定义的异常。 用户执行 <b>SetAlarm</b> （发生用户异常）指令后发生事件。
2		User fault Level2	
3		User fault Level3	
4		User fault Level4	
5		User fault Level5	
6		User fault Level6	
7		User fault Level7	
8		User fault Level8	
9	低	User Information	该事件表示应用上用户定义的信息。 用户执行 <b>SetInfo</b> （发生用户信息）指令后发生事件。

## 事件日志的显示

Sysmac Studio 或触摸屏根据“控制器日志”“用户日志”的类别显示事件日志一览。“控制器日志”中显示“访问日志”和“系统日志”。

此外，Sysmac Studio 还可显示 EtherCAT 从站中记录的异常记录。

Sysmac Studio 中将如下进行一览显示。从一览中选择事件后，将显示对应的详细信息。



### 参考

控制器上发生 Sysmac Studio 或触摸屏的版本不支持的事件时，发生位置显示为“不详”，事件名称显示为“不详事件”。事件代码和附属信息直接显示。

### 事件日志的删除

---

- **通过 Sysmac Studio/ 触摸屏删除**

通过 Sysmac Studio 或触摸屏，可按事件日志的控制器事件和用户事件分别删除事件日志。



#### 使用注意事项

---

- 通过 Sysmac Studio / 触摸屏删除控制器中的事件日志时，请确认之前获取的所有事件信息均可删除后，再执行。特别是一般信息等级 / 监视信息等级的控制器事件或用户事件，可能尚未确认，敬请注意。
  - 通过触摸屏删除事件日志时的限制事项请参考  《NY 系列 故障排除手册 (SBCA-CN5-438)》。
- 

- **通过存储器全部清除来删除**

通过 Sysmac Studio 对 NY 系列控制器进行存储器全部清除操作后，可选择是否删除事件日志。

### 事件日志的导出

---

可使用 Sysmac Studio 或触摸屏将显示的事件日志内容导出为 CSV 文件。

### 8-5-3 控制器事件（控制器异常/信息）

#### 概要

NY 系列系统中定义的异常/信息。

NY 系列系统检测到异常原因、信息原因后发生。

#### ● 控制器异常

系统定义的异常。

“全部停止故障等级”、“部分停止故障等级”、“轻度故障等级”、“监视信息等级”的控制器事件的总称。

检测NY系列控制器的各功能模块、EtherCAT从站的各异常。此时，控制器异常将记录到事件日志中。

在用户程序上确认发生的控制器异常状况时，通过异常状态获取指令来参照系统定义变量的“异常状态变量”。

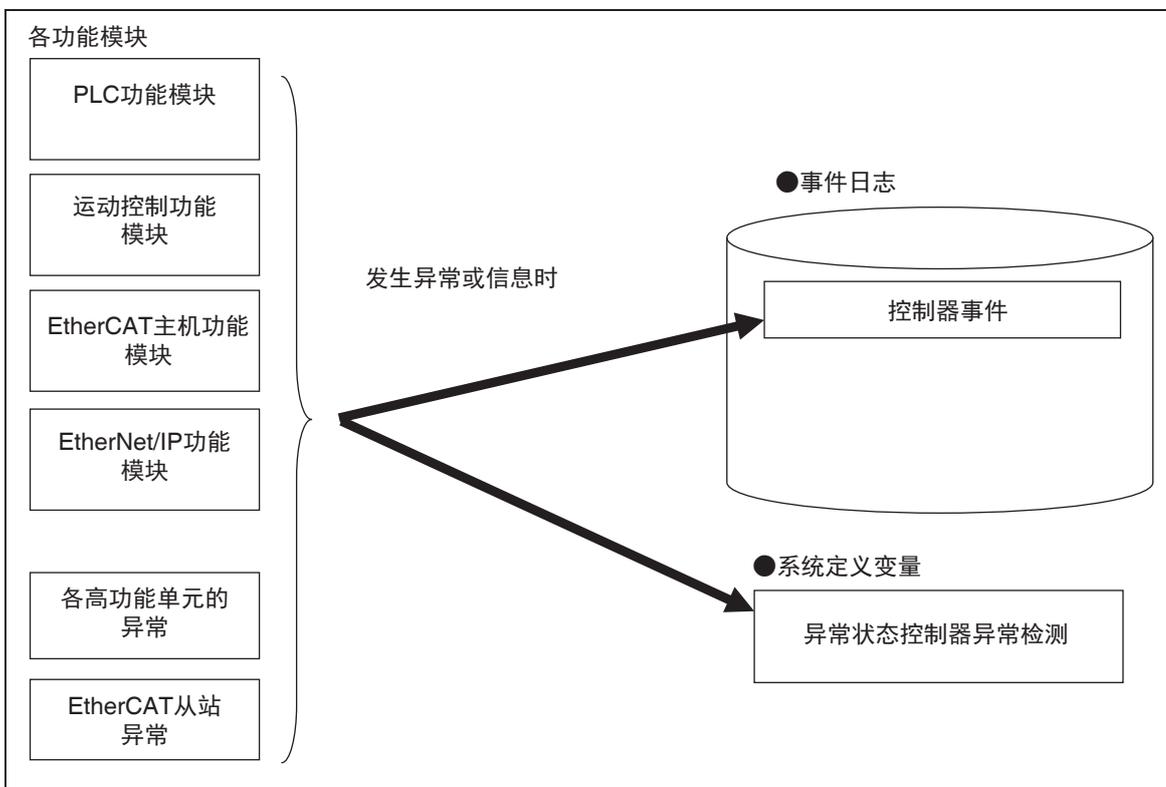
控制器异常不会因动作模式的变更而解除。

控制器异常的详情请参考□《NY 系列 故障排除手册（SBCA-CN5-438）》。

#### ● 控制器信息

系统定义的通知信息，并非异常。“一般信息等级”的控制器事件。

例如，电源 ON/OFF、运行开始/停止、Sysmac Studio 的在线连接、用户程序的下载等异常以外的现象。



## 8-5-4 用户事件（用户异常 / 信息）

## 概要

用户定义的异常 / 信息。可通过指令发生。

## ● 用户异常

用户定义的异常。

通过用户异常发生（SetAlarm）指令，发生用户异常。此时，用户异常将记录到事件日志中。

对应的系统定义变量变为 TRUE。

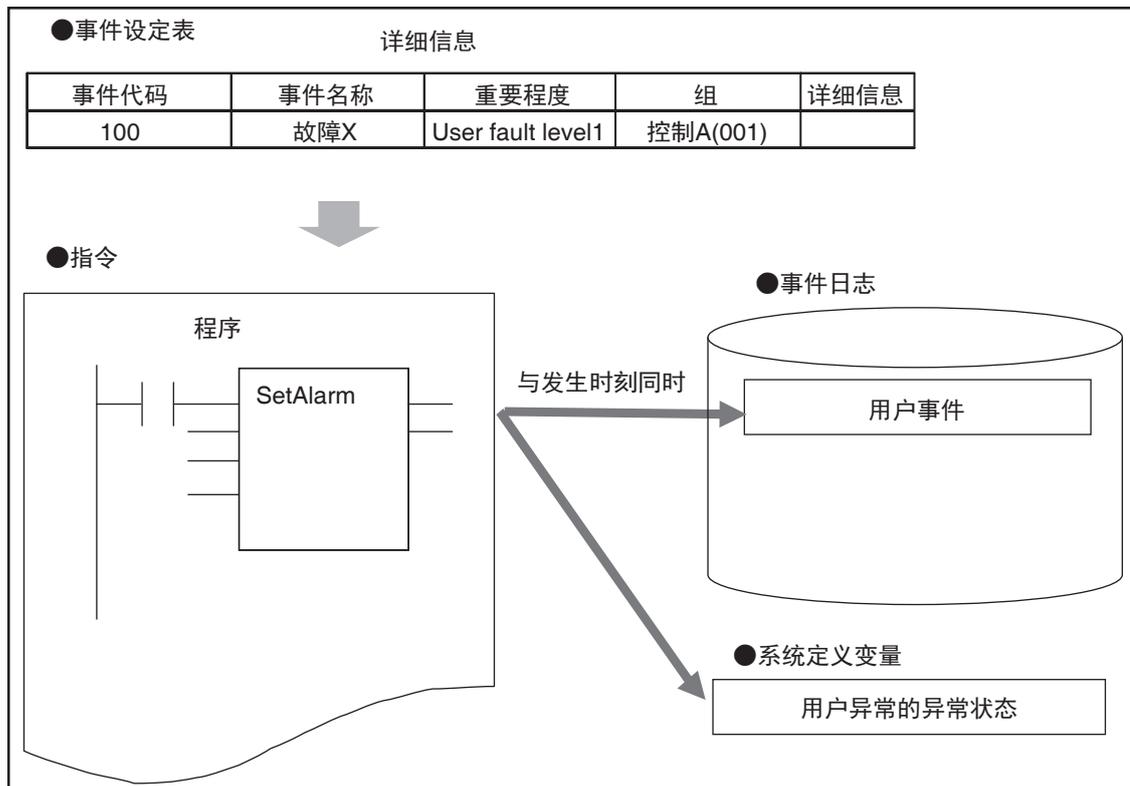
用户异常不会因动作模式的变更而解除。

## ● 用户信息

用户定义的通知信息，并非异常。

通过用户信息发生（SetInfo）指令，发生用户信息。

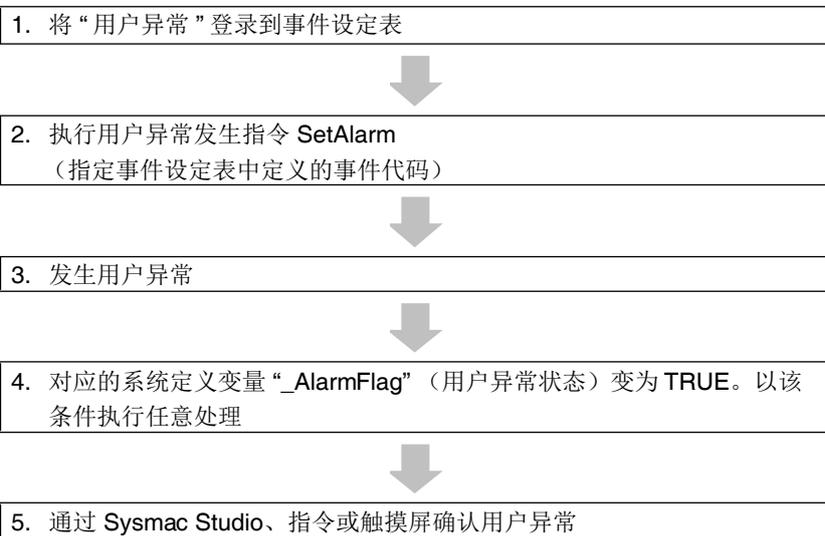
此时，用户信息将记录到事件日志中。



## 使用步骤

按以下步骤使用。

### ● 用户异常



### ● 用户信息



## 事件设定表的设定

要发生“用户异常”或“用户信息”，首先需要通过 Sysmac Studio，事先在“事件设定表”中登录“用户异常”或“用户信息”。

这里设定的用户事件可在 Sysmac Studio 或触摸屏中显示相同的内容。

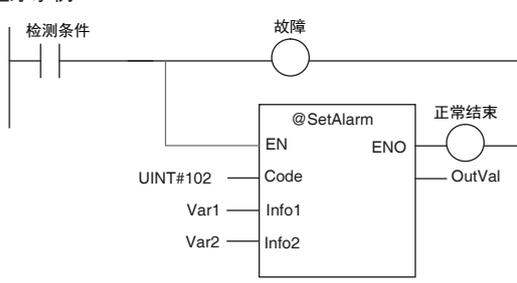
事件设定表中可登录的事件总数最多为 5120 条。

事件设定表的设定画面

事件设定表

事件代码	事件名称	重要程度	组	详细信息
10001	故障X	User fault Level3	控制A(001)	详细信息 ■内容 发生故障X。 ■处理 请确认安全，然后按照原因代码进行处理。
:	:	:	:	

程序示例



在事件设定表中设定以下内容。



## ● 事件设定表的内容

项目	内容	值
事件代码	可根据事件重要程度指定识别事件的编号。	用户异常：1 ~ 40000 用户信息：40001 ~ 60000
事件名称	可记载表示事件名称的标题。	最多 128 个字符
重要程度	指定事件的重要程度。 用数值表示重要程度，数值越小，重要程度越高。	用户异常：User fault Level1 ~ 8 用户信息：User Information
组	可指定表示事件发生位置和分类的组名。可根据用户定义对各事件进行分组。	最多 32 个字符 可使用的字符无限制。 区分大小写。 无保留字。
详细信息	可记载对事件内容进行说明的信息。 用户可自由填入任意字符串。在 Sysmac Studio / 触摸屏上显示事件时使用。	最多 1024 个字符 可使用的字符无限制。 区分大小写。 无保留字。
发生全部停止故障等级的控制器异常时，在触摸屏中显示的异常详细信息	请参考后述的“参考：发生全部停止故障等级的控制器异常时，在触摸屏上显示用户信息的功能：”。	最多 128 个字符 可使用的字符无限制。 区分大小写。 无保留字。
注释	可对多个表添加信息。	



### 参考

可对同一事件代码，向多个区域或多个用户设定最多 9 种语言（9 种）。事件设定表可利用 Sysmac Studio，通过 Microsoft Excel 文件的剪贴板导入。



### 参考

**发生全部停止故障等级的控制器异常时，在触摸屏上显示用户信息的功能：**

发生全部停止故障等级的控制器异常时，停止用户程序的执行。

NY 系列中，发生全部停止故障等级的控制器异常时，能在触摸屏上显示面向用户的信息。该信息在 Sysmac Studio 的事件设定表的用户事件一览下段中设定。

## ● 事件重要程度和事件代码

事件种类	重要程度	事件重要程度的分类名称 (*)	对应的事件代码的范围	说明
用户异常	高  低	User fault Level1	1 ~ 5000	用户从 8 个级别中选择。
		User fault Level2	5001 ~ 10000	
		User fault Level3	10001 ~ 15000	
		User fault Level4	15001 ~ 20000	
		User fault Level5	20001 ~ 25000	
		User fault Level6	25001 ~ 30000	
		User fault Level7	30001 ~ 35000	
		User fault Level8	35001 ~ 40000	
用户信息	最低	User Information	40001 ~ 60000	事件种类为“用户信息”的重要程度

\* 用户异常的重要程度与控制器异常的重要程度分类不同。



### 使用注意事项

更新事件设定表，并传送到控制器时，之前的用户事件的事件日志将以旧内容记录。由于事件设定表发生更新，因此可能发生不匹配。敬请注意。

## 相关指令

包括发生和解除用户异常的指令以及确认所发生用户异常的指令。

### ● 发生和解除用户异常

利用以下指令发生 / 解除用户异常、发生用户信息。

可同时变为发生中状态的用户异常数量为每个事件重要程度 32 件，总共最多 256 件。

指令名称	指令	功能
用户异常发生指令	SetAlarm	发生用户异常。
用户异常解除指令	ResetAlarm	解除用户异常。
用户信息发生指令	SetInfo	将任意用户信息记录到事件日志中。

### ● 确认用户异常

通过用户异常状态获取（GetAlarm）指令，可获取用户异常的发生有无、发生中的用户异常中最重要的事件重要程度及其事件代码。

例)

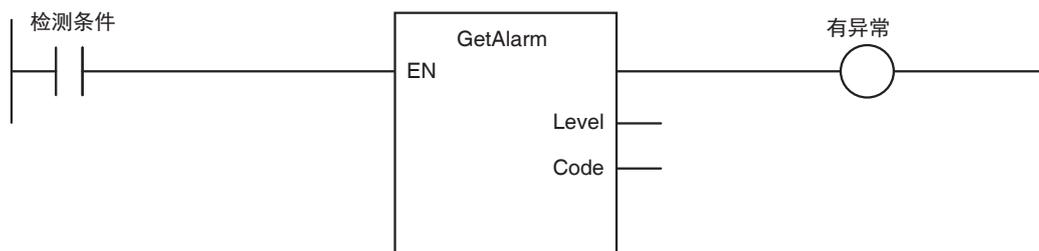
事件设定表

事件代码	事件名称	重要程度	组	详细信息
10001	故障X	User fault Level3	控制A(001)	
:	:	:	:	

详细信息

■内容  
发生故障X。  
■处理  
请确认安全，然后按照原因代码进行处理。

实际程序





## 参考

可根据用户异常，对发生的控制器异常，添加处理方法等信息。

关于发生中的控制器异常，通过 **GetPLCError** 等控制器异常的状态获取指令，获取异常状态和事件代码，并将该信息作为触发发生用户异常。

## 例 1

电池电压降低时，获取事件代码（16#000B0000），显示以下信息。

```

电池用完。
更换电池前，请通电 5 分钟以上，并在电源关闭后
5 分钟内更换新电池。

```

## 例 2

发生部分停止故障等级的控制器异常时，获取事件重要程度（最重要的状态：2），显示以下信息。

```

装置发生故障。请咨询经销商。
<维修联系方式>
受理时间 AM8: 00 ~ PM9: 00
TEL *** - **** - ****

```

## 用户异常相关的系统定义变量

变量名称	名称	功能	数据类型	R/W
_AlarmFlag	用户异常状态	发生用户异常时，与事件重要程度对应的位将变为 TRUE。 User fault Level1 ~ 8 对应位 00 ~ 07。	WORD	R

## 记录到事件日志中

发生用户异常或用户信息时，如果通过 **ResetAlarm** 解除，将记录到事件日志中。此时，事件日志的用户事件日志中将记录发生时刻、事件代码、事件重要程度、附属信息 1、2。

## 用户异常的解除

用户异常可通过 NY 系列控制器电源的 OFF → ON 来解除。

也可通过 **Sysmac Studio**、指令（用户异常解除指令 **ResetAlarm**）或触摸屏解除异常。

详情请参考 □《Sysmac Studio Version 1 操作手册（SBCA-CN5-362）》。

# 8-6 事件重要程度变更功能

NY 系列控制器中发生的异常、状态变化、用户定义现象等的总称，称为事件。用户可根据 Sysmac Studio 的显示、工业电脑的 LED 动作，知道发生了什么样的事件。

事件包括由系统事先定义的控制器事件和由用户任意定义的用户事件。各控制器事件分为 5 个事件重要程度。事件重要程度的详情请参考 □□「事件重要程度」(P.8-48)。

事件重要程度变更功能是指用户任意变更各控制器事件中分配的事件重要程度的功能。

## 8-6-1 事件重要程度变更的用途

根据发生的控制器事件的重要程度，事先规定工业电脑 LED 的动作。使用事件重要程度变更功能，可对发生的任意事件，变更控制器的动作。

例如，ERROR LED 在事件重要程度为轻度故障等级时闪烁，监视信息等级时保持熄灭。若使用事件重要程度变更功能，可对发生的任意事件，设定为 ERROR LED 熄灭或闪烁。

控制器的不同事件重要程度的动作请参考 □□《NY 系列 故障排除手册 (SBCA-CN5-438)》。

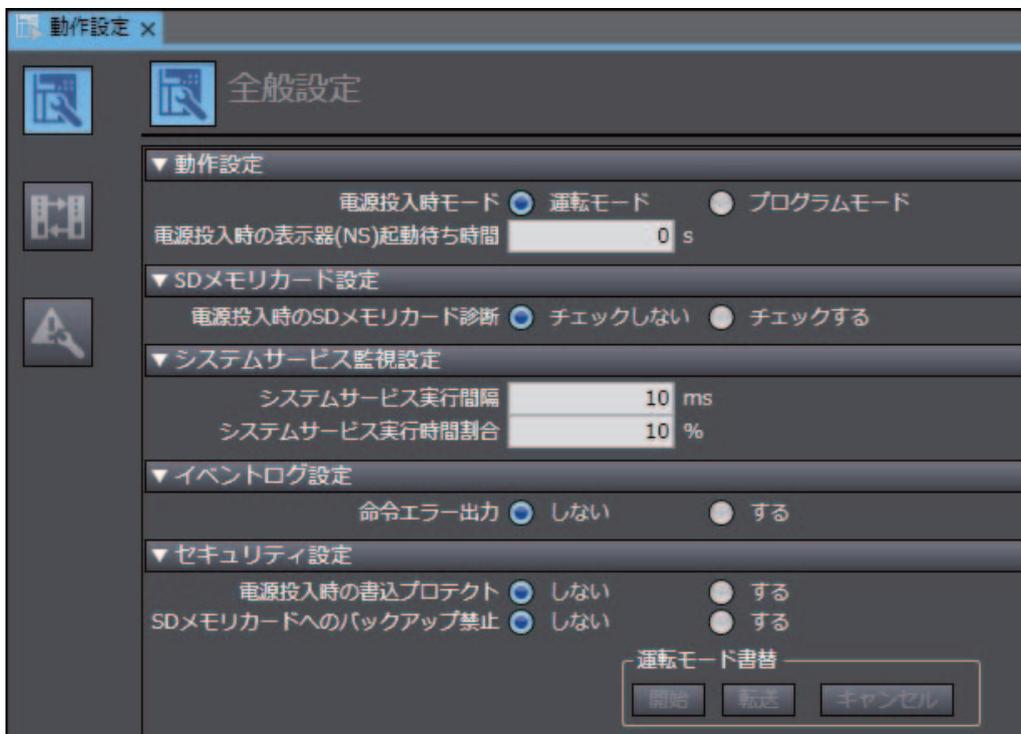
## 8-6-2 可变更事件重要程度的事件

事件的重要程度是否可以变更，取决于各事件。

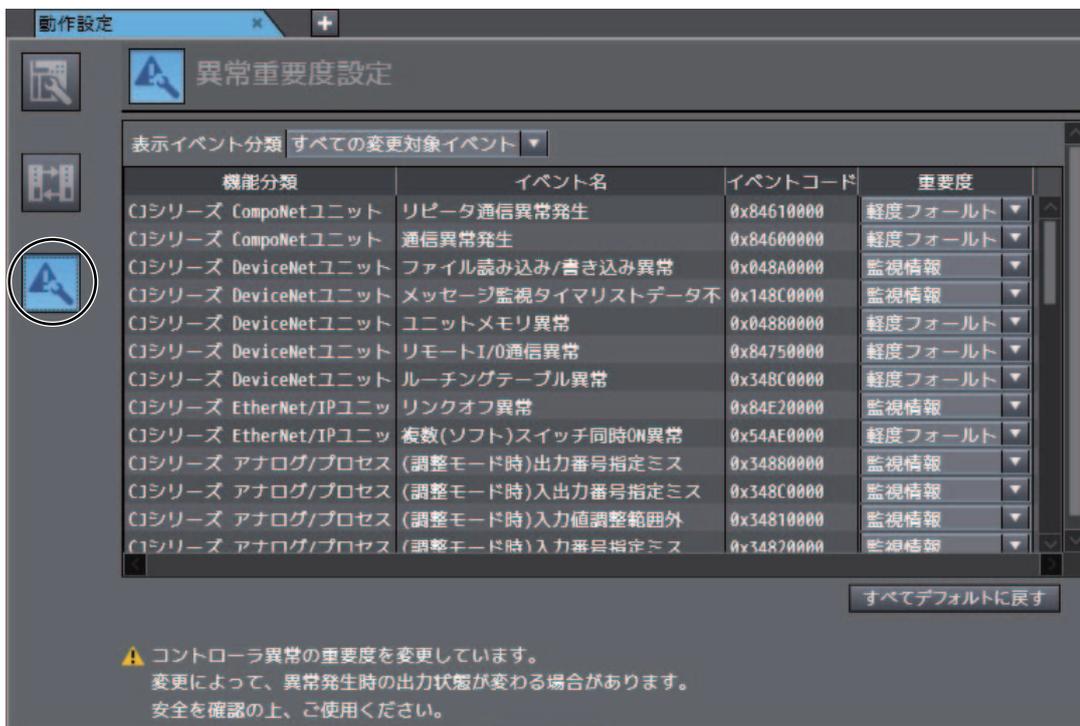
控制器事件的种类、重要程度、重要程度变更的可否请参考 □□《NY 系列 故障排除手册 (SBCA-CN5-438)》。

## 8-6-3 事件重要程度变更的操作方法

- 1 双击或右击 **Sysmac Studio** 的 [配置和设定]-[控制器设定]-[动作设定]，选择 [编辑]。编辑窗口中将显示动作设定的 [整体设定] 视图。



- 2 单击 [异常重要程度设定] 按钮。  
显示可变更事件重要程度的事件一览。



- 3 将要变更重要程度的事件之 [重要程度] 设定为变更目标的重要程度。



### 使用注意事项

---

通过 **Sysmac Studio** 进行事件重要程度变更设定，并下载到控制器时，如果已发生该事件，开始下载时，将先解除该事件。下载过程中，如果再次发生事件，将根据变更前的重要程度动作。下载完成后，如果发生该事件，将切换为与变更后的重要程度对应的动作。

---

# 备份功能

本章介绍 NY 系列控制器的各种设定数据的备份功能。备份功能包括不同的种类，各自的备份对象数据和保存位置不同。首先，对备份功能进行整体说明，然后对不同种类进行说明。

<b>9-1 备份功能的概要</b> .....	<b>9-3</b>
9-1-1 备份功能的用途 .....	9-3
9-1-2 备份功能的对象数据 .....	9-4
9-1-3 备份功能的种类 .....	9-5
9-1-4 备份功能的种类和数据表的关系 .....	9-7
9-1-5 备份功能的对象范围 .....	9-8
<b>9-2 SD 存储卡的备份功能</b> .....	<b>9-10</b>
9-2-1 备份（从控制器到虚拟 SD 存储卡） .....	9-11
9-2-2 核对（控制器和虚拟 SD 存储卡之间） .....	9-15
<b>9-3 SD 存储卡备份禁止功能</b> .....	<b>9-19</b>
<b>9-4 Sysmac Studio 控制器备份功能</b> .....	<b>9-20</b>
9-4-1 备份（从控制器到计算机） .....	9-21
9-4-2 恢复（从计算机到控制器） .....	9-22
9-4-3 核对（控制器和计算机之间） .....	9-23
<b>9-5 Sysmac Studio 备份文件的导入 / 导出功能</b> .....	<b>9-24</b>
<b>9-6 Sysmac Studio 变量 / 存储器的备份功能</b> .....	<b>9-25</b>
9-6-1 Sysmac Studio 变量 / 存储器备份功能的对象数据 .....	9-25
9-6-2 Sysmac Studio 变量 / 存储器备份功能的操作方法 .....	9-25
9-6-3 NY 系列控制器的型号兼容性 .....	9-25
<b>9-7 连接 EtherCAT 从站时的备份功能</b> .....	<b>9-26</b>
9-7-1 EtherCAT 从站的备份对象数据 .....	9-26
9-7-2 控制器在各种状态下的备份功能执行可否 .....	9-27
9-7-3 EtherCAT 从站的恢复执行条件 .....	9-28
9-7-4 备份功能的对象 EtherCAT 从站 .....	9-29
<b>9-8 连接 EtherCAT 从站终端时的备份功能</b> .....	<b>9-31</b>
9-8-1 EtherCAT 从站终端的备份对象数据 .....	9-31
9-8-2 EtherCAT 从站终端在各状态下的备份功能执行可否 .....	9-32
9-8-3 EtherCAT 从站终端的恢复执行条件 .....	9-32

<b>9-9 备份相关文件</b> .....	<b>9-33</b>
9-9-1 备份相关文件的种类 .....	9-33
9-9-2 备份文件的规格 .....	9-34
9-9-3 恢复指令文件的规格 .....	9-35
9-9-4 控制器核对结果文件的规格 .....	9-37
9-9-5 EtherCAT 核对结果文件的规格 .....	9-38
9-9-6 EtherCAT 从站终端核对结果文件的规格 .....	9-39
<b>9-10 备份功能的文件间兼容性</b> .....	<b>9-40</b>
9-10-1 备份功能间的兼容性 .....	9-40
9-10-2 NY 系列控制器的型号兼容性 .....	9-41
9-10-3 NY 系列控制器的单元版本兼容性 .....	9-42
<b>9-11 无法与备份功能同时执行的功能</b> .....	<b>9-43</b>

## 9-1 备份功能的概要

NY 系列控制器的备份功能是指以下 3 个。

功能	内容
备份	将控制器中的各种设定数据汇总保存到虚拟 SD 存储卡或计算机中的功能。要保存的文件称为备份文件。
恢复	将虚拟 SD 存储卡或计算机中的备份文件传送到控制器的功能。控制器中的各种设定数据将替换为备份时的数据。
核对	对虚拟 SD 存储卡或计算机中的备份文件内容与当前控制器中的各种设定数据是否一致进行判定的功能。

下面介绍整体的备份功能。

项目	内容
备份功能的用途	介绍备份功能的有效使用方法。
备份功能的操作步骤示例	备份功能可通过简单的操作执行。表示其示例。
备份功能的对象数据	介绍在使用备份功能后，可以保存所连接单元或从站中的哪些数据。
备份功能的种类	备份功能因数据保存位置等不同，有以下几种。介绍有哪些种类以及它们的区别。
备份功能的种类和数据表的关系	因备份功能的种类不同，对象数据表也不同。介绍备份功能的种类和数据表的关系。
备份功能的对象范围	介绍在使用备份功能后，可以保存哪个单元或从站的设定数据。

### 9-1-1 备份功能的用途

备份功能可在以下用途中使用。

项目	用途
程序、设定变更	变更运行中装置的用户程序或设定时。
硬件更换	更换工业电脑、单元、从站的硬件时。
装置异常时的原因调查	装置发生异常时，为分析原因而保存控制器的数据时。
装置的备份和复原	装置发生异常时，将装置恢复为正常的状态数据时。或者事先备份运行中装置的数据时。
装置的制造	为制造相同的装置，将复制源装置的数据传送到购入后初始状态的装置时。

## 9-1-2 备份功能的对象数据

备份功能的对象数据如下所示。

在备份功能的详细说明中，表示对象数据时，使用“数据组”。

数据组	数据项目
用户程序设定	EtherCAT 构成（EtherCAT 从站构成、EtherCAT 主机设定） 单元构成和单元设定 I/O 映射 控制器设定（动作设定、内置 EtherNet/IP 端口设定） 运动控制设定 凸轮数据设定 事件设定 任务设置 数据追踪设定 标签数据链接表 控制器名称 操作权限认证 用户程序执行用 ID（用户程序内） POU 数据（数据型、全局变量）
变量的当前值	有保持属性的变量值 <sup>*1</sup>
单元和从站设定	EtherCAT 从站的备份参数 <sup>*2</sup>
绝对编码器原点位置偏置	用于恢复带绝对编码器的伺服驱动器的当前位置的设定值

\*1. 有保持属性的系统定义变量中，包含不在备份功能的对象范围内的系统定义变量。详情请通过  「A-4 系统定义变量的个别规格」(P.A-45) 确认个别规格。

\*2. EtherCAT 从站中，有些机型的部分或全部设定参数可能不在备份对象内。备份功能的对象 EtherCAT 从站详情请参考  「9-7-4 备份功能的对象 EtherCAT 从站」(P.9-29)。



### 安全要点

#### 绝对编码器原点位置偏置相关的注意事项

产品利用非易失性存储器来保持绝对编码器的信息，即“绝对编码器原点位置偏置”。满足以下任一条件时，请将绝对编码器原点位置偏置从恢复对象中排除，再执行恢复。然后，请重新设定绝对编码器原点。若未设定绝对编码器原点，控制对象可能发生意外动作。

- 伺服电机或驱动器的个体变为创建备份时的个体。
- 创建备份后，执行了绝对编码器的设置。
- 绝对编码器的绝对值数据将丢失。

### 9-1-3 备份功能的种类

NY 系列控制器中可使用的备份功能的数据保存位置分为 SD 存储卡和计算机。操作方法有以下三种：通过系统定义变量操作、通过 Sysmac Studio 画面操作、通过 Industrial PC Support Utility 操作。

#### 将数据保存到虚拟 SD 存储卡的功能

向虚拟 SD 存储卡进行数据备份、核对的功能为 SD 存储卡备份功能。相关功能有 SD 存储卡备份禁止功能。

功能名称		内容	操作方法		参考
			系统定义变量	Sysmac Studio 画面	
SD 存储卡的备份功能	备份	将控制器的数据作为备份文件保存到虚拟 SD 存储卡中。	○	○	☐「9-2-1 备份（从控制器到虚拟 SD 存储卡）」（P.9-11）
	核对	对控制器的数据和虚拟 SD 存储卡的备份文件数据进行核对。	○	○	☐「9-2-2 核对（控制器和虚拟 SD 存储卡之间）」（P.9-15）
SD 存储卡备份禁止功能		禁止向虚拟 SD 存储卡备份。		○	☐「9-3 SD 存储卡备份禁止功能」（P.9-19）



#### 参考

- 将虚拟SD存储卡中保存的备份文件恢复到控制器时，在Industrial PC Support Utility的控制器备份功能的恢复中，指定虚拟 SD 存储卡中设定的共享文件夹。
- 利用 Sysmac Studio 的控制器备份功能也可恢复。此时，需要将虚拟 SD 存储卡中设定的共享文件夹中保存的备份文件，通过外部存储器等，预先保存到安装有 Sysmac Studio 的计算机中。

## 将数据保存到计算机的功能

将控制器的数据向计算机进行备份、恢复、核对的功能为 Sysmac Studio 控制器备份功能。

不使用控制器，通过 Sysmac Studio 的项目将各数据保存、读取到计算机的功能为 Sysmac Studio 备份文件的导入 / 导出功能。

只将非易失性存储器保存的当前值向计算机进行备份、恢复的功能为 Sysmac Studio 变量 / 存储器的备份功能。

功能名称	内容	操作方法		参考	
		系统定义 变量	Sysmac Studio 画面		
Sysmac Studio 控制器备份 功能	备份	将控制器的数据作为备份文件保存到计算机中。		○	☐「9-4-1 备份（从控制器到计算机）」 (P.9-21)
	恢复	将计算机中的备份文件数据传送到控制器中。		○	☐「9-4-2 恢复（从计算机到控制器）」 (P.9-22)
	核对	对控制器的数据和计算机的备份文件数据进行核对。		○	☐「9-4-3 核对（控制器和计算机之间）」 (P.9-23)
Sysmac Studio 备份文件的 导入 / 导出 功能	导出	不使用控制器，通过 Sysmac Studio 的项目创建备份文件。		○	☐「9-5 Sysmac Studio 备份文件的导入 / 导出功能」 (P.9-24)
	导入	不使用控制器，将备份文件读取到 Sysmac Studio 的项目中。		○	
Sysmac Studio 变量 / 存储 器的备份 功能	备份	将非易失性存储器保持的数据当前值作为文件（XML 格式）备份（读取）到计算机中。		○	☐「9-6 Sysmac Studio 变量 / 存储器的备份功能」 (P.9-25)
	恢复	将非易失性存储器保持的数据当前值由计算机对控制器进行恢复（写入）。		○	

## 将数据保存到 Windows 文件夹的功能

将控制器的数据向 Windows 的文件夹进行备份、恢复、核对的功能为 Industrial PC Support Utility 控制器备份功能。

关于 Industrial PC Support Utility 控制器备份功能的详情，请参考☐《NY 系列 工业平板电脑 / 工业 BOX PC 用户手册 安装篇（SBCA-CN5-441）》。

## 9-1-4 备份功能的种类和数据表的关系

因备份功能的种类不同，对象数据表也不同。备份功能的种类和数据表的关系如下所示。

(○：对象 x：对象外)

备份功能的种类		数据组			
		用户程序设定	变量的当前值	单元和从站设定	绝对编码器原点位置偏置
SD 存储卡的备份功能	备份	○	○ <sup>*1</sup>	○	○
	恢复 <sup>*2</sup>	○	○ <sup>*1</sup>	○	○
	核对 <sup>*2</sup>	○ <sup>*3</sup>	x	○	x
Sysmac Studio 控制器备份功能	备份	○	○ <sup>*1</sup>	○ <sup>*4</sup>	○
	恢复	○	○ <sup>*1</sup>	○ <sup>*4</sup>	○
	核对	○ <sup>*3</sup>	x	○ <sup>*4</sup>	x
Industrial PC Support Utility 控制器备份功能	备份	○	○ <sup>*1</sup>	○ <sup>*4</sup>	○
	恢复	○	○ <sup>*1</sup>	○ <sup>*4</sup>	○
	核对	○ <sup>*3</sup>	x	○ <sup>*4</sup>	x
Sysmac Studio 备份文件的导入 / 导出功能	备份文件的导出	○ <sup>*5</sup>	x	x	x
	备份文件导入	○ <sup>*5</sup>	x	○	x
Sysmac Studio 变量 / 存储器的备份功能	备份恢复	x	○ <sup>*1</sup>	x	○

\*1. 备份的对象数据仅在保持属性的变量当前值时处理。

\*2. 对象数据组中，用户程序、设定以外的数据仅将恢复指令文件中指定为“恢复”的数据作为对象。

\*3. 用户程序、设定数据组中，不核对数据追踪设定。

\*4. 备份对象为 EtherCAT 从站时，将 EtherCAT 从站的备份参数作为对象。

\*5. 以下数据不在处理对象范围内。内置 EtherNet/IP 用标签数据链接设定、操作权限认证、数据追踪设定。



## 参考

Sysmac Studio 变量 / 存储器的备份功能中处理的文件与其他备份功能中处理的文件不同，因此不兼容。

Sysmac Studio 变量 / 存储器的备份功能详情请参考 □ 「9-6 Sysmac Studio 变量 / 存储器的备份功能」(P.9-25)。

## 9-1-5 备份功能的对象范围

因备份功能不同，对象从站或单元也不同。各备份功能的对象从站、单元如下所示。

(○：对象 x：对象外)

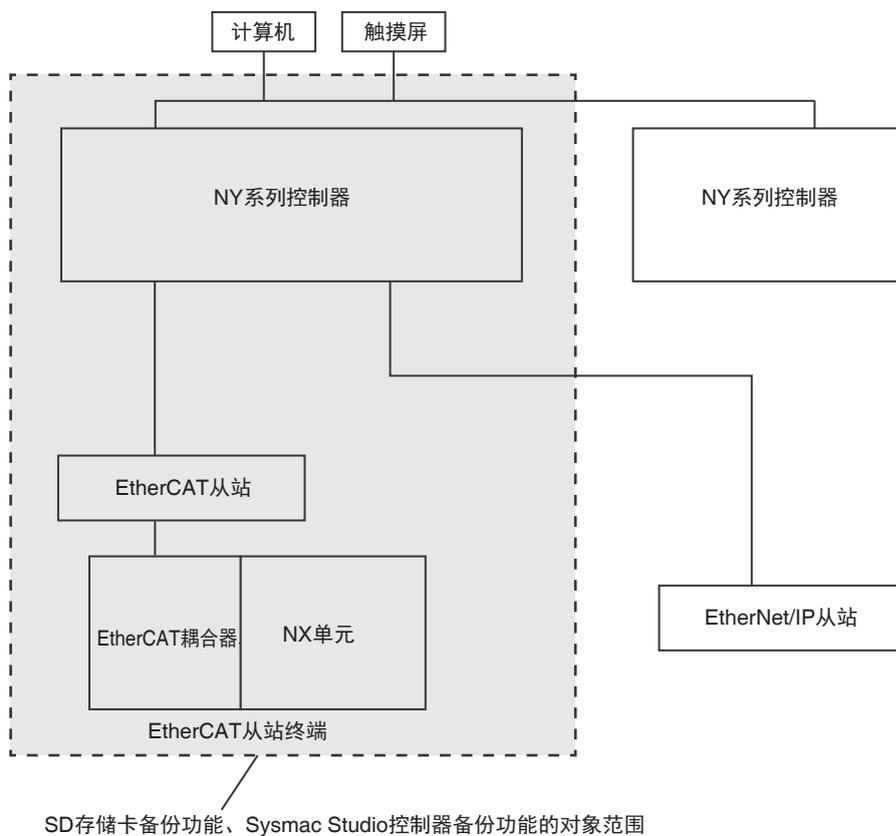
备份功能的种类	从站、单元			
	NY 系列控制器	EtherCAT 从站*1	EtherNet/IP 从站	计算机、触摸屏
SD 存储卡的备份功能	○	○*2	x	x
Sysmac Studio 控制器备份功能	○	○*2	x	x
Sysmac Studio 备份文件的导入 / 导出功能	○	○*3	x	x
Sysmac Studio 变量 / 存储器的备份功能	○	x	x	x

\*1. 包括 EtherCAT 从站终端。备份功能的对象为 EtherCAT 从站终端时，EtherCAT 耦合器单元和各 NX 单元均为备份功能的对象。

\*2. 安全控制单元不在对象范围内。安全控制单元的设定数据导入 / 导出功能请参考□□《NX 系列 安全控制单元 用户手册 (SGFM-CN5-710)》。

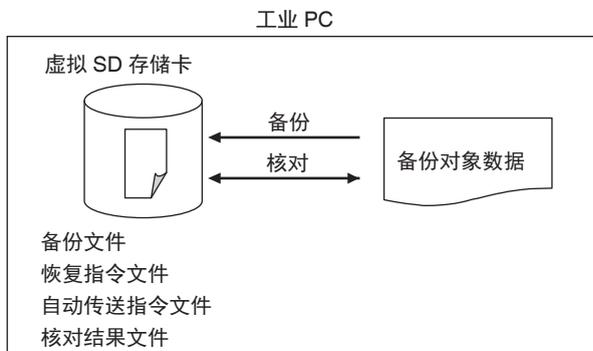
\*3. 只能导入。导出功能不在对象范围内。

SD 存储卡备份功能、Sysmac Studio 控制器备份功能的对象从站和单元如下图所示。



## 9-2 SD 存储卡的备份功能

使用虚拟 SD 存储卡，进行控制器数据的备份、核对的功能。



执行备份后，将在虚拟 SD 存储卡的指定目录下创建“备份文件”、“恢复指令文件”、“自动传送指令文件”。此外，若执行核对，将在指定目录下创建“核对结果文件”。

这些统称为“备份相关文件”。

备份相关文件的作用如下。

文件	作用		
	内容	备份	核对
备份文件	用备份相关功能中处理的控制器数据制作的文件。	创建	参考
恢复指令文件	执行恢复时，用于指定恢复对象数据组的文件。在计算机的文本编辑器中编辑文件，可指定恢复对象。	创建	参考
自动传送指令文件	NY 系列控制器中不使用本文件。	创建	无操作
核对结果文件	执行核对后，用于保存核对结果的文件。	无操作	创建

本功能的操作方法、对象目录及可执行的动作模式如下。

操作方法	目录 <sup>*1</sup>	可执行的动作模式		
		备份	恢复	核对
系统定义变量 <sup>*2*3</sup>	系统定义变量中指定的目录	运行模式、程序模式	不可执行	运行模式、程序模式
Sysmac Studio SD 存储卡画面	画面中指定的目录	运行模式、程序模式	不可执行	运行模式、程序模式

\*1. 仅可在虚拟 SD 存储卡内指定。

\*2. 通过触摸屏发出指示时使用。无法通过用户程序访问该系统定义变量。

\*3. 备份中或核对中时，请在用户侧进行管理，防止从其他的触摸屏发出备份或核对指示。若非如此，可能不按预期动作。

## 9-2-1 备份（从控制器到虚拟 SD 存储卡）

将控制器的数据保存到虚拟 SD 存储卡中。

### 处理内容

- 备份对象数据为所有数据组。
- 执行备份后，将在虚拟 SD 存储卡的指定目录下创建“备份文件”、“恢复指令文件”、“自动传送指令文件”。
- 指定目录下已经存在备份相关文件时，将被覆盖。
- 正在向虚拟 SD 存储卡写入备份相关文件时如果发生错误，已有的备份相关文件将被删除。然后，也不会创建新的备份相关文件。
- 创建新的备份相关文件前如果发生错误，将保持已有的文件，不创建新文件。
- 备份完成后，虚拟 SD 存储卡将保持识别状态。

### 操作方法

#### ● 通过系统定义变量 `_Card1BkupCmd`（SD 存储卡备份指示）操作的方法

处理阶段	操作方法
开始指示	在系统定义变量 <code>_Card1BkupCmd.DirName</code> （目录名称）中保存保存位置的目录名称。 （例）指定目录 <code>dirA</code> 下的子目录 <code>dirB</code> ：“ <code>dirA/dirB</code> ”  系统定义变量 <code>_Card1BkupCmd.ExecBkup</code> （备份启动标志）从 <code>FALSE</code> 变为 <code>TRUE</code> ，则开始执行。
取消指示	取消备份执行指示。 将系统定义变量 <code>_Card1BkupCmd.CancelBkup</code> （备份取消标志）设为 <code>TRUE</code> 后，备份处理将异常结束。
执行中	系统定义变量 <code>_Card1BkupSta.Active</code> （执行中标志）变为 <code>TRUE</code> 。 系统定义变量 <code>_BackupBusy</code> （备份相关功能执行中标志）的值变为 <code>TRUE</code> 。
执行结果	正常结束时 系统定义变量 <code>_Card1BkupSta.Done</code> （完成标志）变为 <code>TRUE</code> 。  异常结束时 系统定义变量 <code>_Card1BkupSta.Err</code> （错误标志）变为 <code>TRUE</code> 。

（注）无法通过用户程序访问该系统定义变量。

#### ● 通过 Sysmac Studio 的 SD 存储卡画面操作的方法

处理阶段	操作方法
开始指示	按下 Sysmac Studio SD 存储卡画面的 [SD 存储卡备份] 按钮，指定保存位置的目录名称，发出执行备份的指示。
执行中	显示备份进度状况画面。 系统定义变量 <code>_BackupBusy</code> （备份相关功能执行中标志）的值变为 <code>TRUE</code> 。
执行结果	显示完成信息，接着确认执行备份数据的核对。

操作的详情请参考 □《Sysmac Studio Version 1 操作手册（SBCA-CN5-362）》。

## ● 通过专用指令操作的方法

处理阶段	操作方法
开始指示	在用户程序中执行 BackupToMemoryCard 指令。
执行中	BackupToMemoryCard 指令的输出变量 Busy 的值变为 TRUE。 系统定义变量 _BackupBusy（备份相关功能执行中标志）的值变为 TRUE。
执行结果	正常结束时 BackupToMemoryCard 指令的输出变量 Done 的值变为 TRUE。  异常结束时 BackupToMemoryCard 指令的输出变量 Error 的值变为 TRUE。 BackupToMemoryCard 指令的输出变量 ErrorID 中将保存错误代码。

## 相关系统定义变量

利用系统定义变量进行备份时，相关的系统定义变量如下。系统定义变量的详情请参考□□「A-4 系统定义变量的个别规格」（P.A-45）。

变量名称 成员名称	名称	功能	数据类型	R/W
_Card1BkupCmd <sup>*1</sup>	SD 存储卡 备份指示		_sBKUP_ CMD	RW
ExecBkup <sup>*1</sup>	备份启动标志	向虚拟 SD 存储卡备份控制器的数据时，变为 TRUE。	BOOL	RW
CancelBkup <sup>*1</sup>	备份取消标志	停止向虚拟 SD 存储卡备份时，变为 TRUE。	BOOL	RW
DirName <sup>*1</sup>	目录名	指定备份对象的虚拟 SD 存储卡目录名称。	STRING (64)	RW
_Card1BkupSta <sup>*1</sup>	SD 存储卡 备份状态		_sBKUP_ STA	R
Done <sup>*1</sup>	完成标志	备份执行完成时为 TRUE。	BOOL	R
Active <sup>*1</sup>	执行中标志	正在执行备份时为 TRUE。	BOOL	R
Err <sup>*1</sup>	错误标志	备份处理异常结束时为 TRUE。	BOOL	R
_BackupBusy	备份相关功能执 行中标志	正在执行备份、恢复、核实时为 TRUE。	BOOL	R

\*1. 无法通过用户程序访问该系统定义变量。



### 参考

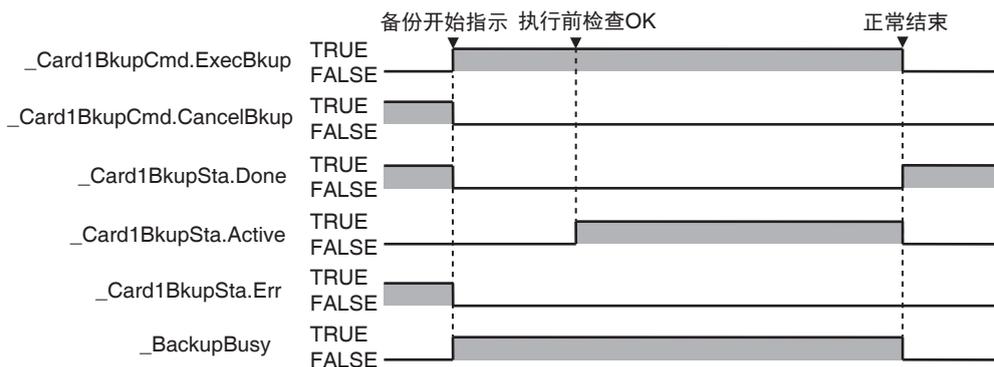
- 连接 NY 系列控制器和 NA 系列触摸屏使用时，变量映射请参考□□《可编程终端 NA 系列 用户手册 软件篇 (SBSA-CN5-546)》。
- 连接 NY 系列控制器和 NS 系列触摸屏使用时，将这些系统定义变量登录到 CX-Designer 变量表的方法请参考□□「A-7 CX-Designer 的变量表登录方法」（P.A-94）。

## 时序图

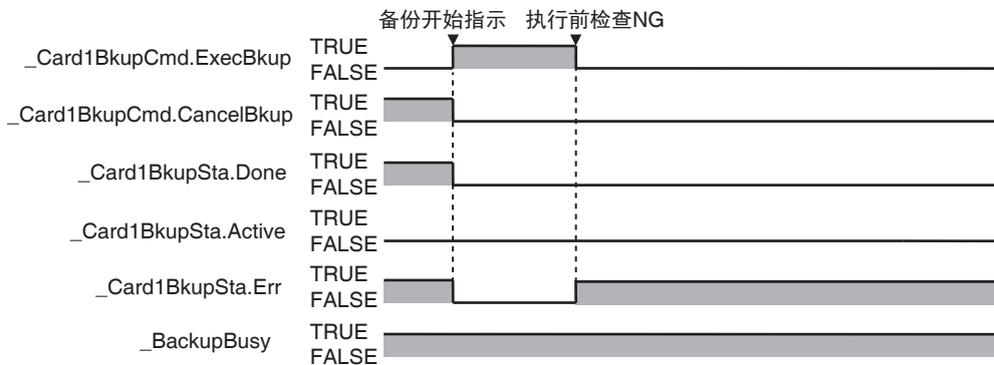
利用系统定义变量进行备份时，系统定义变量的动作如下。

图中的“执行前检查”是指在开始备份前，检查是否存在虚拟 SD 存储卡。\_Card1BkupSta.Active（执行中标志）的值变为 TRUE，则表示执行前检查的结果为 OK，已实际开始备份。

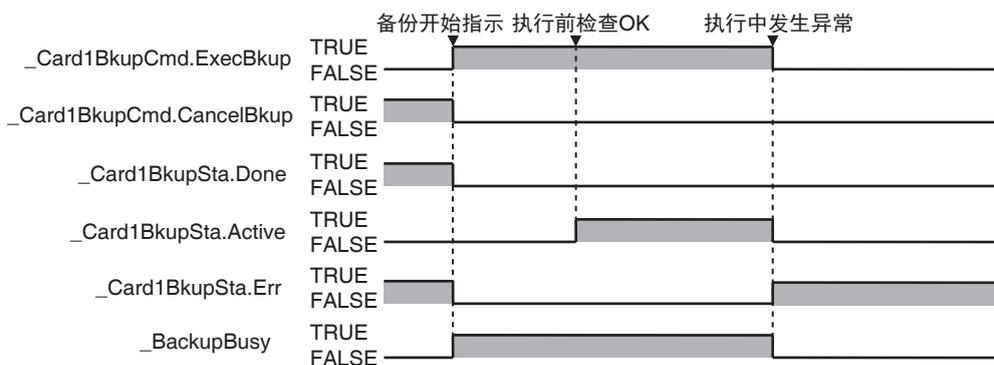
### ● 正常动作



### ● 因正在执行其他备份功能，无法开始备份时

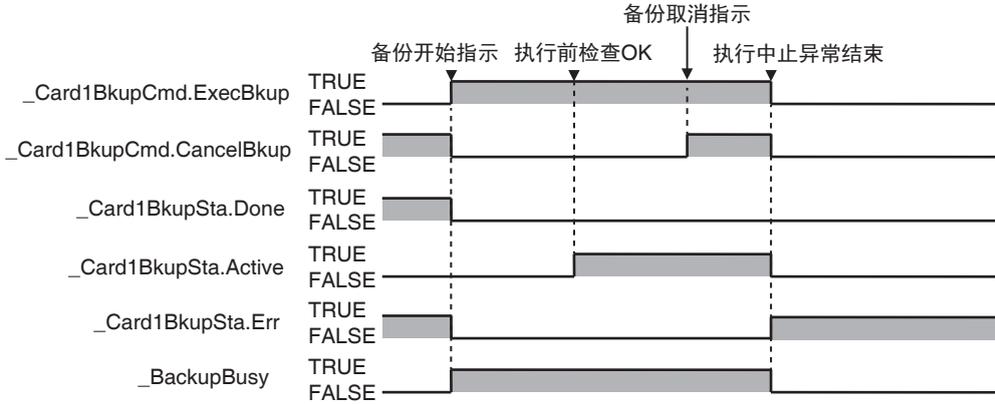


### ● 开始正常备份后，因某些原因导致备份失败时



● 备份执行中发出取消指示时

从发出取消指示到停止备份为止的时间因备份处理的进度状况不同而异。



**处理时间和备份文件的大小**

备份所需的时间因动作模式、单元构成、用户程序的内容不同而异。此外，备份文件的大小因单元构成、用户程序的内容不同而异。以下表示备份时间和备份文件大小的参考值。

控制器	动作模式	所连接的 EtherCAT 从站	用户定义的 POU 数	用户程序的存储大小 [MB]	备份时间 [s]	备份文件的大小 [MB]
NY5 □ 2-1 □ □ □ □	程序模式	*1	72	3.82	约 155	20.27

\*1. 由 AC 伺服驱动器 R88D-KNA-ECT: 32 台、EtherCAT 耦合器单元 NX-ECC201: 5 台构成。  
 EtherCAT 耦合器单元上的 NX 单元构成如下。  
 NX-PD1000: 5 台、NX-AD4608: 5 台、NX-DA3605: 5 台、NX-ID3343: 64 台、NX-OD3153: 64 台

## 9-2-2 核对（控制器和虚拟 SD 存储卡之间）

对控制器的数据和虚拟 SD 存储卡的备份文件数据进行核对。

### 处理内容

- 对控制器的数据和虚拟 SD 存储卡的指定目录下保存的备份文件数据进行核对。
- 核对对象数据为恢复指令文件（RestoreCommand.ini）中指定为“恢复”的数据组。
- 变量的当前值、绝对编码器原点位置偏置可能在执行核对的过程中发生替换，因此不核对。
- 执行核对后，将在指定目录下创建“核对结果文件（VerifyResult.log）”，并保存核对结果。执行核对前，指定目录下已经存核对结果文件时，将被覆盖。
- 虚拟 SD 存储卡的指定目录下不存在恢复指令文件时，将核对指定目录下存在的所有可核对的备份文件数据。
- 备份文件的单元和从站构成与核对对象的实际单元和从站构成不同时，将发生核对错误。
- 核对完成后，虚拟 SD 存储卡将保持识别状态。

### 操作方法

#### ● 系统定义变量 `_Card1BkupCmd`（SD 存储卡备份指示）

处理阶段	操作步骤
开始指示	<p>在系统定义变量 <code>_Card1BkupCmd.DirName</code>（目录名称）中保存保存位置的目录名称。 （例）指定目录 <code>dirA</code> 下的子目录 <code>dirB</code>：“<code>dirA/dirB</code>”</p> <p>系统定义变量 <code>_Card1BkupCmd.ExecVefy</code>（核对启动标志）从 <code>FALSE</code> 变为 <code>TRUE</code>，则开始执行。</p>
取消指示	<p>取消核对执行指示。 将系统定义变量 <code>_Card1BkupCmd.CancelVefy</code>（核对取消标志）设为 <code>TRUE</code>，核对处理将异常结束。</p>
执行中	<p>系统定义变量 <code>_Card1VefySta.Active</code>（执行中标志）变为 <code>TRUE</code>。 系统定义变量 <code>_BackupBusy</code>（备份相关功能执行中标志）的值变为 <code>TRUE</code>。</p>
执行结果	<p>正常结束，核对一致时 系统定义变量 <code>_Card1BkupSta.Done</code>（完成标志）和 <code>_Card1BkupSta.VefyRslt</code>（核对结果标志）变为 <code>TRUE</code>。</p> <p>正常结束，核对不一致时 系统定义变量 <code>_Card1BkupSta.Done</code>（完成标志）变为 <code>TRUE</code>，<code>_Card1BkupSta.VefyRslt</code>（核对结果标志）变为 <code>FALSE</code>。</p> <p>异常结束时 系统定义变量 <code>_Card1BkupSta.Err</code>（错误标志）变为 <code>TRUE</code>。</p>

（注）无法通过用户程序访问该系统定义变量。

## ● 通过 Sysmac Studio 的 SD 存储卡画面操作的方法

处理阶段	处理步骤
开始指示	按下 Sysmac Studio SD 存储卡画面的 [SD 存储卡核对] 按钮，指定核对目标保存位置的目录名称，发出执行核对的指示。
执行中	显示核对进度状况画面。系统定义变量 _BackupBusy（备份相关功能执行中标志）的值变为 TRUE。
执行结果	显示核对结果画面。

操作的详情请参考 [□□《Sysmac Studio Version 1 操作手册（SBCA-CN5-362）》](#)。

## 相关系统定义变量

利用系统定义变量进行核实时，相关的系统定义变量如下。系统定义变量的详情请参考 [□□「A-4 系统定义变量的个别规格」\(P.A-45\)](#)。

变量名称 成员名称	名称	功能	数据类型	R/W
_Card1BkupCmd <sup>*1</sup>	SD 存储卡 备份指示		_sBKUP_ CMD	RW
ExecVefy <sup>*1</sup>	核对启动标志	核对控制器和虚拟 SD 存储卡中的备份文件时，变为 TRUE。	BOOL	RW
CancelVefy <sup>*1</sup>	核对取消标志	停止核对控制器和虚拟 SD 存储卡中的备份文件时，变为 TRUE。	BOOL	RW
DirName <sup>*1</sup>	目录名	指定备份对象的虚拟 SD 存储卡目录名称。	STRING (64)	RW
_Card1VefySta <sup>*1</sup>	SD 存储卡 核对状态		_sVEFY_ STA	R
Done <sup>*1</sup>	完成标志	核对执行完成时为 TRUE。	BOOL	R
Active <sup>*1</sup>	执行中标志	正在执行核实时为 TRUE。	BOOL	R
VefyRslt <sup>*1</sup>	核对结果标志	核对一致时为 TRUE，核对不一致时为 FALSE。	BOOL	R
Err <sup>*1</sup>	错误标志	核对处理异常结束时为 TRUE。	BOOL	R
_BackupBusy	备份相关功能执行中标志	正在执行备份、恢复、核实时为 TRUE。	BOOL	R

\*1. 无法通过用户程序访问该系统定义变量。



## 参考

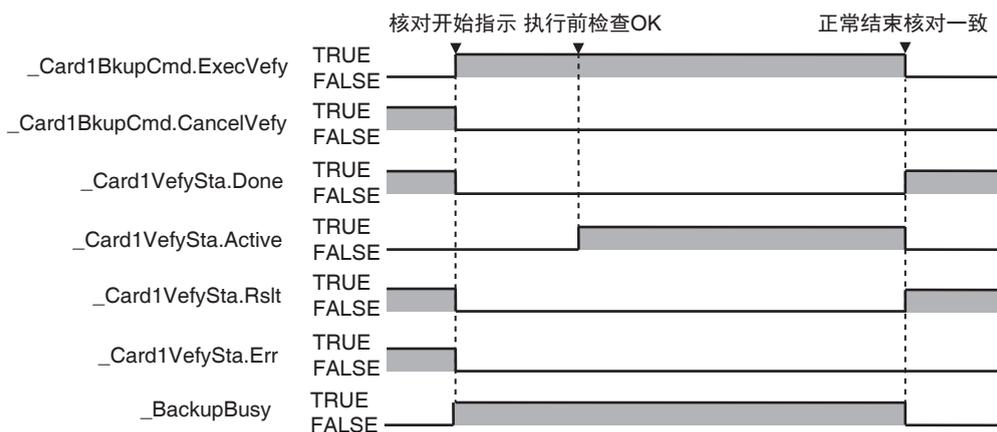
- 连接 NY 系列控制器和 NA 系列触摸屏使用时，变量映射请参考 □□《可编程终端 NA 系列 用户手册 软件篇 (SBSA-CN5-546)》。
- 连接 NY 系列控制器和 NS 系列触摸屏使用时，将这些系统定义变量登录到 CX-Designer 变量表的方法请参考 □□「A-7 CX-Designer 的变量表登录方法」(P.A-94)。

## 时序图

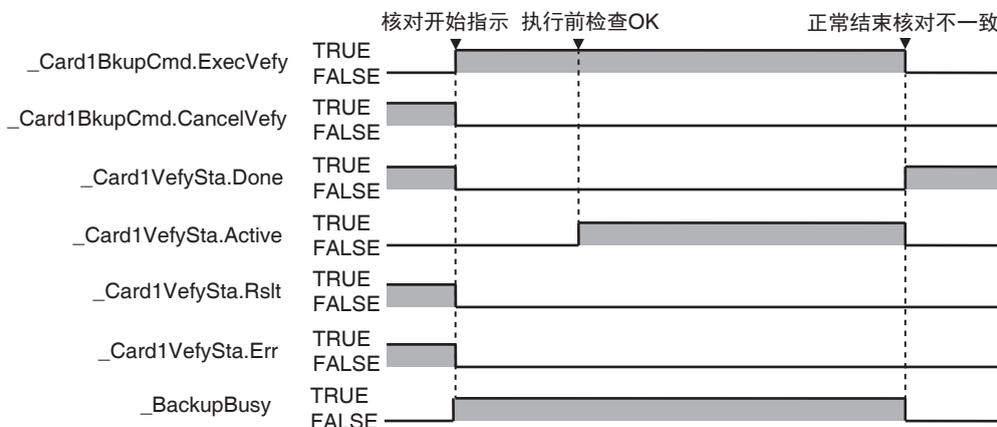
利用系统定义变量进行核对时，系统定义变量的动作如下。

图中的“执行前检查”是指在开始核对前，检查是否存在虚拟 SD 存储卡。\_Card1VefySta.Active（执行中标志）的值变为 TRUE，则表示执行前检查的结果为 OK，已实际开始核对。

### ● 正常动作，核对一致时



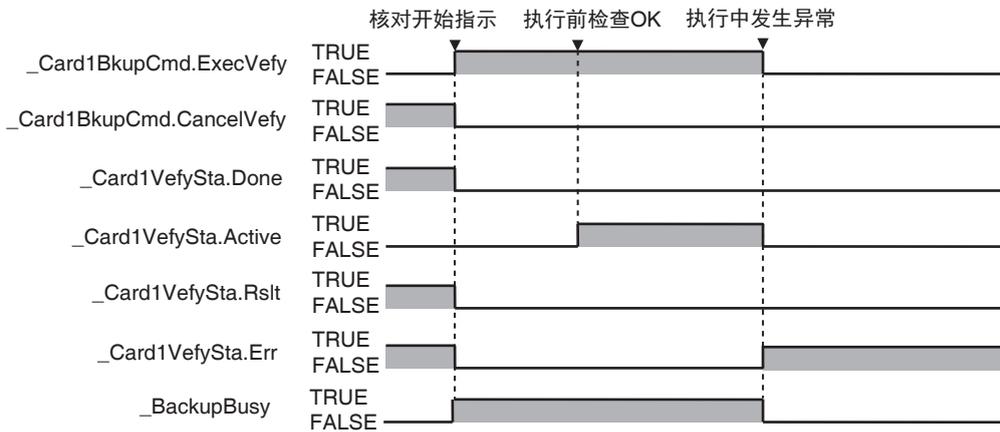
### ● 正常动作，核对不一致时



● 因正在执行其他备份功能，无法开始核实时

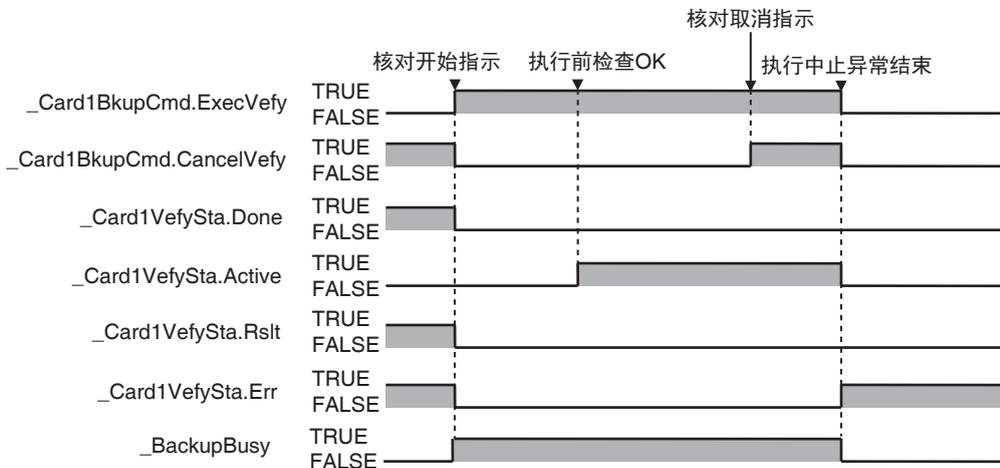


● 开始正常核对后，因某些原因导致核对失败时



● 核对执行中发出取消指示时

从发出取消指示到停止核对为止的时间因核对处理的进度状况不同而异。



## 9-3 SD 存储卡备份禁止功能

禁止向虚拟 SD 存储卡备份的功能。使用本功能，可保护用户资产信息。

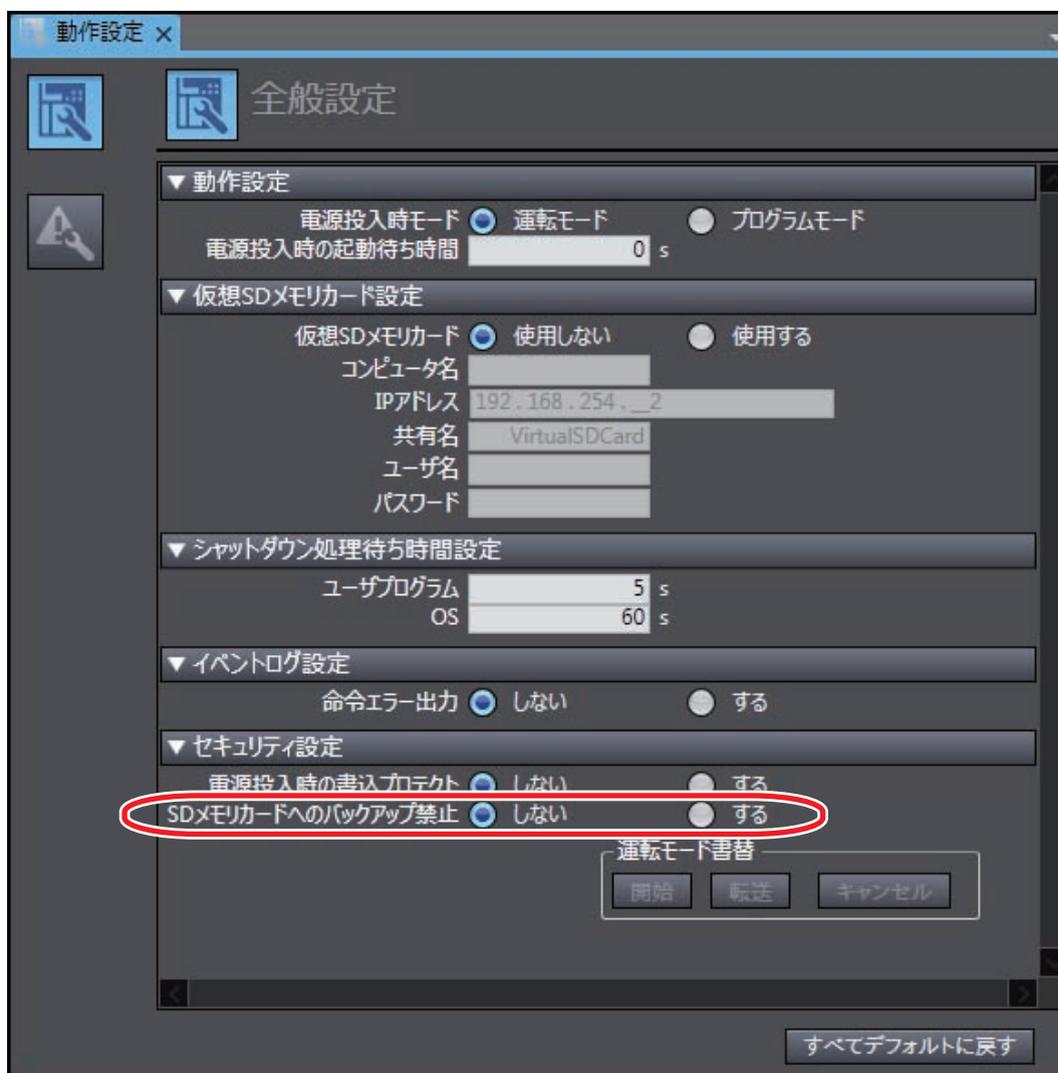
SD 存储卡备份禁止功能的对象为以下 2 个。

- 通过系统定义变量进行备份
- 通过 Sysmac Studio 的 SD 存储卡画面进行备份

通过 BackupToMemoryCard 指令的备份不在备份禁止功能的对象范围内。即，即使设定为 SD 存储卡备份禁止，也可通过 BackupToMemoryCard 指令执行备份。

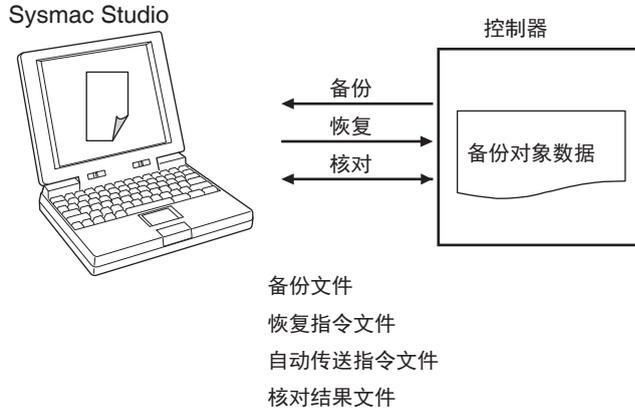
SD 存储卡备份禁止的操作方法如下。

在 Sysmac Studio 的 [ 配置和设定 ]-[ 控制器设定 ]-[ 动作设定 ] 的整体设定选项卡中，将 SD 存储卡的备份禁止设定为 [ 是 ]。



## 9-4 Sysmac Studio 控制器备份功能

使用 Sysmac Studio，通过计算机进行控制器数据的备份、恢复、核对的功能。



执行备份后，将在计算机的指定目录下创建“备份文件”、“恢复指令文件”、“自动传送指令文件”。备份相关文件的作用如下。

文件	作用			
	内容	备份	恢复	核对
备份文件	用备份相关功能中处理的控制器数据制作的文件。	创建	参考	参考
恢复指令文件	执行恢复时，用于指定传送对象数据组的文件。在计算机的文本编辑器中编辑文件，可指定传送对象。	创建	参考	参考
自动传送指令文件	NY 系列控制器中不使用本文件。	创建	无操作	无操作

本功能的可执行动作模式如下。

处理	可执行的动作模式
备份	运行模式、程序模式
恢复	程序模式
核对	运行模式、程序模式



### 参考

正在执行备份或核实时，可变更控制器的动作模式。  
但是，如控制器的存储器发生故障，无法正常执行备份或核实时，将发生错误。

## 9-4-1 备份（从控制器到计算机）

将控制器的数据保存到计算机的指定目录中。

### 处理内容

- 备份数据中的单元和从站设定只能选择所有连接的 EtherCAT 从站。
- 执行备份后，将在计算机的指定目录下创建“备份文件”、“恢复指令文件”、“自动传送指令文件”。
- 指定目录下已经存在备份相关文件时，将被覆盖。
- 正在向指定目录写入备份相关文件时如果发生错误，已有的备份相关文件将被删除。然后，也不会创建新的备份相关文件。
- 创建新的备份相关文件前如果发生错误，将保持已有的文件，不创建新文件。
- 执行过程中，系统定义变量 \_BackupBusy（备份相关功能执行中标志）的值变为 TRUE。

### 操作方法

- 1** 执行 Sysmac Studio 的 [工具]-[备份]-[控制器的备份]。
- 2** 指定创建备份文件、恢复指令文件、自动传送指令文件的目录。
- 3** 在备份确认画面中按下 [执行] 按钮。  
将执行备份，并创建备份文件、恢复指令文件、自动传送指令文件。

## 9-4-2 恢复（从计算机到控制器）

将计算机的指定目录下的备份文件数据恢复到控制器中。  
 仅可在程序模式下执行。

### 处理内容

- 将计算机的指定目录下的备份文件数据恢复到控制器中。
- 恢复对象数据组可在 Sysmac Studio 上选择。恢复所需的条件如下表所示。

数据组	恢复条件
用户程序设定	选择“CPU 单元”时
变量的当前值	选择“保持属性的变量当前值”时
单元和从站设定	选择“EtherCAT 从站”时
绝对编码器原点位置偏置	选择“绝对编码器原点位置偏置”时

- 在恢复处理的开始前检查中发生错误时，控制器中的数据将保持恢复前的数据。
- 正在执行恢复处理时如果控制器发生断电，将发生“用户程序/控制器配置和设定传送错误（全部停止故障等级的控制器异常）”。此时，控制器中的数据将变得不确定，请按以下方法排除异常。
  - 重新执行恢复。
  - 通过 Sysmac Studio 执行存储器完全清除、项目的下载。
- 有保持属性的变量当前值不是恢复对象时，该变量的当前值保持恢复前的值。但是，不满足维持变量本身的条件时，变量的值将被初始化。维持变量本身的条件如下所示。
  - 恢复前的变量和要恢复的变量的变量名称、数据类型名称、数据类型大小应全部相同。
- 恢复执行完成后，请重新接通所有 EtherCAT 从站的电源。

### 操作方法

- 1** 执行 Sysmac Studio 的 [ 工具 ]-[ 备份 ]-[ 控制器的恢复 ]。
- 2** 指定保存有备份文件、恢复指令文件的目录。
- 3** 在恢复确认画面中按下 [ 执行 ] 按钮。  
 执行恢复。

### 9-4-3 核对（控制器和计算机之间）

对控制器的数据和计算机中指定目录下的备份文件数据进行核对。

#### 处理内容

- 对控制器的数据和计算机中指定目录下的备份文件数据进行核对。核对对象数据组可在 Sysmac Studio 上选择。核对所需的条件如下表所示。如果批量指定，将对指定的所有数据进行核对。

数据组	核对条件
用户程序设定	选择“CPU 单元”时
单元和从站设定	选择“EtherCAT 从站”时

- 核对结果将显示在 Sysmac Studio 的画面中。
- 执行过程中，系统定义变量 \_BackupBusy（备份相关功能执行中标志）的值变为 TRUE。

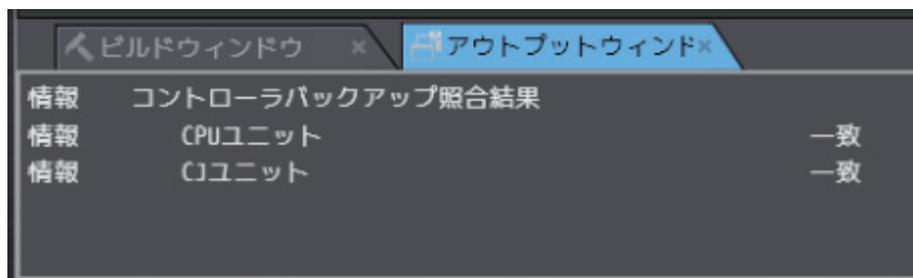
#### 操作方法

**1** 执行 Sysmac Studio 的 [ 工具 ]-[ 备份 ]-[ 备份文件的核对 ]。

**2** 指定保存有备份文件的目录。

**3** 在核对确认画面中按下 [ 执行 ] 按钮。

将指定核对，并将核对结果文件保存到保存有备份文件的目录中。或者在输出窗口中显示核对结果。



## 9-5 Sysmac Studio 备份文件的导入 / 导出功能

不使用控制器，通过 Sysmac Studio 的项目，在计算机的指定目录下创建和读取备份文件的功能。对象数据如下所示。

功能		数据组			
		用户程序设定	变量的当前值	单元和从站设定	绝对编码器原点位置偏置
Sysmac Studio 备份文件的导入 / 导出功能	备份文件的导出	○ <sup>*1</sup>	×	×	×
	备份文件的导入	○ <sup>*2</sup>	×	○	×

\*1. 以下数据不在处理对象范围内。

- 控制器名称中的内置 EtherNet/IP 端口名称
- 控制器设定中的内置 EtherNet/IP 端口用标签数据链接设定
- 操作权限认证
- 数据追踪设定

\*2. 以下数据不在处理对象范围内。

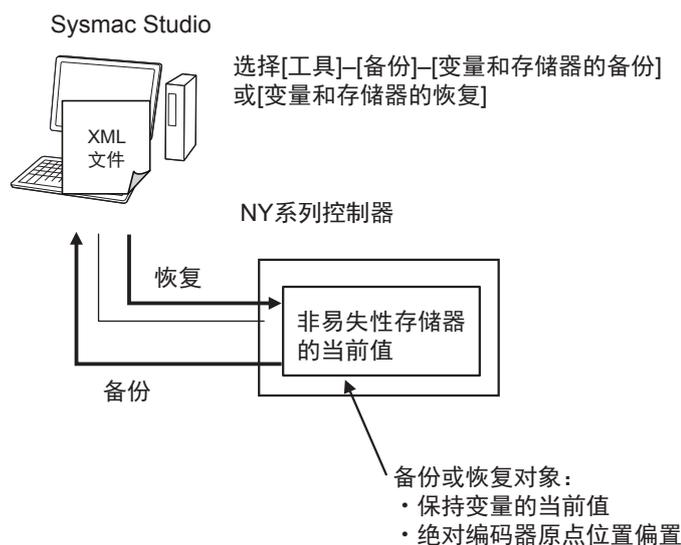
- 控制器名称中的内置 EtherNet/IP 端口名称
- 操作权限认证
- 数据追踪设定

本功能的详情请参考 □ 《Sysmac Studio Version 1 操作手册（SBCA-CN5-362D 之后）》。

## 9-6 Sysmac Studio 变量 / 存储器的备份功能

可将 NY 系列控制器中的非易失性存储器的当前值以文件（XML 格式）的形式备份（读取）到计算机中，或者从计算机恢复（写入）到控制器。

下面介绍 Sysmac Studio 变量 / 存储器备份功能的对象数据、操作方法及控制器型号之间的兼容性。



### 9-6-1 Sysmac Studio 变量 / 存储器备份功能的对象数据

Sysmac Studio 变量 / 存储器备份功能的对象为以下数据。

- 保持属性的变量当前值
- 绝对编码器原点位置偏置

### 9-6-2 Sysmac Studio 变量 / 存储器备份功能的操作方法

Sysmac Studio 上的操作方法如下所示。

在在线连接状态下，从 [ 工具 ] 菜单中选择 [ 备份 ]-[ 变量 • 存储器的备份 ] 或 [ 变量 • 存储器的恢复 ]。

详情请参考 □□《Sysmac Studio Version 1 操作手册（SBCA-CN5-362）》。

### 9-6-3 NY 系列控制器的型号兼容性

关于 Sysmac Studio 变量 / 存储器备份功能，即使备份源和恢复目标的 NY 系列型号不同，也可恢复。

## 9-7 连接 EtherCAT 从站时的备份功能

EtherCAT 从站为 SD 存储卡备份功能、Sysmac Studio 控制器备份功能及 Sysmac Studio 备份导入功能的对象。

下面以连接 EtherCAT 从站时为例，介绍备份对象数据、控制器各状态的备份功能执行可否、恢复执行条件、各型号 EtherCAT 从站的注意事项。



### 参考

备份功能的对象为 EtherCAT 从站终端时，请参考□□「9-8 连接 EtherCAT 从站终端时的备份功能」(P.9-31)。

### 9-7-1 EtherCAT 从站的备份对象数据

EtherCAT 从站的备份对象数据如下所示。

设定	备份对象数据
EtherCAT 主机设定	型号名称、产品名称、从站数、PDO 通信周期、低效运行设定、所有从站加入等待时间、PDO 通信连续超时检测次数、修订版本检查基准、序列号检查基准
EtherCAT 从站设定	设备名称、型号名称、产品名称、修订版本、节点地址、有效 / 无效设定、序列号、PDO 映射设定、DC 有效、参考时钟功能、初始化参数设定

## 9-7-2 控制器在各种状态下的备份功能执行可否

控制器处于各种状态时，可否执行 EtherCAT 从站的备份、恢复、核对如下所示。

控制器的状态	执行可否		
	备份	恢复	核对
链接断开中	不可 <sup>*1</sup>	不可 <sup>*2</sup>	可 <sup>*3</sup>
主机状态不正确 <sup>*4</sup>	不可 <sup>*1</sup>	不可 <sup>*2</sup>	可 <sup>*3</sup>
网络信息与网络配置不一致 <sup>*5</sup>	不可 <sup>*1</sup>	不可 <sup>*2</sup>	可 <sup>*3</sup>
与备份时的网络配置不一致	可	不可 <sup>*2</sup>	可 <sup>*3</sup>
网络配置中存在无效设定的从站	实际配置中存在从站	可	可
	实际配置中不存在从站	可	可
网络配置中存在根据“脱离”指示变为脱离状态的从站	实际配置中存在从站	不可 <sup>*1</sup>	可 脱离状态的从站也在对象范围内
	实际配置中不存在从站	不可 <sup>*1</sup>	不可 <sup>*2</sup> 可 <sup>*3</sup>
从站初始化异常	不可 <sup>*1</sup>	不可 <sup>*2</sup>	可 <sup>*3</sup>

\*1. 事件日志中将记录备份执行失败（EtherCAT 从站）。

\*2. 事件日志中将记录恢复执行失败（EtherCAT 从站）。

\*3. 核对结果为不一致。

\*4. “从站节点地址重复”、“网络配置信息异常”、“网络配置异常”、“从站初始化异常”、“低效运行设定为“停止设定”时的网络配置核对异常”、“链接断开异常”。

\*5. “备份时和网络配置不完全一致（分支从站上的连接端口不同将视为不一致）”、“网络配置信息和网络配置不一致（分支从站上的连接端口不同将视为不一致）”。

### 9-7-3 EtherCAT 从站的恢复执行条件

为了恢复 EtherCAT 从站的备份数据，必须满足以下条件。

- 备份文件中存在 EtherCAT 从站的数据。
- 网络配置信息和恢复目标的实际网络配置一致。
- EtherCAT 从站中事先设定的修订版本值一致。但是，一致条件遵照备份文件中保存的“修订版本检查基准”。此外，即使“修订版本检查基准”设定为“不检查”，“设定修订版本” > “实际机器修订版本”时，也会变为不可恢复。此外，用户无法变更修订版本值。
- 备份文件中保存的“序列号核对的有无”设定为“核对”时，序列号一致。
- 通过硬件开关设定节点地址时，节点地址一致。



#### 使用注意事项

- 恢复执行完成后，请重新接通所有 EtherCAT 从站的电源。
- 恢复执行完成后，所有从站变为脱离状态。要解除对象从站的脱离状态，需要再次加入对象从站的指示。
- 备份文件中保存的“序列号核对的有无”设定为“核对”时，若更换 EtherCAT 从站的硬件，将无法恢复。此时，请通过 Sysmac Studio 变更网络配置并下载，然后传送从站参数，恢复原样。此外，用硬件开关设定节点地址时，请设定为与备份时相同。

## 9-7-4 备份功能的对象 EtherCAT 从站

备份功能的对象 EtherCAT 从站如下所示。使用时请了解注意事项。

EtherCAT 从站的机型	注意事项
NX 系列 EtherCAT 耦合器单元 NX-ECC	EtherCAT 从站终端上的安全控制单元不在备份、恢复、核对的对象范围内。 安全控制单元的设定数据导入 / 导出功能请参考 <a href="#">□□</a> 《NX 系列 安全控制单元 用户手册 (SGFM-CN5-710)》。
AC 伺服驱动器 R88D-1SN □□□ -ECT	*1
AC 伺服驱动器 (旋转型) R88D-KN □□□ -ECT	*1*2
AC 伺服驱动器 (直线型) R88D-KN □□□ -ECT-L	*1*2
变频器 3G3AX-MX2-ECT 3G3AX-RX-ECT	变频器的参数不在恢复对象范围内。请按照后述的 <a href="#">□□</a> 「变频器 3G3AX-MX2-ECT、3G3AX-RX-ECT 的参数写入方法」(P.9-30)，将来自 Sysmac Studio 的参数写入变频器中。 若不写入参数直接执行核对，变频器的 EtherCAT 从站核对结果文件中，可能出现“Result=Not matched”。
视觉传感器 FH-3 □□□□ FH-1 □□□□	传感器的设定数据 (场景数据、系统数据等) 不在备份、恢复、核对的对象范围内。 要将传感器的设定数据传送到外部文件或传送到传感器时，请从 Sysmac Studio 配置和设定编辑画面，执行 [工具]-[传感器数据]-[保存到文件] 或 [从文件中读取]。 详情请参考 <a href="#">□□</a> 《图像处理系统 FH/FZ5 系列 用户手册 (SDNB-CN5-712)》。
视觉传感器 FQ-M □□□□ -ECT FQ-M □□□□ -M-ECT	传感器的设定数据 (场景数据、系统数据等) 不在备份、恢复、核对的对象范围内。 要将传感器的设定数据传送到外部文件或传送到传感器时，请从 Sysmac Studio 配置和设定编辑画面，执行 [工具]-[传感器数据]-[保存到文件] 或 [从文件中读取]。 详情请参考 <a href="#">□□</a> 《视觉传感器 FQ-M 系列 定位专用视觉传感器 用户手册 (SDNB-CN5-706)》。
视觉传感器 FZM1- □□□ -ECT	传感器的设定数据 (场景数据、系统数据等) 不在备份、恢复、核对的对象范围内。 要将传感器的设定数据保存到 USB 存储器或写入到控制器时，请使用专用工具。 详情请参考 <a href="#">□□</a> 《视觉传感器 FZ3 系列 用户手册 (SCHB-CN5-762)》。
数字 I/O 终端 GX- □ D16 □□ GX- □ D32 □ 8 GX-OC1601	*1
模拟 I/O 终端 GX-AD0471 GX-DA0271	*1
编码器输入终端 GX-EC0211 GX-EC0241	—
EtherCAT 分支从站 GX-JC0 □	没有内置需要备份的数据。
光纤同轴位移传感器 ZW-CE1 □	所有设定数据不在备份、恢复、核对的对象范围内。 设定数据的保存、写入到控制器的方法请参考 <a href="#">□□</a> 《位移传感器 ZW 系列 光纤同轴位移传感器 用户手册 (SDNE-CN5-703)》。

EtherCAT 从站的机型	注意事项
数字型传感器 E3NW-ECT E3X-ECT	传感器的参数不在备份、恢复、核对的对象范围内。
其他公司从站	<ul style="list-style-type: none"> <li>• 只有在 ESI 中正确定义的数据是备份、恢复、核对的对象。要备份、恢复、核对未在 ESI 中定义的数据时，请使用专用工具。</li> <li>• 备份、恢复、核对失败时，请向从站的制造商咨询处理方法。</li> </ul>

- \*1. 恢复执行完成后，请重新接通从站的电源。恢复后进行核实时，请在重新接通从站的电源后再执行。若在重新接通电源前执行核对，将发生核对不一致。
- \*2. 符合以下任一条件时，从备份到核对期间，或从恢复到核对期间，请勿设为伺服 ON 状态。从备份到核对期间，或从恢复到核对期间，如果设为伺服 ON 状态，可能在核对前变更参数，发生核对不一致。
- 实时自动调谐模式设定为有效（对象 3002h 的值为 1 ~ 4 或 6）。
  - 适用过滤器设定为有效（对象 3200h 的值为 1 或 2）。

## 变频器 3G3AX-MX2-ECT、3G3AX-RX-ECT 的参数写入方法

变频器的参数不在恢复对象范围内。

要将备份的参数写入到变频器时，请对 Sysmac Studio 进行以下操作。详情请参考 [□□](#) 《Sysmac Studio Version 1 操作手册（SBCA-CN5-362）》。

- 1** 使用 Sysmac Studio 备份文件的导入功能，读取备份文件中的变频器参数。  
Sysmac Studio 的配置和设定的变频器参数选项卡中将显示变频器的参数。
- 2** 确认所读取参数的变频器型号和实际连接的变频器型号一致。
- 3** 使用配置和设定的变频器参数选项卡的传送 [ 计算机 → 驱动器 ] 功能，将参数写入到变频器中。



### 使用注意事项

使用变频器 3G3AX-MX2-ECT 的变频器模式选择（参数 No.b171）功能，写入参数前，请将所连接变频器的模式切换为备份时的模式。此外，要使模式切换变为有效，需要在模式切换操作后，进行变频器初始化。

## 9-8 连接 EtherCAT 从站终端时的备份功能

EtherCAT 从站终端为 SD 存储卡备份功能、Sysmac Studio 控制器备份功能及 Sysmac Studio 备份导入功能的对象。

下面以连接 EtherCAT 从站终端时为例，介绍备份对象数据、控制器各状态的备份功能执行可否、恢复执行条件。



### 使用注意事项

EtherCAT 从站终端上的安全控制单元不在备份、恢复、核对的对象范围内。安全控制单元的设定数据导入 / 导出功能请参考 [《NX 系列 安全控制单元 用户手册 \(SGFM-CN5-710\)》](#)。

### 9-8-1 EtherCAT 从站终端的备份对象数据

EtherCAT 从站终端的备份对象数据因 EtherCAT 耦合器单元和 NX 单元不同而异。各备份对象数据如下所示。

(○：对象 x：对象外)

单元	数据	备份	恢复	核对
EtherCAT 耦合器单元	构成信息 <sup>*1</sup>	○	○	○
	单元动作设定	○	○	○
NX 单元	构成信息 <sup>*1</sup>	○	○	○
	单元动作设定	○	○	○
	单元应用数据 <sup>*2</sup>	○	○	○

\*1. 单元构成信息及 I/O 分配信息的总称。

\*2. 各 NX 单元固有的数据。有的 NX 单元没有单元应用数据。



### 使用注意事项

将备份数据恢复到与备份源相同单元构成的 EtherCAT 从站终端时，请将硬件开关设定为与备份源相同的设定。硬件开关设定与备份数据不同时，无法恢复。将发生监视信息事件“恢复开始失败 (EtherCAT 从站)”。

### 9-8-2 EtherCAT 从站终端在各状态下的备份功能执行可否

EtherCAT 从站终端处于各种状态时，可否执行备份、恢复、核对如下所示。

EtherCAT 从站终端的状态	执行可否		
	备份	恢复	核对
单元构成信息自动生成时	可 <sup>*1</sup>	可 <sup>*2</sup>	可
等待 NX 单元加入	不可 <sup>*3</sup>	不可 <sup>*4</sup>	可 <sup>*5</sup>
EtherCAT 耦合器单元或 NX 单元发生看门狗定时器异常	不可 <sup>*3</sup>	不可 <sup>*4</sup>	可 <sup>*5</sup>
发生总线控制器异常	不可 <sup>*3</sup>	不可 <sup>*4</sup>	可 <sup>*5</sup>
发生单元构成信息异常	不可 <sup>*3</sup>	可	可 <sup>*5</sup>
发生单元构成核对异常	不可 <sup>*3</sup>	可	可 <sup>*5</sup>
与备份时的单元构成信息不一致	—	不可 <sup>*4</sup>	可 <sup>*5</sup>

- \*1. 将备份“单元构成信息不存在”这一信息。
- \*2. 恢复后，也变为单元构成自动生成的状态。
- \*3. 事件日志中将记录备份执行失败。
- \*4. 事件日志中将记录恢复执行失败。
- \*5. 核对结果为不一致。

### 9-8-3 EtherCAT 从站终端的恢复执行条件

为了恢复 EtherCAT 从站终端的备份数据，必须满足以下条件。

- 备份文件中存在 EtherCAT 耦合器单元及 NX 单元的数据。
- 备份源的单元构成信息和恢复目标的实际单元构成一致。
- 备份源的 EtherCAT 耦合器单元的序列号和恢复目标的 EtherCAT 耦合器单元的序列号一致。但是，仅限备份文件中的[通信耦合器单元] | [单元动作设定] | [序列号检查基准]的设定为“设定值=实际机器”时。
- 备份源的 NX 单元的序列号和恢复目标的 NX 单元的序列号一致。但是，仅限备份文件中的[通信耦合器单元] | [单元动作设定] | [序列号检查基准]的设定为“设定值=实际机器”时。
- 备份源的 EtherCAT 耦合器单元的硬件开关信息和恢复目标的 EtherCAT 耦合器单元的硬件开关设定完全一致。
- 备份源的 EtherCAT 耦合器单元的单元版本设定值和恢复目标的 EtherCAT 耦合器单元的实际单元版本一致或较旧。
- 备份源的 NX 单元的单元版本设定值和恢复目标的 NX 单元的实际单元版本一致或较旧。

## 9-9 备份相关文件

下面介绍备份相关文件的规格。

备份相关文件是指在“Sysmac Studio 变量 / 存储器的备份功能”以外的备份功能中使用的文件。

### 9-9-1 备份相关文件的种类

备份相关文件包括备份文件、恢复指令文件、自动传送指令文件、核对结果文件 4 种。

#### ● 备份文件

用备份相关功能中处理的控制器数据制作的文件。执行备份时创建。

#### ● 恢复指令文件

从虚拟 SD 存储卡执行恢复时，用于指定传送对象数据组的文件。在计算机的文本编辑器中编辑文件，可指定传送对象。执行备份时创建。

#### ● 自动传送指令文件

NY 系列控制器中不使用本文件。

#### ● 核对结果文件

该文件用于保存对控制器的数据和虚拟 SD 存储卡的备份文件数据进行核对的核对结果。

核对结果文件分为以下 3 种。均在执行 SD 存储卡备份功能中的核对时创建。

核对结果文件	内容
控制器核对结果文件	记述了恢复指令文件中指定的所有备份数据核对结果的文件。
EtherCAT 从站核对结果文件	记述了各 EtherCAT 从站核对结果的文件。恢复指令对象文件中的恢复对象为单元和从站设定，且备份文件中含有 EtherCAT 从站设定时创建。
EtherCAT 从站终端核对结果文件	<p>记述了 EtherCAT 耦合器单元及各 NX 单元核对结果的文件。同时满足以下 3 个条件时创建。</p> <ul style="list-style-type: none"> <li>• 恢复指令对象文件中的恢复对象为单元和从站设定。</li> <li>• 备份文件中包含 EtherCAT 从站设定。</li> <li>• 连接了 EtherCAT 从站终端。</li> </ul> <p>创建 EtherCAT 从站终端核对结果文件时，必定会同时创建 EtherCAT 从站核对结果文件。</p>

## 9-9-2 备份文件的规格

备份文件的文件名、创建时间、创建目录如下。

### 文件名

NY 系列控制器的备份文件的文件名如下。

文件	文件名
备份文件	NYBackup.dat

### 文件创建时间和创建目录

功能	操作方法	创建时间	创建目录
SD 存储卡的备份功能	系统定义变量	执行备份时	虚拟 SD 存储卡中，以系统定义变量指定的目录
	Sysmac Studio SD 存储卡画面	执行备份时	虚拟 SD 存储卡中，由用户在画面中指定的目录
	专用指令	执行备份时	虚拟 SD 存储卡中，由用户以 BackupToMemoryCard 指令的输入变量指定的目录
SysmacStudio 控制器备份功能	Sysmac Studio 控制器备份画面	执行备份时	计算机中，由用户在画面中指定的目录
Industrial PC Support Utility 控制器备份功能	Industrial PC Support Utility 控制器备份画面	执行备份时	工业电脑中，由用户在画面中指定的目录
SysmacStudio 备份文件的导入 / 导出功能	SysmacStudio 备份文件的导出画面	执行导出时	计算机中，由用户在画面中指定的目录

### 9-9-3 恢复指令文件的规格

恢复指令文件的文件名、创建时间、创建目录、数据组的指定方法如下。

#### 文件名

文件	文件名
恢复指令文件	RestoreCommand.ini

#### 文件创建时间和创建目录

功能	操作方法	创建时间	创建目录
SD 存储卡的备份功能	系统定义变量	执行备份时	与备份文件相同的目录
	Sysmac Studio SD 存储卡画面	执行备份时	与备份文件相同的目录
SysmacStudio 控制器备份功能	Sysmac Studio 控制器备份画面	执行备份时	与备份文件相同的目录
Industrial PC Support Utility 控制器备份功能	Industrial PC Support Utility 控制器备份画面	执行备份时	与备份文件相同的目录
SysmacStudio 备份文件的导入 / 导出功能	SysmacStudio 备份文件的导出画面	执行导出时	与备份文件相同的目录

## 要备份的数据组的指定方法

恢复指令文件是指按数据组进行恢复指定所需的文件。

在计算机的文本编辑器中，对按数据组进行指定的部分进行编辑，可以变更指定。

例如，将下表中“文件内容”第8行的“Variable=yes”变更为“Variable=no”后，“变量的当前值”将排除在恢复对象之外。

文件内容（创建文件时的默认内容）	说明
<pre>[Restore] ; --- User Program and Configuration. --- ; Always select "yes". UserProgram=yes  ; --- Present values of variables (Retained variables only). --- ; "yes":will be restored, "no":will not be restored Variable=yes  :---Unit/Slave Parameters.--- ; "yes";will be restored."no";will not be restored UnitConfig=yes  ; --- Absolute encoder home offset. --- ; "yes":will be restored, "no":will not be restored AbsEncoder=yes</pre>	<p>用户程序设定 yes 固定：必须恢复</p> <p>变量的当前值（仅保持属性） yes/no：恢复 / 不恢复</p> <p>单元和从站设定 yes/no：恢复 / 不恢复</p> <p>绝对编码器原点位置偏置 yes/no：恢复 / 不恢复</p>

(注) 1 本表的文件内容为创建恢复指令文件时的内容。记述的所有数据组均为恢复对象。

2 记述在创建备份文件时已在备份文件中，且指定了可恢复数据组的文件。

3 全部为半角英文和数字。不区分大小写。



### 使用注意事项

编辑恢复指令文件时，请只记述可选择的数据组的“yes/no”。若编辑其他记述后执行恢复，控制器可能发生意外动作。

## 9-9-4 控制器核对结果文件的规格

控制器核对结果文件的文件名、创建时间、创建目录、核对结果的确认方法如下。

### 文件名

文件	文件名
控制器核对结果文件	VerifyResult.log

### 文件创建时间和创建目录

功能	操作方法	创建时间	创建目录
SD 存储卡的 备份功能	Sysmac Studio SD 存储卡画面	执行核对时	与备份文件相同的目录
	系统定义变量	执行核对时	与备份文件相同的目录

### 核对结果的确认方法

核对结果文件用于保存按数据组对控制器的数据和虚拟SD存储卡的备份文件数据进行核对的核对结果。

在保存了各数据组确认结果的部分，可确认核对结果。

若结果为“Result=Matched”，则表示核对一致；若结果为“Result=Not matched”，则表示核对不一致。

文件内容为以下内容时，用户程序设定为核对一致，单元 / 从站设定为核对不一致。

文件内容	说明
[UserProgram] ; --- User Program and Configuration. --- Result=Matched	用户程序设定 Matched: 核对一致、 Not matched: 核对不一致
[UnitConfig] ; --- Unit/Slave Parameters. --- Result=Not matched	单元 / 从站设定 Matched: 核对一致、 Not matched: 核对不一致

(注) 仅记述执行了核对的数据组的核对结果。

### 9-9-5 EtherCAT 核对结果文件的规格

EtherCAT 核对结果文件的文件名、创建时间、创建目录、核对结果的确认方法如下。

#### 文件名

文件	文件名
EtherCAT 核对结果文件	VerifyResult_ECAT.log

#### 文件创建时间和创建目录

功能	操作方法	创建时间	创建目录
SD 存储卡的 备份功能	Sysmac Studio SD 存储卡画面	执行核对时	与备份文件相同的目录
	系统定义变量	执行核对时	与备份文件相同的目录

#### 核对结果的确认方法

核对结果文件用于保存按数据组对控制器的数据和虚拟 SD 存储卡的备份文件数据进行核对的核对结果。在保存了各 EtherCAT 从站确认结果的部分，可确认核对结果。

若结果为“Result=Matched”，则表示核对一致；若结果为“Result=Not matched”，则表示核对不一致。文件内容为以下内容时，核对结果如下。

- 核对一致 : EtherCAT 从站 Master、EtherCAT 从站终端 E002
- 核对不一致 : EtherCAT 从站 E001

文件内容	说明
[Verification Results] ; --- EtherCAT Parameters. --- : --- See the VerifyResult_ECAT_NX.log about detail result if NX mark is included in square brackets.  [Master] Result=Matched  [E001] Result=Not matched Factor=Verification error  [E002:NX] Result=Matched	从站以用户设定的设备名称表示。 EtherCAT 从站终端时，设备名称后面带 ":NX"。 <sup>*1</sup>  核对结果用以下表述表示。 Result=Matched      核对一致 Result=Not matched      核对不一致

\*1. EtherCAT 从站终端为核对对象时，将创建 EtherCAT 从站终端核对结果文件。EtherCAT 从站终端的详细核对结果记载于 EtherCAT 从站终端核对结果文件中。

(注) 仅记述执行了核对的 EtherCAT 从站的核对结果。

## 9-9-6 EtherCAT 从站终端核对结果文件的规格

EtherCAT 从站终端核对结果文件的文件名、创建时间、创建目录、核对结果的确认方法如下。

### 文件名

文件	文件名
EtherCAT 从站终端核对结果文件	VerifyResult_ECANT_NX.log

### 文件创建时间和创建目录

功能	操作方法	创建时间	创建目录
SD 存储卡的备份功能	Sysmac Studio SD 存储卡画面	执行核对时	与备份文件相同的目录
	系统定义变量	执行核对时	与备份文件相同的目录

### 核对结果的确认方法

核对结果文件用于保存按数据组对控制器的数据和虚拟SD存储卡的备份文件数据进行核对的核对结果。在保存了 EtherCAT 耦合器单元及各 NX 单元确认结果的部分，可确认核对结果。

若结果为“Result=Matched”，则表示核对一致；若结果为“Result=Not matched”，则表示核对不一致。文件内容为以下内容时，核对结果如下。

- 核对一致 : EtherCAT 耦合器单元 E002、NX 单元 N1、NX 单元 N2
- 核对不一致 : EtherCAT 耦合器单元 E005、NX 单元 N3

文件内容	说明
[Verification Results] ; --- NX Parameters. ---	单元按以下规格表述。 {设备名称};UnitNo.{单元编号}[空格]{单元型号}
[E002:UnitNo.0 NX-ECC201] Result=Matched	{设备名称} 用户设定的设备名称。
[N1:UnitNo.1 NX-AD2203] Result=Matched	{单元编号} 10 进制数的字符串。值的范围为 0-125。
[N2:UnitNo.2 NX-DA2203] Result=Matched	{单元型号} 表示单元型号的字符串。 从型号后面开始的连续空格将删除。
[N3:UnitNo.3 NX-TS3201] Result=Not matched Factor=Verification error	核对结果用以下表述表示。 Result=Matched 核对一致 Result=Not matched 核对不一致
[E005:UnitNo.0 NX-ECC201] Result=Not matched Factor=Verification error	

## 9-10 备份功能的文件间兼容性

若执行备份时和执行恢复时的条件不同，需要考虑文件间的兼容性。

以下 3 种情况下需要考虑兼容性。

- 备份的功能和恢复功能不同时
- 备份源和恢复目标的控制器型号不同时
- 备份源和恢复目标的控制器或单元、从站的单元版本不同时

此外，这里我们将各备份功能的“恢复”、“自动传送”、“读取”统称为“恢复”。

### 9-10-1 备份功能间的兼容性

备份的功能和恢复功能不同时，文件的兼容性如下。

(○：有兼容性、×：无兼容性)

备份的功能	恢复功能				
	SD 存储卡备份功能（恢复） 从 SD 存储卡到控制器	Sysmac Studio 控制器备份功能（恢复） 从计算机到控制器	Industrial PC Support Utility 控制器备份功能（恢复） 从 Windows 到控制器	Sysmac Studio 变量 / 存储器的备份功能（恢复） 从计算机到控制器	Sysmac Studio 备份文件的导入 从计算机到项目
SD 存储卡的备份功能（备份） 从控制器到 SD 存储卡	○	○	○	×	○ <sup>*1</sup>
Sysmac Studio 控制器备份功能（备份） 从控制器到计算机	○	○	○	×	○ <sup>*1</sup>
Industrial PC Support Utility 控制器备份功能（备份） 从控制器到 Windows	○	○	○	×	○ <sup>*1</sup>
Sysmac Studio 变量 / 存储卡的备份功能（备份） 从控制器到计算机	×	×	×	○	×
Sysmac Studio 备份文件的导出 从项目到计算机	○ <sup>*1</sup>	○ <sup>*1</sup>	○ <sup>*1</sup>	×	○

\*1. 但是，以下数据不在对象范围内。

- 控制器设定的内置 EtherNet/IP 端口名称、内置 EtherNet/IP 用标签数据链接设定
- 操作权限认证
- 数据追踪设定
- 变量的当前值
- 绝对编码器原点位置偏置



## 参考

Sysmac Studio 变量 / 存储器的备份功能中处理的文件与其他备份功能中处理的文件不同，因此不兼容。

本功能的详情请参考□□「9-6 Sysmac Studio 变量 / 存储器的备份功能」(P.9-25)。

## 9-10-2 NY 系列控制器的型号兼容性

备份源和备份目标的控制器型号不同时，文件的兼容性如下。

(○：有兼容性、×：无兼容性)

备份源的控制器型号	备份目标的控制器型号
	NY5□2-1500 NY5□2-1400 NY5□2-1300

即使控制器的型号兼容时，各控制器型号之间仍可能有所限制。

因控制器型号间兼容性的有无，恢复功能的动作如下所示。

(○：执行恢复、×：不执行恢复)

控制器的型号兼容性	恢复功能				
	SD 存储卡备份功能 (恢复) 从 SD 存储卡到控制器	Sysmac Studio 控制器备份功能 (恢复) 从计算机到控制器 <sup>*1</sup>	Industrial PC Support Utility 控制器备份功能 (恢复) 从 Windows 到控制器	Sysmac Studio 变量 / 存储器的备份功能 (恢复) 从计算机到控制器	Sysmac Studio 备份文件的导入功能 从计算机到项目
有兼容性	○ <sup>*2</sup>	○	○	○	○
无兼容性	× <sup>*3</sup>	×	×	×	×

\*1. 仅通过本功能备份的文件可以恢复。

\*2. 备份文件的内容超出恢复目标的规格范围时，将无法正常工作，并发生全部停止故障等级的控制器异常或部分停止故障等级的控制器异常。例如，使用的控制轴数超出规格时，动作如下。

\*3. 将发生监视信息“恢复开始失败”。

## 9-10-3 NY 系列控制器的单元版本兼容性

备份源和恢复目标的控制器单元版本不同时，备份文件的兼容性如下。

此外，备份、恢复前后单元版本相同时，可无限制执行恢复。

(○：执行恢复、×：不执行恢复)

控制器的单元版本	恢复功能			
	SD 存储卡备份功能（恢复） 从 SD 存储卡到 控制器	Sysmac Studio 控制器备份 功能（恢复） 从计算机到 控制器	Industrial PC Support Utility 控制器备份 功能（恢复） 从 Windows 到 控制器	Sysmac Studio 变量 / 存储器的备份 功能（恢复） 从计算机到 控制器
恢复目标较新	○	○	○	○
恢复目标较旧	×	×	×	○

## 9-11 无法与备份功能同时执行的功能

以下功能无法与备份功能同时执行。控制器正在执行这些功能时，请勿执行备份功能。此外，正在执行备份功能时，请勿执行这些功能。

- 正在执行备份功能
- 同步传送（计算机→控制器）
- 执行在线编辑
- 执行存储器全部清除
- 执行凸轮表保存指令（MC\_SaveCamTable）
- 执行 CPU 单元名称写入
- 执行从站终端的参数设定传送



# 通信设定

本章介绍 NY 系列控制器的在线连接以及与其他设备的连接。

---

<b>10-1 通信系统的概要</b> .....	<b>10-2</b>
10-1-1 概要 .....	10-3
<b>10-2 Sysmac Studio 的连接</b> .....	<b>10-4</b>
10-2-1 可在线连接的构成 .....	10-4
10-2-2 无法在线连接的构成 .....	10-5
<b>10-3 与其他控制器或从站连接</b> .....	<b>10-6</b>
10-3-1 控制器与控制器之间的连接构成 .....	10-6
10-3-2 控制器与从站的连接构成 .....	10-9
<b>10-4 与触摸屏的连接</b> .....	<b>10-10</b>

## 10-1 通信系统的概要

下面介绍 NY 系列的通信系统概要。

在整体步骤中，与以下阴影标示的位置对应。

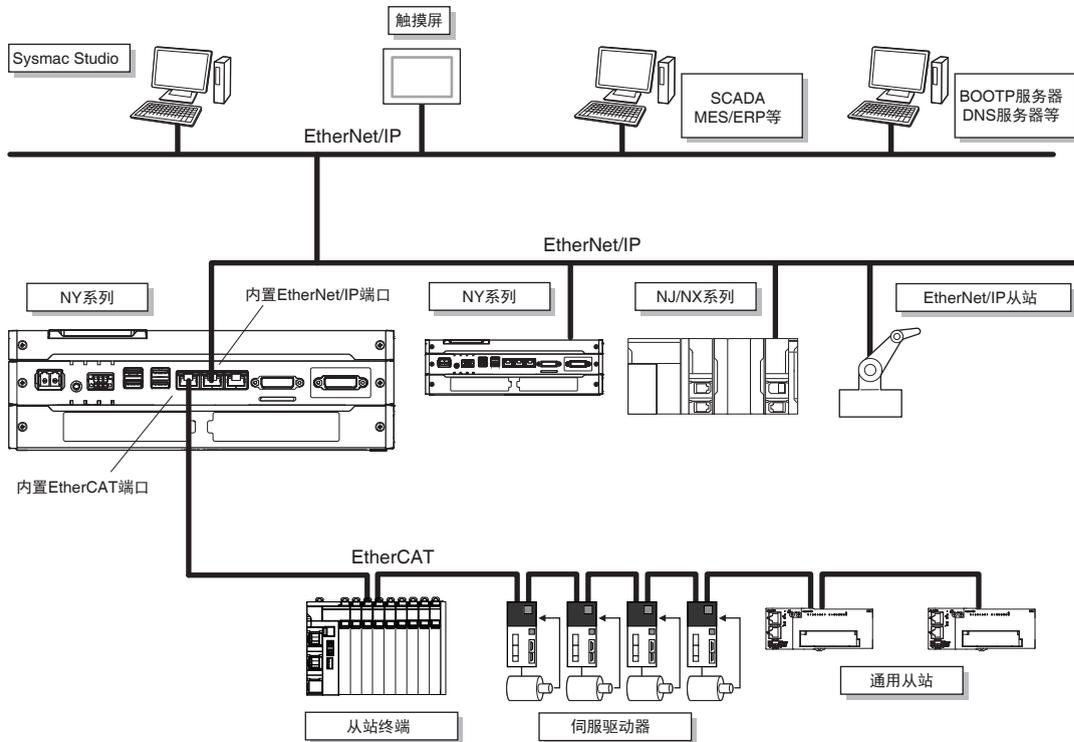


详情请参考□□「1-4 NY 系列的整体使用步骤」(P.1-10)。

## 10-1-1 概要

## ● NY 系列控制器系统

NY 系列的控制器功能可构建以下通信系统。



连接对象		连接方法
Sysmac Studio 的连接		使用内置 EtherNet/IP 端口
控制器之间的连接	与 NJ/NX/NY 系列控制器连接	使用内置 EtherNet/IP 端口
控制器与从站的连接	连接伺服驱动器、通用从站	使用内置 EtherCAT 端口
触摸屏的连接		使用内置 EtherNet/IP 端口
与各种系统连接	连接 SCADA、MES/ERP	使用内置 EtherNet/IP 端口
与各种服务器的连接	FTP 服务器、BOOTP 服务器、DNS 服务器	使用内置 EtherNet/IP 端口



## 参考

关于 Windows 侧可使用的通信系统，请参考《搭载机器自动化控制软件的 NY 系列工业平板电脑 / 工业 BOX PC 用户手册 安装篇 (SBCA-CN5-441)》。

## 10-2 Sysmac Studio 的连接

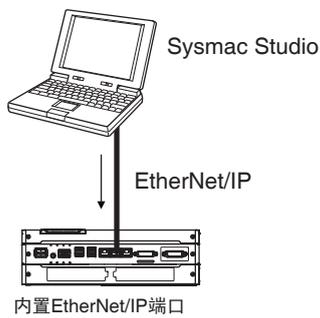
下面介绍 Sysmac Studio 与 NY 系列的连接构成。

### 10-2-1 可在线连接的构成

Sysmac Studio 可通过 NY 系列的内置 EtherNet/IP 端口，进行在线连接。

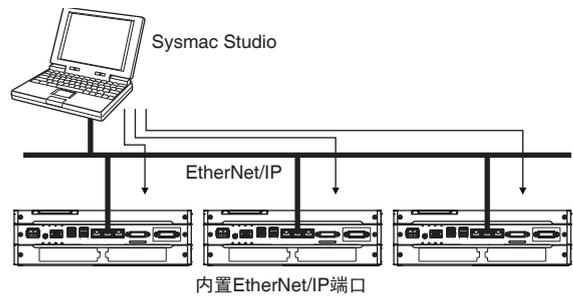
#### ● 用 EtherNet/IP 连接

1: 1 的连接:



- 通过 Sysmac Studio 直接连接。无需指定 IP 地址、连接设备。
- 无论有无交换式集线器均可连接。
- 可使用交叉电缆、直通电缆中的任意一种。

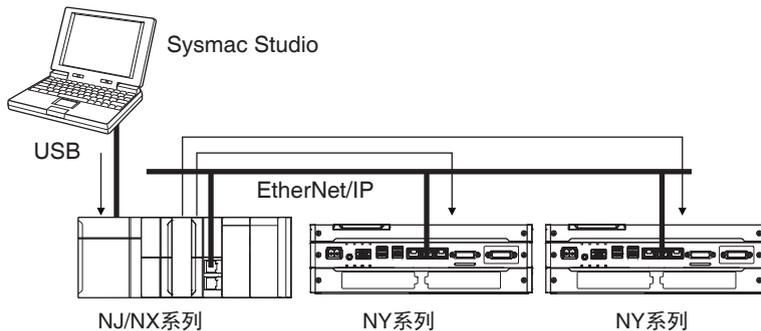
1: N 的连接:



- 通过 Sysmac Studio 指定连接对象的 IP 地址。
- 可使用交叉电缆、直通电缆中的任意一种。

#### ● 通过 USB 用 EtherNet/IP 连接

可通过 NJ/NX 系列的 USB 端口，与 NY 系列内置 EtherNet/IP 端口连接。



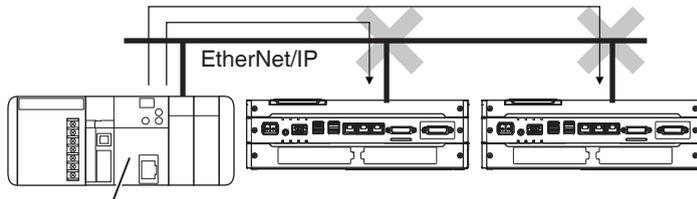
- 通过 Sysmac Studio 指定连接对象的 IP 地址。
- 可使用交叉电缆、直通电缆中的任意一种。

(注) 计算机和 NJ/NX 系列 CPU 单元之间请使用 USB2.0 标准的电缆，不可经过 USB 集线器连接。

## 10-2-2 无法在线连接的构成

- 对 CS/CJ 系列的 EtherNet/IP 单元 / 端口进行路由

不可对 CS/CJ 系列的 EtherNet/IP 单元 / 端口（CS1W-EIP21、CJ1W-EIP21、CJ2 内置 EtherNet/IP 端口、CJ2M 内置 EtherNet/IP 端口）进行路由后连接到 NY 系列控制器。



CJ2内置 EtherNet/IP、EtherNet/IP单元

## 10-3 与其他控制器或从站连接

下面介绍控制器与控制器、控制器与从站之间的连接构成。

### 10-3-1 控制器与控制器之间的连接构成

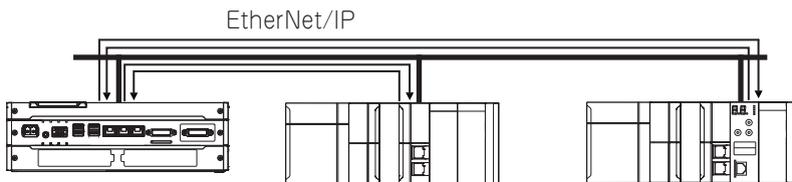
#### EtherNet/IP

使用内置 EtherNet/IP 端口。

关于内置 EtherNet/IP 端口，请参考 □ 《NY 系列 工业平板电脑 / 工业 BOX PC 用户手册 内置 EtherNet/IP 端口篇（SBCD-CN5-369）》。

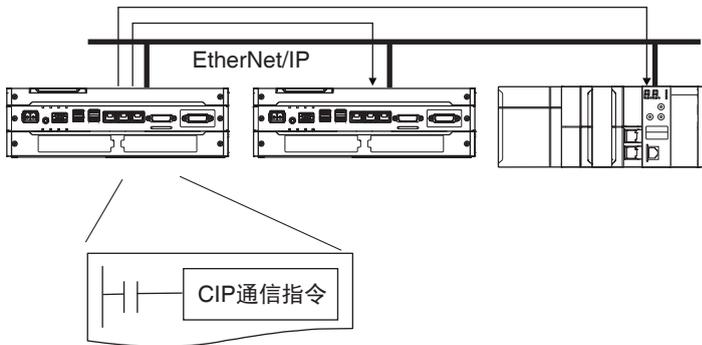
#### ● 标签数据链接

可与 EtherNet/IP 上的 NJ/NX/NY 系列控制器进行标签数据链接。



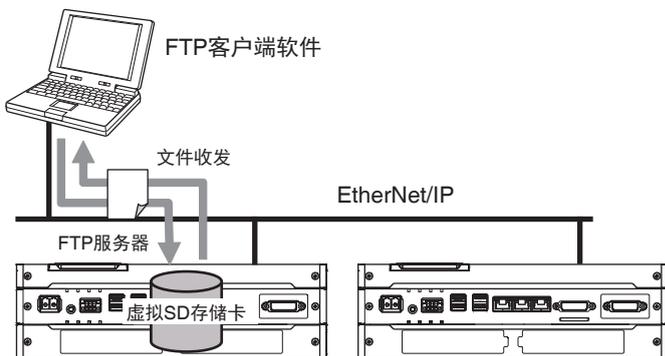
#### ● 信息通信

可通过程序发行 CIP 信息。



### ● 文件收发

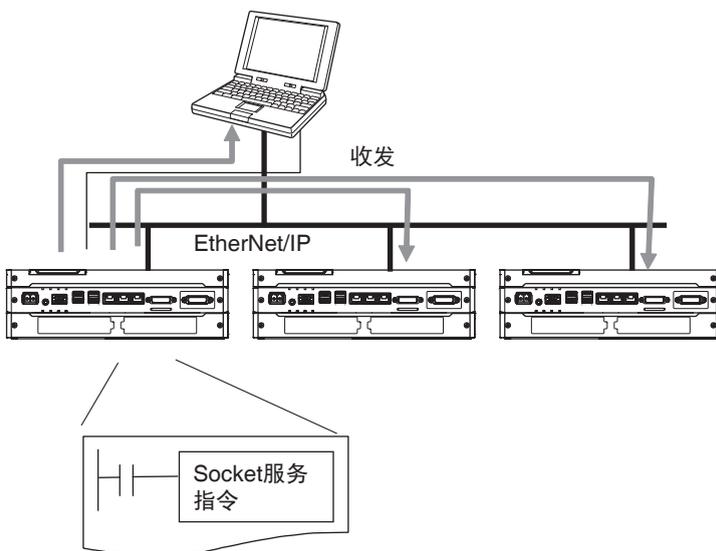
可通过 FTP 客户端软件，对 NY 系列控制器的虚拟 SD 存储卡中的文件进行收发。



### ● Socket 服务

直接通过用户程序使用 TCP 或 UDP 的功能，可在主机 PC — PLC 之间或控制器之间与对象节点进行任意数据的收发。

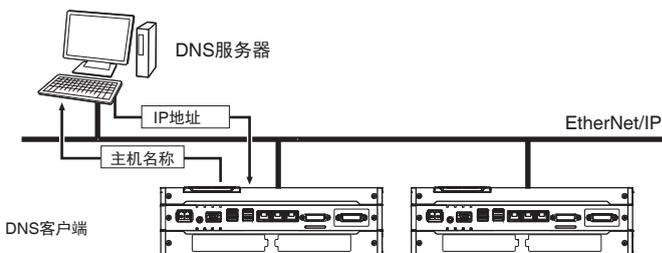
本功能只能在内置 EtherNet/IP 端口上使用。



### ● 以主机名称指定

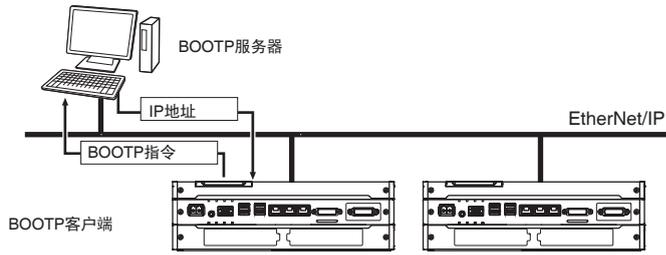
对于 SNMP 管理器的 IP 地址或 Socket 指令、CIP 通信指令的接收方，可用主机名称指定，而不是 IP 地址（DNS 客户端功能或 Hosts 设定）。

例：在 DNS 服务器上设定主机名称



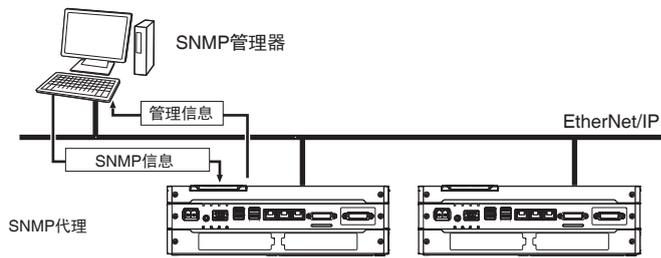
### ● 电源接通时 IP 地址的获取

电源接通时，可从 BOOTP 服务器获取内置 EtherNet/IP 端口的 IP 地址。



### ● SNMP 代理指定

向使用 SNMP 管理器的网络管理软件提供内置 EtherNet/IP 端口内部状态信息。

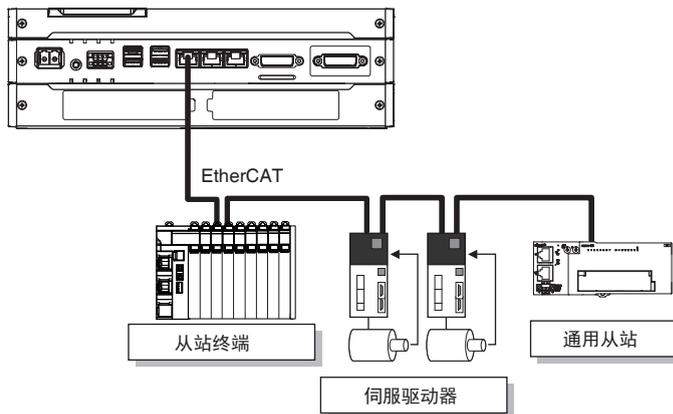


## 10-3-2 控制器与从站的连接构成

### EtherCAT

可与伺服驱动器、通用从站进行高速、高精度通信。

详情请参考《NY 系列 工业平板电脑 / 工业 BOX PC 用户手册 内置 EtherCAT 端口篇 (SBCD-CN5-368)》。

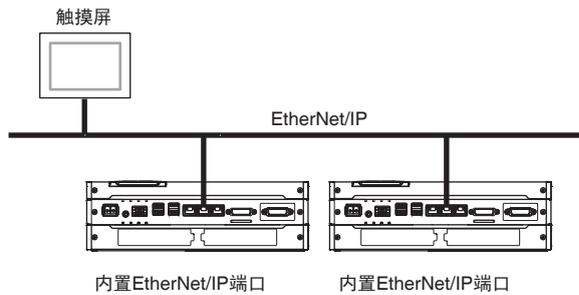


## 10-4 与触摸屏的连接

下面介绍 NS/NA 系列等触摸屏与 NY 系列控制器的连接构成。

### ● EtherNet/IP

使用内置 EtherNet/IP 端口。



关于内置 EtherNet/IP 端口，请参考□□《NY 系列 工业平板电脑 / 工业 BOX PC 用户手册 内置 EtherNet/IP 端口篇（SBCD-CN5-369）》。

# 实际使用步骤示例

本章介绍实际使用 NY 系列控制器时的使用步骤示例。

---

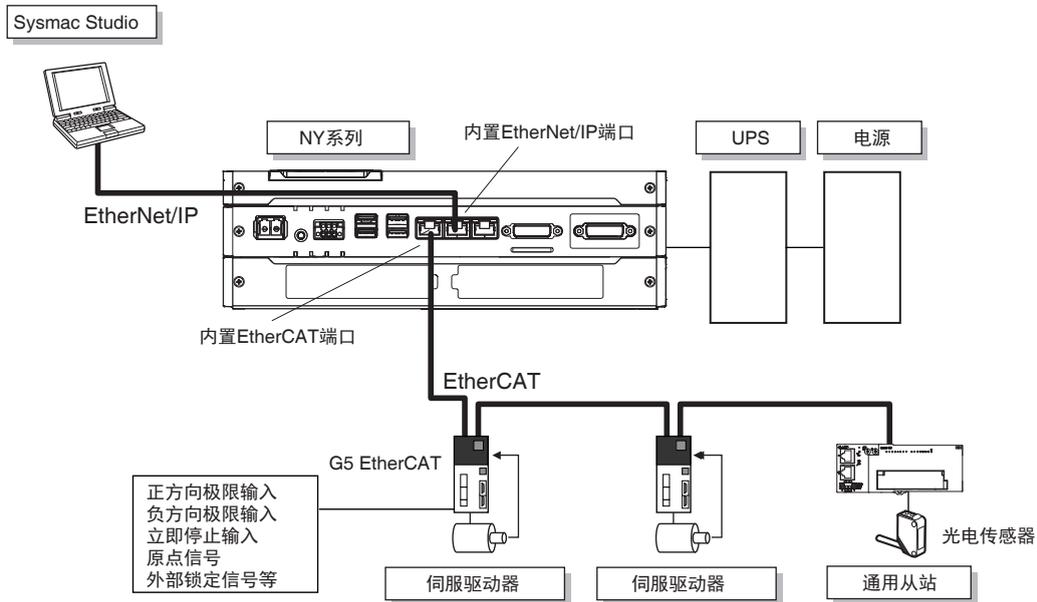
<b>11-1 实际使用示例</b> .....	<b>11-2</b>
11-1-1 系统构成 .....	11-2
11-1-2 动作 .....	11-2
<b>11-2 操作顺序示例的概要</b> .....	<b>11-3</b>
11-2-1 配线和设定 .....	11-3
11-2-2 软件设计 .....	11-3
11-2-3 Sysmac Studio 上的软件设定 .....	11-4
11-2-4 Sysmac Studio 上的编程 .....	11-8
11-2-5 上机确认动作和运行 .....	11-9

# 11-1 实际使用示例

下面介绍 NY 系列的实际使用示例。

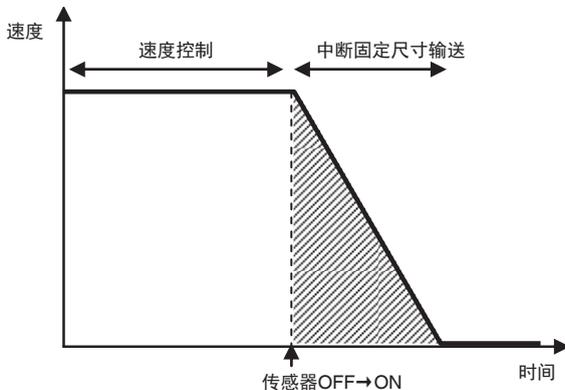
## 11-1-1 系统构成

单元名称	台数	连接设备
电源	1 台	—
UPS（不间断电源装置）	1 台	—
NY 系列 工业 BOX PC	1 台	—
EtherCAT 从站	伺服驱动器 G5EtherCAT 型	2 台
	I/O 终端	1 台



## 11-1-2 动作

速度控制过程中，传感器信号从 OFF 变为 ON，进行中断固定尺寸输送。



根据光电传感器的输入，变更纵向的位置。

## 11-2 操作顺序示例的概要

下面介绍 NY 系列的实际操作步骤示例。

### 11-2-1 配线和设定

进行配线及硬件的设定。

### 11-2-2 软件设计

进行 I/O 的设计、任务设计、POU 设计及变量设计。

#### I/O 的设计

- 设计外部 I/O 和单元的构成关系。
- 设定以多少间隔刷新哪个外部 I/O。

#### 任务及 POU 的设计

研究以下内容。

- 如何设计任务构成
- 将哪个程序分配到哪种周期的任务中
- 将哪个单元分配到哪个任务中
- 将哪个处理编入程序，将哪个处理设计为功能块、函数

#### 设计变量

研究以下内容。

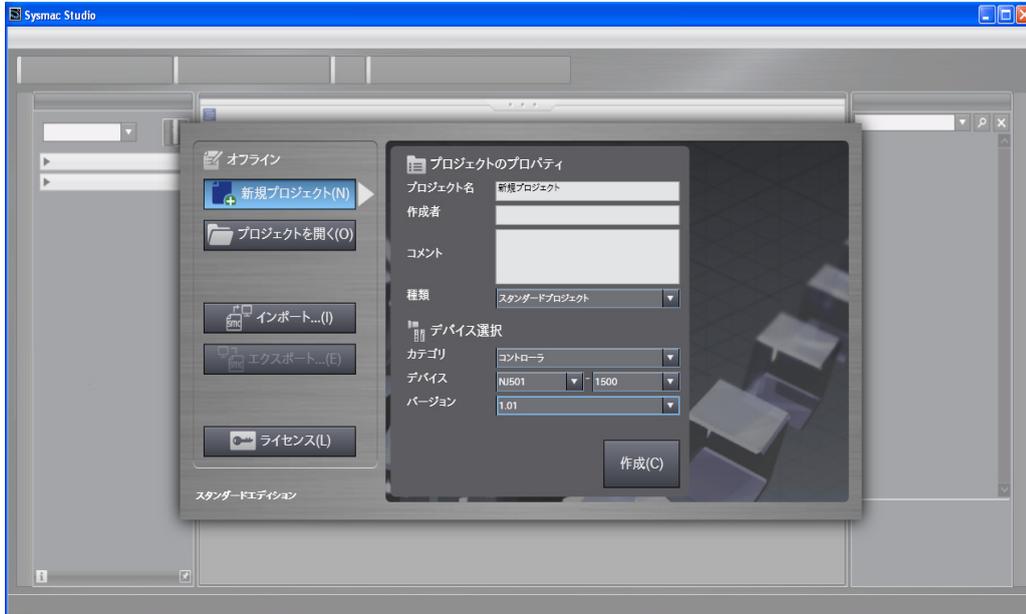
- 区分 POU 共同使用的变量（全局变量）和各 POU 单独使用的变量（本地变量）
- 设计用于访问从站的设备变量之变量名称
- 设计变量的名称、保持等属性
- 设计变量的数据类型

### 11-2-3 Sysmac Studio 上的软件设定

在 Sysmac Studio 上, 进行从站构成、全局变量 / 设备变量的登录、轴 (轴变量) 的创建及控制器设定。

## 启动 Sysmac Studio

新建项目。



## 构成 EtherCAT 从站

- 1 双击 [配置和设定] 下的 [EtherCAT]。
- 2 拖放从站，创建从站构成。



- 3 选择主机，设定主机参数。
- 4 选择各从站，设定从站的参数。



### 参考

此时，可通过 I/O 映射进行强制值刷新，以进行配线确认。

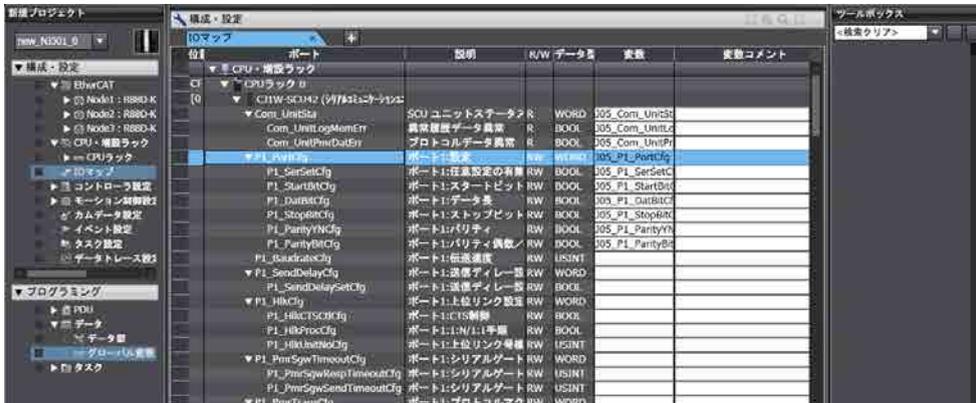
## 登录全局变量 / 设备变量。

### ● 全局变量的登录

- 1 双击 [编程] - [数据] 下的 [全局变量]。
- 2 将全局变量登录到全局变量表中。

### ● 设备变量的登录

- 1 双击 [配置和设定] 下的 [IO 映射]。
- 2 利用 I/O 映射，对（根据从站构成自动创建的）I/O 端口分配变量。  
此外，还可通过 Sysmac Studio 自动生成设备变量名称。此时，选择 I/O 端口并右击，选择 [生成设备变量]。

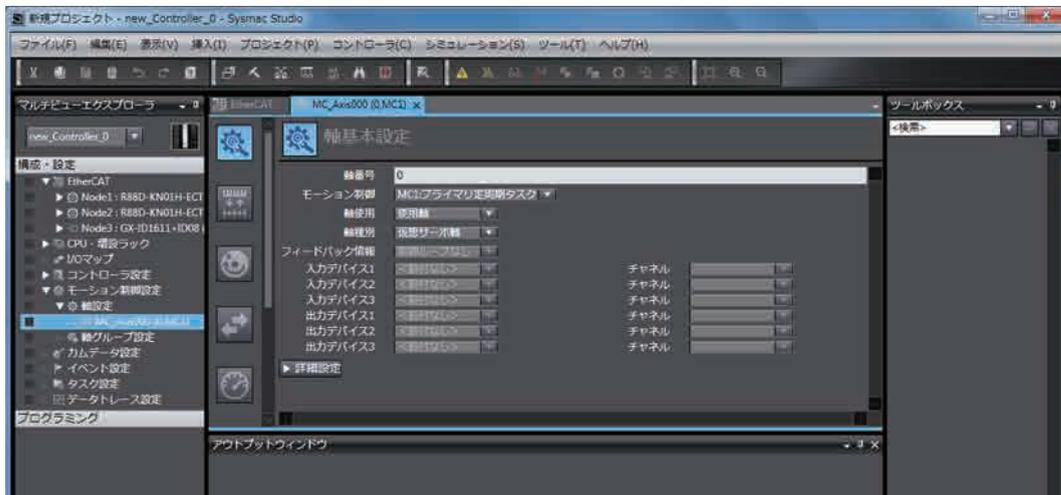


默认情况下，生成的设备变量将登录到全局变量表中。

根据需要，将 [变量种类] 从“全局变量”更改为各 POU 的本地变量（内部变量）。

## 创建“轴”（轴变量）。

- 1 选择 [ 配置和设定 ] 下的 [ 运动控制设定 ]-[ 轴设定 ] 并右击，选择 [ 添加 ]-[ 轴设定 ]。
- 2 对创建的轴（轴变量）分配 EtherCAT 从站上的伺服驱动器。



- 将 [ 轴使用 ] 选择为 [ 使用轴 ]。
  - [ 轴种类 ] 选择为 [ 伺服轴 ]。
  - [ 输入设备 ]、[ 输出设备 ] 选择为从站构成中登录的 EtherCAT 从站。
- 此外，设定单位转换、动作设定等轴参数。

## 设定控制器

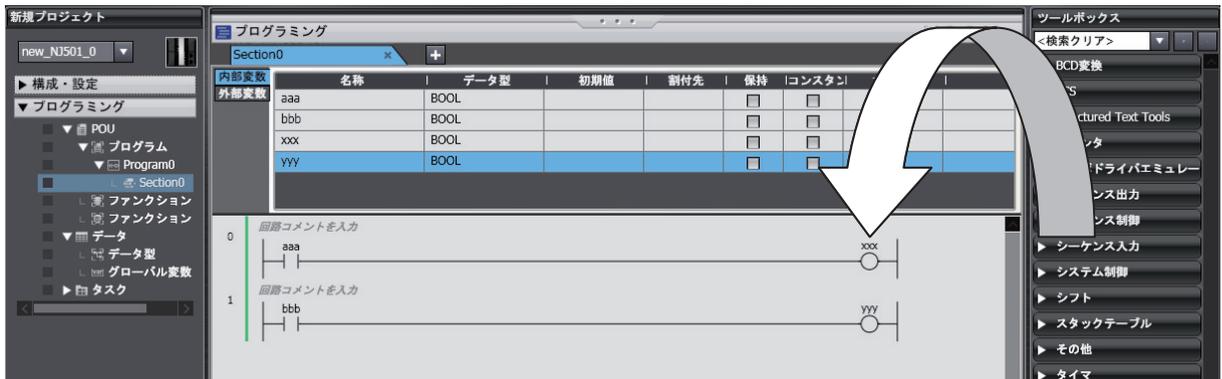
- PLC 模块相关的初始设定：  
控制器设定：电源接通时模式等

## 11-2-4 Sysmac Studio 上的编程

通过 Sysmac Studio 进行程序创建、任务设定和链接。

### 编程

- 1 选择 [编程]-[POU] 下的 [程序] 并右击, 选择 [添加]-[梯形图] 或 [ST]。
- 2 双击所创建“程序”下的“Section□”。
- 3 登录各程序的本地变量。
- 4 创建程序。



创建记述有以下指令的程序。

- 定原点: MC\_Home 指令
- 速度控制: MC\_MoveVelocity 指令
- 中断固定尺寸输送: MC\_MoveFeed 指令
- 定位: MC\_Move 指令

- 5 根据需要, 选择 [编程]-[POU] 下的 [函数] 或 [功能块], 选择 [添加]-[梯形图] 或 [ST]。双击创建的“Function□”或“FunctionBlock□”。登录各函数或功能块的本地变量。创建各算法。  
(注) 梯形图时, 按 [R] 键, 创建以下回路。

### 任务设置

双击 [配置和设定] 下的 [任务设置]。

- 在 [任务设置] 中, 从下拉菜单中选择主固定周期任务的 [周期 / 执行条件] 的任务周期。
- 在 [控制 I/O 的任务设置] 中, 选择分配各单元 / 从站的任务名称。
- 在 [程序的分配设定] 中, 将程序分配到主固定周期任务或固定周期任务 (执行优先级 16) 中。

### 链接

从 [项目] 菜单中选择 [链接]。

## 11-2-5 上机确认动作和运行

在线连接后，进行项目的下载、配线确认和调试，再进行正式运行。

### 在线连接

- 1 接通 NY 系列控制器的电源。
- 2 通过 EtherNet/IP 连接电脑和 NY 系列控制器。
- 3 从 [ 控制器 ] 菜单中选择 [ 通信设置 ]。从 [ 连接方法 ] 中选择与连接形态对应的连接方法。
- 4 从 [ 控制器 ] 菜单中选择 [ 在线 ]。

### 项目的下载（利用“同步”功能）

从 [ 控制器 ] 菜单中选择 [ 同步 ]，将项目下载到控制器中。

（注）“下载”和“上传”时使用 Sysmac Studio 的“同步”功能。

### 配线确认

通过 I/O 映射或梯形图编辑器对实际 I/O 进行强制值刷新，确认配线。

### MC 调试

- 1 打开 MC 调试画面。
- 2 将 NY 系列控制器设定为程序模式。
- 3 通过画面监视输入信号，检查配线。
- 4 通过画面进行点动进给。



### 手动运行

---

将 NY 系列控制器设定为运行模式。

- 伺服 ON/OFF：执行运动控制指令的 MC\_Power
- 点动进给：执行运动控制指令的 MC\_MoveJog

### 定原点

---

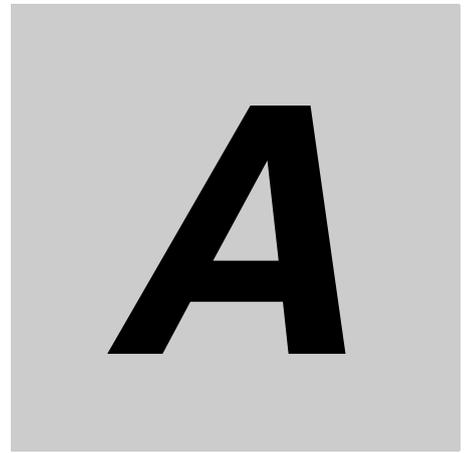
定原点：执行 MC\_Home 指令

### 正式运行

---

从 [ 控制器 ] 菜单中选择 [ 动作模式 ]-[ 运行模式 ]。

发生异常时，调查原因，修正用户程序。



# 附录



本章介绍 NY 系列控制器的详细规格、任务执行处理时间、系统定义变量的个别规格等文本的补充信息。

<b>A-1 规格</b>	<b>A-3</b>
A-1-1 性能规格	A-3
A-1-2 功能规格	A-6
<b>A-2 NY 系列中任务实际处理时间的计算标准</b>	<b>A-12</b>
A-2-1 任务实际处理时间平均值的计算方法	A-13
A-2-2 任务实际处理时间平均值的计算和任务周期的设定示例	A-19
<b>A-3 系统定义变量一览</b>	<b>A-22</b>
A-3-1 NY 系列控制器的系统整体（类别名称：无）	A-22
A-3-2 PLC 功能模块（类别名称：_PLC）	A-30
A-3-3 运动控制功能模块（类别名称：_MC）	A-31
A-3-4 EtherCAT 主机功能模块（类别名称：_EC）	A-32
A-3-5 EtherNet/IP 功能模块（类别名称：_EIP）	A-36
A-3-6 异常状态中各个位的含义	A-44
<b>A-4 系统定义变量的个别规格</b>	<b>A-45</b>
A-4-1 NY 系列控制器的系统整体（类别名称：无）	A-46
A-4-2 PLC 功能模块（类别名称：_PLC）	A-57
A-4-3 运动控制功能模块（类别名称：_MC）	A-59
A-4-4 EtherCAT 主机功能模块（类别名称：_EC）	A-61
A-4-5 EtherNet/IP 功能模块（类别名称：_EIP）	A-69
<b>A-5 控制器内数据的属性</b>	<b>A-79</b>
<b>A-6 变量的存储器确保方法</b>	<b>A-82</b>
A-6-1 变量的存储器确保规则	A-82
A-6-2 应注意的案例	A-91
<b>A-7 CX-Designer 的变量表登录方法</b>	<b>A-94</b>
<b>A-8 EtherCAT 从站或轴的有效 / 无效切换</b>	<b>A-97</b>
A-8-1 使用 EtherCAT 从站和轴时的项目设定	A-97
A-8-2 执行 EtherCAT 从站或轴的有效 / 无效切换的指令	A-97
A-8-3 表示 EtherCAT 从站或轴状态的系统定义变量	A-98
A-8-4 程序的执行 / 停止指令	A-99
A-8-5 确认程序的执行 / 停止	A-99
A-8-6 Sysmac Studio 上的设定	A-100

A-8-7 EtherCAT 从站或轴的有效 / 无效切换应用示例 .....A-101

**A-9 用户程序的大小相关的限制 .....A-104**

A-9-1 用户程序的各部件限制 .....A-104

A-9-2 用户程序的各部件数量 .....A-106

**A-10 NY 系列的版本相关信息 .....A-108**

# A-1 规格

下面介绍搭载机器自动化控制软件的 NY 系列工业平板电脑 / 工业 BOX PC 的控制功能之性能规格和功能规格。

关于一般规格，请参考□□《NY 系列 工业平板电脑 用户手册 硬件篇（SBCA-CN5-435）》、□□《NY 系列 工业 BOX PC 用户手册 硬件篇（SBCA-CN5-434）》。

## A-1-1 性能规格

下面介绍 NY 系列工业平板电脑 / 工业 BOX PC 的控制功能之性能规格。

项目			NY5□2-			
			15□□	14□□	13□□	
处理时间	指令执行时间	LD 指令	0.33ns ~			
		算术指令（双精度实数型）	1.2ns ~			
编程	程序容量 <sup>*1</sup>	大小	40MB			
		数量	POU 定义数	3,000		
			POU 实例数	24,000		
	变量容量	有保存属性	大小	4MB		
			变量数量	40,000		
		无保存属性	大小	64MB		
	变量数量	180,000				
	数据类型	数据类型的数量	4,000			
运动控制	控制轴数	最大控制轴数 <sup>2</sup>	64 轴	32 轴	16 轴	
		最大使用实轴数 <sup>3</sup>	64 轴	32 轴	16 轴	
		最大单轴控制数	64 轴	32 轴	16 轴	
		直线插补控制最大数	每 1 轴组 4 轴			
		圆弧插补控制轴数	每 1 轴组 2 轴			
		轴组最大数	32 组			
		运动控制周期	与 EtherCAT 通信的过程数据通信周期相同			
	凸轮	凸轮数据点数	每 1 凸轮表的 最大点数	65,535 点		
			所有凸轮表的 最大点数	1,048,560 点		
			凸轮表最大表数	640 表		
	位置单位	脉冲、mm、μm、nm、degree、inch				
	超驰	0.00、0.01 ~ 500.00%				

项目		NY5□2-			
		15□□	14□□	13□□	
内置 EtherNe t/IP 端口	端口数量	1			
	物理层	10BASE-T/100BASE-TX/1000BASE-T			
	帧长度	最大 1,514 字节			
	介质存取方式	CSMA/CD			
	调制方式	基带			
	拓扑结构	星形			
	传送速度	1Gb/s (1000BASE-T)			
	传送介质	双绞线电缆 (带屏蔽: STP): 类别 5、5e 以上			
	传输距离 (集线器和节点之间的距离) 最大值	100m			
	串联连接最大数量	使用交换式集线器时无限制			
	CIP 服务: 标 签数据链接 (周期链接通 信)	最大连接数	128		
		Packet 间隔 <sup>*4</sup>	各连接可设定 1 ~ 1000ms (以 1ms 为单位)		
		单元允许通信带宽	20,000 pps <sup>*5</sup> (含 Heartbeat)		
		最大标签集数	128		
		标签种类	网络变量		
		1 连接 (=1 标签集) 的标签数	8 (标签集中包含控制器状态时为 7)		
		标签最大数量	256		
		每 1 节点的最大链接数 数据容量 (所有标签的总 大小)	184,832 字节		
		每 1 连接的最大数据容 量	1,444 字节		
		最大可登录的标签集数	128 (1 连接 =1 标签集)		
	1 标签集的最大容量	1,444 字节 (标签集中包含控制器状态时使用 2 字节)			
多点传送 Packet 过滤 功能 <sup>*6</sup>	支持				
CIP 信息服务: Explicit 信息	Class3 (连接数)	64 (客户端 + 服务器)			
	UCMM (非连接 型)	可同时通 信的最大 客户端数	32		
		可同时通 信的最大 服务器数	32		
TCP Socket 数	30				
内置 EtherC AT 端口	通信标准	IEC 61158 Type12			
	EtherCAT 主站规格	符合 Class B (符合 Feature Pack Motion Control)			
	物理层	100BASE-TX			
	调制方式	基带			
	传送速度	100Mb/s (100BASE-TX)			
	Duplex 模式	Auto			
	拓扑结构	线、菊花链、分支布线			
	传送介质	类别 5 以上 双绞线电缆 (推荐铝带编织双重隔离屏蔽电缆、直线型)			
	节点间距离最大值	100m			
	最大从站数	128			
	可设定的节点地址范围	1 ~ 512			
	过程数据的最大容量	IN: 5,736 字节 OUT: 5,736 字节 但要在 4 帧 (最大过程数据帧数) 范围内			
	1 从站的最大容量	IN: 1,434 字节 OUT: 1,434 字节			
内置 EtherC AT 端口	通信周期	500/1,000/2,000/4,000/8,000μs			
	同步速度偏差	1μs 以下			

项目			NY5□2-		
			15□□	14□□	13□□
单元构成	可安装的单元数	系统整体的 NX 单元最大数	4,096 (EtherCAT 从站终端上)		
	最大扩展装置数		0		
内置时钟			环境温度 55 °C : 月差 -3.5 分~ +0.5 分 环境温度 25 °C : 月差 -1.5 分~ +1.5 分 环境温度 0 °C : 月差 -3 分~ +1 分		

\*1. 执行对象、变量表（变量名等）的容量。

\*2. 所有轴种类的合计。

\*3. 轴种类为 [伺服轴] 或 [编码器轴]，且轴使用设定为 [使用轴] 的轴合计。

\*4. 与节点数无关，以设定周期更新线路中的数据。

\*5. pps 代表 Packet Per Second，表示 1 秒内可处理的接收发送 Packet 数。

\*6. EtherNet/IP 端口安装了 IGMP 客户端，使用了支持 IGMP Snooping 的交换式集线器，所以会进行不必要的多点传送数据包的过滤。

## A-1-2 功能规格

下面介绍 NY 系列工业电脑 / 工业 BOX PC 的控制功能之功能规格。

项目			NY5□2-1□□□		
任务功能	功能	以指定执行条件和执行优先度的“任务”单位，执行 I/O 刷新及用户程序的功能			
		以固定周期执行的任务	主固定周期任务最大数	1	
			固定周期任务最大数	3	
		条件成立时执行的任务	事件任务最大数	32	
执行条件	事件任务启动指令执行时、变量条件式一致时				
设定	系统服务监视设定		—		
编程功能	POU (Program Organization Unit)	程序	任务中分配单位的 POU		
		功能块	编写带状态元件时使用的 POU		
		功能	运算处理等，编写输出与输入一对一的元件时使用的 POU		
	程序语言	种类	梯形图 <sup>1</sup> 结构化文本 (ST)		
	名称空间		按照名称进行 POU 定义分组的功能		
	变量	变量的外部参照功能	网络变量	许可来自触摸屏、高位电脑以及其它控制器等的访问的功能	
		数据类型	基本数据类型	布尔型	BOOL
	位列型			BYTE、WORD、DWORD、LWORD	
	整数型			INT、SINT、DINT、LINT、UINT、USINT、UDINT、ULINT	
	实型数据			REAL、LREAL	
	持续时间型			TIME	
	日期型			DATE	
	时刻型			TIME_OF_DAY	
	日期时刻型			DATE_AND_TIME	
	字符串型		STRING		
	派生数据类型		结构体型、联合体型、列举型		
	结构体型		功能	将数据型的多个不同数据汇总为 1 个进行处理的功能	
			成员最大数	2,048	
			嵌套最大数	8	
			成员的数据类型	基本数据类型、结构体型、联合体型、列举型、排列变量	
	联合体型	成员的偏置指定	将结构体成员配置到任意存储器位置的功能		
		功能	可通过多个不同数据类型对同一数据进行访问的功能		
		成员最大数	4		
	列举型	成员的数据类型	BOOL、BYTE、WORD、DWORD、LWORD		
		功能	将变量值称为“列举值”以标签 (字符串) 表示的功能		
	数据类型的属性	排列指定	功能	汇总相同数据类型的要素，从开头起使用编号 (后缀) 进行指定的功能	
			次元最大数	3	
要素最大数			65,535		
FB 实例的排列指定		可以			
范围指定	明确表示只能提取预先设定的范围内数值的功能				
库	可使用 (用户) 库				

项目			NY5□2-1□□□	
运动控制功能	单轴	控制模式	位置控制、速度控制、转矩控制	
		轴种类	伺服轴、虚拟伺服轴、编码器轴、虚拟编码器轴	
		可管理位置	指令位置、反馈位置	
		单轴位置控制	绝对值定位	指定绝对坐标的目标位置，进行定位的功能
			相对值定位	指定指令当前位置起的移动距离，进行定位的功能
			中断固定尺寸定位	指定基于外部输入的中断输入发生位置起的移动距离，进行定位的功能
			周期性同步绝对定位控制	在定位控制模式下，按控制周期输出指令位置的功能
		单轴速度控制	速度控制	在位置控制模式下进行速度控制的功能
			周期性同步速度控制	在速度控制模式下，按控制周期输出速度指令的功能
		单轴扭矩控制	扭矩控制	执行电机转矩控制的功能
		单轴同步控制	凸轮动作开始	使用指定的凸轮表开始凸轮动作的功能
			凸轮动作解除	完成输入参数指定轴的凸轮动作的功能
			齿轮动作开始	设定主轴与从轴间的齿轮比，进行齿轮动作的功能
			位置指定齿轮动作	设定主轴与从轴间的齿轮比和要同步的位置，进行齿轮动作的功能
			齿轮动作解除	中止执行中的齿轮动作、位置指定齿轮动作的功能
			梯形模式凸轮	与指定的主轴同步进行定位的功能
			主轴相对值相位补偿	执行同步控制中的主轴相位补偿的功能
			加减法定位	将 2 个轴的指令位置相加或相减得到的数值，作为指令位置输出的功能
		单轴手动操作	可运行	将伺服驱动器的状态切换为伺服 ON 状态，实现轴动作的功能
			点动进给	依据指定的目标速度执行点动进给的功能
		单轴控制辅助	轴错误复位	解除轴异常的功能
			原点复位	驱动电机，使用极限信号、近原点信号、原点信号确定机械原点的功能
			参数指定原点复位	指定参数驱动电机，使用极限信号、近原点信号、原点信号确定机械原点的功能
			高速原点复位	将绝对坐标“0”作为目标位置，进行定位并返回原点的功能
			强制停止	使轴减速停止的功能
			立即停止	使轴立即停止的功能
			超调值设定	变更轴目标速度的功能
			当前位置变更	将轴的指令当前位置和反馈当前位置变更为任意数值的功能
			外部锁定有效	发生触发，记录轴位置的功能
			外部锁定无效	使执行中的锁定无效的功能
			区域监视	判断轴的指令位置或反馈当前位置是否存在于指定范围（区域）内的功能
			数字凸轮开关有效	根据轴的位置将数字输出设为 ON 或 OFF 的功能
			轴间偏差监视	监视指定的 2 轴指令位置或反馈位置的差异量是否超出了容许值的功能
偏差计数器复位	将指令当前位置和反馈当前位置间的偏差归零的功能			
转矩限制	通过切换伺服驱动器转矩限制功能的有效 / 无效和设定转矩限制值，限制输出转矩的功能			
指令位置补偿	对动作中的轴进行位置补偿的功能			
启动速度	设定轴动作开始时初速度的功能			

项目			NY5□2-1□□□		
运动控制功能	轴组	多轴协调控制	绝对值直线插补	指定绝对位置进行直线插补的功能	
			相对值直线插补	指定相对位置进行直线插补的功能	
			2 轴圆弧插补	进行 2 轴圆弧插补的功能	
			轴组周期性同步绝对位置控制	在定位控制模式下，按控制周期输出位置指令的功能	
		多轴协调控制辅助	轴组错误复位	解除轴组及轴异常的功能	
			轴组有效	将轴组动作设为有效的功能	
			轴组无效	将轴组动作设为无效的功能	
			轴组强制停止	使插补动作中的所有轴减速停止的功能	
			轴组即停	使插补动作中的所有轴即停的功能	
			轴组超驰值设定	变更插补动作中合成目标速度的功能	
			轴组位置获取	获取轴组指令当前位置和反馈当前位置的功能	
			轴组构成轴写入	暂时改写轴组参数的 [构成轴] 的功能	
		通用	凸轮	凸轮表属性更新	对输入参数指定的凸轮表的终点索引进行更新的功能
				凸轮表保存	将输入参数指定的凸轮表保存至 CPU 单元内非易失性存储器中的功能
	凸轮表生成			根据输入参数指定的凸轮属性和凸轮节点生成凸轮表的功能	
	参数		MC 设定写入	暂时改写部分轴参数及轴组参数的功能	
			轴参数的变更	通过用户程序查看、变更轴参数的功能	
			计数模式	可选择线性模式（有限长）或旋转模式（无限长）	
	辅助功能	单位转换	可根据机械设定各轴的显示单位		
		加减速控制	自动加减速控制	以跃度设定轴及轴组动作时加减速曲线的功能	
			变更加减速速度	即使在加减速动作中仍可变更加减速速度的功能	
		位置检查	旨在检查定位完成，设定位置宽度和位置检查时间的功能		
		停止方法选择	设定即停输入信号及极限输入信号有效时的停止方法的功能		
		运动控制指令的重启	变更执行中运动控制指令的输入变量并重启，在动作中变更目标值的功能		
		运动控制指令的多重启动（缓冲模式）	对动作中启动其它运动控制指令时的执行开始时间和动作间速度的连接方法进行指定的功能		
		轴组动作的连续动作（转换模式）	对基于轴组动作多重启动的连续动作方法进行指定的功能		
		监视功能	软件限位	监视轴动作范围的功能	
位置偏差			监视轴的指令当前值和反馈当前值之间的位置偏差的功能		
速度 / 加减速速度 / 扭矩 / 插补速度 / 插补加减速速度			按轴及轴组分别设定警告值并监视的功能		
适用于绝对值编码器		使用欧姆龙制伺服驱动器 1S 系列或 G5 系列的带绝对值编码器电机，无需接通电源时原点复位的功能			
输入信号的逻辑反转		使立即停止输入信号、正方向极限输入信号、负方向极限输入信号、近原点输入信号的逻辑反转的功能			
外部 I/F 信号	可使用伺服驱动器侧的下列输入信号 原点信号、近原点信号、正方向极限信号、负方向极限信号、即停信号、中断输入信号				
单元（输入输出）管理功能	EtherCAT 从站	最大从站数	128		

项目			NY5□2-1□□□	
通信功能	Ethernet/IP 端口	通信协议	TCP/IP、UDP/IP	
		TCP/IP 功能	CIDR	对不使用 IP 地址类别（A 类~C 类）的 IP 地址进行分配的功能
			IP Forwarding	在接口之间传送 IP Packet 的功能
			Packet Filter*2	对 IP Packet 进行检查，判断是否可通过发送源 IP 地址或 TCP 端口编号等进行收发的功能
			NAT	将 2 个 IP 地址进行转换并发送的功能
		CIP 通信服务	标签数据链接	无需程序即可与 EtherNet/IP 网络上的设备进行周期性的数据交换的功能。
			信息通信	可与 EtherNet/IP 网络上的设备进行任意的 CIP 指令接收 / 发送的功能
		TCP/IP 应用	Socket 服务	通过 UDP 或 TCP 协议与 Ethernet 上的任意节点之间接收 / 发送任意数据的功能通过 Socket 通信用指令执行的功能
			FTP 客户端	通过 CPU 单元使用 FTP 向 Ethernet 上的其它计算机及控制器传输文件的功能。通过 FTP 客户端通信指令执行
			FTP 服务器	通过 Ethernet 上的其它计算机读写控制器的 CPU 单元的 SD 存储卡内文件的功能
	SNMP 代理		向使用 SNMP 管理器的网络管理软件提供内置 EtherNet/IP 端口内部状态信息的功能	
	EtherCAT 端口	支持服务	过程数据通信	在通过 CoE 定义的 EtherCAT 主站和从站之间，将控制信息作为一定周期内周期性通信进行数据交换的通信方式
			SDO 通信	在通过 CoE 定义的 EtherCAT 主站和从站之间，将控制信息作为非固定周期的事件通信进行数据交换的通信方式
		网络扫描	读取连接的从站设备的信息，自动生成从站构成的功能	
		DC (Distributed Clock)	所有的 EtherCAT 设备（包含主站）共享相同“EtherCAT System Time”，进行时刻同步的功能	
		Packet 监视器	保存主站发送和接收的帧的功能。可通过 WireShark 等查看已保存的数据	
		设定从站有效 / 无效	将从站作为通信对象，设定为有效或无效的功能	
		从站脱离 / 再加入	出于更换从站等保养的目的，暂时将相应从站从 EtherCAT 网络中脱离 / 再加入的功能	
		支持应用协议	CoE	在 EtherCAT 中，向从站发送 CAN 应用信息（SDO）的功能
	通信用指令			FTP 客户端指令、CIP 通信指令、Socket 通信用指令、SDO 信息指令、Modbus RTU 协议指令
系统管理功能	事件日志	功能	记录事件发生的功能	
		最大件数	系统事件日志	2,048
			访问事件日志	1,024
			用户事件日志	1,024

项目			NY5□2-1□□□		
调试功能	在线编辑		在线变更程序、功能块、功能、全局变量的功能 经由网络多个操作者可变更个别 POU		
	强制值刷新		以强制值刷新特定接点的功能		
	最大点数	EtherCAT 从站设备变量	64		
	MC 调试		通过 Sysmac Studio 确认电机电动作及接线的功能		
	同步		在线连接时将 Sysmac Studio 的项目文件和 CPU 单元的数据实现一致的功能		
	微分监控		监控接点的上升沿或下降沿的功能		
	最大点数		8		
	数据跟踪	种类	触发跟踪	建立触发后，达到设定的采样数时自动停止跟踪的功能	
			连续跟踪	继续执行数据跟踪， Sysmac Studio 始终收集跟踪数据的功能	
		同时启动最大数		4	
		记录最大数		10,000	
		最大采样点数		192 变量	
		采样时间		指定任务周期、指定时间、采样指令执行时	
		触发跟踪	触发条件		<ul style="list-style-type: none"> <li>• BOOL 型变量的上升沿 / 下降沿</li> <li>• BOOL 型变量以外的常数值比较</li> </ul> 比较方法：=、>、≥、<、≤、≠
	延时值			设定触发成立前 / 成立后采样数比例的功能	
仿真功能			通过 Sysmac Studio 模拟 CPU 单元动作的功能		
高可靠性功能	自诊断	控制器异常	重要程度	全部停止故障、部分停止故障、轻度故障、监视信息、一般信息	
			信息语言最大数	9(Sysmac Studio) 2 (NS 系列触摸屏)	
		用户异常	功能	设计任意异常并预先登录，通过执行指令留下记录的功能	
			重要程度	8 级	
			信息语言最大数	9	
安全功能	顾客资产保护 / 防止误操作	CPU 单元名称功能及串行 ID 功能		通过 Sysmac Studio 进行在线连接时，确认项目中的 CPU 单元名称与连接对象 CPU 单元的 CPU 名称是否一致的功能	
		保护功能	无用户程序复原信息传送功能	进行设置使其不能通过 Sysmac Studio 读取 CPU 单元内数据的功能	
			对 CPU 单元的写入保护功能	进行设置使其不能通过 Sysmac Studio/SD 存储卡写入 CPU 单元内数据的功能	
			项目文件整体保护功能	通过密码禁止在 Sysmac Studio 中打开 .smc 文件的功能	
			数据保护功能	通过密码在 Sysmac Studio 中对 POU 进行保护的功能	
		操作权限的认证功能			因操作失误可能对装置或人施加危害时，根据操作权限限制在线操作功能的功能
			组数		5
用户程序执行用 ID 认证功能			对于特定的硬件 (CPU 单元)，只要不从 Sysmac Studio 输入用户程序执行用 ID，就不能执行用户程序的功能		
SD 存储卡功能	保存位置		共享文件夹 <sup>3</sup>		
	应用程序	SD 存储卡操作指令	通过用户程序中的指令访问虚拟 SD 存储卡的功能		
		通过 Sysmac Studio 的文件操作	用户不仅可以向虚拟 SD 存储卡保存及读取控制器用文件，还可以保存及读取通用文档文件等		
	通过 FTP Client/Server 功能进行的文件操作	可利用 FTP 客户端功能或 FTP 服务器功能进行保存和读取			

项目			NY5□2-1□□□	
备份功能	SD 存储卡的备份功能	操作方法	通过系统定义变量进行指示	通过系统定义变量的操作进行备份及比对的功能
			SysmacStudio 存储卡画面	通过 Sysmac Studio 的 SD 存储卡操作画面进行备份及比对的功能
			专用指令	通过专用指令进行备份的功能
		保护功能	禁止向 SD 存储卡备份	禁止向虚拟 SD 存储卡备份的功能
	Sysmac Studio 控制器备份功能			使用 Sysmac Studio 进行控制器备份、恢复、比对的功能

\*1. 可联机 ST（梯形图中描述 ST 语言的梯形图语言的要素）。

\*2. 仅内部通信端口。

\*3. 运行 Windows 的 HDD/SSD 上存在的文件夹。控制器侧作为虚拟 SD 存储卡处理。

## A-2 NY 系列中任务实际处理时间的计算标准

---

下面介绍 NY 系列系统中任务实际处理时间平均值的参考计算方法。

任务实际处理时间和任务执行时间需要在实际设备的控制器上确认。详情请参考□「5-9 任务设计方法及输入输出响应时间」(P.5-56)。



### 使用注意事项

---

在实际的控制器上，因用户程序中搭载的逻辑是否动作、是否发生通信指令、数据链接动作、是否使用数据追踪等各种功能、是否连接 Windows 侧的负载和工业监视器、是否使用 PCIe 卡等不同，任务执行时间会发生变动。

请务必使用实际机器在各种条件下确认性能，确认不会发生任务周期超限，确保良好的通信响应性。

---



### 参考

---

正在执行优先度高于固定周期任务的任务时，将暂停。“任务实际处理时间”不含暂停的时间。“任务执行时间”中包含暂停时间，表示任务从开始到结束的时间。“任务实际处理时间”和“任务执行时间”的详细区别请参考□「任务实际时间和任务实际处理时间的含义」(P.5-55)。

---

## A-2-1 任务实际处理时间平均值的计算方法

任务实际处理时间的平均值为 I/O 刷新处理时间、用户程序执行时间、运动控制处理时间、共通处理时间的总和。

任务实际处理时间的平均值 = I/O 刷新处理时间 + 用户程序执行时间  
+ 运动控制处理时间 + 共通处理时间

各处理内容如下所示。

○：有，—：无

处理		处理内容	主固定周期任务	固定周期任务 (执行优先级 16)	固定周期任务 (执行优先级 17、18)
I/O 刷新处理		执行与 EtherCAT 从站的刷新。	○	—	—
用户程序执行		• 按分配顺序执行任务中分配的 程序。	○	○	○
运动控制处理		• 执行用户程序发出的运动控制 指令。 • 运动输出处理	○	—	—
共通处理 时间	系统 共通处理 1	• 变量的更新处理 (参照任务 时) • 运动输入处理 • 数据追踪处理	○	○	○
	系统 共通处理 2	• 变量的更新处理 (更新任务 时) • 为确保与任务执行的同时性, 从控制器外部进行的变量访问 处理	○	○	○
	系统额外时 间	上述以外的系统共通处理	○	○	○

下面表示各处理时间的计算标准。

## I/O 刷新处理时间

I/O 刷新处理时间按以下公式计算。

I/O 刷新处理时间 = EtherCAT 从站处理时间

下面表示 EtherCAT 从站处理时间的计算方法。



### 参考

在 EtherCAT 中未分配从站的任务中， EtherCAT 从站处理时间为“0”。

### ● 主固定周期任务中的 EtherCAT 从站处理时间

主固定周期任务中的 EtherCAT 从站处理时间按以下公式计算。

$$\text{EtherCAT 从站处理时间 } [\mu\text{s}] = 0.0008 \times \text{pDout} + 0.0028 \times \text{pDin} + 0.082 \times \text{pDinout} + (1.24 \times \text{Snum} + 0.01 \times \text{Clen})$$

- pDout : 分配到主固定周期任务的 EtherCAT 从站的输出过程数据总大小 [ 字节 ]
- pDin : 分配到主固定周期任务的 EtherCAT 从站的输入过程数据总大小 [ 字节 ]
- pDinout : 分配到主固定周期任务的各 EtherCAT 从站的输出过程数据和输入过程数据中总大小较大者 [ 字节 ]
- Snum : 内置 EtherCAT 端口上连接的 EtherCAT 从站总台数 [ 台 ]
- Clen : 内置 EtherCAT 端口上连接的电缆总长度 [m]

## 用户程序执行时间

用户程序执行时间取决于指令种类 × 指令数。

作为参考，按以下 3 大分类来检测和预计指令数。

- 标准指令
- 双精度实数（LREAL）型的四则运算指令
- 双精度实数（LREAL）型的三角函数指令

此外，梯形图程序和 ST 语言程序中使用的指令不同。各指令的构成请参考「“梯形图语言的标准指令”的指令种类构成」（P.A-15）及「“ST 语言的标准指令”的指令种类构成」（P.A-17）。

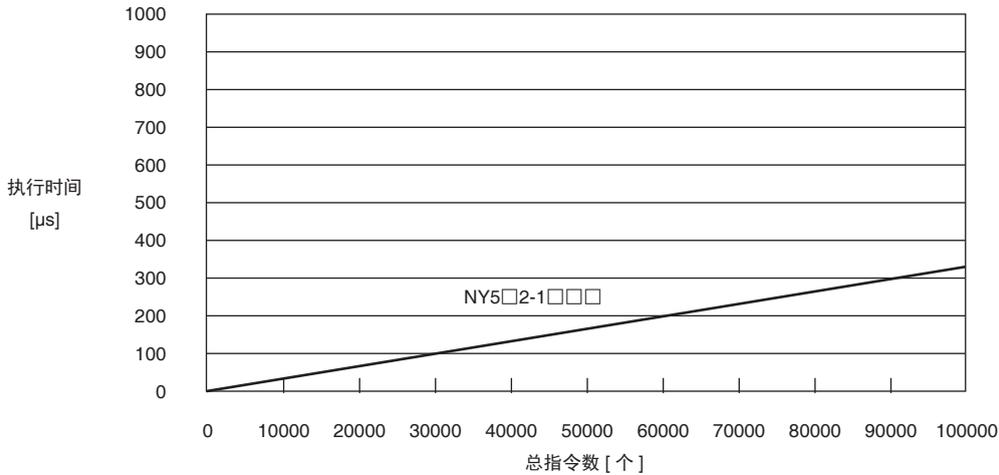
### ● 简单估算

根据各分类的指令数，按以下图表，读取以下各指令的执行时间并合计。

- 标准指令的执行时间
- 双精度实数（LREAL）型的四则运算指令执行时间
- 双精度实数（LREAL）型的三角函数指令执行时间

由此可估算出用户程序的执行时间。

### 梯形图语言的标准指令执行时间

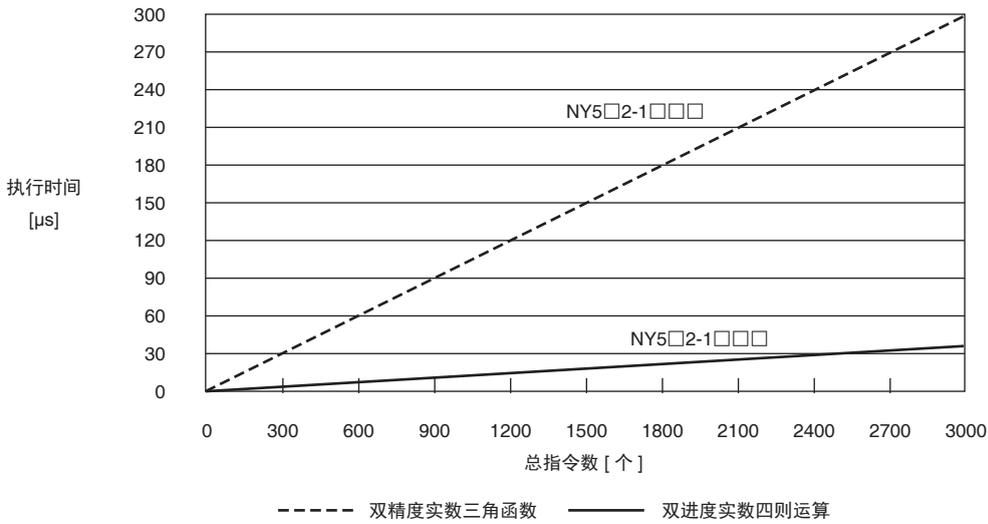


- “梯形图语言的标准指令”的指令种类构成

本构成的指令执行率为 20%。

指令种类	指令名称	指令个数的比率 [%]	各指令种类的执行时间比率 [%]
梯形图指令	LD, AND, OUT, SET, RESET	81.0	40.2
比较指令	EQ, LT	4.1	8.3
计时指令、计数指令	Timer, TON/TOF, CTU/CTD	1.6	7.3
算术指令	+, -, *, /, ADD, SUB, MUL, DIV	2.4	6.5
BCD 转换指令、数据转换指令	INT_TO_DINT, WORD_BCD_TO_UINT	0.2	1.2
位列运算指令	AND, OR	6.2	13.0
数据传送指令	MOVE	4.6	23.5
总计		100.0	100.0

### 梯形图语言双精度实数（LREAL）型的四则运算指令和三角函数指令执行时间



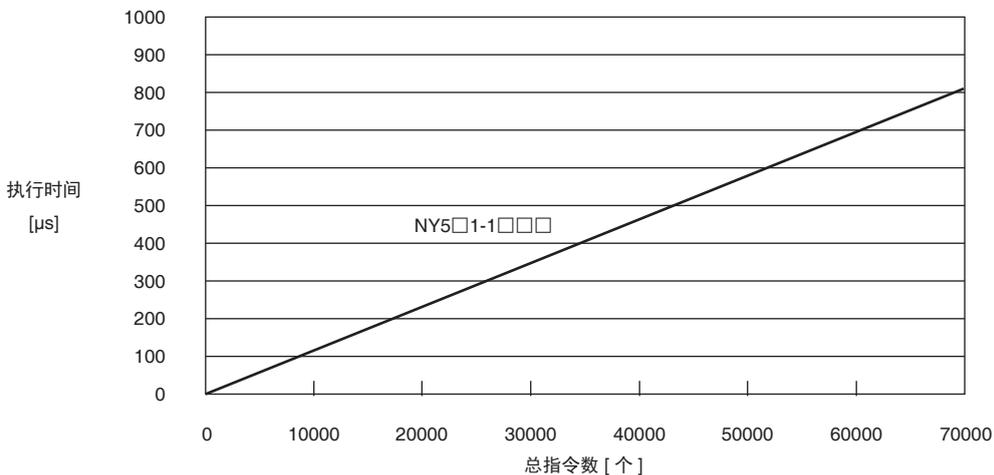
• “双精度实数的四则运算”构成

指令名称	指令个数的比率 [%]
双精度实数的加法	20.0
双精度实数的减法	20.0
双精度实数的乘法	30.0
双精度实数的除法	30.0
总计	100.0

• “双精度实数的三角函数”构成

指令名称	指令个数的比率 [%]
双精度实数的 Sin	16.7
双精度实数的 Cos	16.7
双精度实数的 Tan	16.7
双精度实数的 Sin <sup>-1</sup>	16.7
双精度实数的 Cos <sup>-1</sup>	16.7
双精度实数的 Tan <sup>-1</sup>	16.7
总计	100.0

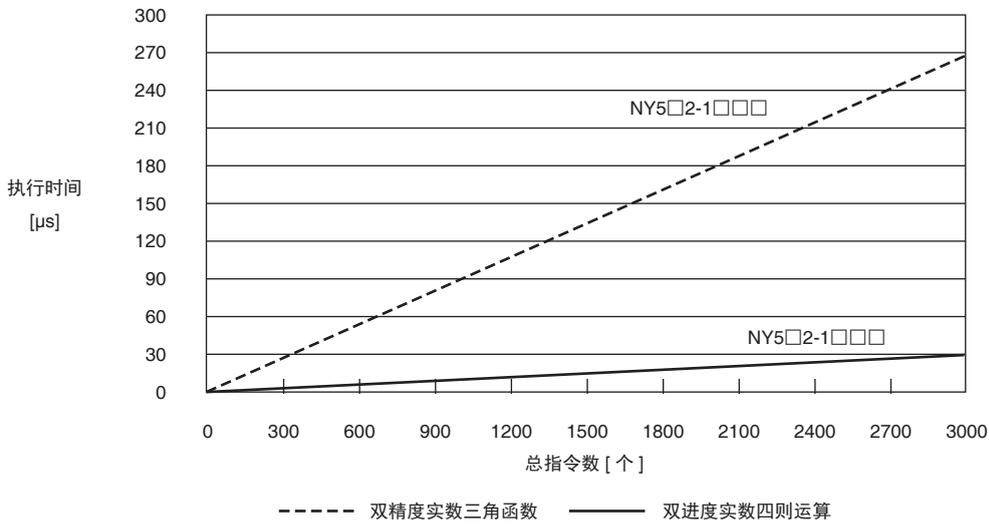
### ST 语言的标准指令执行时间



- “ST 语言的标准指令”的指令种类构成

指令种类	指令名称	指令个数的比率 [%]	各指令种类的执行时间比率 [%]
ST 语法指令	IF ELSIF END_IF	75.4	41.6
比较指令	EQ, LT	5.2	8.7
计时指令、计数指令	Timer, TON/TOF, CTU/CTD	2.1	18.8
算术指令	+, -, *, /	3.1	10.2
BCD 转换指令、数据转换指令	INT_TO_DINT, WORD_BCD_TO_UINT	0.2	1.6
位列运算指令	AND, OR	8.0	11.7
数据传送指令	:=	5.9	7.3
总计		100.0	100.0

### ST 语言双精度实数（LREAL）型的四则运算指令和三角函数指令执行时间



- “双精度实数的四则运算”构成

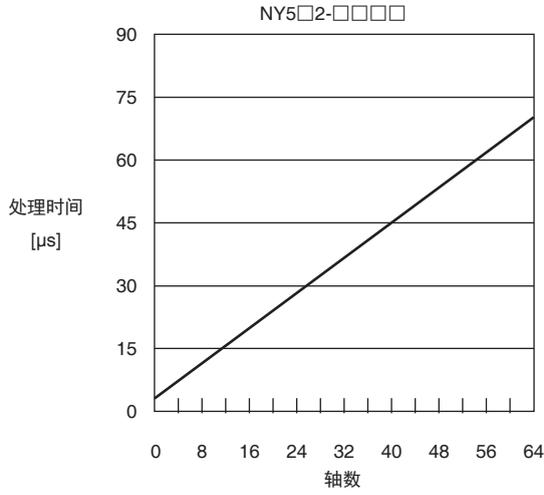
指令名称	指令个数的比率 [%]
双精度实数的加法	20.0
双精度实数的减法	20.0
双精度实数的乘法	30.0
双精度实数的除法	30.0
总计	100.0

- “双精度实数的三角函数”构成

指令名称	指令个数的比率 [%]
双精度实数的 Sin	16.7
双精度实数的 Cos	16.7
双精度实数的 Tan	16.7
双精度实数的 $\text{Sin}^{-1}$	16.7
双精度实数的 $\text{Cos}^{-1}$	16.7
双精度实数的 $\text{Tan}^{-1}$	16.7
总计	100.0

## 运动控制处理时间

运动控制处理时间取决于所用伺服的轴数（包括虚拟轴）。  
根据使用的伺服轴数，按以下图表，读取运动控制处理时间。



## 共通处理时间

共通处理时间是指系统额外时间、系统共通处理 1 及系统共通处理 2 的总和，值如下。共通处理时间因任务类型不同而异。

任务类型	共通处理时间 [μs] (参考值)	
	NY5 □ 2-1 □□□	
主固定周期任务	258	
固定周期任务 (执行优先级 16、17、18)	20	

## A-2-2 任务实际处理时间平均值的计算和任务周期的设定示例

### 任务实际处理时间平均值的计算示例

计算使用单元版本为 1.12 的 NY5□2-1□□□、以下条件下的任务实际处理时间的平均值。

任务类型为主固定周期任务。

项目		条件	
使用的从站 / 单元	EtherCAT 从站	输入从站	GX-ID1611(Ver.1.1): 1 台
		输出从站	GX-OD1611(Ver.1.1): 1 台
		伺服驱动器	R88D-1SN □□□ -ECT: 4 台
		EtherCAT 从站终端: 1 台	
		EtherCAT 耦合器单元	NX-ECC203: 1 台
		DC 输入单元	NX-ID5342: 1 台
		晶体管输出单元	NX-OD5121: 1 台
		模拟输入单元	NX-AD3608: 1 台
		模拟输出单元	NX-DA2605: 1 台
		通讯接口单元	NX-CIF101: 1 台 NX-CIF105: 1 台
用户程序:	使用语言	梯形图语言	
	标准指令构成	指令数: 5,000 个	
	双进度实数的四则运算	指令数: 200 个	
	双精度实数的三角函数	指令数: 100 个	
运动控制处理	轴数	4 轴	

(注) 内置 EtherCAT 端口上连接的电缆总长度为 10[m]。

#### ● I/O 刷新时间

##### EtherCAT 从站处理时间:

输入从站 GX-ID1611(Ver.1.1)、输出从站 GX-OD1611(Ver.1.1)、伺服电机 R88D-1SN □□□ -ECT 及上述构成下 EtherCAT 从站终端的输出过程数据大小 pDout、输入过程数据大小 pDin、输出数据大小和输入数据大小中的较大者 pDinout 的值如下所示。

EtherCAT 从站	输出过程数据大小 pDout [字节]	输入过程数据大小 pDin [字节]	输出数据大小和输入数据大小的较大者 pDinout
GX-ID1611(Ver.1.1)	0	3	3
GX-OD1611(Ver.1.1)	2	1	2
R88D-1SN □□□ -ECT	23	26	26
EtherCAT 从站终端合计	62	88	88
NX-ECC203	0	18	
NX-ID5342	0	2	
NX-OD5121	2	0	
NX-AD3608	0	8	
NX-DA2605	4	0	
NX-CIF101	28	30	
NX-CIF105	28	30	

pDout、pDin 及 pDinout 各自的总计值如下。

$$pDout=2 + 23 \times 4 + 62=156[\text{字节}]$$

$$pDin=3 + 1 + 26 \times 4 + 88=196[\text{字节}]$$

$$pDinout=3 + 2 + 26 \times 4 + 88=197[\text{字节}]$$

根据上述值，按以下公式计算 I/O 刷新时间。

I/O 刷新处理时间

$$\begin{aligned}
 &= \text{EtherCAT 从站处理时间} \\
 &= 0.0008 \times pDout + 0.0028 \times pDin + 0.082 \times pDinout + (1.24 \times Snum + 0.01 \times Clen) \\
 &= 0.0008 \times 156 + 0.0028 \times 196 + 0.082 \times 197 + (1.24 \times 7 + 0.01 \times 10) \\
 &= 0.1248 + 0.5488 + 16.154 + (8.68 + 0.1) \\
 &\approx 26[\mu\text{s}]
 \end{aligned}$$

## ● 用户程序执行时间

从各图表中读取以下数值。

- 标准指令构成

根据梯形图语言的标准指令执行时间图表，NY5□2-1□□□ 中指令数为 5000 个的用户程序执行时间为 16.5μs。

- 双精度实数的四则运算

根据梯形图语言的双精度实数 (LREAL) 型的四则运算指令和三角函数指令的执行时间图表，NY5□2-1□□□ 中指令数为 200 个的用户程序执行时间为 2.5μs。

- 双精度实数的三角函数

根据梯形图语言的双精度实数 (LREAL) 型的四则运算指令和三角函数指令的执行时间图表，NY5□2-1□□□ 中指令数为 100 个的用户程序执行时间为 10μs。

因此，用户程序执行时间为以上的合计值，结果如下。

$$\begin{aligned}
 \text{用户程序执行时间} &= 16.5 + 2.5 + 10 \\
 &= 29[\mu\text{s}]
 \end{aligned}$$

## ● 运动控制处理时间

根据运动控制处理的执行时间图表，NY5□2-1□□□ 的 4 轴运动控制处理的执行时间读取为 7μs。

## ● 共通处理时间

任务类型为主固定周期任务，因此 NY5□2-1□□□ 的共通处理时间为 258μs。

因此，任务实际处理时间的平均值按以下公式计算。

$$\begin{aligned}
 \text{任务实际处理时间的平均值} &= \text{I/O 刷新时间} + \text{用户程序执行时间} \\
 &\quad + \text{运动控制处理} + \text{共通处理时间} \\
 &= 26 + 29 + 7 + 258 \\
 &= 320[\mu\text{s}]
 \end{aligned}$$

## 任务周期的设定示例

根据上面计算得到的任务实际处理时间的平均值，设定任务周期。任务类型为主固定周期任务。任务周期的值应大于计算得到的任务实际处理时间的平均值。此时，请预留出余量，将任务周期的值设定为大于任务实际处理时间平均值的 1.1 倍。

$$\text{任务周期} \geq \text{任务实际处理时间的平均值} \times 1.1$$

由于上面计算得到的任务实际处理时间的平均值为  $320\mu\text{s}$ ，因为要设定为大于其 1.1 倍的值，所以任务周期为  $500\mu\text{s}$ 。

在实际的控制器上，因用户程序中搭载的逻辑是否动作、是否发生通信指令、数据链接动作、是否使用数据追踪等各种功能、是否连接 Windows 侧的负载和工业监视器、是否使用 PCIe 卡等不同，任务执行时间会发生变动。此外，如果是固定周期任务，因是否有执行优先度更高的任务动作，任务执行时间会发生变动。

请务必使用实机的控制器，根据任务执行时间监视器，进行任务执行时间的验证。

## A-3 系统定义变量一览

系统中分配了特定功能的变量。

已事先登录到全局变量表或各 POU 的本地变量表中。

变量名称固定。由以 “\_(下划线)” 开始的部分和以 “P\_” 开始的部分组成。

系统定义变量有只读和可读写两类。

可通过用户程序、外部通信或 Sysmac Studio、NS/NA 系列触摸屏读写。

系统定义变量基本上按功能模块分类。

变量名称以下类别名称开头。

功能模块	类别名称
NY 系列控制器的系统整体	_ (无)
PLC 功能模块	_PLC
运动控制功能模块	_MC
EtherCAT 主机功能模块	_EC
EtherNet/IP 功能模块	_EIP、_EIP1、_EIPIn1

一览表的阅读方法如下所示。

变量名称	名称	功能	数据类型	值的范围	参照页
[系统定义变量名称。 开头带类别名称]	[名称]	[介绍功能]	[表示变量的 数据类型]	[表示变量值 的范围]	[表示个别规 格表的页面]

变量名称栏的 () 中表示追加了该系统定义变量的 NY 系列控制器的单元版本。



### 使用注意事项

因控制器的系列不同，可能有不支持的或排列数等规格不同的系统定义变量。详情请通过□□「A-4 系统定义变量的个别规格」(P.A-45) 确认个别规格。

### A-3-1 NY 系列控制器的系统整体（类别名称：无）

#### ● 功能分类：时钟相关

变量名称	名称	功能	数据类型	值的范围	参照页
_CurrentTime	系统时刻	保存控制器的内部时钟数据。	DATE_AND_ TIME	DT#2000-01-0 1-00:00:00 ~ DT#2099-12-3 1-23:59:59	A-46

## ● 功能分类：任务相关

变量名称	名称	功能	数据类型	值的范围	参照页
_<任务名称> _Active	任务执行中标志	正在执行任务时为 TRUE。 任务非执行中时为 FALSE。 (注) 本系统定义变量在用户程序中不可使用。仅在通过 Sysmac Studio 的数据追踪功能浏览任务状态时使用。	BOOL	TRUE、 FALSE	A-46
_<任务名称> _LastExecTime	上次任务执行时间	表示上次执行任务的时间（以 0.1μs 为单位）。 (注) 本系统定义变量在用户程序中不可使用。仅在通过 Sysmac Studio 的数据追踪功能浏览任务状态时使用。	TIME	遵照数据类型	A-46
_<任务名称> _MaxExecTime	任务执行时间最大值	表示任务执行时间的最大值（以 0.1μs 为单位）。 (注) 本系统定义变量在用户程序中不可使用。仅在通过 Sysmac Studio 的数据追踪功能浏览任务状态时使用。	TIME	遵照数据类型	A-46
_<任务名称> _MinExecTime	任务执行时间最小值	表示任务执行时间的最小值（以 0.1μs 为单位）。 (注) 本系统定义变量在用户程序中不可使用。仅在通过 Sysmac Studio 的数据追踪功能浏览任务状态时使用。	TIME	遵照数据类型	A-46
_<任务名称> _ExecCount	任务执行次数	表示任务的执行次数。 若超出 4294967295 次，将恢复为 0，然后继续计数。 (注) 本系统定义变量在用户程序中不可使用。仅在通过 Sysmac Studio 的数据追踪功能浏览任务状态时使用。	UDINT	遵照数据类型	A-47
_<任务名称> _Exceeded	任务周期超出标志	超出任务周期后，变为 TRUE。 若在任务周期内完成任务，则变为 FALSE。 (注) 本系统定义变量在用户程序中不可使用。仅在通过 Sysmac Studio 的数据追踪功能浏览任务状态时使用。	BOOL	TRUE、 FALSE	A-47
_<任务名称> _ExceedCount	任务周期超出次数	保存超出周期的次数。 若当前值超出数据类型的最大值，当前值将恢复为 0，然后继续计数。 若超出 4294967295 次，将恢复为 0，然后继续计数。 (注) 本系统定义变量在用户程序中不可使用。仅在通过 Sysmac Studio 的数据追踪功能浏览任务状态时使用。	UDINT	遵照数据类型	A-47

## ● 功能分类：异常相关

变量名称	名称	功能	数据类型	值的范围	参照页
_ErrSta	控制器异常状态	正在发生控制器异常时为 TRUE。 未发生控制器异常时为 FALSE。 (注) 出于实时性及与表示各功能模块异常原因的状态之间的同时性考虑, 在用户程序中无法使用。仅在介由通信从外部参考状态时使用。各个位的含义请参考后面介绍的“异常状态中各个位的含义”。	WORD	16#0000 ~ 16#C0F0	A-48
_AlarmFlag	用户异常状态	发生用户异常时, 与事件重要程度对应的位将变为 TRUE。 User fault Level 1 ~ 8 对应位 00 ~ 07。 未发生用户异常时, 为 16 进制的 0000。	WORD	16#0000 ~ 16#00FF	A-48

## ● 功能分类：SD 存储卡相关

变量名称	名称	功能	数据类型	值的范围	参照页
_Card1Ready	SD 存储卡 可使用标志	NY 系列识别到虚拟 SD 存储卡时, 变为 TRUE。 未识别到时, 为 FALSE。TRUE: 可使用 FALSE: 不可使用	BOOL	TRUE、 FALSE	A-48
_Card1Protect	SD 存储卡 写保护标志	NY 系列中不使用。 始终为 FALSE。	BOOL	TRUE、 FALSE	A-48
_Card1Err	SD 存储卡 错误标志	NY 系列中不使用。 始终为 FALSE。	BOOL	TRUE、 FALSE	A-49
_Card1Access	SD 存储卡 访问中标志	NY 系列中不使用。 始终为 FALSE。	BOOL	TRUE、 FALSE	A-49
_Card1Deteriorated	SD 存储卡 寿命警告标志	NY 系列中不使用。 始终为 FALSE。	BOOL	TRUE、 FALSE	A-49
_Card1PowerFail	SD 存储卡 访问中断电标志	正在访问虚拟 SD 存储卡的过程中, 若控制器发生断电, 则变为 TRUE。 TRUE: 正在访问虚拟存储卡时发生断电 FALSE: 正常	BOOL	TRUE、 FALSE	A-49

变量名称 成员名称	名称	功能	数据类型	值的范围	参照页
_Card1BkupCmd	SD 存储卡 备份指示		_sBKUP_ CMD		A-49
ExecBkup	备份启动标志	向虚拟 SD 存储卡备份控制器的数据时, 变为 TRUE。 (注) 本系统定义变量在用户程序中不可 使用。通过 CIP 信息通信, 由触摸屏 或上位计算机发出指令时使用。	BOOL	TRUE、 FALSE	A-49
CancelBkup	备份取消标志	停止向虚拟 SD 存储卡备份时, 变为 TRUE。 (注) 本系统定义变量在用户程序中不可 使用。通过 CIP 信息通信, 由触摸屏 或上位计算机发出指令时使用。	BOOL	TRUE、 FALSE	A-50
ExecVefy	核对启动标志	核对控制器和虚拟 SD 存储卡中的备份文 件时, 变为 TRUE。 (注) 本系统定义变量在用户程序中不可 使用。通过 CIP 信息通信, 由触摸屏 或上位计算机发出指令时使用。	BOOL	TRUE、 FALSE	A-50
CancelVefy	核对取消标志	停止核对控制器和虚拟 SD 存储卡中的备 份文件时, 变为 TRUE。 (注) 本系统定义变量在用户程序中不可 使用。通过 CIP 信息通信, 由触摸屏 或上位计算机发出指令时使用。	BOOL	TRUE、 FALSE	A-50
DirName	目录名	指定备份对象的虚拟 SD 存储卡目录名 称。 (注) 本系统定义变量在用户程序中不可 使用。通过 CIP 信息通信, 由触摸屏 或上位计算机发出指令时使用。	STRING(64)	遵照数据类型	A-50
_Card1BkupSta	SD 存储卡 备份 状态		_sBKUP_ STA		A-50
Done	完成标志	备份执行完成时为 TRUE。 (注) 本系统定义变量在用户程序中不可 使用。通过 CIP 信息通信, 由触摸屏 或上位计算机发出指令时使用。	BOOL	TRUE、 FALSE	A-50
Active	执行中标志	正在执行备份时为 TRUE。 (注) 本系统定义变量在用户程序中不可 使用。通过 CIP 信息通信, 由触摸屏 或上位计算机发出指令时使用。	BOOL	TRUE、 FALSE	A-51
Err	错误标志	备份处理异常结束时为 TRUE。 (注) 本系统定义变量在用户程序中不可 使用。通过 CIP 信息通信, 由触摸屏 或上位计算机发出指令时使用。	BOOL	TRUE、 FALSE	A-51

变量名称 成员名称	名称	功能	数据类型	值的范围	参照页
_Card1VefySta	SD 存储卡 核对状态		_sVEFY_ STA		A-51
Done	完成标志	核对执行完成时为 TRUE。 (注) 本系统定义变量在用户程序中不可使用。通过 CIP 信息通信, 由触摸屏或上位计算机发出指令时使用。	BOOL	TRUE、 FALSE	A-51
Active	执行中标志	正在执行核对时为 TRUE。 (注) 本系统定义变量在用户程序中不可使用。通过 CIP 信息通信, 由触摸屏或上位计算机发出指令时使用。	BOOL	TRUE、 FALSE	A-51
VefyRsIt	核对结果标志	核对一致时为 TRUE, 核对不一致时为 FALSE。 (注) 本系统定义变量在用户程序中不可使用。通过 CIP 信息通信, 由触摸屏或上位计算机发出指令时使用。	BOOL	TRUE、 FALSE	A-51
Err	错误标志	核对处理异常结束时为 TRUE。 (注) 本系统定义变量在用户程序中不可使用。通过 CIP 信息通信, 由触摸屏或上位计算机发出指令时使用。	BOOL	TRUE、 FALSE	A-51

● 功能分类：备份相关

变量名称	名称	功能	数据类型	值的范围	参照页
_BackupBusy	备份相关功能执行中 标志	正在执行备份、恢复、核对时为 TRUE。	BOOL	TRUE、 FALSE	A-52

### ● 功能分类：电源管理相关

变量名称	名称	功能	数据类型	值的范围	参照页
_PowerOnHour	通电时间	表示通电时间。 以 1 小时为单位保存控制器的通电时间。 重置时，请保存 0。 变为 4294967295 后，不会继续更新。 电源 ON 时不会初始化。	UDINT	0 ~ 4294967295	A-52
_PowerOnCount	断电发生次数	表示发生断电的次数。 从第一次打开控制器的电源后，每次发生断电时累加 (+1)。 重置时，请保存 0。 变为 4294967295 后，不会继续更新。 电源 ON 时不会初始化。	UDINT	0 ~ 4294967295	A-52
_RetainFail	断电保持失败标志	以下情况下，将变为 TRUE（断电保持失败）。 <ul style="list-style-type: none"> <li>接通电源时，电池备份存储器 / 非易失性存储器检查错误时</li> </ul> 以下情况下，将变为 FALSE（无断电保持失败）。 <ul style="list-style-type: none"> <li>接通电源时，电池备份存储器 / 非易失性存储器的存储器检查 OK 时</li> <li>下载了用户程序时</li> <li>执行了存储器全部清除时</li> </ul> （注）无法保持绝对编码器原点位置偏置数据时，将反映到轴变量的异常状态中，而不是本标志。	BOOL	TRUE、 FALSE	A-53
_SelfTest_UPSSignal	UPS 信号检测标志	检测到 UPS 发出的临时停电信号时变为 TRUE。	BOOL	TRUE、 FALSE	A-53
_RequestShutdown	关机请求标志	如果在运行过程中按下电源按钮，将变为 TRUE。	BOOL	TRUE、 FALSE	A-53

### ● 功能分类：OS(Windows) 相关

变量名称	名称	功能	数据类型	值的范围	参照页
_OSRunning	OS 运行中标志	控制器观测到 OS(Windows) 的状态为运行中时变为 TRUE。	BOOL	TRUE、 FALSE	A-53
_OSHalted	OS 停止中标志	控制器观测到 OS(Windows) 的状态为停止中时变为 TRUE。	BOOL	TRUE、 FALSE	A-53
_OSErrorState	OS 异常状态标志	控制器判断 OS(Windows) 的状态为异常发生中时变为 TRUE。	BOOL	TRUE、 FALSE	A-54

## ● 功能分类：编程相关

变量名称	名称	功能	数据类型	值的范围	参照页
P_On	始终 ON 标志	表示始终为 ON 的标志。	BOOL	TRUE	A-54
P_Off	始终 OFF 标志	表示始终为 OFF 的标志。	BOOL	FALSE	A-54
P_CY	进位标志	在部分指令中更新的标志。	BOOL	TRUE、 FALSE	A-54
P_First_RunMode	运行开始时 1 个周期 ON 标志	控制器的动作模式从程序模式变更为运行模式时，若该程序正在执行，则在 1 个任务周期内为 TRUE。 若该程序不是正在执行，则保持 FALSE。用于在控制器开始运行时进行初始处理等情况。 (注) 本系统定义变量在函数中不可使用。	BOOL	TRUE、 FALSE	A-54
P_First_Run	程序启动时 1 个周期 ON 标志	该程序开始执行后，仅在 1 个任务周期内为 TRUE。 用于每次开始执行程序时需要进行初始处理的情况。 (注) 本系统定义变量在函数中不可使用。	BOOL	TRUE、 FALSE	A-55
P_PRGER	指令错误发生标志	相应的程序或通过该程序调用的 FB 或 FUN 发生指令错误时，持续变为 TRUE 的标志。一旦变为 TRUE，在通过用户程序变为 FALSE 前，将保持 TRUE。	BOOL	TRUE、 FALSE	A-55

## ● 功能分类：版本管理相关

变量名称	名称	功能	数据类型	值的范围	参照页
_UnitVersion	单元版本	保存控制器的单元版本。 在要素编号 0 中保存单元版本的整数部分。 在要素编号 1 中保存单元版本的小数部分。 例 1) 单元版本为 1.08 时，要素编号 0 中保存 1、要素编号 1 中保存 8。 例 2) 单元版本为 1.10 时，要素编号 0 中保存 1、要素编号 1 中保存 10。	ARRAY[0..1] OF USINT	0 ~ 99	A-55
_HardwareRevision	硬件修订版本	保存控制器的硬件修订版本。 硬件修订版本为“无”时，保存“-”，其他情况下保存“A”~“Z”。	STRING[2]	'-', 'A' ~ 'Z'	A-55

● 功能分类：自诊断相关

变量名称	名称	功能	数据类型	值的范围	参照页
_SelfTest_HighTemperature	CPU 单元内部温度超限标志	控制器内部温度为高温时，变为 TRUE。	BOOL	TRUE、FALSE	A-56
_SelfTest_LowBattery	电池电压低标志	电池拆下或电池电压低时变为 TRUE。	BOOL	TRUE、FALSE	A-56
_SelfTest_LowFanRevolution	风扇转速降低标志	风扇拆下或风扇的转速下降时变为 TRUE。	BOOL	TRUE、FALSE	A-56

## A-3-2 PLC 功能模块（类别名称：\_PLC）

### ● 功能分类：调试相关

变量名称 成员名称	名称	功能	数据类型	值的范围	参照页
_PLC_TraceSta[0..3]			_sTRACE_ STA		A-57
.IsStart	追踪执行中 ON 标志	开始追踪时，变为 TRUE。 (注) 本系统定义变量在用户程序中不可使用。仅在通过 Sysmac Studio 监视数据追踪状态时使用。	BOOL	TRUE、 FALSE	A-57
.IsComplete	追踪完成时 ON 标志	追踪执行完成时，变为 TRUE。 (注) 本系统定义变量在用户程序中不可使用。仅在通过 Sysmac Studio 监视数据追踪状态时使用。	BOOL	TRUE、 FALSE	A-57
.IsTrigger	追踪触发监视标志	追踪触发条件成立时，变为 TRUE。 下次开始追踪时，变为 FALSE。 (注) 本系统定义变量在用户程序中不可使用。仅在通过 Sysmac Studio 监视数据追踪状态时使用。	BOOL	TRUE、 FALSE	A-57
.ParamErr	追踪参数异常标志	开始追踪时，若追踪设定异常，则变为 TRUE。 正常时，为 FALSE。 (注) 本系统定义变量在用户程序中不可使用。仅在通过 Sysmac Studio 监视数据追踪状态时使用。	BOOL	TRUE、 FALSE	A-57

### ● 功能分类：异常相关

变量名称	名称	功能	数据类型	值的范围	参照页
_PLC_ErrSta	PLC 功能模块异常状态	发生与 PLC 功能模块相关的控制器异常时，变为 TRUE。 未发生与 PLC 功能模块相关的控制器异常时，变为 FALSE。 各个位的含义请参考后面介绍的“异常状态中各个位的含义”。	WORD	16#0000 ~ 16#00F0	A-58

### A-3-3 运动控制功能模块（类别名称：\_MC）

#### ● 功能分类：运动控制功能相关

变量名称	名称	功能	数据类型	值的范围	参照页
_MC_ErrSta	运动控制功能模块的异常状态	表示运动控制功能模块中检测到的异常状态。 可在用户程序中使用（直接参照）。各个位的含义请参考后面介绍的“异常状态中各个位的含义”。	WORD	16#0000 ~ 16#40F0	A-59
_MC_ComErrSta	共通异常状态	表示在运动控制的共通处理中检测到的异常状态。 可在用户程序中使用（直接参照）。各个位的含义请参考后面介绍的“异常状态中各个位的含义”。	WORD	16#0000 ~ 16#00F0	A-59
_MC_AX_ErrSta	轴异常状态	表示以轴为单位检测到的异常状态。最多可显示 64 个轴。 可在用户程序中使用（直接参照）。各个位的含义请参考后面介绍的“异常状态中各个位的含义”。	ARRAY [0..63] OF WORD	16#0000 ~ 16#00F0	A-59
_MC_GRP_ErrSta	轴组异常状态	表示以轴组为单位检测到的异常状态。最多可显示 32 个轴组。 可在用户程序中使用（直接参照）。各个位的含义请参考后面介绍的“异常状态中各个位的含义”。	ARRAY [0..31] OF WORD	16#0000 ~ 16#00F0	A-59
_MC_COM	通用变量	进行运动控制功能模块共通的状态监视。结构体的成员详情请参考□□《NY 系列指令基准手册 运动篇 (SBCE-CN5-380)》。	_sCOMMON_REF	—	A-60
_MC_GRP	轴组变量	NY 系列控制器时，进行运动控制指令的轴组指定、多轴协调控制的状态监视、多轴协调控制的设定监视。 若用 Sysmac Studio 创建“轴组”，将生成用户自定义名称的轴组变量。 通常使用自定义名称的轴组变量。 结构体的成员详情请参考□□《NY 系列指令基准手册 运动篇 (SBCE-CN5-380)》。	ARRAY[0..31] OF _sGROUP_REF	—	A-60
_MC_AX	轴变量	NY 系列控制器时，进行运动控制指令的轴指定、单轴控制的状态监视、单轴控制的设定监视。 若用 Sysmac Studio 创建“轴”，将生成用户自定义名称的轴变量。 通常使用自定义名称的轴变量。 结构体的成员详情请参考□□《NY 系列指令基准手册 运动篇 (SBCE-CN5-380)》。	ARRAY[0..63] OF _sAXIS_REF	—	A-60

### A-3-4 EtherCAT 主机功能模块（类别名称：\_EC）

#### ● 功能分类：EtherCAT 通信的异常相关

变量名称	名称	功能	数据类型	值的范围	参照页
_EC_ErrSta	内置 EtherCAT 异常	集合了 EtherCAT 主机功能模块异常的系统定义变量。 各个位的含义请参考后面介绍的“异常状态中各个位的含义”。	WORD	16#0000 ~ 16#40F0	A-61
_EC_PortErr	通信端口异常	集合了 EtherCAT 主机用通信端口异常的系统定义变量。 各个位的含义请参考后面介绍的“异常状态中各个位的含义”。	WORD	16#0000 ~ 16#00F0	A-61
_EC_MstrErr	主机异常	集合了 EtherCAT 主机异常和 EtherCAT 主机检测到的从站异常的系统定义变量。 各个位的含义请参考后面介绍的“异常状态中各个位的含义”。	WORD	16#0000 ~ 16#00F0	A-61
_EC_SlavErr	从站异常	集合了 EtherCAT 从站整体异常状态的系统定义变量。 各个位的含义请参考后面介绍的“异常状态中各个位的含义”。	WORD	16#0000 ~ 16#00F0	A-61
_EC_SlavErrTbl	从站异常表	以 EtherCAT 从站为单位通知异常状态。本异常状态表示实际系统构成中以从站为单位的异常状态，本变量的排列与 EtherCAT 从站的节点地址（1 ~ 512）对应，表示发生异常的从站。 各个位的含义请参考后面介绍的“异常状态中各个位的含义”。	Array [1..512] OF WORD	16#0000 ~ 16#00F0	A-62
_EC_MacAdrErr	MAC 地址异常	MAC 地址不正确时，变为 TRUE。	BOOL	TRUE、FALSE	A-62
_EC_LanHwErr	通信控制器异常	检测到通信控制器的硬件异常时，变为 TRUE。	BOOL	TRUE、FALSE	A-62
_EC_LinkOffErr	链接关闭异常	检测到通信控制器未建立链接时，变为 TRUE。	BOOL	TRUE、FALSE	A-62
_EC_NetCfgErr	网络构成信息异常	检测到网络配置信息不正确时，变为 TRUE。	BOOL	TRUE、FALSE	A-62
_EC_NetCfgCmpErr	网络构成核对异常	网络配置信息和网络实际配置不一致时，变为 TRUE。	BOOL	TRUE、FALSE	A-62
_EC_NetTopologyErr	网络构成异常	发生网络配置异常（连接台数超限、环形连接）时，变为 TRUE。	BOOL	TRUE、FALSE	A-63
_EC_PDCommErr	过程数据通信异常	过程数据通信中，发生从站意外脱离、加入，或检测到从站 WDT 异常时，变为 TRUE。	BOOL	TRUE、FALSE	A-63
_EC_PDTimeoutErr	过程数据接收超时异常	若在接收过程数据时发生超时，则变为 TRUE。	BOOL	TRUE、FALSE	A-63
_EC_PDSendErr	过程数据发送异常	发生过程数据发送异常（未在过程数据通信周期内发送、发送抖动超限）时，变为 TRUE。	BOOL	TRUE、FALSE	A-63
_EC_SlavAdrDupErr	从站节点地址重复异常	若从站的节点地址重复，则变为 TRUE。	BOOL	TRUE、FALSE	A-63
_EC_SlavInitErr	从站初始化异常	向从站发出初始化指令时若发生异常，则变为 TRUE。	BOOL	TRUE、FALSE	A-63
_EC_SlavAppErr	从站应用异常	检测到从站的应用程序状态寄存器异常时，变为 TRUE。	BOOL	TRUE、FALSE	A-64
_EC_MsgErr	EtherCAT 信息异常	向不支持消息功能的从站发送消息或向从站发送的消息返回的响应格式异常时，变为 TRUE。	BOOL	TRUE、FALSE	A-64
_EC_SlavEmergErr	紧急消息检测	主机检测到从站发出紧急消息时，变为 TRUE。	BOOL	TRUE、FALSE	A-64
_EC_CommErrTbl	通信异常从站表	以从站节点地址顺序的表格表示。主机检测到异常的从站对应的位置变为 TRUE。	Array [1..512] OF BOOL	TRUE、FALSE	A-64

变量名称	名称	功能	数据类型	值的范围	参照页
_EC_CycleExceeded	EtherCAT 通信周期设定异常	启动时，若无法在设定的通信周期内进行通信，则变为 TRUE。	BOOL	TRUE、FALSE	A-64



## 参考

## 内置 EtherCAT 异常标志的典型关系

变量名称	名称	变量名称	名称	变量名称	名称	事件重要程度
_EC_ErrSta	内置 EtherCAT 异常	_EC_PortErr	通信端口异常	_EC_MacAdrErr	MAC 地址异常	部分停止故障等级
				_EC_LanHwErr	通信控制器异常	
				_EC_LinkOffErr	链接关闭异常	
		_EC_MstrErr	主机异常	_EC_NetCfgErr	网络配置信息异常	轻度故障等级
				_EC_NetCfgCmpErr	网络配置核对异常	
				_EC_NetTopologyErr	网络配置异常	
				_EC_PDCommErr	过程数据通信异常	
				_EC_PDTimeoutErr	过程数据接收超时异常	
				_EC_PDSendErr	过程数据发送异常	
				_EC_SlavAdrDupErr	从站节点节点地址重复	
				_EC_SlavInitErr	从站初始化异常	
				_EC_SlavAppErr	从站应用异常	
				_EC_CommErrTbl	通信异常从站表	
				_EC_CycleExceeded	EtherCAT 通信周期设定异常	
		_EC_MsgErr	EtherCAT 消息异常	监视信息		
_EC_SlavEmergErr	紧急信息检测					
_EC_SlavErr	从站异常	_EC_SlavErrTbl	从站异常表	在从站上定义		

(注) 与 EtherCAT 通信异常相关的所有系统定义变量，在排除异常原因并解除发生的控制器异常（Sysmac Studio 的故障排除 /ResetECError 指令）前，值不会变化。

## ● 功能分类：EtherCAT 通信的状态相关

变量名称	名称	功能	数据类型	值的范围	参照页
_EC_RegSlavTbl	登录从站表	网络配置信息中登录的从站一览。以从站节点地址顺序的表格表示。登录了从站时，对应的位置变为 TRUE。	Array [1..512] OF BOOL	TRUE、 FALSE	A-65
_EC_EntrySlavTbl	网络加入从站表	通知网络加入从站一览。以从站节点地址顺序的表格表示。从站加入网络时，对应的位置变为 TRUE。	Array [1..512] OF BOOL	TRUE、 FALSE	A-65
_EC_MBXSlavTbl	可进行信息通信的从站表	可进行信息通信的从站一览。以从站节点地址顺序的表格表示。从站变为可进行信息通信的状态（可预操作、可安全操作、可操作）时，对应的位置变为 TRUE。 （注）与 EtherCAT 从站进行信息通信时，请参照本数据，确认是否可与相应的从站进行信息通信。	Array [1..512] OF BOOL	TRUE、 FALSE	A-65
_EC_PDSlavTbl	过程数据通信中从站表	正在进行过程数据通信的从站一览。以从站节点地址顺序的表格表示。从站的 IN、OUT 均变为过程数据有效状态（可操作）时，对应的位置变为 TRUE。 （注）控制 EtherCAT 从站时，请参照本数据，确认对应从站数据的有效性。	Array [1..512] OF BOOL	TRUE、 FALSE	A-65
_EC_DisconnSlavTbl	脱离指令中的从站表	以从站节点地址顺序的表格表示。脱离指令中的从站对应的位置变为 TRUE。	Array [1..512] OF BOOL	TRUE、 FALSE	A-66
_EC_DisableSlavTbl	无效设定从站表	以从站节点地址顺序的表格表示。设定为无效的从站对应的位置变为 TRUE。	Array [1..512] OF BOOL	TRUE、 FALSE	A-66
_EC_PDActive	过程数据通信状态	正在与所有从站 (*) 进行过程数据通信时，变为 TRUE。 * 不含设定为无效的从站。	BOOL	TRUE、 FALSE	A-66
_EC_PktMonStop	信息包监视停止	信息包监视停止时变为 TRUE。	BOOL	TRUE、 FALSE	A-66
_EC_LinkStatus	链路状态	通信控制器的链接状态为 Link ON 时，变为 TRUE。	BOOL	TRUE、 FALSE	A-66
_EC_PktSaving	信息包数据文件保存中	表示是否正在执行信息包数据文件保存处理。 TRUE：正在执行信息包数据文件保存处理 FALSE：未执行信息包数据文件保存处理	BOOL	TRUE、 FALSE	A-67
_EC_InDataInvalid	输入数据无效	主固定周期任务中执行的过程数据通信未正常进行，输入值不正确时，变为 TRUE。	BOOL	TRUE、 FALSE	A-67
_EC_InData1Invalid	输入数据 1 无效	主固定周期任务中执行的过程数据通信未正常进行，输入值不正确时，变为 TRUE。	BOOL	TRUE、 FALSE	A-67

（注）与 EtherCAT 通信的状态相关的所有系统定义变量均表示当前的状态。

● 功能分类：EtherCAT 通信的诊断统计信息相关

变量名称	名称	功能	数据类型	值的范围	参照页
_EC_StatisticsLogEnable	诊断统计信息日志许可	开始诊断统计信息日志功能时，从 FALSE 变为 TRUE。 结束诊断统计信息日志功能时，从 TRUE 变为 FALSE。	BOOL	TRUE、FALSE	A-68
_EC_StatisticsLogCycleSec	诊断统计信息日志保存周期	在诊断统计信息日志功能中，以秒为单位指定统计诊断信息的写入周期。 若指定为“0”，则仅在结束诊断统计信息日志功能时写入 1 次。 (注) 正在执行诊断统计信息日志功能时，即使变更本系统定义变量的值，正在执行的写入周期也不会发生变化。	UINT	0、30 ~ 1800	A-68
_EC_StatisticsLogBusy	诊断统计信息日志执行状态标志	正在执行诊断统计信息日志功能时变为 TRUE。	BOOL	TRUE、FALSE	A-68
_EC_StatisticsLogErr	诊断统计信息日志异常结束标志	开始诊断统计信息日志功能失败或执行过程中写入日志失败时，变为 TRUE。 开始诊断统计信息日志功能后，若 _EC_StatisticsLogBusy (诊断统计信息日志执行状态标志) 变为 FALSE，则本标志的值确定。 异常结束可能有以下原因。 <ul style="list-style-type: none"> <li>• 虚拟 SD 存储卡的容量达到上限，无法将记录添加到日志文件中。</li> <li>• 未识别虚拟 SD 存储卡。</li> <li>• 由于 _EC_StatisticsLogCycleSec (诊断统计信息日志保存周期) 中指定的值不正确，因此无法开始。</li> </ul>	BOOL	TRUE、FALSE	A-68

## A-3-5 EtherNet/IP 功能模块（类别名称：\_EIP）

## ● 功能分类：EtherNet/IP 通信的异常相关

变量名称	名称	功能	数据类型	值的范围	参照页
_EIP_ErrSta	内置 EtherNet/IP 异常	内置 EtherNet/IP 端口的异常状态变量。 NY 系列控制器时，集合了以下异常标志。 <ul style="list-style-type: none"> <li>• _EIP1_PortErr（通信端口 1 异常）</li> <li>• _EIPIn1_PortErr（内部通信端口 1 异常）</li> <li>• _EIP_CipErr（CIP 通信异常）</li> <li>• _EIP_TcpAppErr（TCP 应用通信异常）</li> </ul> （注）各个位的含义请参考后面介绍的“异常状态中各个位的含义”。	WORD	16#0000 ~ 16#00F0	A-69
_EIP_PortErr	通信端口异常	通信端口的异常状态变量。 NY 系列控制器时，集合了以下异常标志。 <ul style="list-style-type: none"> <li>• _EIP1_MacAdrErr（通信端口 1 MAC 地址异常）</li> <li>• _EIP1_LanHwErr（通信端口 1 通信控制器异常）</li> <li>• _EIP1_EtnCfgErr（通信端口 1 Ethernet 基本设定异常）</li> <li>• _EIP1_IPAdrCfgErr（通信端口 1 IP 地址设定异常）</li> <li>• _EIP1_IPAdrDupErr（通信端口 1 IP 地址重复异常）</li> <li>• _EIP1_BootpErr（通信端口 1 BOOTP 服务器连接异常）</li> <li>• _EIP_DNSCfgErr（DNS 设定异常）</li> <li>• _EIP_DNSSrvErr（DNS 服务器连接异常）</li> <li>• _EIP_IPRTblErr（IP 路由表异常）</li> </ul> （注）“链接断开检测”或“内置 EtherNet/IP 处理异常”时，登录到事件日志中，本异常的位变为 ON。 各个位的含义请参考后面介绍的“异常状态中各个位的含义”。	WORD	16#0000 ~ 16#00F0	A-69
_EIP1_PortErr	通信端口 1 异常	通信端口的异常状态变量。 集合了以下异常标志。 <ul style="list-style-type: none"> <li>• _EIP1_MacAdrErr（通信端口 1 MAC 地址异常）</li> <li>• _EIP1_LanHwErr（通信端口 1 通信控制器异常）</li> <li>• _EIP1_EtnCfgErr（通信端口 1 Ethernet 基本设定异常）</li> <li>• _EIP1_IPAdrCfgErr（通信端口 1 IP 地址设定异常）</li> <li>• _EIP1_IPAdrDupErr（通信端口 1 IP 地址重复异常）</li> <li>• _EIP1_BootpErr（通信端口 1 BOOTP 服务器连接异常）</li> <li>• _EIP_DNSCfgErr（DNS 设定异常）</li> <li>• _EIP_DNSSrvErr（DNS 服务器连接异常）</li> <li>• _EIP_IPRTblErr（IP 路由表异常）</li> </ul> （注）“链接断开检测”或“内置 EtherNet/IP 处理异常”时，登录到事件日志中，本异常的位变为 ON。 各个位的含义请参考后面介绍的“异常状态中各个位的含义”。	WORD	16#0000 ~ 16#00F0	A-69

变量名称	名称	功能	数据类型	值的范围	参照页
_EIPIn1_PortErr	内部通信端口 1 异常	内部通信端口 1 的异常状态变量。 集合了以下异常标志。 <ul style="list-style-type: none"> <li>• _EIPIn1_IPAdrCfgErr (内部通信端口 1 IP 的地址设定异常)</li> <li>• _EIPIn1_IPAdrDupErr (内部通信端口 1 IP 地址重复异常)</li> <li>• _EIP_DNSCfgErr (DNS 设定异常)</li> <li>• _EIP_DNSSrvErr (DNS 服务器连接异常)</li> <li>• _EIP_IPRTblErr (IP 路由表异常)</li> </ul> (注) “链接断开检测”或“内置 EtherNet/IP 处理异常”时, 登录到事件日志中, 本异常的位变为 ON。 各个位的含义请参考后面介绍的“异常状态中各个位的含义”。	WORD	16#0000 ~ 16#00F0	A-70
_EIP_CipErr	CIP 通信异常	CIP 通信的异常状态变量。 集合了以下异常标志。 <ul style="list-style-type: none"> <li>• _EIP_IdentityErr (Identity 信息不正确)</li> <li>• _EIP_TDLinkCfgErr (标签数据链接设定异常)</li> <li>• _EIP_TDLinkOpnErr (建立标签数据链接失败)</li> <li>• _EIP_TDLinkErr (标签数据链接通信异常)</li> <li>• _EIP_TagAdrErr (标签解决异常)</li> <li>• _EIP_MultiSwONErr (多个开关同时 ON 异常)</li> </ul> (注) “标签解决异常”时登录到事件日志中, 本异常变为 ON。各个位的含义请参考后面介绍的“异常状态中各个位的含义”。	WORD	16#0000 ~ 16#00F0	A-70
_EIP_TcpAppErr	TCP 应用通信异常	TCP 应用通信的异常状态变量。 集合了以下异常标志。 <ul style="list-style-type: none"> <li>• _EIP_TcpAppCfgErr (TCP 应用程序设定异常)</li> <li>• _EIP_NTPSrvErr (NTP 服务器连接异常)</li> </ul> (注) 各个位的含义请参考后面介绍的“异常状态中各个位的含义”。	WORD	16#0000 ~ 16#00F0	A-70
_EIP_MacAdrErr	MAC 地址异常	NY 系列控制器时, 表示电源 ON 时发生通信端口 1 的 MAC 地址读取异常。 TRUE: 发生异常 FALSE: 正常	BOOL	TRUE、FALSE	A-70
_EIP1_MacAdrErr	通信端口 1 MAC 地址异常	表示电源 ON 时发生通信端口 1 的 MAC 地址读取异常。 TRUE: 发生异常 FALSE: 正常	BOOL	TRUE、FALSE	A-71
_EIP_LanHwErr	通信控制器异常	NY 系列控制器时, 表示通信端口 1 上发生通信控制器故障。 TRUE: 发生故障 FALSE: 正常	BOOL	TRUE、FALSE	A-71
_EIP1_LanHwErr	通信端口 1 通信控制器异常	表示通信端口 1 上发生通信控制器故障。 TRUE: 发生故障 FALSE: 正常	BOOL	TRUE、FALSE	A-71
_EIP_EtnCfgErr	Ethernet 基本设定异常	NY 系列控制器时, 表示通信端口 1 的 Ethernet 通信速度设定 (Speed / Duplex) 不正确。或者读取失败。 TRUE: 不正确或读取失败 FALSE: 正常	BOOL	TRUE、FALSE	A-71

变量名称	名称	功能	数据类型	值的范围	参照页
_EIP1_EtnCfgErr	通信端口 1 Ethernet 基本设定异常	表示通信端口 1 的 Ethernet 通信速度设定 (Speed / Duplex) 不正确。或者读取失败。 TRUE: 不正确或读取失败 FALSE: 正常	BOOL	TRUE、FALSE	A-71
_EIP_IPAdrCfgErr	IP 地址设定错误	NY 系列控制器时, 表示通信端口 1 的 IP 地址设定异常。 TRUE: <ul style="list-style-type: none"> <li>IP 地址的设定值不正确</li> <li>读取失败</li> <li>从 BOOTP 服务器获取的 IP 地址不匹配</li> </ul> FALSE: 正常	BOOL	TRUE、FALSE	A-72
_EIP1_IPAdrCfgErr	通信端口 1 IP 地址设定异常	表示通信端口 1 的 IP 地址设定异常。 TRUE: <ul style="list-style-type: none"> <li>IP 地址的设定值不正确</li> <li>读取失败</li> <li>从 BOOTP 服务器获取的 IP 地址不匹配</li> </ul> FALSE: 正常	BOOL	TRUE、FALSE	A-72
_EIPIn1_IPAdrCfgErr	内部通信端口 1 IP 地址设定异常	表示内部通信端口 1 的 IP 地址设定异常。 TRUE: <ul style="list-style-type: none"> <li>IP 地址的设定值不正确</li> <li>读取失败</li> <li>从 BOOTP 服务器获取的 IP 地址不匹配</li> </ul> FALSE: 正常	BOOL	TRUE、FALSE	A-72
_EIP_IPAdrDupErr	IP 地址重复异常	NY 系列控制器时, 表示通信端口 1 上发生 IP 地址重复。 TRUE: 发生重复 FALSE: 上述以外	BOOL	TRUE、FALSE	A-72
_EIP1_IPAdrDupErr	通信端口 1 IP 地址重复异常	表示通信端口 1 上发生 IP 地址重复。 TRUE: 发生重复 FALSE: 上述以外	BOOL	TRUE、FALSE	A-72
_EIPIn1_IPAdrDupErr	内部通信端口 1 IP 地址重复异常	表示内部通信端口 1 上发生 IP 地址重复。 TRUE: 发生重复 FALSE: 上述以外	BOOL	TRUE、FALSE	A-73
_EIP_DNSCfgErr	DNS 设定异常	DNS 设定值或 hosts 设定不正确。或者读取失败。 TRUE: 不正确或读取失败 FALSE: 正常	BOOL	TRUE、FALSE	A-73
_EIP_BootpErr	BOOTP 服务器连接异常	NY 系列控制器时, 表示通信端口 1 上发生 BOOTP 服务器连接失败。 TRUE: BOOTP 服务器的连接失败 (超时)。 FALSE: 未进行 BOOTP 设定, 或已设定 BOOTP, 已正常从 BOOTP 服务器获取 IP 地址。	BOOL	TRUE、FALSE	A-73
_EIP1_BootpErr	通信端口 1 BOOTP 服务器连接异常	表示通信端口 1 上发生 BOOTP 服务器连接失败。 TRUE: BOOTP 服务器的连接失败 (超时)。 FALSE: 未进行 BOOTP 设定, 或已设定 BOOTP, 已正常从 BOOTP 服务器获取 IP 地址。	BOOL	TRUE、FALSE	A-73
_EIP_IPRTblErr	IP 路由表异常	NY 系列控制器时, 表示默认网关的设定或 IP 路由表的设定不正确。或者读取失败。 TRUE: 不正确或读取失败 FALSE: 正常	BOOL	TRUE、FALSE	A-73
_EIP_IdentityErr	Identity 信息不正确	NY 系列控制器时, 表示 CIP 通信 1 的 Identity 信息 (用户不可改写) 不正确。或者读取失败。 TRUE: 不正确或读取失败 FALSE: 正常	BOOL	TRUE、FALSE	A-74

变量名称	名称	功能	数据类型	值的范围	参照页
_EIP_TDLinkCfgErr	标签数据链接设定异常	NY 系列控制器时, 表示 CIP 通信 1 的标签数据链接设定值不正确。或者读取失败。 TRUE: 不正确或读取失败 FALSE: 正常	BOOL	TRUE、FALSE	A-74
_EIP_TDLinkOpnErr	建立标签数据链接的连接失败	NY 系列控制器时, 表示建立 CIP 通信 1 的标签数据链接失败。 TRUE: 因以下原因, 导致建立标签数据链接失败。 • 标签数据连接参数中登录的配对节点信息与实际的节点信息不同。 • 配对节点无响应。 FALSE: 上述以外	BOOL	TRUE、FALSE	A-74
_EIP_TDLinkErr	标签数据链接通信异常	NY 系列控制器时, 表示建立 CIP 通信 1 的标签数据链接时发生超时。 TRUE: 发生超时 FALSE: 上述以外	BOOL	TRUE、FALSE	A-74
_EIP_TagAdrErr	标签解决异常	NY 系列控制器时, 表示解决 CIP 通信 1 的标签 (根据标签名称找出对应的地址) 失败。 TRUE: 标签解决 (根据标签名称找出对应的地址) 失败。原因如下: • 网络变量和标签设定之间的大小不同。 • 标签数据链接设定输入输出方向和控制器中变量输入输出方向不一致。 • 控制器中不存在与标签设定相符的网络变量。 FALSE: 上述以外	BOOL	TRUE、FALSE	A-74
_EIP_MultiSwONErr	多个开关同时 ON 异常	NY 系列控制器时, 表示 CIP 通信 1 中有多个开关同时变为 ON。 TRUE: 有多个数据链接开始 / 数据链接停止开关同时变为 TRUE。 FALSE: 上述以外	BOOL	TRUE、FALSE	A-75
_EIP_TcpAppCfgErr	TCP 应用程序设定异常	TRUE: 至少有 1 个或以上 TCP 应用程序功能 (FTP、NTP、SNMP) 的设定值不正确。或者读取失败。 FALSE: 正常	BOOL	TRUE、FALSE	A-75
_EIP_NTPSrvErr	NTP 服务器连接异常	NY 系列控制器时, 始终为 FALSE。	BOOL	TRUE、FALSE	A-75
_EIP_DNSSrvErr	DNS 服务器连接异常	TRUE: DNS 客户端与服务器连接失败 (超时)。 FALSE: 未设定 DNS。或者已设定 DNS 且连接成功。	BOOL	TRUE、FALSE	A-75

## NY 系列中与 EtherNet/IP 异常相关的系统定义变量的层级关系

与 EtherNet/IP 异常相关的系统定义变量的层级关系如下表所示。例如，第 2 层级中的 \_EIP1\_PortErr、\_EIPIn1\_PortErr、\_EIP\_CipErr、\_EIP\_TcpAppErr 有任意一个不是 0 时，其上一层中的 \_EIP\_ErrSta 不为 0。即通过参照上位层级中的系统定义变量值，可知道下面的层级是否发生异常。

1 级菜单		2 级菜单		3 级菜单	
变量名称	名称	变量名称	名称	变量名称	名称
_EIP_ErrSta	内置 EtherNet/IP 异常	_EIP1_PortErr	通信端口 1 异常	_EIP1_MacAdrErr	通信端口 1 MAC 地址异常
				_EIP1_LanHwErr	通信端口 1 通信控制器异常
				_EIP1_EtnCfgErr	通信端口 1 Ethernet 基本设定异常
				_EIP1_IPAdrCfgErr	通信端口 1 IP 地址设定异常
				_EIP1_IPAdrDupErr	通信端口 1 IP 地址重复异常
				_EIP1_BootpErr	通信端口 1 BOOTP 服务器连接异常
				_EIP_DNSCfgErr	DNS 设定异常
				_EIP_DNSSrvErr	DNS 服务器连接异常
				_EIP_IPRTblErr	IP 路由表异常
		_EIPIn1_PortErr	内部通信端口 1 异常	_EIPIn1_IPAdrCfgErr	内部通信端口 1 IP 地址设定异常
				_EIPIn1_IPAdrDupErr	内部通信端口 1 IP 地址重复异常
				_EIP_DNSCfgErr	DNS 设定异常
				_EIP_DNSSrvErr	DNS 服务器连接异常
				_EIP_IPRTblErr	IP 路由表异常
		_EIP_CipErr	CIP 通信异常	_EIP_IdentityErr	Identity 信息不正确
				_EIP_TDLinkCfgErr	标签数据链接设定异常
				_EIP_TDLinkOpnErr	建立标签数据链接的连接失败
				_EIP_TDLinkErr	标签数据链接通信异常
				_EIP_TagAdrErr	标签解决异常
		_EIP_TcpAppErr	TCP 应用通信异常	_EIP_MultiSwONErr	多个开关同时 ON
_EIP_TcpAppCfgErr	TCP 应用程序设定异常				
				_EIP_NTPSrvErr	NTP 服务器连接异常

(注) 变量名称以“\_EIP1”开始的系统定义变量与变量名称以“\_EIP”开始的同名系统定义变量参照相同的值。例如，\_EIP1\_PortErr（通信端口 1 异常）与 \_EIP\_PortErr（通信端口异常）参照相同的值。

### ● 功能分类：EtherNet/IP 通信的状态相关

变量名称	名称	功能	数据类型	值的范围	参照页
_EIP_EtnOnlineSta	在线	NY 系列控制器时，表示可使用内置 EtherNet/IP 端口中通信端口 1 的通信功能（链接 ON、IP 地址已确定。且未发生异常）。 TRUE：可使用 FALSE：因初始化处理中、重新启动中、发生异常、链接断开，导致内置 EtherNet/IP 端口的通信功能无法使用。	BOOL	TRUE、FALSE	A-75
_EIP1_EtnOnlineSta	通信端口 1 在线	表示可使用内置 EtherNet/IP 端口中通信端口 1 的通信功能（链接 ON、IP 地址已确定。且未发生异常）。 TRUE：可使用 FALSE：因初始化处理中、重新启动中、发生异常、链接断开，导致内置 EtherNet/IP 端口的通信功能无法使用。	BOOL	TRUE、FALSE	A-76
_EIPIn1_EtnOnlineSta	内部通信端口 1 在线	表示可使用内置 EtherNet/IP 端口中内部通信端口 1 的通信功能（链接 ON、IP 地址已确定。且未发生异常）。 TRUE：可使用 FALSE：因初始化处理中、重新启动中、发生异常、链接断开，导致内置 EtherNet/IP 端口的通信功能无法使用。	BOOL	TRUE、FALSE	A-76
_EIP_TDLinkRunSta	标签数据链接通信中	NY 系列控制器时，表示 CIP 通信 1 中至少有 1 个或以上的连接正在正常动作。 TRUE：正常动作 FALSE：上述以外	BOOL	TRUE、FALSE	A-76
_EIP_TDLinkAllRunSta	所有标签数据链接通信中	NY 系列控制器时，表示 CIP 通信 1 的所有标签数据链接正在通信。 TRUE：表示作为始发端，所有连接正在进行标签数据链接通信。 FALSE：至少有 1 个或以上的连接发生异常。	BOOL	TRUE、FALSE	A-76
_EIP_RegTargetSta [255]	登录目标节点信息	NY 系列控制器时，表示在内置 EtherNet/IP 的 CIP 通信 1 中登录了连接的节点一览。 本变量仅在始发端为内置 EtherNet/IP 时有效。 排列 [x] 为 TRUE： 已登录与目标节点 ID“x”的节点之间的连接。 排列 [x] 为 FALSE： 未登录与目标节点 ID“x”的节点之间的连接。	ARRAY [0..255] OF BOOL	TRUE、FALSE	A-76
_EIP_EstbTargetSta [255]	正常目标节点信息	NY 系列控制器时，表示在内置 EtherNet/IP 的 CIP 通信 1 中已正常建立连接的节点一览。 排列 [x] 为 TRUE： 已正常与目标节点 ID“x”的节点建立连接。 排列 [x] 为 FALSE： 未与目标节点 ID“x”的节点建立连接或发生异常。	ARRAY [0..255] OF BOOL	TRUE、FALSE	A-77

变量名称	名称	功能	数据类型	值的范围	参照页
_EIP_TargetPLCModeSta [255]	目标 PLC 动作模式	NY 系列控制器时，表示在始发端为内置 EtherNet/IP 的 CIP 通信 1 的连接中，对应目标节点的控制器运行信息。 本区域的信息仅在正常目标节点信息为 TRUE 时有效。若为 FALSE，则保持之前的值。 排列 [x] 为 TRUE： 目标节点 ID“x”的目标控制器正在运行的状态。 排列 [x] 为 FALSE： 上述以外	ARRAY [0..255] OF BOOL	TRUE、FALSE	A-77
_EIP_TargetPLCErr [255]	目标 PLC 异常信息	NY 系列控制器时，表示在始发端为内置 EtherNet/IP 的 CIP 通信 1 的连接中，对应目标节点的控制器本体的异常状态（运行停止、持续异常的逻辑或）。本区域的信息仅在正常目标节点信息为 TRUE 时有效。若为 FALSE，则保持之前的值。 排列 [x] 为 TRUE： 目标节点 ID“x”的目标控制器继续运行或发生停止异常。 排列 [x] 为 FALSE： 上述以外	ARRAY [0..255] OF BOOL	TRUE、FALSE	A-77
_EIP_TargetNodeErr [255]	目标节点异常信息	NY 系列控制器时，表示与 CIP 通信 1 的登录目标节点信息对应的连接未建立或目标控制器发生异常。 本区域仅在登录目标节点信息为 TRUE 时有效。 排列 [x] 为 TRUE： 未正常与目标节点 ID“x”的目标节点建立连接（登录目标节点信息为 TRUE，正常目标节点信息为 FALSE）， 已与目标节点建立连接，但目标控制器发生异常。 排列 [x] 为 FALSE： 目标节点 ID“x”的目标节点未登录（登录目标节点信息为 FALSE），或已与目标节点建立连接，（登录目标节点信息为 TRUE、正常目标节点信息为 TRUE） 但目标控制器发生异常（目标 PLC 异常信息为 TRUE）。	ARRAY [0..255] OF BOOL	TRUE、FALSE	A-77
_EIP_NTPResult	NTP 的动作信息	—	_sNTP_RESULT		A-78
.ExecTime	NTP 的最终动作时间	NY 系列控制器时，保持初始值不变。	DATE_AND_TIME	遵照数据类型	A-78
.ExecNormal	NTP 的动作结果	NY 系列控制器时，保持初始值不变。	BOOL	TRUE、FALSE	A-78



## 参考

## 与目标节点的通信状态

NY 系列控制器和目标节点的通信状态以下 4 个系统定义变量值的组合，按下表决定。

- \_EIP\_RegTargetSta（登录目标节点信息）
- \_EIP\_EstbTargetSta（正常目标节点信息）
- \_EIP\_TargetPLCErr（目标 PLC 异常信息）
- \_EIP\_TargetNodeErr（目标节点异常信息）

_EIP_Reg TargetSta 的值	_EIP_Estb TargetSta 的值	_EIP_Target PLCErr 的值	_EIP_Target NodeErr 的值	与目标节点的通信状态
TRUE	TRUE	FALSE	FALSE	已与目标节点正常建立连接，且目标 PLC 未发生异常。
		TRUE	TRUE	已与目标节点建立连接，但目标 PLC 发生异常。
	FALSE	—	TRUE	未正常与目标节点建立连接。
FALSE	—	—	—	由于未登录目标节点，因此信息无效。

### ● 功能分类：EtherNet/IP 通信的开关相关

变量名称	名称	功能	数据类型	值的范围	参照页
_EIP_TDLINKStartCmd	标签数据链接通信开始开关	NY 系列控制器时，从 FALSE 变为 TRUE 后，开始 CIP 通信 1 的标签数据链接。开始标签数据链接动作后，自动变为 FALSE。 (注) 自动变为 FALSE 前，请勿通过用户程序或 Sysmac Studio 等强制操作本开关。	BOOL	TRUE、FALSE	A-78
_EIP_TDLINKStopCmd	标签数据链接通信停止开关	NY 系列控制器时，从 FALSE 变为 TRUE 后，停止 CIP 通信 1 的标签数据链接。停止标签数据链接动作后，自动变为 FALSE。 (注) 自动变为 FALSE 前，请勿通过用户程序或 Sysmac Studio 等强制操作本开关。	BOOL	TRUE、FALSE	A-78

## A-3-6 异常状态中各个位的含义

以下异常状态中各个位的含义共通。

- “\_ErrSta”（控制器异常状态）
- “\_PLC\_ErrSta”（PLC 功能模块异常状态）
- “\_MC\_ErrSta”（MC 异常状态）
- “\_MC\_ComErrSta”（MC 共通异常状态）
- “\_MC\_AX\_ErrSta”（轴异常状态）
- “\_MC\_GRP\_ErrSta”（轴组异常状态）
- “\_EC\_ErrSta”（内置 EtherCAT 异常）
- “\_EC\_PortErr”（通信端口异常）
- “\_EC\_MstrErr”（主机异常）
- “\_EC\_SlavErr”（从站异常）
- “\_EC\_SlavErrTbl”（从站异常表）
- “\_EIP\_ErrSta”（内置 EtherNet/IP 异常）
- “\_EIP\_PortErr”（通信端口异常）、“\_EIP1\_PortErr”（通信端口 1 异常）
- “\_EIP\_CipErr”（CIP 通信异常）
- “\_EIP\_TcpAppErr”（TCP 应用通信异常）

如下所示。但是，出于实时性及与表示各功能模块异常原因的状态之间的同时性考虑，“\_ErrSta”（控制器异常状态）在用户程序中无法使用。仅在介由通信从外部参考状态时使用。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WORD			-	-	-	-	-	-					-	-	-	-

位	内容
15	保留
14	从站集：表示在事件发生源（各功能模块）的下位层级（单元 / 从站 / 轴 / 轴组等）中，是否发生控制器异常。 TRUE：下位层级中有控制器异常 FALSE：下位层级中无控制器异常 （_MC_ErrSta/_EC_ErrSta 有效。）
13 ~ 8	保留
7	表示是否发生全部停止故障（Major fault）等级的控制器异常。 TRUE：发生全部停止故障等级的控制器异常 FALSE：未发生全部停止故障等级的控制器异常
6	表示是否发生部分停止故障（Partial fault）等级的控制器异常。 TRUE：发生部分停止故障等级的控制器异常 FALSE：未发生部分停止故障等级的控制器异常
5	表示是否发生轻度故障（Minor fault）等级的控制器异常。 TRUE：发生轻度故障等级的控制器异常 FALSE：未发生轻度故障等级的控制器异常
4	表示是否发生监视信息（Observation）等级的控制器异常。 TRUE：发生监视信息等级的控制器异常 FALSE：未发生监视信息等级的控制器异常
3 ~ 0	保留

## A-4 系统定义变量的个别规格

系统定义变量的个别规格表的阅读方法如下。

<b>变量名称</b>	[系统定义变量名称。开头带类别名称]		<b>成员（结构体型时）</b>	[仅在结构体型时表示成员名称]	
<b>名称</b>	[名称]		<b>全局/本地</b>	“全局：全局变量、本地：本地变量”	
<b>功能</b>	[介绍功能]				
<b>数据类型</b>	[表示变量的数据类型]		<b>值的范围</b>	[表示变量值的范围]	
<b>R/W 访问</b>	[R：只读 RW：可读写]	<b>保持</b>	[表示变量的保持属性]	<b>网络公开</b>	[表示变量的网络公开属性]
<b>是否可在用户程序中使用</b>	[表示是否可在用户程序上直接指定后使用]	<b>相关指令</b>	[表示与变量相关的指令。 “用户程序中的使用”为不可时，表示用于访问的专用指令]		

## A-4-1 NY 系列控制器的系统整体（类别名称：无）

### ● 功能分类：时钟相关

变量名称	_CurrentTime				
名称	系统时刻	全局 / 本地		全局	
功能	保存控制器的内部时钟数据。				
数据类型	DATE_AND_TIME		值的范围	DT#2000-01-01-00:00:00 ~ DT#2099-12-31-23:59:59	
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	时钟功能用指令		

### ● 功能分类：任务相关

变量名称	_<任务名称>_Active				
名称	任务执行中标志	全局 / 本地		全局	
功能	正在执行任务时为 TRUE。 任务非执行中时为 FALSE。 (注) 本系统定义变量在用户程序中不可使用。仅在通过 Sysmac Studio 的数据追踪功能浏览任务状态时使用。				
数据类型	BOOL		值的范围	TRUE、FALSE	
R/W 访问	R	保持	非保持	网络公开	非公开
是否可在用户程序中使用	不支持	相关指令	<ul style="list-style-type: none"> <li>ActEventTask</li> </ul> 在用户程序中，只能通过以下专用指令访问。 <ul style="list-style-type: none"> <li>Task_IsActive</li> </ul>		

变量名称	_<任务名称>_LastExecTime				
名称	上次任务执行时间	全局 / 本地		全局	
功能	表示上次执行任务的时间（以 0.1 $\mu$ s 为单位）。 (注) 本系统定义变量在用户程序中不可使用。仅在通过 Sysmac Studio 的数据追踪功能浏览任务状态时使用。				
数据类型	TIME		值的范围	遵照数据类型	
R/W 访问	R	保持	非保持	网络公开	非公开
是否可在用户程序中使用	不支持	相关指令	在用户程序中，只能通过以下专用指令访问。 <ul style="list-style-type: none"> <li>GetMyTaskStatus</li> </ul>		

变量名称	_<任务名称>_MaxExecTime				
名称	任务执行时间最大值	全局 / 本地		全局	
功能	表示任务执行时间的最大值（以 0.1 $\mu$ s 为单位）。 (注) 本系统定义变量在用户程序中不可使用。仅在通过 Sysmac Studio 的数据追踪功能浏览任务状态时使用。				
数据类型	TIME		值的范围	遵照数据类型	
R/W 访问	R	保持	非保持	网络公开	非公开
是否可在用户程序中使用	不支持	相关指令	在用户程序中，只能通过以下专用指令访问。 <ul style="list-style-type: none"> <li>GetMyTaskStatus</li> </ul>		

变量名称	_<任务名称>_MinExecTime				
名称	任务执行时间最小值	全局 / 本地		全局	
功能	表示任务执行时间的最小值（以 0.1 $\mu$ s 为单位）。 (注) 本系统定义变量在用户程序中不可使用。仅在通过 Sysmac Studio 的数据追踪功能浏览任务状态时使用。				
数据类型	TIME		值的范围	遵照数据类型	
R/W 访问	R	保持	非保持	网络公开	非公开
是否可在用户程序中使用	不支持	相关指令	在用户程序中，只能通过以下专用指令访问。 <ul style="list-style-type: none"> <li>GetMyTaskStatus</li> </ul>		

变量名称	_<任务名称>_ExecCount		
名称	任务执行次数	全局 / 本地	全局
功能	表示任务的执行次数。 若超出 4294967295 次, 将恢复为 0, 然后继续计数。 (注) 本系统定义变量在用户程序中不可使用。仅在通过 Sysmac Studio 的数据追踪功能浏览任务状态时使用。		
数据类型	UDINT	值的范围	遵照数据类型
R/W 访问	R	保持	非保持
是否可在用户程序中使用	不支持	相关指令	在用户程序中, 只能通过以下专用指令访问。 • GetMyTaskStatus

变量名称	_<任务名称>_Exceeded		
名称	任务周期超出标志	全局 / 本地	全局
功能	超出任务周期后, 变为 TRUE。 若在任务周期内完成任务, 则变为 FALSE。 (注) 本系统定义变量在用户程序中不可使用。仅在通过 Sysmac Studio 的数据追踪功能浏览任务状态时使用。		
数据类型	BOOL	值的范围	TRUE、FALSE
R/W 访问	R	保持	非保持
是否可在用户程序中使用	不支持	相关指令	在用户程序中, 只能通过以下专用指令访问。 • GetMyTaskStatus

变量名称	_<任务名称>_ExceedCount		
名称	任务周期超出次数	全局 / 本地	全局
功能	保存超出周期的次数。 若当前值超出数据类型的最大值, 当前值将恢复为 0, 然后继续计数。 若超出 4294967295 次, 将恢复为 0, 然后继续计数。 (注) 本系统定义变量在用户程序中不可使用。仅在通过 Sysmac Studio 的数据追踪功能浏览任务状态时使用。		
数据类型	UDINT	值的范围	遵照数据类型
R/W 访问	R	保持	非保持
是否可在用户程序中使用	不支持	相关指令	在用户程序中, 只能通过以下专用指令访问。 • GetMyTaskStatus

## ● 功能分类：异常相关

变量名称	_ErrSta				
名称	控制器异常状态		全局 / 本地	全局	
功能	正在发生控制器异常时为 TRUE。 未发生控制器异常时为 FALSE。 (注) 出于实时性及与表示各功能模块异常原因的状态之间的同时性考虑, 在用户程序中无法使用。仅在介由通信从外部参考状态时使用。各个位的含义请参考 □ 「A-3-6 异常状态中各个位的含义」(P.A-44)。				
数据类型	WORD		值的范围	16#0000 ~ 16#C0F0	
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	不支持	相关指令	<ul style="list-style-type: none"> <li>• ResetPLCError</li> <li>• ResetECError</li> <li>• ResetMCError</li> <li>• MC_Reset</li> <li>• MC_GroupReset</li> </ul> 在用户程序中, 只能通过以下专用指令访问。 <ul style="list-style-type: none"> <li>• GetPLCError</li> <li>• GetECError</li> <li>• GetMCError</li> <li>• GetEIPError</li> </ul>		

变量名称	_AlarmFlag				
名称	用户异常状态		全局 / 本地	全局	
功能	发生用户异常时, 与事件重要程度对应的位将变为 TRUE。 User fault Level 1 ~ 8 对应位 00 ~ 07。 未发生用户异常时, 为 16 进制的 0000。				
数据类型	WORD		值的范围	16#0000 ~ 16#00FF	
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	<ul style="list-style-type: none"> <li>• SetAlarm</li> <li>• ResetAlarm</li> <li>• GetAlarm</li> </ul>		

## ● 功能分类：SD 存储卡相关

变量名称	_Card1Ready				
名称	SD 存储卡可使用标志		全局 / 本地	全局	
功能	NY 系列识别到虚拟 SD 存储卡时, 变为 TRUE。 未识别到时, 为 FALSE。 TRUE: 可使用 FALSE: 不可使用				
数据类型	BOOL		值的范围	TRUE、FALSE	
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	—		

变量名称	_Card1Protect				
名称	SD 存储卡写保护标志		全局 / 本地	全局	
功能	NY 系列中不使用。 始终为 FALSE。				
数据类型	BOOL		值的范围	TRUE、FALSE	
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	—		

变量名称	_Card1Err				
名称	SD 存储卡错误标志	全局 / 本地		全局	
功能	NY 系列中不使用。 始终为 FALSE。				
数据类型	BOOL	值的范围		TRUE、FALSE	
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	—		

变量名称	_Card1Access				
名称	SD 存储卡访问中标志	全局 / 本地		全局	
功能	NY 系列中不使用。 始终为 FALSE。				
数据类型	BOOL	值的范围		TRUE、FALSE	
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	—		

变量名称	_Card1Deteriorated				
名称	SD 存储卡寿命警告标志	全局 / 本地		全局	
功能	NY 系列中不使用。 始终为 FALSE。				
数据类型	BOOL	值的范围		TRUE、FALSE	
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	—		

变量名称	_Card1PowerFail				
名称	SD 存储卡访问中断电标志	全局 / 本地		全局	
功能	正在访问虚拟 SD 存储卡的过程中，若控制器发生断电，则变为 TRUE。 TRUE：正在访问虚拟 SD 存储卡时发生断电 FALSE：正常				
数据类型	BOOL	值的范围		TRUE、FALSE	
R/W 访问	RW	保持	保持 <sup>*1</sup>	网络公开	公开
是否可在用户程序中使用	可以	相关指令	—		

\*1. 有保持属性的系统定义变量，但不在备份功能的对象范围内。

变量名称	_Card1BkupCmd		成员名称	.ExecBkup	
名称	备份启动标志	全局 / 本地		全局	
功能	向虚拟 SD 存储卡备份控制器的数据时，变为 TRUE。 (注) 本系统定义变量在用户程序中不可使用。通过 CIP 信息通信，由触摸屏或上位计算机发出指令时使用。				
数据类型	结构体：_sBKUP_CMD、成员：BOOL		值的范围	TRUE、FALSE	
R/W 访问	RW	保持	非保持	网络公开	公开
是否可在用户程序中使用	不支持	相关指令	—		

变量名称	_Card1BkupCmd			成员名称	.CancelBkup
名称	备份取消标志			全局 / 本地	全局
功能	停止向虚拟 SD 存储卡备份时，变为 TRUE。 (注) 本系统定义变量在用户程序中不可使用。通过 CIP 信息通信，由触摸屏或上位计算机发出指令时使用。				
数据类型	结构体：_sBKUP_CMD、成员：BOOL			值的范围	TRUE、FALSE
R/W 访问	RW	保持	非保持	网络公开	公开
是否可在用户程序中使用	不支持	相关指令	—		

变量名称	_Card1BkupCmd			成员名称	.ExecVefy
名称	核对启动标志			全局 / 本地	全局
功能	核对控制器和虚拟 SD 存储卡中的备份文件时，变为 TRUE。 (注) 本系统定义变量在用户程序中不可使用。通过 CIP 信息通信，由触摸屏或上位计算机发出指令时使用。				
数据类型	结构体：_sBKUP_CMD、成员：BOOL			值的范围	TRUE、FALSE
R/W 访问	RW	保持	非保持	网络公开	公开
是否可在用户程序中使用	不支持	相关指令	—		

变量名称	_Card1BkupCmd			成员名称	.CancelVefy
名称	核对取消标志			全局 / 本地	全局
功能	停止核对控制器和虚拟 SD 存储卡中的备份文件时，变为 TRUE。 (注) 本系统定义变量在用户程序中不可使用。通过 CIP 信息通信，由触摸屏或上位计算机发出指令时使用。				
数据类型	结构体：_sBKUP_CMD、成员：BOOL			值的范围	TRUE、FALSE
R/W 访问	RW	保持	非保持	网络公开	公开
是否可在用户程序中使用	不支持	相关指令	—		

变量名称	_Card1BkupCmd			成员名称	.DirName
名称	目录名			全局 / 本地	全局
功能	指定备份对象的虚拟 SD 存储卡目录名称。 (注) 本系统定义变量在用户程序中不可使用。通过 CIP 信息通信，由触摸屏或上位计算机发出指令时使用。				
数据类型	结构体：_sBKUP_CMD、成员：STRING(64)			值的范围	遵照数据类型
R/W 访问	RW	保持	非保持	网络公开	公开
是否可在用户程序中使用	不支持	相关指令	—		

变量名称	_Card1BkupSta			成员名称	.Done
名称	完成标志			全局 / 本地	全局
功能	备份执行完成时为 TRUE。 (注) 本系统定义变量在用户程序中不可使用。通过 CIP 信息通信，从触摸屏或上位计算机读取时使用。				
数据类型	结构体：_sBKUP_STA、成员：BOOL			值的范围	TRUE、FALSE
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	不支持	相关指令	—		

变量名称	_Card1BkupSta			成员名称	.Active
名称	执行中标志			全局 / 本地	全局
功能	正在执行备份时为 TRUE。 (注) 本系统定义变量在用户程序中不可使用。通过 CIP 信息通信, 从触摸屏或上位计算机读取时使用。				
数据类型	结构体: _sBKUP_STA、成员: BOOL			值的范围	TRUE、FALSE
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	不支持	相关指令	—		

变量名称	_Card1BkupSta			成员名称	.Err
名称	错误标志			全局 / 本地	全局
功能	备份处理异常结束时为 TRUE。 (注) 本系统定义变量在用户程序中不可使用。通过 CIP 信息通信, 从触摸屏或上位计算机读取时使用。				
数据类型	结构体: _sBKUP_STA、成员: BOOL			值的范围	TRUE、FALSE
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	不支持	相关指令	—		

变量名称	_Card1VefySta			成员名称	.Done
名称	完成标志			全局 / 本地	全局
功能	核对执行完成时为 TRUE。 (注) 本系统定义变量在用户程序中不可使用。通过 CIP 信息通信, 从触摸屏或上位计算机读取时使用。				
数据类型	结构体: _sVEFY_STA、成员: BOOL			值的范围	TRUE、FALSE
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	不支持	相关指令	—		

变量名称	_Card1VefySta			成员名称	.Active
名称	执行中标志			全局 / 本地	全局
功能	正在执行核实时为 TRUE。 (注) 本系统定义变量在用户程序中不可使用。通过 CIP 信息通信, 从触摸屏或上位计算机读取时使用。				
数据类型	结构体: _sVEFY_STA、成员: BOOL			值的范围	TRUE、FALSE
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	不支持	相关指令	—		

变量名称	_Card1VefySta			成员名称	.VefyRslt
名称	核对结果标志			全局 / 本地	全局
功能	核对一致时为 TRUE, 核对不一致时为 FALSE。 (注) 本系统定义变量在用户程序中不可使用。通过 CIP 信息通信, 从触摸屏或上位计算机读取时使用。				
数据类型	结构体: _sVEFY_STA、成员: BOOL			值的范围	TRUE、FALSE
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	不支持	相关指令	—		

变量名称	_Card1VefySta			成员名称	.Err
名称	错误标志			全局 / 本地	全局
功能	核对处理异常结束时为 TRUE。 (注) 本系统定义变量在用户程序中不可使用。通过 CIP 信息通信, 从触摸屏或上位计算机读取时使用。				
数据类型	结构体: _sVEFY_STA、成员: BOOL			值的范围	TRUE、FALSE
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	不支持	相关指令	—		

### ● 功能分类：备份相关

变量名称	_BackupBusy				
名称	备份相关功能执行中标志			全局 / 本地	全局
功能	正在执行备份、恢复、核对时为 TRUE。				
数据类型	BOOL			值的范围	TRUE、FALSE
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	—		

### ● 功能分类：电源管理相关

变量名称	_PowerOnHour				
名称	通电时间			全局 / 本地	全局
功能	表示通电时间。 以 1 小时为单位保存控制器的通电时间。 重置时, 请保存 0。 变为 4294967295 后, 不会继续更新。 电源 ON 时不会初始化。				
数据类型	UDINT			值的范围	0 ~ 4294967295
R/W 访问	RW	保持	保持 <sup>*1</sup>	网络公开	公开
是否可在用户程序中使用	可以	相关指令	—		

\*1. 有保持属性的系统定义变量, 但不在备份功能的对象范围内。

变量名称	_PowerOnCount				
名称	断电发生次数			全局 / 本地	全局
功能	表示发生断电的次数。 从第一次打开控制器的电源后, 每次发生断电时累加 (+1)。 重置时, 请保存 0。 变为 4294967295 后, 不会继续更新。 电源 ON 时不会初始化。				
数据类型	UDINT			值的范围	0 ~ 4294967295
R/W 访问	RW	保持	保持 <sup>*1</sup>	网络公开	公开
是否可在用户程序中使用	可以	相关指令	—		

\*1. 有保持属性的系统定义变量, 但不在备份功能的对象范围内。

变量名称	_RetainFail		
名称	断电保持失败标志	全局 / 本地	全局
功能	以下情况下, 将变为 TRUE (断电保持失败)。 <ul style="list-style-type: none"> <li>接通电源时, 电池备份存储器 / 非易失性存储器检查错误时</li> </ul> 以下情况下, 将变为 FALSE (无断电保持失败)。 <ul style="list-style-type: none"> <li>接通电源时, 电池备份存储器 / 非易失性存储器的存储器检查 OK 时</li> <li>下载了用户程序时</li> <li>执行了存储器全部清除时</li> </ul> (注) 无法保持绝对编码器原点位置偏置数据时, 将反映到轴变量的异常状态中, 而不是本标志。		
数据类型	BOOL	值的范围	TRUE、FALSE
R/W 访问	R	保持	非保持
是否可在用户程序中使用	可以	相关指令	—

变量名称	_SelfTest_UPSSignal		
名称	UPS 信号检测标志	全局 / 本地	全局
功能	检测到 UPS 发出的临时停电信号时变为 TRUE。		
数据类型	BOOL	值的范围	TRUE、FALSE
R/W 访问	R	保持	非保持
是否可在用户程序中使用	可以	相关指令	<ul style="list-style-type: none"> <li>IPC_EnableShutdown</li> </ul>

变量名称	_RequestShutdown		
名称	关机请求标志	全局 / 本地	全局
功能	如果在运行过程中按下电源按钮, 将变为 TRUE。		
数据类型	BOOL	值的范围	TRUE、FALSE
R/W 访问	R	保持	非保持
是否可在用户程序中使用	可以	相关指令	<ul style="list-style-type: none"> <li>IPC_EnableShutdown</li> </ul>

## ● 功能分类: OS(Windows) 相关

变量名称	_OSRunning		
名称	OS 运行中标志	全局 / 本地	全局
功能	控制器观测到 OS(Windows) 的状态为运行中时变为 TRUE。		
数据类型	BOOL	值的范围	TRUE、FALSE
R/W 访问	R	保持	非保持
是否可在用户程序中使用	可以	相关指令	<ul style="list-style-type: none"> <li>IPC_GetOSStatus</li> <li>IPC_ShutdownOS</li> <li>IPC_RebootOS</li> </ul>

变量名称	_OSHalted		
名称	OS 停止中标志	全局 / 本地	全局
功能	控制器观测到 OS(Windows) 的状态为停止中时变为 TRUE。		
数据类型	BOOL	值的范围	TRUE、FALSE
R/W 访问	R	保持	非保持
是否可在用户程序中使用	可以	相关指令	<ul style="list-style-type: none"> <li>IPC_GetOSStatus</li> <li>IPC_ShutdownOS</li> <li>IPC_RebootOS</li> </ul>

变量名称	_OSErrorState		
名称	OS 异常状态标志	全局 / 本地	全局
功能	控制器判断 OS(Windows) 的状态为异常发生时变为 TRUE。		
数据类型	BOOL	值的范围	TRUE、FALSE
R/W 访问	R	保持	非保持
是否可在用户程序中使用	可以	相关指令	网络公开 公开
			<ul style="list-style-type: none"> <li>IPC_GetOSStatus</li> <li>IPC_ShutdownOS</li> <li>IPC_RebootOS</li> </ul>

### ● 功能分类：编程相关

变量名称	P_On		
名称	始终 ON 标志	全局 / 本地	全局
功能	表示始终为 ON 的标志。		
数据类型	BOOL	值的范围	TRUE
R/W 访问	R	保持	非保持
是否可在用户程序中使用	可以	相关指令	网络公开 非公开
			—

变量名称	P_Off		
名称	始终 OFF 标志	全局 / 本地	全局
功能	表示始终为 OFF 的标志。		
数据类型	BOOL	值的范围	FALSE
R/W 访问	R	保持	非保持
是否可在用户程序中使用	可以	相关指令	网络公开 非公开
			—

变量名称	P_CY		
名称	进位标志	全局 / 本地	本地
功能	在部分指令中更新的标志。		
数据类型	BOOL	值的范围	TRUE、FALSE
R/W 访问	R	保持	非保持
是否可在用户程序中使用	可以	相关指令	网络公开 非公开
			—

变量名称	P_First_RunMode		
名称	运行开始时 1 个周期 ON 标志	全局 / 本地	本地
功能	控制器的动作模式从程序模式变更为运行模式时，若该程序正在执行，则在 1 个任务周期内为 TRUE。若该程序不是正在执行，则保持 FALSE。用于在控制器开始运行时进行初始处理等情况。 (注) 本系统定义变量在函数中不可使用。		
数据类型	BOOL	值的范围	TRUE、FALSE
R/W 访问	R	保持	非保持
是否可在用户程序中使用	可以	相关指令	网络公开 非公开
			—

变量名称	P_First_Run				
名称	程序启动时 1 个周期 ON 标志	全局 / 本地		本地	
功能	该程序开始执行后, 仅在 1 个任务周期内为 TRUE。 用于每次开始执行程序时需要进行初始处理的情况。 (注) 本系统定义变量在函数中不可使用。				
数据类型	BOOL	值的范围		TRUE、FALSE	
R/W 访问	R	保持	非保持	网络公开	非公开
是否可在用户程序中使用	可以	相关指令	—		

变量名称	P_PRGER				
名称	指令错误发生标志	全局 / 本地		本地	
功能	相应的程序或通过该程序调用的 FB 或 FUN 发生指令错误时, 持续变为 TRUE 的标志。一旦变为 TRUE, 在通过用户程序变为 FALSE 前, 将保持 TRUE。				
数据类型	BOOL	值的范围		TRUE、FALSE	
R/W 访问	RW	保持	非保持	网络公开	非公开
是否可在用户程序中使用	可以	相关指令	—		

### ● 功能分类：版本管理相关

变量名称	_UnitVersion				
名称	单元版本	全局 / 本地		全局	
功能	保存控制器的单元版本。 在要素编号 0 中保存单元版本的整数部分。 在要素编号 1 中保存单元版本的小数部分。 例 1) 单元版本为 1.08 时, 要素编号 0 中保存 1、要素编号 1 中保存 8。 例 2) 单元版本为 1.10 时, 要素编号 0 中保存 1、要素编号 1 中保存 10。				
数据类型	ARRAY[0..1] OF USINT	值的范围		0 ~ 99	
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	—		

变量名称	_HardwareRevision				
名称	硬件修订版本	全局 / 本地		全局	
功能	保存控制器的硬件修订版本。 硬件修订版本为“无”时, 保存“-”, 其他情况下保存“A' ~ 'Z'”。				
数据类型	STRING[2]	值的范围		'-', 'A' ~ 'Z'	
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	—		

## ● 功能分类：自诊断相关

变量名称	_SelfTest_HighTemperature				
名称	CPU 单元内部温度超限标志			全局 / 本地	全局
功能	控制器内部温度为高温时，变为 TRUE。				
数据类型	BOOL			值的范围	TRUE、FALSE
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	—		

变量名称	_SelfTest_LowBattery				
名称	电池电压低标志			全局 / 本地	全局
功能	电池拆下或电池电压低时变为 TRUE。				
数据类型	BOOL			值的范围	TRUE、FALSE
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	—		

变量名称	_SelfTest_LowFanRevolution				
名称	风扇转速降低标志			全局 / 本地	全局
功能	风扇拆下或风扇的转速下降时变为 TRUE。				
数据类型	BOOL			值的范围	TRUE、FALSE
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	—		

## A-4-2 PLC 功能模块（类别名称：\_PLC）

### ● 功能分类：调试相关

变量名称	_PLC_TraceSta[0..3]			成员名称	.IsStart
名称	追踪执行中 ON 标志			全局 / 本地	全局
功能	开始追踪时，变为 TRUE。 (注) 本系统定义变量在用户程序中不可使用。仅在通过 Sysmac Studio 监视数据追踪状态时使用。				
数据类型	结构体：_sTRACE_STA、成员：BOOL			值的范围	TRUE、FALSE
R/W 访问	R	保持	非保持	网络公开	非公开
是否可在用户程序中使用	不支持	相关指令	<ul style="list-style-type: none"> <li>TraceTrig</li> <li>TraceSamp</li> </ul> 在用户程序中，只能通过以下专用指令访问。 <ul style="list-style-type: none"> <li>GetTraceStatus</li> </ul>		

变量名称	_PLC_TraceSta[0..3]			成员名称	.IsComplete
名称	追踪完成时 ON 标志			全局 / 本地	全局
功能	追踪执行完成时，变为 TRUE。 (注) 本系统定义变量在用户程序中不可使用。仅在通过 Sysmac Studio 监视数据追踪状态时使用。				
数据类型	结构体：_sTRACE_STA、成员：BOOL			值的范围	TRUE、FALSE
R/W 访问	R	保持	非保持	网络公开	非公开
是否可在用户程序中使用	不支持	相关指令	<ul style="list-style-type: none"> <li>TraceTrig</li> <li>TraceSamp</li> </ul> 在用户程序中，只能通过以下专用指令访问。 <ul style="list-style-type: none"> <li>GetTraceStatus</li> </ul>		

变量名称	_PLC_TraceSta[0..3]			成员名称	.IsTrigger
名称	追踪触发监视标志			全局 / 本地	全局
功能	追踪触发条件成立时，变为 TRUE。 下次开始追踪时，变为 FALSE。 (注) 本系统定义变量在用户程序中不可使用。仅在通过 Sysmac Studio 监视数据追踪状态时使用。				
数据类型	结构体：_sTRACE_STA、成员：BOOL			值的范围	TRUE、FALSE
R/W 访问	R	保持	保持	网络公开	非公开
是否可在用户程序中使用	不支持	相关指令	<ul style="list-style-type: none"> <li>TraceTrig</li> <li>TraceSamp</li> </ul> 在用户程序中，只能通过以下专用指令访问。 <ul style="list-style-type: none"> <li>GetTraceStatus</li> </ul>		

变量名称	_PLC_TraceSta[0..3]			成员名称	.ParamErr
名称	追踪参数异常标志			全局 / 本地	全局
功能	开始追踪时，若追踪设定异常，则变为 TRUE。 正常时，为 FALSE。 (注) 本系统定义变量在用户程序中不可使用。仅在通过 Sysmac Studio 监视数据追踪状态时使用。				
数据类型	结构体：_sTRACE_STA、成员：BOOL			值的范围	TRUE、FALSE
R/W 访问	R	保持	非保持	网络公开	非公开
是否可在用户程序中使用	不支持	相关指令	在用户程序中，只能通过以下专用指令访问。 <ul style="list-style-type: none"> <li>GetTraceStatus</li> </ul>		

## ● 功能分类：异常相关

变量名称	_PLC_ErrSta				
名称	PLC 功能模块异常状态	全局 / 本地		全局	
功能	发生与 PLC 功能模块相关的控制器异常时，变为 TRUE。 未发生与 PLC 功能模块相关的控制器异常时，变为 FALSE。 各个位的含义请参考 □□「A-3-6 异常状态中各个位的含义」(P.A-44)。				
数据类型	WORD	值的范围		16#0000 ~ 16#00F0	
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	<ul style="list-style-type: none"> <li>• GetPLCError</li> <li>• ResetPLCError</li> </ul> 可通过以下专用指令解除		

## A-4-3 运动控制功能模块（类别名称：\_MC）

### ● 功能分类：运动控制功能相关

变量名称	_MC_ErrSta				
名称	运动控制功能模块的异常状态			全局 / 本地	全局
功能	表示运动控制功能模块中检测到的异常状态。 可在用户程序中使用（直接参照）。各个位的含义请参考 □□「A-3-6 异常状态中各个位的含义」（P.A-44）。				
数据类型	WORD		值的范围	16#0000 ~ 16#40F0	
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	<ul style="list-style-type: none"> <li>• GetMCError</li> <li>• ResetMCError</li> <li>• MC_Reset</li> <li>• MC_GroupReset</li> </ul>		

变量名称	_MC_ComErrSta				
名称	共通异常状态			全局 / 本地	全局
功能	表示在运动控制的共通处理中检测到的异常状态。 可在用户程序中使用（直接参照）。各个位的含义请参考 □□「A-3-6 异常状态中各个位的含义」（P.A-44）。				
数据类型	WORD		值的范围	16#0000 ~ 16#00F0	
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	<ul style="list-style-type: none"> <li>• GetMCError</li> <li>• ResetMCError</li> </ul>		

变量名称	_MC_AX_ErrSta				
名称	轴异常状态			全局 / 本地	全局
功能	表示以轴为单位检测到的异常状态。最多可显示 64 个轴。 可在用户程序中使用（直接参照）。各个位的含义请参考 □□「A-3-6 异常状态中各个位的含义」（P.A-44）。				
数据类型	ARRAY [0..63] OF WORD		值的范围	16#0000 ~ 16#00F0	
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	<ul style="list-style-type: none"> <li>• GetMCError</li> <li>• ResetMCError</li> <li>• MC_Reset</li> </ul>		

变量名称	_MC_GRP_ErrSta				
名称	轴组异常状态			全局 / 本地	全局
功能	表示以轴组为单位检测到的异常状态。最多可显示 32 个轴组。 可在用户程序中使用（直接参照）。各个位的含义请参考 □□「A-3-6 异常状态中各个位的含义」（P.A-44）。				
数据类型	ARRAY [0..31] OF WORD		值的范围	16#0000 ~ 16#00F0	
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	<ul style="list-style-type: none"> <li>• GetMCError</li> <li>• ResetMCError</li> <li>• MC_GroupReset</li> </ul>		

变量名称	_MC_COM				
名称	通用变量	全局 / 本地		全局	
功能	进行运动控制功能模块共通的状态监视。 结构体的成员详情请参考 <a href="#">□□</a> 《NY 系列 指令基准手册 运动篇 (SBCE-CN5-380)》。				
数据类型	_sCOMMON_REF		值的范围	—	
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	—		

变量名称	_MC_GRP				
名称	轴组变量	全局 / 本地		全局	
功能	NY 系列控制器时, 进行运动控制指令的轴组指定、多轴协调控制的状态监视、多轴协调控制的设定监视。 若用 Sysmac Studio 创建“轴组”, 将生成用户自定义名称的轴组变量。 通常使用自定义名称的轴组变量。 结构体的成员详情请参考 <a href="#">□□</a> 《NY 系列 指令基准手册 运动篇 (SBCE-CN5-380)》。				
数据类型	ARRAY[0..31] OF _sGROUP_REF		值的范围	—	
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	—		

变量名称	_MC_AX				
名称	轴变量	全局 / 本地		全局	
功能	NY 系列控制器时, 进行运动控制指令的轴指定、单轴控制的状态监视、单轴控制的设定监视。 若用 Sysmac Studio 创建“轴”, 将生成用户自定义名称的轴变量。 通常使用自定义名称的轴变量。 结构体的成员详情请参考 <a href="#">□□</a> 《NY 系列 指令基准手册 运动篇 (SBCE-CN5-380)》。				
数据类型	ARRAY[0..63] OF _sAXIS_REF		值的范围	—	
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	—		

## A-4-4 EtherCAT 主机功能模块（类别名称：\_EC）

### ● 功能分类：EtherCAT 通信的异常相关

变量名称	_EC_ErrSta				
名称	内置 EtherCAT 异常	全局 / 本地		全局	
功能	集成了 EtherCAT 主机功能模块异常的系统定义变量。 各个位的含义请参考 □□「A-3-6 异常状态中各个位的含义」(P.A-44)。				
数据类型	WORD	值的范围		16#0000 ~ 16#40F0	
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	EtherCAT 异常状态获取指令 • GetECError EtherCAT 异常解除指令 • ResetECError		

变量名称	_EC_PortErr				
名称	通信端口异常	全局 / 本地		全局	
功能	集成了 EtherCAT 主机用通信端口异常的系统定义变量。 各个位的含义请参考 □□「A-3-6 异常状态中各个位的含义」(P.A-44)。				
数据类型	WORD	值的范围		16#0000 ~ 16#00F0	
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	EtherCAT 异常状态获取指令 • GetECError EtherCAT 异常解除指令 • ResetECError		

变量名称	_EC_MstrErr				
名称	主机异常	全局 / 本地		全局	
功能	集成了 EtherCAT 主机异常和 EtherCAT 主机检测到的从站异常的系统定义变量。 各个位的含义请参考 □□「A-3-6 异常状态中各个位的含义」(P.A-44)。				
数据类型	WORD	值的范围		16#0000 ~ 16#00F0	
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	EtherCAT 异常状态获取指令 • GetECError EtherCAT 异常解除指令 • ResetECError		

变量名称	_EC_SlavErr				
名称	从站异常	全局 / 本地		全局	
功能	集成了 EtherCAT 从站整体异常状态的系统定义变量。 各个位的含义请参考 □□「A-3-6 异常状态中各个位的含义」(P.A-44)。				
数据类型	WORD	值的范围		16#0000 ~ 16#00F0	
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	EtherCAT 异常状态获取指令 • GetECError EtherCAT 异常解除指令 • ResetECError		

变量名称	_EC_SlavErrTbl		
名称	从站异常表	全局 / 本地	全局
功能	以 EtherCAT 从站为单位通知异常状态。 本异常状态表示实际系统构成中以从站为单位的异常状态，本变量的排列与 EtherCAT 从站的节点地址（1 ~ 512）对应，表示发生异常的从站。 各个位的含义请参考 □□「A-3-6 异常状态中各个位的含义」（P.A-44）。		
数据类型	Array [1..512] OF WORD	值的范围	16#0000 ~ 16#00F0
R/W 访问	R	保持	非保持
是否可在用户程序中使用	可以	相关指令	EtherCAT 异常状态获取指令 • GetECError EtherCAT 异常解除指令 • ResetECError

变量名称	_EC_MacAdrErr		
名称	MAC 地址异常	全局 / 本地	全局
功能	MAC 地址不正确时，变为 TRUE。		
数据类型	BOOL	值的范围	TRUE、FALSE
R/W 访问	R	保持	非保持
是否可在用户程序中使用	可以	相关指令	EtherCAT 异常解除指令 • ResetECError

变量名称	_EC_LanHwErr		
名称	通信控制器异常	全局 / 本地	全局
功能	检测到通信控制器的硬件异常时，变为 TRUE。		
数据类型	BOOL	值的范围	TRUE、FALSE
R/W 访问	R	保持	非保持
是否可在用户程序中使用	可以	相关指令	EtherCAT 异常解除指令 • ResetECError

变量名称	_EC_LinkOffErr		
名称	链接关闭异常	全局 / 本地	全局
功能	检测到通信控制器未建立链接时，变为 TRUE。		
数据类型	BOOL	值的范围	TRUE、FALSE
R/W 访问	R	保持	非保持
是否可在用户程序中使用	可以	相关指令	EtherCAT 异常解除指令 • ResetECError

变量名称	_EC_NetCfgErr		
名称	网络构成信息异常	全局 / 本地	全局
功能	检测到网络配置信息不正确时，变为 TRUE。		
数据类型	BOOL	值的范围	TRUE、FALSE
R/W 访问	R	保持	非保持
是否可在用户程序中使用	可以	相关指令	EtherCAT 异常解除指令 • ResetECError

变量名称	_EC_NetCfgCmpErr		
名称	网络构成核对异常	全局 / 本地	全局
功能	网络配置信息和网络实际配置不一致时，变为 TRUE。		
数据类型	BOOL	值的范围	TRUE、FALSE
R/W 访问	R	保持	非保持
是否可在用户程序中使用	可以	相关指令	EtherCAT 异常解除指令 • ResetECError

变量名称	_EC_NetTopologyErr		
名称	网络构成异常	全局 / 本地	全局
功能	发生网络配置异常 (连接台数超限、环形连接) 时, 变为 TRUE。		
数据类型	BOOL	值的范围	TRUE、FALSE
R/W 访问	R	保持 非保持	网络公开 公开
是否可在用户程序中使用	可以	相关指令	EtherCAT 异常解除指令 • ResetECError

变量名称	_EC_PDCommErr		
名称	过程数据通信异常	全局 / 本地	全局
功能	过程数据通信中, 发生从站意外脱离、加入, 或检测到从站 WDT 异常时, 变为 TRUE。		
数据类型	BOOL	值的范围	TRUE、FALSE
R/W 访问	R	保持 非保持	网络公开 公开
是否可在用户程序中使用	可以	相关指令	EtherCAT 异常解除指令 • ResetECError

变量名称	_EC_PDTimeoutErr		
名称	过程数据接收超时异常	全局 / 本地	全局
功能	若在接收过程数据时发生超时, 则变为 TRUE。		
数据类型	BOOL	值的范围	TRUE、FALSE
R/W 访问	R	保持 非保持	网络公开 公开
是否可在用户程序中使用	可以	相关指令	EtherCAT 异常解除指令 • ResetECError

变量名称	_EC_PDSendErr		
名称	过程数据发送异常	全局 / 本地	全局
功能	发生过程数据发送异常 (未在过程数据通信周期内发送、发送抖动超限) 时, 变为 TRUE。		
数据类型	BOOL	值的范围	TRUE、FALSE
R/W 访问	R	保持 非保持	网络公开 公开
是否可在用户程序中使用	可以	相关指令	EtherCAT 异常解除指令 • ResetECError

变量名称	_EC_SlavAdrDupErr		
名称	从站节点地址重复异常	全局 / 本地	全局
功能	若从站的节点地址重复, 则变为 TRUE。		
数据类型	BOOL	值的范围	TRUE、FALSE
R/W 访问	R	保持 非保持	网络公开 公开
是否可在用户程序中使用	可以	相关指令	EtherCAT 异常解除指令 • ResetECError

变量名称	_EC_SlavInitErr		
名称	从站初始化异常	全局 / 本地	全局
功能	向从站发出初始化指令时若发生异常, 则变为 TRUE。		
数据类型	BOOL	值的范围	TRUE、FALSE
R/W 访问	R	保持 非保持	网络公开 公开
是否可在用户程序中使用	可以	相关指令	EtherCAT 异常解除指令 • ResetECError

变量名称	_EC_SlavAppErr		
名称	从站应用异常	全局 / 本地	全局
功能	检测到从站的应用程序状态寄存器异常时，变为 TRUE。		
数据类型	BOOL	值的范围	TRUE、FALSE
R/W 访问	R	保持	非保持
是否可在用户程序中使用	可以	相关指令	EtherCAT 异常解除指令 • ResetECError

变量名称	_EC_MsgErr		
名称	EtherCAT 信息异常	全局 / 本地	全局
功能	向不支持消息功能的从站发送消息或向从站发送的消息返回的响应格式异常时，变为 TRUE。		
数据类型	BOOL	值的范围	TRUE、FALSE
R/W 访问	R	保持	非保持
是否可在用户程序中使用	可以	相关指令	CoE 消息 (CoE SDO 读取) • EC_CoESDORead CoE 消息 (CoE SDO 写入) • EC_CoESDOWrite

变量名称	_EC_SlavEmergErr		
名称	紧急信息检测	全局 / 本地	全局
功能	主机检测到从站发出紧急消息时，变为 TRUE。		
数据类型	BOOL	值的范围	TRUE、FALSE
R/W 访问	R	保持	非保持
是否可在用户程序中使用	可以	相关指令	EtherCAT 异常解除指令 • ResetECError

变量名称	_EC_CommErrTbl		
名称	通信异常从站表	全局 / 本地	全局
功能	以从站节点地址顺序的表格表示。主机检测到异常的从站对应的位置变为 TRUE。		
数据类型	Array [1..512] OF BOOL	值的范围	TRUE、FALSE
R/W 访问	R	保持	非保持
是否可在用户程序中使用	可以	相关指令	EtherCAT 异常解除指令 • ResetECError

(注) 与 EtherCAT 通信异常相关的所有系统定义变量，在排除异常原因并解除发生的控制器异常 (Sysmac Studio 的故障排除 /ResetECError 指令) 前，值不会变化。

变量名称	_EC_CycleExceeded		
名称	EtherCAT 通信周期设定异常	全局 / 本地	全局
功能	启动时，若无法在设定的通信周期内进行通信，则变为 TRUE。		
数据类型	BOOL	值的范围	TRUE、FALSE
R/W 访问	R	保持	非保持
是否可在用户程序中使用	可以	相关指令	—

## ● 功能分类：EtherCAT 通信的状态相关

变量名称	_EC_RegSlavTbl				
名称	登录从站表	全局 / 本地		全局	
功能	网络配置信息中登录的从站一览。 以从站节点地址顺序的表格表示。 登录了从站时，对应的位置变为 TRUE。				
数据类型	Array [1..512] OF BOOL		值的范围	TRUE、FALSE	
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	—		

变量名称	_EC_EntrySlavTbl				
名称	网络加入从站表	全局 / 本地		全局	
功能	通知网络加入从站一览。以从站节点地址顺序的表格表示。 从站加入网络时，对应的位置变为 TRUE。				
数据类型	Array [1..512] OF BOOL		值的范围	TRUE、FALSE	
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	—		

变量名称	_EC_MBXSlavTbl				
名称	可进行信息通信的从站表	全局 / 本地		全局	
功能	可进行信息通信的从站一览。以从站节点地址顺序的表格表示。 从站变为可进行信息通信的状态（可预操作、可安全操作、可操作）时，对应的位置变为 TRUE。 （注）与 EtherCAT 从站进行信息通信时，请参照本数据，确认是否可与相应的从站进行信息通信。				
数据类型	Array [1..512] OF BOOL		值的范围	TRUE、FALSE	
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	EtherCAT 从站脱离 • EC_DisconnectSlave EtherCAT 从站再加入 • EC_ConnectSlave		

变量名称	_EC_PDSlavTbl				
名称	过程数据通信中从站表	全局 / 本地		全局	
功能	正在进行过程数据通信的从站一览。以从站节点地址顺序的表格表示。 从站的 IN、OUT 均变为过程数据有效状态（可操作）时，对应的位置变为 TRUE。 （注）控制 EtherCAT 从站时，请参照本数据，确认对应从站数据的有效性。				
数据类型	Array [1..512] OF BOOL		值的范围	TRUE、FALSE	
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	EtherCAT 从站脱离 • EC_DisconnectSlave EtherCAT 从站再加入 • EC_ConnectSlave		

变量名称	_EC_DisconnSlavTbl		
名称	脱离指令中的从站表	全局 / 本地	全局
功能	以从站节点地址顺序的表格表示。 脱离指令中的从站对应的位置变为 TRUE。		
数据类型	Array [1..512] OF BOOL	值的范围	TRUE、FALSE
R/W 访问	R	保持	非保持
网络公开	网络公开	公开	
是否可在用户程序中使用	可以	相关指令	EtherCAT 从站脱离 • EC_DisconnectSlave EtherCAT 从站再加入 • EC_ConnectSlave

变量名称	_EC_DisableSlavTbl		
名称	无效设定从站表	全局 / 本地	全局
功能	以从站节点地址顺序的表格表示。 设定为无效的从站对应的位置变为 TRUE。		
数据类型	Array [1..512] OF BOOL	值的范围	TRUE、FALSE
R/W 访问	R	保持	非保持
网络公开	网络公开	公开	
是否可在用户程序中使用	可以	相关指令	—

变量名称	_EC_PDActive		
名称	过程数据通信状态	全局 / 本地	全局
功能	正在与所有从站*1 进行过程数据通信时，变为 TRUE。		
数据类型	BOOL	值的范围	TRUE、FALSE
R/W 访问	R	保持	非保持
网络公开	网络公开	公开	
是否可在用户程序中使用	可以	相关指令	EtherCAT 从站脱离 • EC_DisconnectSlave EtherCAT 从站再加入 • EC_ConnectSlave

\*1. 不含设定为无效的从站。

变量名称	_EC_PktMonStop		
名称	信息包监视停止	全局 / 本地	全局
功能	信息包监视停止时变为 TRUE。		
数据类型	BOOL	值的范围	TRUE、FALSE
R/W 访问	R	保持	非保持
网络公开	网络公开	公开	
是否可在用户程序中使用	可以	相关指令	信息包监视停止 • EC_StopMon 信息包监视启动 • EC_StartMon

变量名称	_EC_LinkStatus		
名称	链路状态	全局 / 本地	全局
功能	通信控制器的链接状态为 Link ON 时，变为 TRUE。		
数据类型	BOOL	值的范围	TRUE、FALSE
R/W 访问	R	保持	非保持
网络公开	网络公开	公开	
是否可在用户程序中使用	可以	相关指令	—

变量名称	_EC_PktSaving				
名称	信息包数据文件保存中	全局 / 本地		全局	
功能	表示是否正在执行信息包数据文件保存处理。 TRUE: 正在执行信息包数据文件保存处理 FALSE: 未执行信息包数据文件保存处理				
数据类型	BOOL	值的范围		TRUE、FALSE	
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	信息包数据文件保存 • EC_SaveMon		

变量名称	_EC_InDataInvalid				
名称	输入数据无效	全局 / 本地		全局	
功能	主固定周期任务中执行的过程数据通信未正常进行, 输入值不正确时, 变为 TRUE。				
数据类型	BOOL	值的范围		TRUE、FALSE	
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	—		

(注) 与 EtherCAT 通信的状态相关的所有系统定义变量均表示当前的状态。

变量名称	_EC_InData1Invalid				
名称	输入数据 1 无效	全局 / 本地		全局	
功能	主固定周期任务中执行的过程数据通信未正常进行, 输入值不正确时, 变为 TRUE。				
数据类型	BOOL	值的范围		TRUE、FALSE	
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	—		

(注) 与 EtherCAT 通信的状态相关的所有系统定义变量均表示当前的状态。

## ● 功能分类：EtherCAT 通信的诊断统计信息相关

变量名称	_EC_StatisticsLogEnable				
名称	诊断统计信息日志许可	全局 / 本地		全局	
功能	开始诊断统计信息日志功能时，从 FALSE 变为 TRUE。 结束诊断统计信息日志功能时，从 TRUE 变为 FALSE。				
数据类型	BOOL		值的范围	TRUE、FALSE	
R/W 访问	RW	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	—		

变量名称	_EC_StatisticsLogCycleSec				
名称	诊断统计信息日志保存周期	全局 / 本地		全局	
功能	在诊断统计信息日志功能中，以秒为单位指定统计诊断信息的写入周期。 若指定为“0”，则仅在结束诊断统计信息日志功能时写入 1 次。 (注) 正在执行诊断统计信息日志功能时，即使变更本系统定义变量的值，正在执行的写入周期也不会发生变化。				
数据类型	UINT		值的范围	0、30 ~ 1800	
R/W 访问	RW	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	—		

变量名称	_EC_StatisticsLogBusy				
名称	诊断统计信息日志执行状态标志	全局 / 本地		全局	
功能	正在执行诊断统计信息日志功能时变为 TRUE。				
数据类型	BOOL		值的范围	TRUE、FALSE	
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	—		

变量名称	_EC_StatisticsLogErr				
名称	诊断统计信息日志异常结束标志	全局 / 本地		全局	
功能	开始诊断统计信息日志功能失败或执行过程中写入日志失败时，变为 TRUE。 开始诊断统计信息日志功能后，若 _EC_StatisticsLogBusy（诊断统计信息日志执行状态标志）变为 FALSE，则本标志的值确定。 异常结束可能有以下原因。 • 虚拟 SD 存储卡的容量达到上限，无法将记录添加到日志文件中。 • 未识别虚拟 SD 存储卡。 • 由于 _EC_StatisticsLogCycleSec（诊断统计信息日志保存周期）中指定的值不正确，因此无法开始。				
数据类型	BOOL		值的范围	TRUE、FALSE	
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	—		

## A-4-5 EtherNet/IP 功能模块（类别名称：\_EIP）

### ● 功能分类：EtherNet/IP 通信的异常相关

变量名称	_EIP_ErrSta		
名称	内置 EtherNet/IP 异常	全局 / 本地	全局
功能	内置 EtherNet/IP 端口的异常状态变量。 NY 系列控制器时，集合了以下异常标志。 <ul style="list-style-type: none"> <li>• _EIP1_PortErr（通信端口 1 异常）</li> <li>• _EIPIn1_PortErr（内部通信端口 1 异常）</li> <li>• _EIP_CipErr（CIP 通信异常）</li> <li>• _EIP_TcpAppErr（TCP 应用通信异常）</li> </ul> （注）各个位的含义请参考 □ 「A-3-6 异常状态中各个位的含义」（P.A-44）。		
数据类型	WORD	值的范围	16#0000 ~ 16#00F0
R/W 访问	R	保持	非保持
是否可在用户程序中使用	可以	相关指令	在用户程序中，可通过以下专用指令访问。 <ul style="list-style-type: none"> <li>• GetEIPError</li> </ul>

变量名称	_EIP_PortErr		
名称	通信端口异常	全局 / 本地	全局
功能	通信端口的异常状态变量。 NY 系列控制器时，集合了以下异常标志。 <ul style="list-style-type: none"> <li>• _EIP1_MacAdrErr（通信端口 1 MAC 地址异常）</li> <li>• _EIP1_LanHwErr（通信端口 1 通信控制器异常）</li> <li>• _EIP1_EtnCfgErr（通信端口 1 Ethernet 基本设定异常）</li> <li>• _EIP1_IPAdrCfgErr（通信端口 1 IP 地址设定异常）</li> <li>• _EIP1_IPAdrDupErr（通信端口 1 IP 地址重复异常）</li> <li>• _EIP1_BootpErr（通信端口 1 BOOTP 服务器连接异常）</li> <li>• _EIP_DNSCfgErr（DNS 设定异常）</li> <li>• _EIP_DNSSrvErr（DNS 服务器连接异常）</li> <li>• _EIP_IPRTblErr（IP 路由表异常）</li> </ul> （注）“链接断开检测”或“内置 EtherNet/IP 异常”时，登录到事件日志中，本异常的位变为 ON。各个位的含义请参考 □ 「A-3-6 异常状态中各个位的含义」（P.A-44）。		
数据类型	WORD	值的范围	16#0000 ~ 16#00F0
R/W 访问	R	保持	非保持
是否可在用户程序中使用	可以	相关指令	在用户程序中，可通过以下专用指令访问。 <ul style="list-style-type: none"> <li>• GetEIPError</li> </ul>

变量名称	_EIP1_PortErr		
名称	通信端口 1 异常	全局 / 本地	全局
功能	通信端口 1 的异常状态变量。 集合了以下异常标志。 <ul style="list-style-type: none"> <li>• _EIP1_MacAdrErr（通信端口 1 MAC 地址异常）</li> <li>• _EIP1_LanHwErr（通信端口 1 通信控制器异常）</li> <li>• _EIP1_EtnCfgErr（通信端口 1 Ethernet 基本设定异常）</li> <li>• _EIP1_IPAdrCfgErr（通信端口 1 IP 地址设定异常）</li> <li>• _EIP1_IPAdrDupErr（通信端口 1 IP 地址重复异常）</li> <li>• _EIP1_BootpErr（通信端口 1 BOOTP 服务器连接异常）</li> <li>• _EIP_DNSCfgErr（DNS 设定异常）</li> <li>• _EIP_DNSSrvErr（DNS 服务器连接异常）</li> <li>• _EIP_IPRTblErr（IP 路由表异常）</li> </ul> （注）“链接断开检测”或“内置 EtherNet/IP 异常”时，登录到事件日志中，本异常的位变为 ON。各个位的含义请参考 □ 「A-3-6 异常状态中各个位的含义」（P.A-44）。		
数据类型	WORD	值的范围	16#0000 ~ 16#00F0
R/W 访问	R	保持	非保持
是否可在用户程序中使用	可以	相关指令	在用户程序中，可通过以下专用指令访问。 <ul style="list-style-type: none"> <li>• GetEIPError</li> </ul>

变量名称	_EIPIn1_PortErr		
名称	内部通信端口 1 异常	全局 / 本地	全局
功能	内部通信端口 1 的异常状态变量。 集合了以下异常标志。 <ul style="list-style-type: none"> <li>• _EIPIn1_IPAdrCfgErr (内部通信端口 1 IP 的地址设定异常)</li> <li>• _EIPIn1_IPAdrDupErr (内部通信端口 1 IP 地址重复异常)</li> <li>• _EIP_DNSCfgErr (DNS 设定异常)</li> <li>• _EIP_DNSSrvErr (DNS 服务器连接异常)</li> <li>• _EIP_IPRTblErr (IP 路由表异常)</li> </ul> (注) “链接断开检测”或“内置 EtherNet/IP 异常”时，登录到事件日志中，本异常的位变为 ON。各个位的含义请参考 □□ 「A-3-6 异常状态中各个位的含义」(P.A-44)。		
数据类型	WORD	值的范围	16#0000 ~ 16#00F0
R/W 访问	R	保持	非保持
是否可在用户程序中使用	可以	相关指令	在网络公开 在用户程序中，可通过以下专用指令访问。 <ul style="list-style-type: none"> <li>• GetEIPError</li> </ul>

变量名称	_EIP_CipErr		
名称	CIP 通信异常	全局 / 本地	全局
功能	CIP 通信的异常状态变量。 集合了以下异常标志。 <ul style="list-style-type: none"> <li>• _EIP_IdentityErr (Identity 信息不正确)</li> <li>• _EIP_TDLinkCfgErr (标签数据链接设定异常)</li> <li>• _EIP_TDLinkOpnErr (建立标签数据链接失败)</li> <li>• _EIP_TDLinkErr (标签数据链接通信异常)</li> <li>• _EIP_TagAdrErr (标签解决异常)</li> <li>• _EIP_MultiSwONErr (多个开关同时 ON 异常)</li> </ul> (注) “标签解决异常”时登录到事件日志中，本异常变为 ON。各个位的含义请参考 □□ 「A-3-6 异常状态中各个位的含义」(P.A-44)。		
数据类型	WORD	值的范围	16#0000 ~ 16#00F0
R/W 访问	R	保持	非保持
是否可在用户程序中使用	可以	相关指令	在网络公开 在用户程序中，可通过以下专用指令访问。 <ul style="list-style-type: none"> <li>• GetEIPError</li> </ul>

变量名称	_EIP_TcpAppErr		
名称	TCP 应用通信异常	全局 / 本地	全局
功能	TCP 应用通信的异常状态变量。 集合了以下异常标志。 <ul style="list-style-type: none"> <li>• _EIP_TcpAppCfgErr (TCP 应用程序设定异常)</li> <li>• _EIP_NTPSrvErr (NTP 服务器连接异常)</li> </ul> (注) 各个位的含义请参考 □□ 「A-3-6 异常状态中各个位的含义」(P.A-44)。		
数据类型	WORD	值的范围	16#0000 ~ 16#00F0
R/W 访问	R	保持	非保持
是否可在用户程序中使用	可以	相关指令	在网络公开 在用户程序中，可通过以下专用指令访问。 <ul style="list-style-type: none"> <li>• GetEIPError</li> </ul>

变量名称	_EIP_MacAdrErr		
名称	MAC 地址异常	全局 / 本地	全局
功能	NY 系列控制器时，表示电源 ON 时发生通信端口 1 的 MAC 地址读取异常。 TRUE: 发生异常 FALSE: 正常		
数据类型	BOOL	值的范围	TRUE、FALSE
R/W 访问	R	保持	非保持
是否可在用户程序中使用	可以	相关指令	—

变量名称	_EIP1_MacAdrErr				
名称	通信端口 1 MAC 地址异常	全局 / 本地		全局	
功能	表示电源 ON 时发生通信端口 1 的 MAC 地址读取异常。 TRUE: 发生异常 FALSE: 正常				
数据类型	BOOL	值的范围		TRUE、FALSE	
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	—		

变量名称	_EIP_LanHwErr				
名称	通信控制器异常	全局 / 本地		全局	
功能	NY 系列控制器时, 表示通信端口 1 上发生通信控制器故障。 TRUE: 发生故障 FALSE: 正常				
数据类型	BOOL	值的范围		TRUE、FALSE	
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	—		

变量名称	_EIP1_LanHwErr				
名称	通信端口 1 通信控制器异常	全局 / 本地		全局	
功能	表示通信端口 1 上发生通信控制器故障。 TRUE: 发生故障 FALSE: 正常				
数据类型	BOOL	值的范围		TRUE、FALSE	
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	—		

变量名称	_EIP_EtnCfgErr				
名称	Ethernet 基本设定异常	全局 / 本地		全局	
功能	NY 系列控制器时, 表示通信端口 1 的 Ethernet 通信速度设定 (Speed / Duplex) 不正确。或者读取失败。 TRUE: 不正确或读取失败 FALSE: 正常				
数据类型	BOOL	值的范围		TRUE、FALSE	
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	—		

变量名称	_EIP1_EtnCfgErr				
名称	通信端口 1 Ethernet 基本设定异常	全局 / 本地		全局	
功能	表示通信端口 1 的 Ethernet 通信速度设定 (Speed / Duplex) 不正确。或者读取失败。 TRUE: 不正确或读取失败 FALSE: 正常				
数据类型	BOOL	值的范围		TRUE、FALSE	
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	—		

变量名称	_EIP_IPAdrCfgErr		
名称	IP 地址设定错误	全局 / 本地	全局
功能	NY 系列控制器时，表示通信端口 1 的 IP 地址设定异常。 TRUE: <ul style="list-style-type: none"> <li>IP 地址的设定值不正确             <ul style="list-style-type: none"> <li>读取失败</li> <li>从 BOOTP 服务器获取的 IP 地址不匹配</li> </ul> </li> </ul> FALSE: 正常		
数据类型	BOOL	值的范围	TRUE、FALSE
R/W 访问	R	保持	非保持
是否可在用户程序中使用	可以	相关指令	—

变量名称	_EIP1_IPAdrCfgErr		
名称	通信端口 1 IP 地址设定异常	全局 / 本地	全局
功能	表示通信端口 1 的 IP 地址设定异常。 TRUE: <ul style="list-style-type: none"> <li>IP 地址的设定值不正确             <ul style="list-style-type: none"> <li>读取失败</li> <li>从 BOOTP 服务器获取的 IP 地址不匹配</li> </ul> </li> </ul> FALSE: 正常		
数据类型	BOOL	值的范围	TRUE、FALSE
R/W 访问	R	保持	非保持
是否可在用户程序中使用	可以	相关指令	—

变量名称	_EIPIn1_IPAdrCfgErr		
名称	内部通信端口 1 IP 地址设定异常	全局 / 本地	全局
功能	表示内部通信端口 1 的 IP 地址设定异常。 TRUE: <ul style="list-style-type: none"> <li>IP 地址的设定值不正确             <ul style="list-style-type: none"> <li>读取失败</li> <li>从 BOOTP 服务器获取的 IP 地址不匹配</li> </ul> </li> </ul> FALSE: 正常		
数据类型	BOOL	值的范围	TRUE、FALSE
R/W 访问	R	保持	非保持
是否可在用户程序中使用	可以	相关指令	—

变量名称	_EIP_IPAdrDupErr		
名称	IP 地址重复异常	全局 / 本地	全局
功能	NY 系列控制器时，表示通信端口 1 上发生 IP 地址重复。 TRUE: 发生重复 FALSE: 上述以外		
数据类型	BOOL	值的范围	TRUE、FALSE
R/W 访问	R	保持	非保持
是否可在用户程序中使用	可以	相关指令	—

变量名称	_EIP1_IPAdrDupErr		
名称	通信端口 1 IP 地址重复异常	全局 / 本地	全局
功能	表示通信端口 1 上发生 IP 地址重复。 TRUE: 发生重复 FALSE: 上述以外		
数据类型	BOOL	值的范围	TRUE、FALSE
R/W 访问	R	保持	非保持
是否可在用户程序中使用	可以	相关指令	—

变量名称	_EIPIn1_IPAdrDupErr		
名称	内部通信端口 1 IP 地址重复异常	全局 / 本地	全局
功能	表示内部通信端口 1 上发生 IP 地址重复。 TRUE: 发生重复 FALSE: 上述以外		
数据类型	BOOL	值的范围	TRUE、FALSE
R/W 访问	R	保持	非保持
是否可在用户程序中使用	可以	相关指令	—

变量名称	_EIP_DNSCfgErr		
名称	DNS 设定异常	全局 / 本地	全局
功能	DNS 设定值或 hosts 设定不正确。或者读取失败。 TRUE: 不正确或读取失败 FALSE: 正常		
数据类型	BOOL	值的范围	TRUE、FALSE
R/W 访问	R	保持	非保持
是否可在用户程序中使用	可以	相关指令	—

变量名称	_EIP_BootpErr		
名称	BOOTP 服务器连接异常	全局 / 本地	全局
功能	NY 系列控制器时, 表示通信端口 1 上发生 BOOTP 服务器连接失败。 TRUE: BOOTP 服务器的连接失败 (超时)。 FALSE: 未进行 BOOTP 设定, 或已设定 BOOTP, 已正常从 BOOTP 服务器获取 IP 地址。		
数据类型	BOOL	值的范围	TRUE、FALSE
R/W 访问	R	保持	非保持
是否可在用户程序中使用	可以	相关指令	—

变量名称	_EIP1_BootpErr		
名称	通信端口 1 BOOTP 服务器连接异常	全局 / 本地	全局
功能	表示通信端口 1 上发生 BOOTP 服务器连接失败。 TRUE: BOOTP 服务器的连接失败 (超时)。 FALSE: 未进行 BOOTP 设定, 或已设定 BOOTP, 已正常从 BOOTP 服务器获取 IP 地址。		
数据类型	BOOL	值的范围	TRUE、FALSE
R/W 访问	R	保持	非保持
是否可在用户程序中使用	可以	相关指令	—

变量名称	_EIP_IPRTblErr		
名称	IP 路由表异常	全局 / 本地	全局
功能	NY 系列控制器时, 表示默认网关的设定或 IP 路由表的设定不正确。或者读取失败。 TRUE: 不正确或读取失败 FALSE: 正常		
数据类型	BOOL	值的范围	TRUE、FALSE
R/W 访问	R	保持	非保持
是否可在用户程序中使用	可以	相关指令	—

变量名称	_EIP_IdentityErr				
名称	Identity 信息不正确	全局 / 本地		全局	
功能	NY 系列控制器时, 表示 CIP 通信 1 的 Identity 信息 (用户不可改写) 不正确。或者读取失败。 TRUE: 不正确或读取失败 FALSE: 正常				
数据类型	BOOL		值的范围	TRUE、FALSE	
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	—		

变量名称	_EIP_TDLinkCfgErr				
名称	标签数据链接设定异常	全局 / 本地		全局	
功能	NY 系列控制器时, 表示 CIP 通信 1 的标签数据链接设定值不正确。或者读取失败。 TRUE: 不正确或读取失败 FALSE: 正常				
数据类型	BOOL		值的范围	TRUE、FALSE	
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	—		

变量名称	_EIP_TDLinkOpnErr				
名称	建立标签数据链接的连接失败	全局 / 本地		全局	
功能	NY 系列控制器时, 表示建立 CIP 通信 1 的标签数据链接失败。 TRUE: 因以下原因, 导致建立标签数据链接失败。 <ul style="list-style-type: none"> <li>• 标签数据连接参数中登录的配对节点信息与实际的节点信息不同。</li> <li>• 配对节点无响应。</li> </ul> FALSE: 上述以外				
数据类型	BOOL		值的范围	TRUE、FALSE	
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	—		

变量名称	_EIP_TDLinkErr				
名称	标签数据链接通信异常	全局 / 本地		全局	
功能	NY 系列控制器时, 表示建立 CIP 通信 1 的标签数据链接时发生超时。 TRUE: 发生超时 FALSE: 上述以外				
数据类型	BOOL		值的范围	TRUE、FALSE	
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	—		

变量名称	_EIP_TagAdrErr				
名称	标签解决异常	全局 / 本地		全局	
功能	NY 系列控制器时, 表示解决 CIP 通信 1 的标签 (根据标签名称找出对应的地址) 失败。 TRUE: 标签解决 (根据标签名称找出对应的地址) 失败。原因如下: <ul style="list-style-type: none"> <li>• 网络变量和标签设定之间的大小不同。</li> <li>• 标签数据链接设定输入输出方向和控制器中变量输入输出方向不一致。</li> <li>• 控制器中不存在与标签设定相符的网络变量。</li> </ul> FALSE: 上述以外				
数据类型	BOOL		值的范围	TRUE、FALSE	
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	—		

变量名称	_EIP_MultiSwONErr				
名称	多个开关同时 ON 异常	全局 / 本地		全局	
功能	NY 系列控制器时, 表示 CIP 通信 1 中有多个开关同时变为 ON。 TRUE: 有多个数据链接开始 / 数据链接停止开关同时变为 TRUE。 FALSE: 上述以外				
数据类型	BOOL	值的范围		TRUE、FALSE	
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	—		

变量名称	_EIP_TcpAppCfgErr				
名称	TCP 应用程序设定异常	全局 / 本地		全局	
功能	TRUE: 至少有 1 个或以上 TCP 应用程序功能 (FTP、NTP、SNMP) 的设定值不正确。或者读取失败。 FALSE: 正常				
数据类型	BOOL	值的范围		TRUE、FALSE	
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	—		

变量名称	_EIP_NTpsrvErr				
名称	NTP 服务器连接异常	全局 / 本地		全局	
功能	NY 系列控制器时, 始终为 FALSE。				
数据类型	BOOL	值的范围		TRUE、FALSE	
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	—		

变量名称	_EIP_DNSSrvErr				
名称	DNS 服务器连接异常	全局 / 本地		全局	
功能	TRUE: DNS 客户端与服务器连接失败 (超时)。 FALSE: 未设定 DNS。或者已设定 DNS 且连接成功。				
数据类型	BOOL	值的范围		TRUE、FALSE	
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	—		

### ● 功能分类: EtherNet/IP 通信的状态相关

变量名称	_EIP_EtnOnlineSta				
名称	在线	全局 / 本地		全局	
功能	NY 系列控制器时, 表示可使用内置 EtherNet/IP 端口中通信端口 1 的通信功能 (链接 ON、IP 地址已确定。且未发生异常)。 TRUE: 可使用 FALSE: 因初始化处理中、重新启动中、发生异常、链接断开, 导致内置 EtherNet/IP 端口的通信功能无法使用。				
数据类型	BOOL	值的范围		TRUE、FALSE	
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	—		

变量名称	_EIP1_EtnOnlineSta		
名称	通信端口 1 在线	全局 / 本地	全局
功能	表示可使用内置 EtherNet/IP 端口中通信端口 1 的通信功能（链接 ON、IP 地址已确定。且未发生异常）。 TRUE：可使用 FALSE：因初始化处理中、重新启动中、发生异常、链接断开，导致内置 EtherNet/IP 端口的通信功能无法使用。		
数据类型	BOOL	值的范围	TRUE、FALSE
R/W 访问	R	保持	非保持
是否可在用户程序中使用	可以	相关指令	—

变量名称	_EIPIn1_EtnOnlineSta		
名称	内部通信端口 1 在线	全局 / 本地	全局
功能	表示可使用内置 EtherNet/IP 端口中内部通信端口 1 的通信功能（链接 ON、IP 地址已确定。且未发生异常）。 TRUE：可使用 FALSE：因初始化处理中、重新启动中、发生异常、链接断开，导致内置 EtherNet/IP 端口的内部通信端口 1 的通信功能无法使用。		
数据类型	BOOL	值的范围	TRUE、FALSE
R/W 访问	R	保持	非保持
是否可在用户程序中使用	可以	相关指令	—

变量名称	_EIP_TDLinkRunSta		
名称	标签数据链接通信中	全局 / 本地	全局
功能	NY 系列控制器时，表示 CIP 通信 1 中至少有 1 个或以上的连接正在正常动作。 TRUE：正常动作 FALSE：上述以外		
数据类型	BOOL	值的范围	TRUE、FALSE
R/W 访问	R	保持	非保持
是否可在用户程序中使用	可以	相关指令	—

变量名称	_EIP_TDLinkAllRunSta		
名称	所有标签数据链接通信中	全局 / 本地	全局
功能	NY 系列控制器时，表示 CIP 通信 1 的所有标签数据链接正在通信。 TRUE：表示作为始发端，所有连接正在进行标签数据链接通信。 FALSE：至少有 1 个或以上的连接发生异常。		
数据类型	BOOL	值的范围	TRUE、FALSE
R/W 访问	R	保持	非保持
是否可在用户程序中使用	可以	相关指令	—

变量名称	_EIP_RegTargetSta [255]		
名称	登录目标节点信息	全局 / 本地	全局
功能	NY 系列控制器时，表示在内置 EtherNet/IP 的 CIP 通信 1 中登录了连接的节点一览。 本变量仅在始发端为内置 EtherNet/IP 时有效。 排列 [x] 为 TRUE：已登录与目标节点 ID“x”的节点之间的连接。 排列 [x] 为 FALSE：未登录与目标节点 ID“x”的节点之间的连接。		
数据类型	ARRAY [0..255] OF BOOL	值的范围	TRUE、FALSE
R/W 访问	R	保持	非保持
是否可在用户程序中使用	可以	相关指令	—

变量名称	_EIP_EstbTargetSta [255]				
名称	正常目标节点信息	全局 / 本地		全局	
功能	NY 系列控制器时, 表示在内置 EtherNet/IP 的 CIP 通信 1 中已正常建立连接的节点一览。 排列 [x] 为 TRUE: 已正常与目标节点 ID“x”的节点建立连接。 排列 [x] 为 FALSE: 未与目标节点 ID“x”的节点建立连接或发生异常。				
数据类型	ARRAY [0..255] OF BOOL	值的范围		TRUE、FALSE	
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	—		

变量名称	_EIP_TargetPLCModeSta [255]				
名称	目标 PLC 动作模式	全局 / 本地		全局	
功能	NY 系列控制器时, 表示在始发端为内置 EtherNet/IP 的 CIP 通信 1 的连接中, 对应目标节点的控制器运行信息。 本区域的信息仅在正常目标节点信息为 TRUE 时有效。若为 FALSE, 则保持之前的值。 排列 [x] 为 TRUE: 目标节点 ID“x”的目标控制器正在运行的状态。 排列 [x] 为 FALSE: 上述以外				
数据类型	ARRAY [0..255] OF BOOL	值的范围		TRUE、FALSE	
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	—		

变量名称	_EIP_TargetPLCErr [255]				
名称	目标 PLC 异常信息	全局 / 本地		全局	
功能	NY 系列控制器时, 表示在始发端为内置 EtherNet/IP 的 CIP 通信 1 的连接中, 对应目标节点的控制器本体的异常状态 (运行停止、持续异常的逻辑或)。本区域的信息仅在正常目标节点信息为 TRUE 时有效。若为 FALSE, 则保持之前的值。 排列 [x] 为 TRUE: 目标节点 ID“x”的目标控制器继续运行或发生停止异常。 排列 [x] 为 FALSE: 上述以外				
数据类型	ARRAY [0..255] OF BOOL	值的范围		TRUE、FALSE	
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	—		

变量名称	_EIP_TargetNodeErr [255]				
名称	目标节点异常信息	全局 / 本地		全局	
功能	NY 系列控制器时, 表示与 CIP 通信 1 的登录目标节点信息对应的连接未建立或目标控制器发生异常。 本区域仅在登录目标节点信息为 TRUE 时有效。 排列 [x] 为 TRUE: 未正常与目标节点 ID“x”的目标节点建立连接 (登录目标节点信息为 TRUE, 正常目标节点信息为 FALSE), 或已与目标节点建立连接, 但目标控制器发生异常。 排列 [x] 为 FALSE: 目标节点 ID“x”的目标节点未登录 (登录目标节点信息为 FALSE), 或已与目标节点建立连接, (登录目标节点信息为 TRUE、正常目标节点信息为 TRUE), 但目标控制器发生异常 (目标 PLC 异常信息为 TRUE)。				
数据类型	ARRAY [0..255] OF BOOL	值的范围		TRUE、FALSE	
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	—		

变量名称	_EIP_NTPResult			成员名称	.ExecTime
名称	NTP 的最终动作时间			全局 / 本地	全局
功能	NY 系列控制器时, 保持初始值不变。				
数据类型	结构体: _sNTP_RESULT 成员: DATE_AND_TIME			值的范围	遵照数据类型
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	不支持	相关指令	可通过 GetNTPStatus 指令读取		

变量名称	_EIP_NTPResult			成员名称	.ExecNormal
名称	NTP 的动作结果			全局 / 本地	全局
功能	NY 系列控制器时, 保持初始值不变。				
数据类型	BOOL			值的范围	TRUE、FALSE
R/W 访问	R	保持	非保持	网络公开	公开
是否可在用户程序中使用	不支持	相关指令	可通过 GetNTPStatus 指令读取		

### ● 功能分类: EtherNet/IP 通信的开关相关

变量名称	_EIP_TDLINKStartCmd				
名称	标签数据链接通信开始开关			全局 / 本地	全局
功能	NY 系列控制器时, 从 FALSE 变为 TRUE 后, 开始 CIP 通信 1 的标签数据链接。 开始标签数据链接动作后, 自动变为 FALSE。 (注) 自动变为 FALSE 前, 请勿通过用户程序或 Sysmac Studio 等强制操作本开关。				
数据类型	BOOL			值的范围	TRUE、FALSE
R/W 访问	R/W	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	—		

变量名称	_EIP_TDLINKStopCmd				
名称	标签数据链接通信停止开关			全局 / 本地	全局
功能	NY 系列控制器时, 从 FALSE 变为 TRUE 后, 停止 CIP 通信 1 的标签数据链接。 停止标签数据链接动作后, 自动变为 FALSE。 (注) 自动变为 FALSE 前, 请勿通过用户程序或 Sysmac Studio 等强制操作本开关。				
数据类型	BOOL			值的范围	TRUE、FALSE
R/W 访问	R/W	保持	非保持	网络公开	公开
是否可在用户程序中使用	可以	相关指令	—		

## A-5 控制器内数据的属性

断电时、电源接通时、动作模式变更时、发生全部停止等级的控制器异常时，控制器内数据的保持 / 非保持属性如下。

控制器内数据项目		断电时的保持有无和方法 ○：断电保持 ×：断电不保存	电源接通时	状态变更		覆盖保护中的写入	Sysmac Studio 的传送功能 同步功能的对象	可写入的动作模式	运行模式下的重写
				程序模式 ↔ 运行模式变更时	发生全部停止故障等级的控制器异常时				
用户程序	POUs、用户程序执行用 ID (用户程序内)	○ (通过非易失性存储器)	与断电前一样	保持	保持	不支持	○	程序模式 / 运行模式 (在线编辑)	○ 在线编辑
任务设定	任务设置	○ (通过非易失性存储器)	与断电前一样	保持	保持	不支持	○	程序模式	×
变量	变量表 (变量的值除外)	设备变量	与断电前一样	保持	保持	不支持	○	程序模式	×
	用户定义变量	○ (通过非易失性存储器)	与断电前一样	保持	保持	不支持	○	程序模式 / 运行模式 (在线编辑)	○ 在线编辑
数据类型		用户定义数据类型	与断电前一样	保持	保持	不支持	○	程序模式 / 运行模式 (在线编辑)	○ 在线编辑
控制器名称	CPU 单元名称	○ (通过非易失性存储器)	与断电前一样	保持	保持	不支持	○	程序模式 / 运行模式	○
	内置 EtherNet/IP 端口名称	○ (通过非易失性存储器)	与断电前一样	保持	保持	支持	×	程序模式 / 运行模式	○

控制器内数据项目			断电时的保持有无和方法 ○：断电保持 ×：断电不保存	电源接通时	状态变更		覆盖保护中的写入	Sysmac Studio 的传送功能	可写入的动作模式	运行模式下的重写
					程序模式 ↔ 运行模式 变更时	发生全部停止故障等级的控制器异常时		同步功能的对象		
控制器设定	动作设定	动作设定 虚拟 SD 存储卡设定 关机处理等待时间设定 事件日志设定 异常设定	○ (通过非易失性存储器)	与断电前一样	保持	保持	不支持	○	CPU 单元名称： 运行模式 / 程序模式 其他：程序模式	×
	安全设定	电源接通时的保护设定	○ (通过非易失性存储器)	与断电前一样	保持	保持	不支持	○	写保护，其他： 程序模式	○
	内置 EtherNet/IP 端口设定	TCP/IP 内置 EtherNet/IP 端口 LINK 设定 服务设定 SNMP 设定 SNMP 捕捉设定 NTP 设定 FTP 设定 IP 路由表	○ (通过非易失性存储器)	与断电前一样	保持	保持	不支持	○	程序模式	×
		内置 EtherNet/IP 端口用标签数据链接设定	○ (通过非易失性存储器)	与断电前一样	保持	保持	支持	×	程序模式 / 运行模式	×
	FINS 设定	节点地址设定 FINS/UDP FINS/TCP FINS 路由表	○ (通过非易失性存储器)	与断电前一样	保持	保持	不支持	○	程序模式	×
运动设定	轴分配 轴参数设定 轴组参数设定 MC 共通参数设定	○ (通过非易失性存储器)	与断电前一样	保持	保持	不支持	○	程序模式	×	
凸轮数据			○ (通过非易失性存储器)	与断电前一样	保持	保持	不支持	○	程序模式	×
事件设定表	事件设定表	用户异常消息	○ (通过非易失性存储器)	与断电前一样	保持	保持	不支持	○	运行模式 / 程序模式	×
EtherCAT 构成	EtherCAT 网络配置	网络配置信息	○ (通过非易失性存储器)	与断电前一样	保持	保持	不支持	○	程序模式	×

控制器内数据项目			断电时的保持有无和方法 ○：断电保持 ×：断电不保存	电源接通时	状态变更		覆盖保护中的写入	Sysmac Studio 的传送功能	可写入的动作模式	运行模式下的重写
					程序模式 ↔ 运行模式 变更时	发生全部停止故障等级的控制器异常时				
EtherCAT 设定	EtherCAT 设定	主机	○ (通过非易失性存储器)	与断电前一样	保持	保持	不支持	○	程序模式	×
		从站自身的设定	○ 从站决定	—	保持	保持	支持	○	运行模式 / 程序模式	○
操作权限认证			○ (通过非易失性存储器)	与断电前一样	保持	保持	不支持	×	程序模式	×
用户程序执行用 ID (控制器内)			○ (通过非易失性存储器)	与断电前一样	保持	保持	不支持	×	程序模式	×
变量的值	非保持变量的值	用户定义变量 / 设备变量	×	初始值	初始值	初始值	支持	×	运行模式 / 程序模式	○
	保持变量的值	用户定义变量 / 设备变量	○ (通过非易失性存储器)	与断电前一样	保持	保持	支持	×	运行模式 / 程序模式	○
事件日志	履历	系统日志 用户日志	○ (通过非易失性存储器)	与断电前一样	保持	保持	支持	×		○
内置时钟	系统时钟的当前时刻		○ (通过电池)	有电池时： 保持 (继续) 无电池时： 不确定 (也可能停止)	保持 (继续)	保持 (继续)	支持	×	运行模式 / 程序模式	○
绝对编码器原点位置偏置			○ (通过非易失性存储器)	与断电前一样	保持 (继续)	保持 (继续)	支持	×		×

## A-6 变量的存储器确保方法

将结构体型或联合体型变量成员在存储器上的位置与其他设备统一时，需要注意变量的存储器确保方法。主要是在使用结构体型和联合体型变量与其他设备进行以下通信时，需要调整。

- NY 系列控制器和其他控制器之间，进行 EtherNet/IP 的标签数据链接或通过 CIP 消息访问变量时
- 通过结构体型、联合体型变量与 ID 标签等控制器以外的设备进行数据交换时

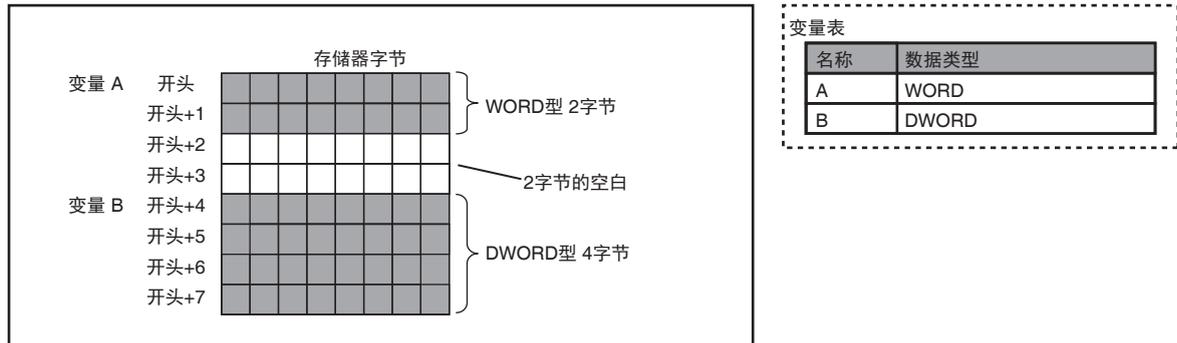
### A-6-1 变量的存储器确保规则

为变量保留的存储器量和存储器上的位置因变量的数据类型而变化。此外，为排列的要素、结构体的成员、联合体的成员保留的存储器量和存储器上的位置，除了数据类型之外，还因排列、结构体、联合体的声明内容而变化。

#### 数据类型的校准和保留的存储器量

不同的数据类型有规定的數據大小。数据大小是指保存该数据类型的值需要的最小存储器容量。

另外，为了有效地访问存储器，与变量对应的存储器由控制器自动配置。因此，为变量保留的存储器总量和这些变量的数据大小总和肯定不一致。例如，声明了 WORD 型和 DWORD 型变量时，数据大小总和为 6 字节，但存储器应如下图所示保留 8 字节。



这种决定存储器上变量配置的信息称为校准。校准根据数据类型而定，为具体变量保留的存储器量和存储器上的位置如下所示。

项目	规格
保留的存储器量	校准的整数倍。但应大于数据大小。
存储器上的位置	从存储器起点起，校准的整数倍位置。

基本数据类型和枚举型的校准和保留的存储器量如下所示。

数据类型	校准 [ 字节 ]	保留的存储器量 [ 字节 ]
BOOL	2	2
BYTE,USINT,SINT	1	1
WORD,UINT,INT	2	2
DWORD,UDINT,DINT	4	4
LWORD,ULINT,LINT	8	8
REAL	4	4
LREAL	8	8
TIME,DATE,TIME_OF_DAT,DATE_ANT_TIME	8	8
STRING[N+1] <sup>*1</sup>	1	N+1
枚举型	4	4

\*1. N 表示处理的最大字符数。例如，半角英文或数字的最大字符数为 10 时，字符串末尾附加 NULL，需要保留 11 个字符的存储器。

为了有效地访问，排列、结构体型、联合体型的各要素和各成员也分别配置到存储器上。排列、结构体型、联合体型的校准和保留的存储器量根据其声明确定如下。

数据类型	校准	保留的存储器量
排列	与要素的数据类型的校准相同	(为要素的数据类型保留的存储器量) × 要素数 <sup>*1</sup>
结构体型	所有成员的校准中，最大的值	将各成员依次配置到该成员数据类型校准的整数倍位置时保留的存储器的总量以上且校准的整数倍
联合体型	所有成员的校准中，最大的值	所有成员保留的存储器量中，最大的值

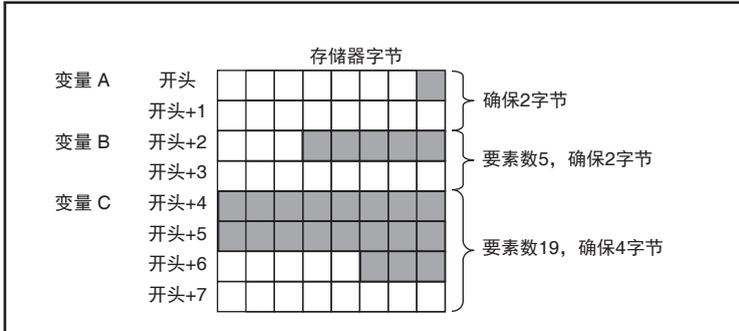
\*1. BOOL 型排列为例外。为 BOOL 型排列保留的存储器量请参考后面的使用注意事项。



**使用注意事项**

**为 BOOL 型排列保留的存储器量**

为 BOOL 型的单独变量、结构体的成员、联合体的成员保留的存储器量为 2 字节。  
但是，BOOL 型排列并非每 1 个要素确保 2 字节的存储器。各要素对应的存储器 1 位 1 位排列。  
排列整体保留 2 字节整数倍的存储空间，包括空格在内。



名称	数据类型
A	BOOL
B	ARRAY [1..5] OF BOOL
C	ARRAY [0..18] OF BOOL

因此，为 BOOL 型排列保留的存储器量可按以下公式计算。要素数为 1 ~ 16 时，保留 2 字节，17 ~ 32 时，保留 4 字节。

$$\text{存储器的量} = 2 \left\lceil \frac{\text{要素数} - 1}{16} \right\rceil + 2$$

括号内的计算结果中，小数点以下舍去

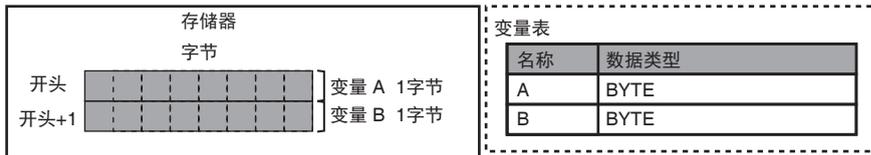
下面以具体案例为例，介绍各数据类型的存储器保留规则。

## 基本数据类型

### ● BYTE 型等校准为 1 字节的变量

以 1 字节为单位的校准，保留 1 字节的存储器。

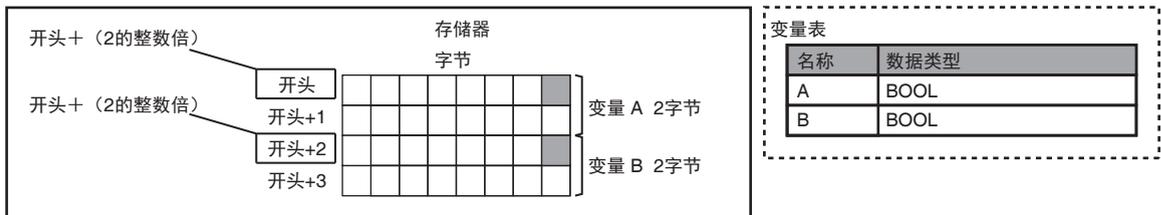
例) BYTE 型变量为连续 2 个时



### ● BOOL 型、WORD 型等校准为 2 字节的变量

以 2 字节为单位的校准，保留 2 字节的存储器。

例) BOOL 型变量为连续 2 个时

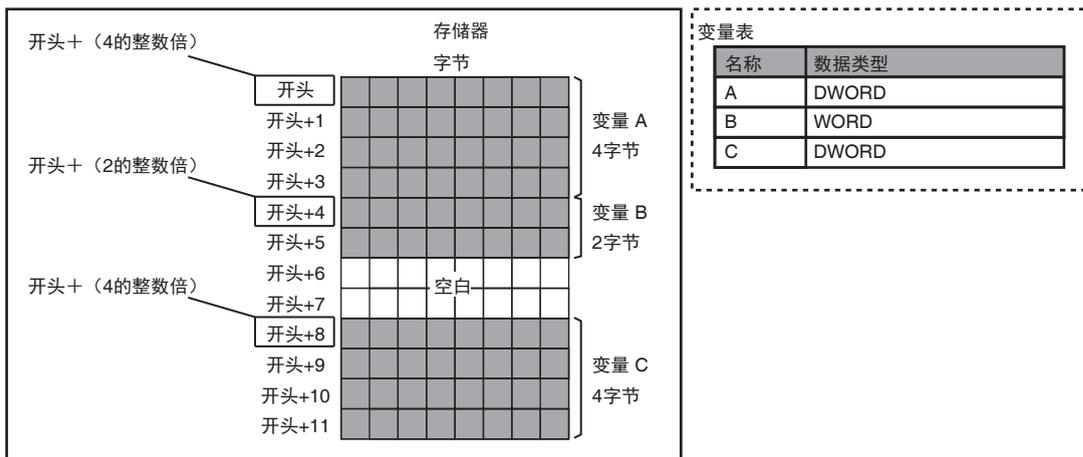


### ● DWORD 型等校准为 4 字节的变量

以 4 字节为单位的校准，保留 4 字节的存储器。

配置了数据了存储器开头位置为 4 字节的整数倍，因此若插入 WORD 型等校准为 2 字节的变量，将出现 2 字节的空白。

例) 变量按 DWORD 型、WORD 型、DWORD 型的顺序连续时

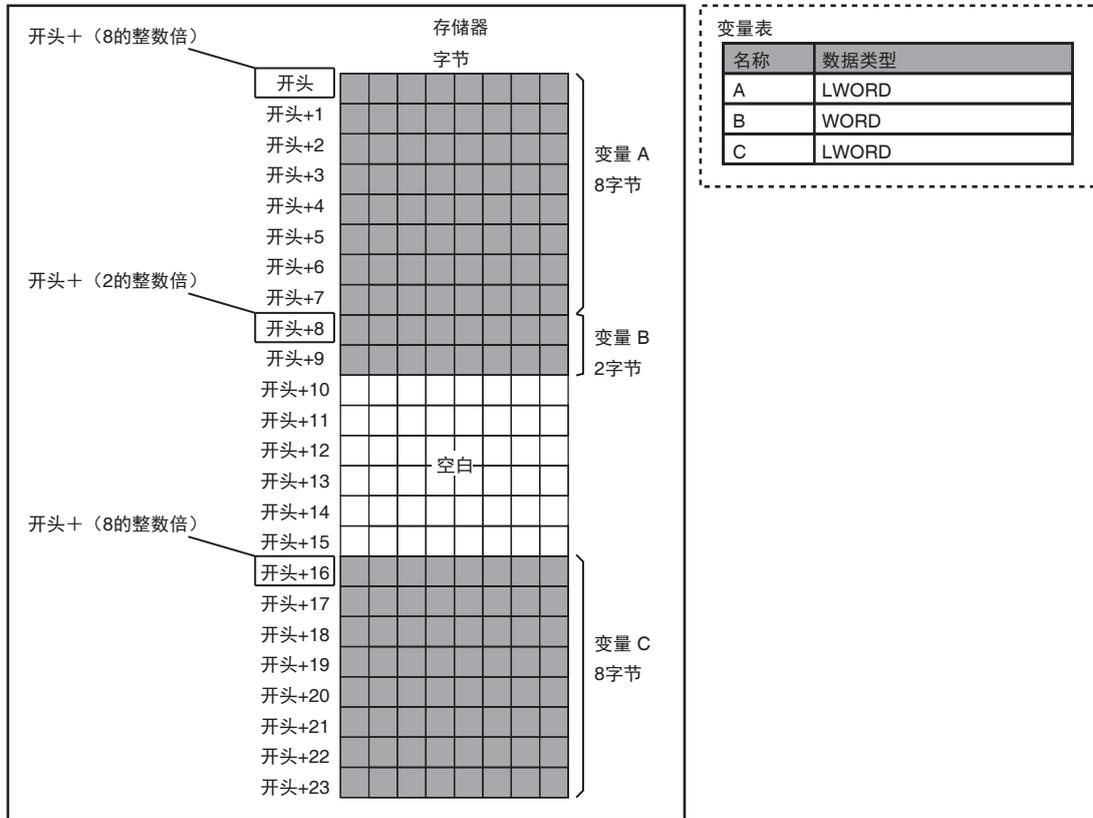


● **LWORD 型等校准为 8 字节的变量**

以 8 字节为单位的校准，保留 8 字节的存储器。

配置了数据了存储器开头位置为 8 字节的整数倍，因此若插入 WORD 型等校准为 2 字节的变量，将出现 6 字节的空白，若插入 DWORD 型等校准为 4 字节的变量，将出现 4 字节的空白。

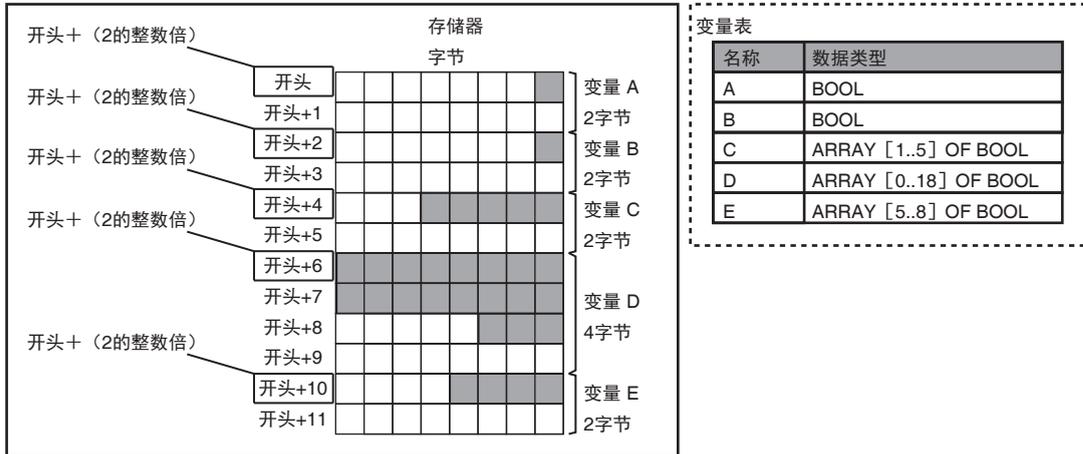
例) 变量按 LWORD 型、WORD 型、LWORD 型的顺序连续时



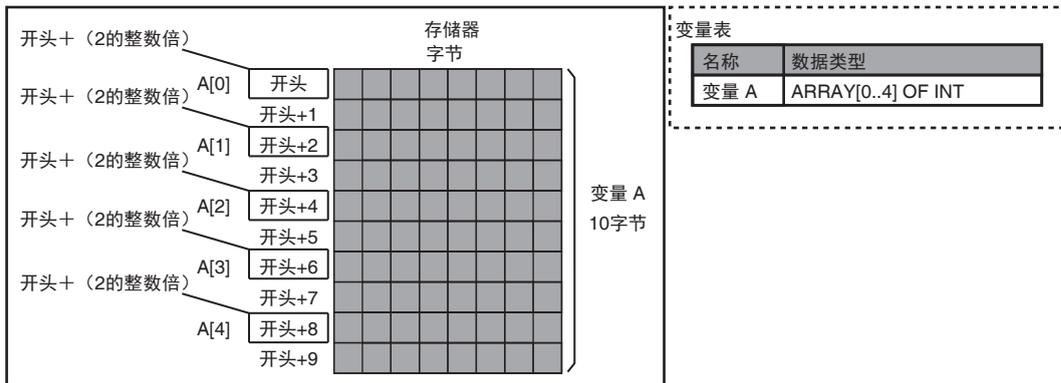
## 排列

排列变量以排列要素数据类型的数据大小为单位，按照要素数连续保留存储空间。排列的校准与要素的数据类型的校准相同。

例) 变量按照 BOOL 型变量 2 个、要素数 5 的 BOOL 型排列、要素数 19 的 BOOL 型排列、要素数 4 的 BOOL 型排列的顺序连续时



例) 要素数 5 的 INT 型排列时

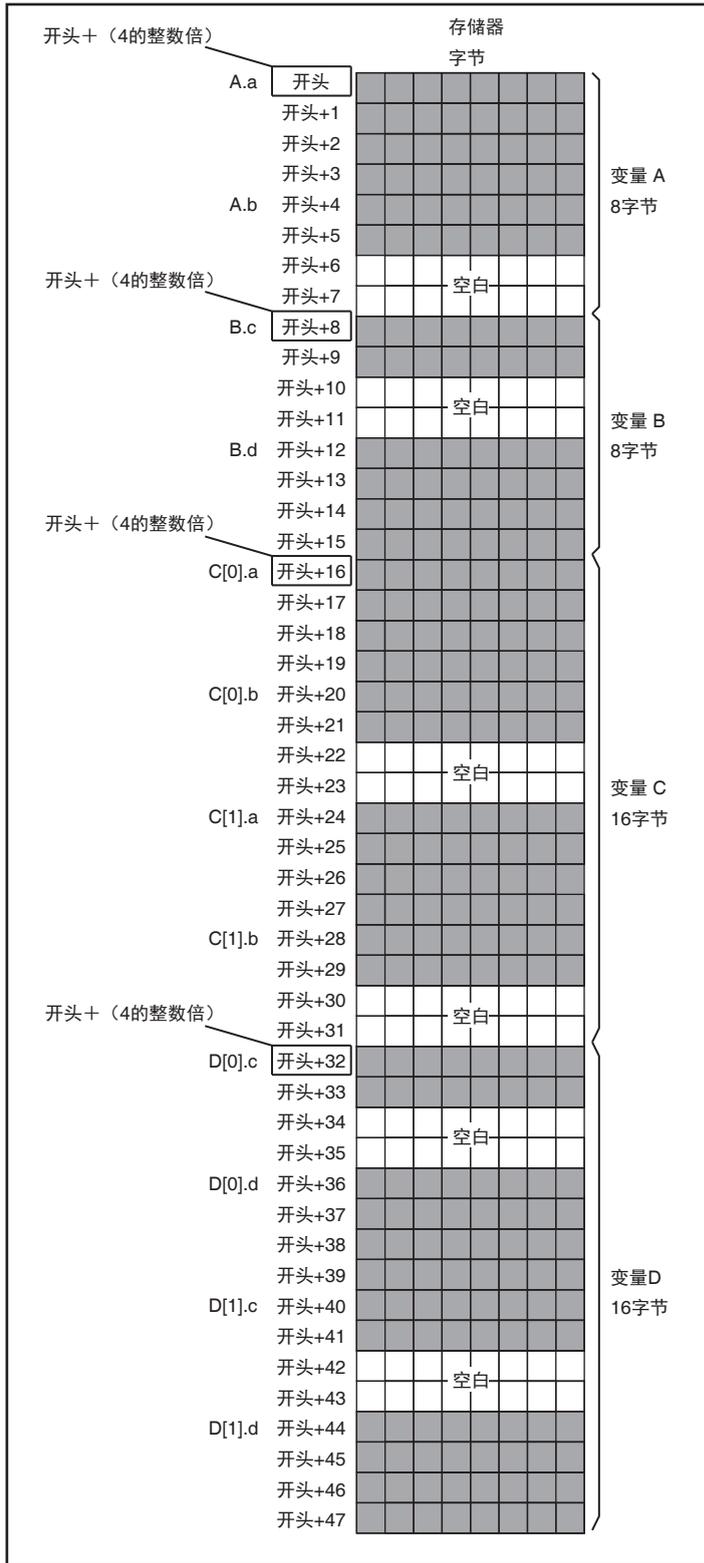


## 结构体型

结构体变量按照各成员声明的顺序配置到存储器中。或者将各成员依次配置到该成员数据类型校准的整数倍位置。因此，成员之间或成员末尾可能会插入存储器的空白。结构体型校准为所有成员的校准中最大的值。保留的存储器量：将各成员依次配置到该成员数据类型校准的整数倍位置时保留的存储器的总量以上且校准的整数倍。

例) 下图 4 种变量声明时，各自的校准和保留的存储器量为以下值。

变量	校准 [ 字节 ]	保留的存储器量 [ 字节 ]
A	4	8
B	4	8
C	4	16
D	4	16



数据类型定义

名称	数据类型
结构体 STR_A	STRUCT
a	DINT
b	INT

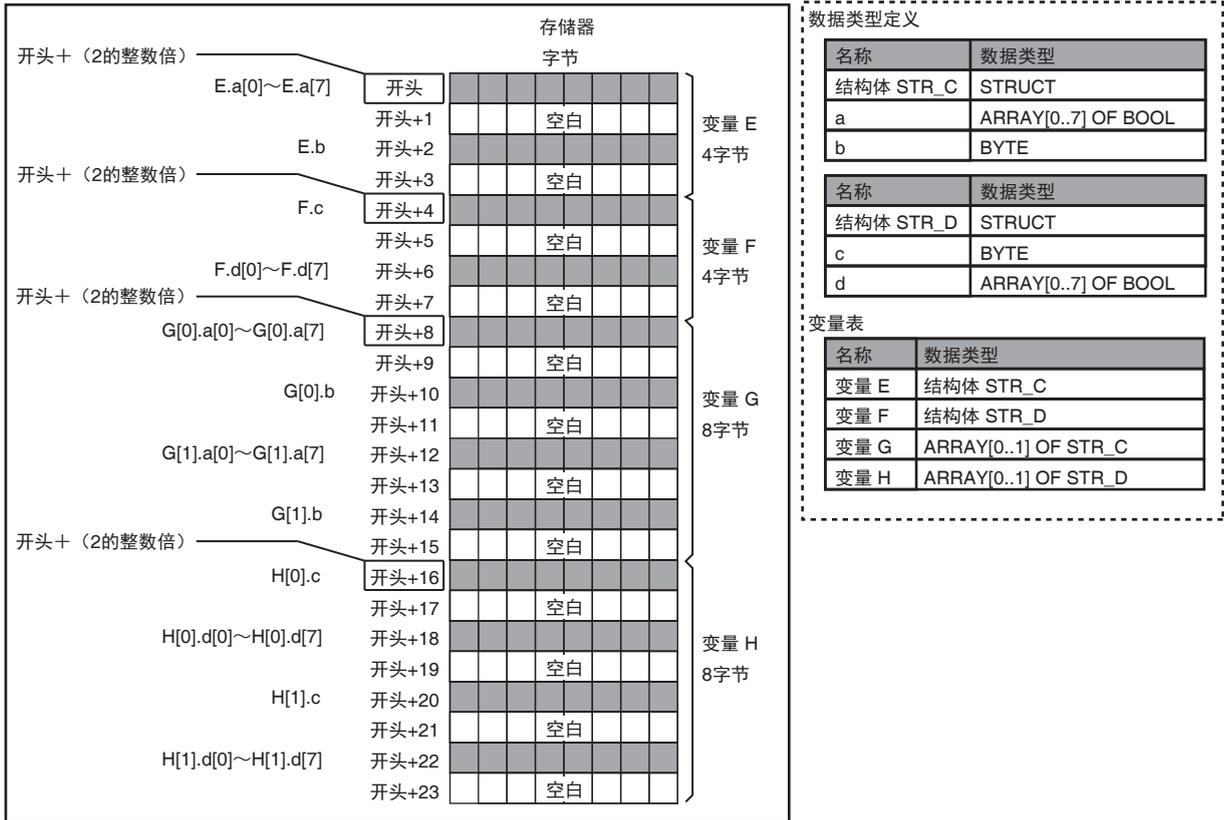
名称	数据类型
结构体 STR_B	STRUCT
c	INT
d	DINT

变量表

名称	数据类型
变量 A	结构体 STR_A
变量 B	结构体 STR_B
变量 C	ARRAY[0..1] OF STR_A
变量 D	ARRAY[0..1] OF STR_B

例) 下图 4 种变量声明时, 各自的校准和保留的存储器量为以下值。

变量	校准 [ 字节 ]	保留的存储器量 [ 字节 ]
E	2	4
F	2	4
G	2	8
H	2	8



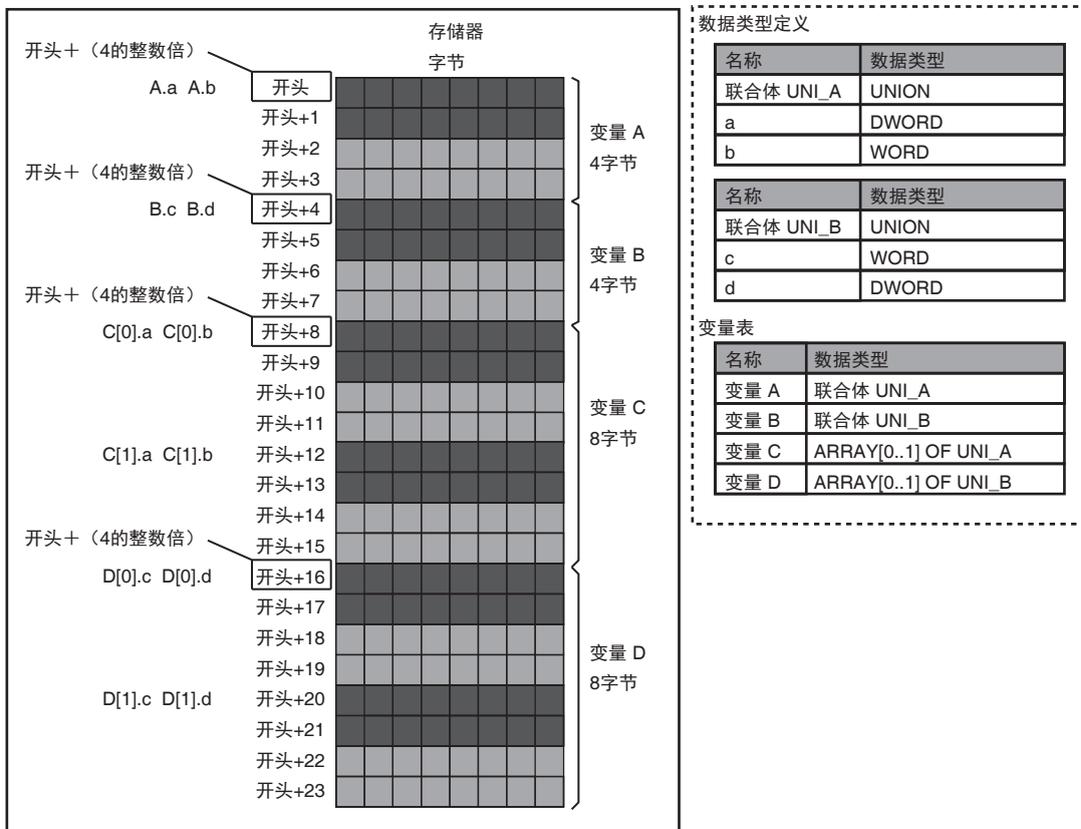
## 联合体型

联合体型变量的各成员重叠配置到存储器上相同的位置。

联合体型校准为所有成员的校准中最大的值。保留的存储器量为所有成员保留的存储器量中最大的值。

例) 下图 4 种变量声明时, 各自的校准和保留的存储器量为以下值。

变量	校准 [ 字节 ]	保留的存储器量 [ 字节 ]
A	4	4
B	4	4
C	4	8
D	4	8



## A-6-2 应注意的案例

NY 系列控制器的结构体型变量，通过与其他设备的通信进行数据交换时，结构体型变量的成员的存储器配置需要与进行通信的对象设备一致。

此时，请在 NY 系列控制器或通信对象的设备上采取本项目中介绍的措施。



### 参考

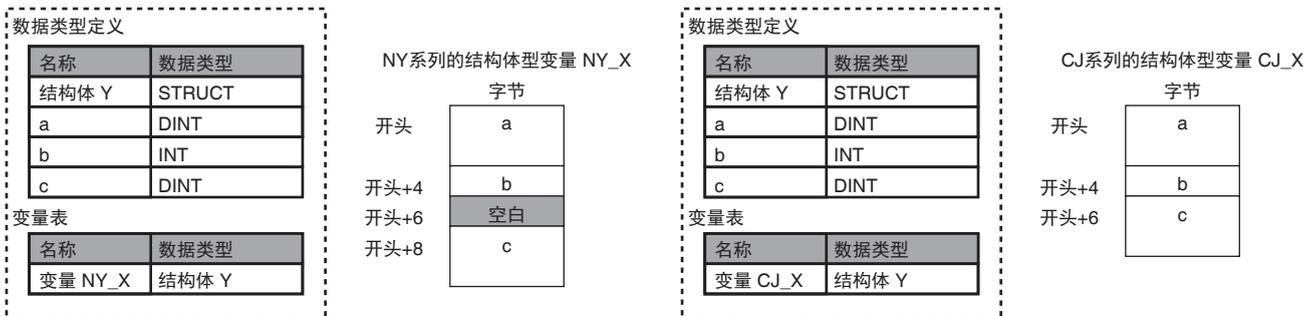
NY 系列控制器之间进行数据交换时，无需特别注意成员的存储器配置。

## 与通信对象设备统一存储器配置的方法

要与通信对象设备统一存储器配置，有两种方法。

例如，在 NY 系列控制器和 CJ 系列的 CPU 中，结构体型变量的存储器配置有以下不同。

本项中举例介绍统一其存储器配置的方法。



### ● 对策 1：变更 NY 系列控制器的结构体型变量成员的存储器配置

NY 系列控制器中，可通过成员的偏置指定，对各结构体型变量设定成员的存储器配置。在 NY 系列控制器上，根据通信对象设备，变更和调整结构体型变量成员的存储器配置。

结构体型变量成员的偏置指定在登录结构体型时进行。

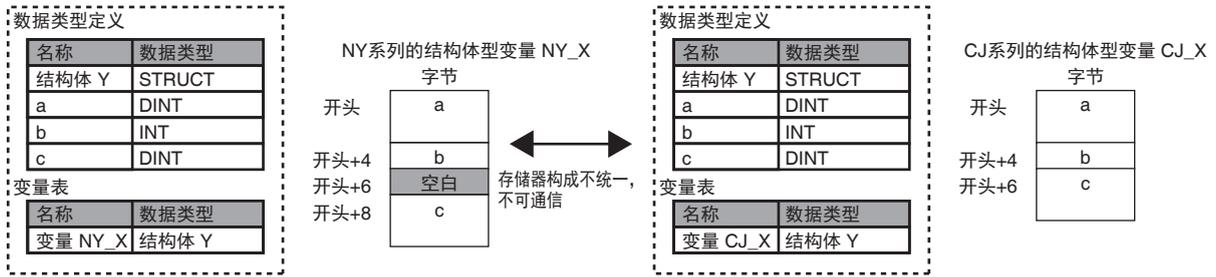
进行通信的对象设备为 CJ 系列 CPU 单元时，若将偏置类型选择为“CJ”，可自动变为 CJ 的存储器配置。

若将偏置类型选择为“任意”，则可自由决定偏置。

在本设定中调整结构体型变量的存储器配置后，请在联网的其他 NY 系列控制器上也进行相同的设定变更。

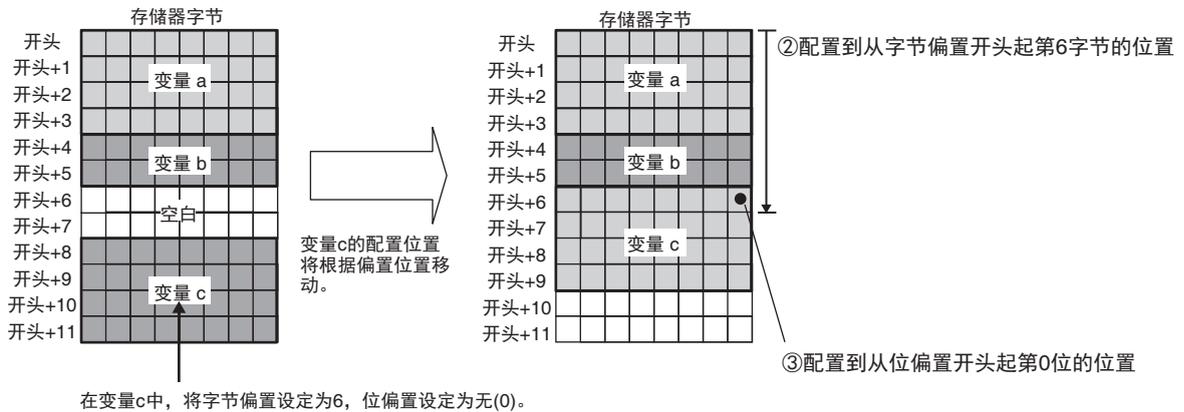
结构体型变量的存储器配置操作方法请参考 □□《Sysmac Studio Version 1 操作手册 (SBCA-CN5-362C 以上)》。

例) 根据 CJ 系列 CPU 单元的结构体型变量的成员变更 NY 系列控制器的结构体型变量成员的存储器配置



为了统一 NY 系列控制器和 CJ 系列 CPU 单元的存储器构成，在 Sysmac Studio 中指定偏置。

在结构体型变量 NY\_X 的数据类型 Y 的成员 c 中指定以下偏置值。

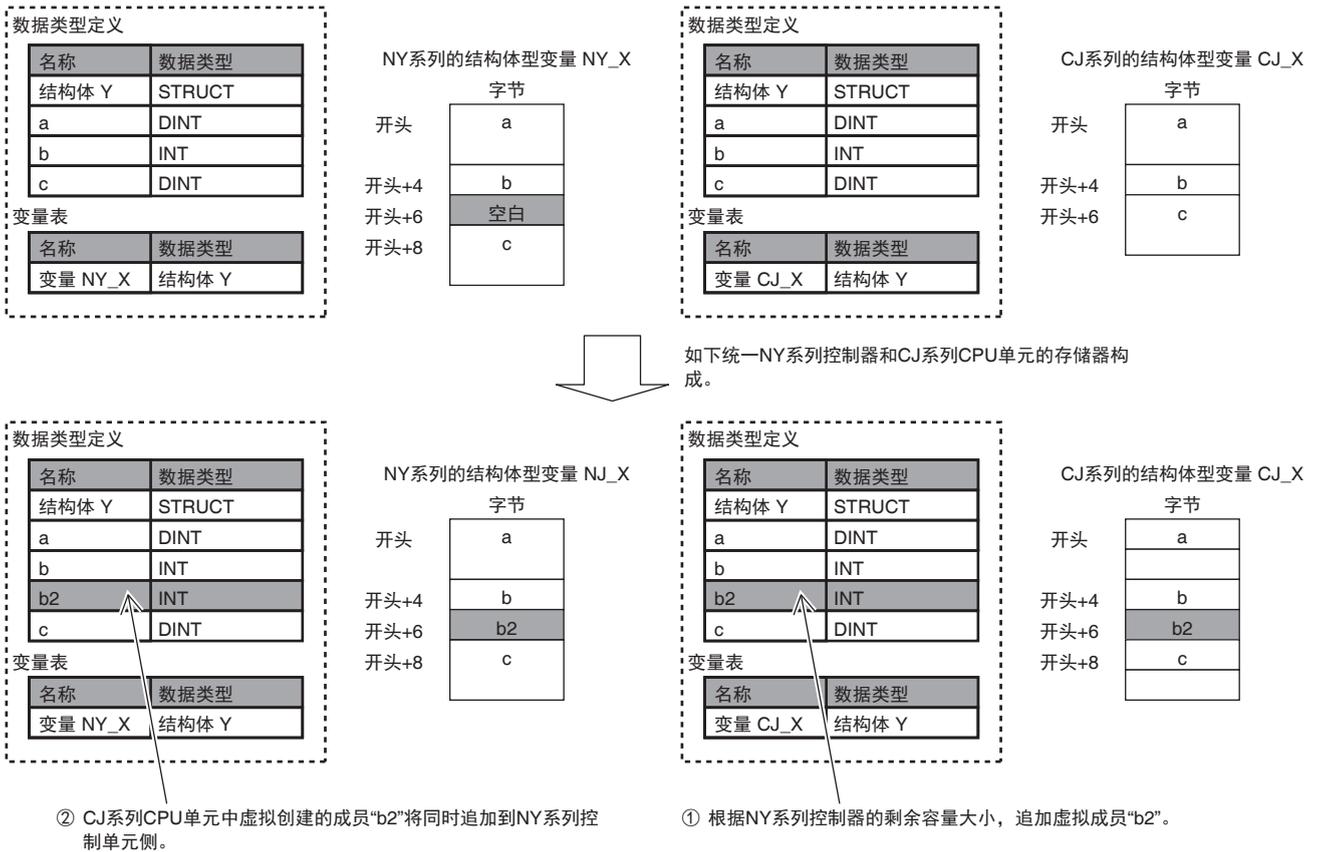


● 对策 2：变更通信对象设备的结构体型变量的存储器配置

根据 NY 系列控制器的结构体型变量的存储器配置，插入配置调整用的成员等，来调整通信对象设备的结构体型变量的存储器配置。

此外，除了存储器配置之外，还需要统一数据类型，因此需要同时创建通信对象设备、NY 系列控制器的配置调整用成员。

例) 根据 NY 系列控制器的结构体型变量的存储器配置变更 CJ 系列 CPU 单元的结构体型变量的存储器配置



## A-7 CX-Designer 的变量表登录方法

连接 NY 系列控制器和 NS 系列触摸屏使用时，在 CX-Designer 上，可将变量作为各功能部件中设定的地址使用。变量通过 [ 变量表 ] 管理。下面介绍通过 Microsoft Excel 复制变量表后，将多个变量批量登录到变量表的方法。CX-Designer 的详情请参考 □ 《CX-Designer 用户手册（SBSA-CN5-532）》。

### 1 在 Microsoft Excel 中创建以下格式的变量表。

表格的列数和配置请遵照以下格式。如 [ 地址类型 / 编号 ] 或 [ I/O 注释 ] 等，全是空白单元格的列也不可省略。

主机	名称	类型	地址类型 / 编号	I/O 注释	标签
HOST3	_Card1BkupCmd.ExecBkup	BOOL			TRUE
HOST3	_Card1BkupCmd.CancelBkup	BOOL			TRUE
HOST3	_Card1BkupCmd.ExecVefy	BOOL			TRUE
HOST3	_Card1BkupCmd.CancelVefy	BOOL			TRUE
HOST3	_Card1BkupCmd.DirName	STRING(64)			TRUE
HOST3	_Card1BkupSta.Done	BOOL			TRUE
HOST3	_Card1BkupSta.Active	BOOL			TRUE
HOST3	_Card1BkupSta.Err	BOOL			TRUE
HOST3	_Card1VefySta.Done	BOOL			TRUE
HOST3	_Card1VefySta.Active	BOOL			TRUE
HOST3	_Card1VefySta.VefyRslt	BOOL			TRUE
HOST3	_Card1VefySta.Err	BOOL			TRUE
HOST3	_BackupBusy	BOOL			TRUE

### 2 启动 CX-Designer，打开变量表画面。



**3 复制 Microsoft Excel 变量表的阴影部分。**  
请务必复制所有的列。

主机	名称	类型	地址类型 / 编号	I/O 注释	标签
HOST3	_Card1BkupCmd.ExecBkup	BOOL			TRUE
HOST3	_Card1BkupCmd.CancelBkup	BOOL			TRUE
HOST3	_Card1BkupCmd.ExecVefy	BOOL			TRUE
HOST3	_Card1BkupCmd.CancelVefy	BOOL			TRUE
HOST3	_Card1BkupCmd.DirName	STRING(64)			TRUE
HOST3	_Card1BkupSta.Done	BOOL			TRUE
HOST3	_Card1BkupSta.Active	BOOL			TRUE
HOST3	_Card1BkupSta.Err	BOOL			TRUE
HOST3	_Card1VefySta.Done	BOOL			TRUE
HOST3	_Card1VefySta.Active	BOOL			TRUE
HOST3	_Card1VefySta.VefyRslt	BOOL			TRUE
HOST3	_Card1VefySta.Err	BOOL			TRUE
HOST3	_BackupBusy	BOOL			TRUE

**4 在 CX-Designer 的变量表画面上右击，执行 [粘贴]。**



**5 在 CX-Designer 的主机选择画面中，选择 NY 系列的主机，然后按下 [OK] 按钮。**



至此，已将变量登录到 CX-Designer 的变量表画面中。

変数テーブル					
追加		検索		未使用検索	
		前へ検索		次へ検索	
検索結果クリア					
ホスト	名称	タイプ	アドレス種別/番号	I/Oコメント	効
全て		全て	全て		全て
PTMEM	AutoGen1	BOOL	\$E0		なし
PTMEM	AutoGen2	CHANNEL	\$W0		なし
HOST3	_Card1BkupCmd.ExecBkup	BOOL			ネットワーク変数
HOST3	_Card1BkupCmd.CancelBkup	BOOL			ネットワーク変数
HOST3	_Card1BkupCmd.ExecVefy	BOOL			ネットワーク変数
HOST3	_Card1BkupCmd.CancelVefy	BOOL			ネットワーク変数
HOST3	_Card1BkupCmd.DirName	STRING(64)			ネットワーク変数
HOST3	_Card1BkupSta.Done	BOOL			ネットワーク変数
HOST3	_Card1BkupSta.Active	BOOL			ネットワーク変数
HOST3	_Card1BkupSta.Err	BOOL			ネットワーク変数
HOST3	_Card1VefySta.Done	BOOL			ネットワーク変数
HOST3	_Card1VefySta.Active	BOOL			ネットワーク変数
HOST3	_Card1VefySta.VefyRsIt	BOOL			ネットワーク変数
HOST3	_Card1VefySta.Err	BOOL			ネットワーク変数
HOST3	_BackupBusy	BOOL			ネットワーク変数

## A-8 EtherCAT 从站或轴的有效 / 无效切换

EtherCAT 从站或轴的有效 / 无效切换可通过指令执行。利用本功能，可实现以下用途。

- 可在一个 Sysmac Studio 项目中管理 EtherCAT 从站构成或轴构成不同的多个装置。
  - 可在一条生产线保持运行的状态下，变更其他生产线的 EtherCAT 从站构成或轴构成。
- 下面介绍本功能中使用的指令、系统定义变量及应用示例。

### A-8-1 使用 EtherCAT 从站和轴时的项目设定

接通电源时或下载项目时，如果存在可能未安装到 EtherCAT 网络的 EtherCAT 从站，请事先设定为无效。同样，轴使用也请设定为未使用轴。若未安装到 EtherCAT 网络的 EtherCAT 从站设定为有效，或轴使用设定为使用轴，执行运行时可能发生异常。



#### 参考

- EtherCAT 从站的有效 / 无效切换还可通过 Sysmac Studio 的 [配置和设定]-[EtherCAT]-[网络配置]-[有效 / 无效设定] 的设定进行。但是，若设定方法为后面介绍的「应用 1 对 EtherCAT 从站构成和轴构成不同的装置进行统一管理」(P.A-101)，可能因装置原因，需要每次手动更改 Sysmac Studio 的设定或更改项目文件。
- 设定为无效的 EtherCAT 从站可从 EtherCAT 网络上拆下或安装到网络上。

### A-8-2 执行 EtherCAT 从站或轴的有效 / 无效切换的指令

EtherCAT 从站或轴的有效 / 无效切换通过用户程序中的指令执行。EtherCAT 从站的有效 / 无效切换和轴的有效 / 无效切换使用不同的指令。指令名称分别如下。

切换对象	指令名称
EtherCAT 从站	EC_ChangeEnableSetting (EtherCAT 从站有效 / 无效切换) 指令
轴	MC_ChangeAxisUse (轴使用变更) 指令

#### EC\_ChangeEnableSetting 指令

EtherCAT 从站的有效 / 无效切换使用 EC\_ChangeEnableSetting(EtherCAT 从站有效无效切换) 指令。EC\_ChangeEnableSetting 指令可将指定节点地址的 EtherCAT 从站设定为无效或有效。执行本指令后，若重新接通控制器的电源，将恢复为指令执行前的设定。

EC\_ChangeEnableSetting 指令的详细规格请参考□□《NY 系列 指令基准手册 基本篇 (SBCA-CN5-437)》。

#### MC\_ChangeAxisUse 指令

轴的有效 / 无效切换使用 MC\_ChangeAxisUse (轴使用变更) 指令。MC\_ChangeAxisUse 指令可将指定的轴参数 [轴使用] 设定为使用轴或未使用轴。执行本指令后，若重新接通控制器的电源，将恢复为指令执行前的设定。

MC\_ChangeAxisUse 指令的详细规格请参考□□《NY 系列 指令基准手册 运动篇 (SBCE-CN5-380)》。

### A-8-3 表示 EtherCAT 从站或轴状态的系统定义变量

EtherCAT 从站或轴的当前状态可通过系统定义变量的值来判断。各自的系统定义变量如下所示。

参照状态的对象	系统定义变量名称
EtherCAT 从站	_EC_DisableSlavTbl[] (无效设定从站表)
轴	_MC_AX[].Cfg.AxEnable (轴使用)

#### **\_EC\_DisableSlavTbl[] (无效设定从站表)**

\_EC\_DisableSlavTbl[] (无效设定从站表) 为表示 EtherCAT 从站的当前状态是否为无效的系统定义变量。在排列的后缀中指定节点地址。\_EC\_DisableSlavTbl[] (无效设定从站表) 的值含义如下。

值	含义
TRUE	指定节点地址的 EtherCAT 从站为无效
FALSE	指定节点地址的 EtherCAT 从站为有效

#### **\_MC\_AX[].Cfg.AxEnable (轴使用)**

\_MC\_AX[].Cfg.AxEnable (轴使用) 为表示轴是否是未创建轴以及是否使用轴的系统定义变量。在排列的后缀中指定轴编号。\_MC\_AX[].Cfg.AxEnable (轴使用) 的值含义如下。

值	含义
0:_mcNoneAxis	指定的轴为未创建轴
1:_mcUnusedAxis	指定的轴为未使用轴
2:_mcUsedAxis	指定的轴为使用轴

## A-8-4 程序的执行 / 停止指令

进行有效 / 无效切换的 EtherCAT 从站或轴中可能有附带的特定程序。此时，在切换 EtherCAT 从站或轴的有效 / 无效的同时，需要执行或停止附带的程序。特定程序的执行 / 停止指令可在用户程序中使用以下指令执行。

功能	指令名称
程序的执行指令	PrgStart 指令
程序的停止指令	PrgStop 指令

PrgStart 指令、PrgStop 指令的详细规格请参考□□《NY 系列 指令基准手册 基本篇 (SBCA-CN5-437)》。



### 使用注意事项

要停止程序时，请在程序停止前，将附带该程序的 EtherCAT 从站或轴设为无效。

A

## A-8-5 确认程序的执行 / 停止

对于进行有效 / 无效切换的 EtherCAT 从站或轴中附带的程序，可通过 PrgStatus 指令确认对其发出的指令时执行还是停止。PrgStatus 指令的详细规格请参考□□《NY 系列 指令基准手册 基本篇 (SBCA-CN5-437)》。

## A-8-6 Sysmac Studio 上的设定

EtherCAT 从站的有效 / 无效、轴的有效 / 无效及程序开始运行时的执行 / 停止等设定在 Sysmac Studio 上也可变更。根据应用不同，有些需要在 Sysmac Studio 上事先设定 EtherCAT 从站、轴及开始运行时程序的状态。

### Sysmac Studio 上的 EtherCAT 从站有效 / 无效设定

若要在 Sysmac Studio 上将 EtherCAT 从站设为有效，应按照以下步骤进行设定。

- 1** 右击 [配置和设定]-[EtherCAT]，从菜单中选择 [编辑]。  
显示 EtherCAT 配置编辑画面。
- 2** 在工具箱中右击要连接的 EtherCAT 从站，从菜单中选择 [插入]。  
在 EtherCAT 配置编辑画面中，选择的 EtherCAT 从站将显示在 EtherCAT 主机下方。此外，EtherCAT 配置编辑画面的右侧将显示 EtherCAT 从站的参数设定画面。
- 3** 在 EtherCAT 从站的参数设定画面中，将 [有效 / 无效设定] 的设定值设为“有效”。

### Sysmac Studio 上的轴有效 / 无效设定

若要在 Sysmac Studio 上将轴设为有效，应按照以下步骤进行设定。

- 1** 右击 [配置和设定]-[运动控制设定]-[轴设定]，从菜单中选择 [添加]-[轴设定]。  
将在 [轴设定] 下方添加“MC\_Axis000(0)”等轴变量。
- 2** 右击“MC\_Axis000(0)”等轴变量名称，从菜单中选择 [编辑]。  
显示轴参数设定的“轴基本设定”画面。
- 3** 将 [轴使用] 设定为“使用轴”。

### Sysmac Studio 上的开始运行时程序启动 / 停止设定

若要在 Sysmac Studio 上、在开始运行时执行程序，应按照以下步骤进行设定。

- 1** 右击 [配置和设定]-[任务设定]，从菜单中选择 [编辑]。  
显示任务设定画面。
- 2** 单击 [程序的分配设定] 按钮。  
显示任务设定的程序分配设定画面。
- 3** 在程序的分配设定画面中，将该程序的 [初始状态] 设为“启动”。

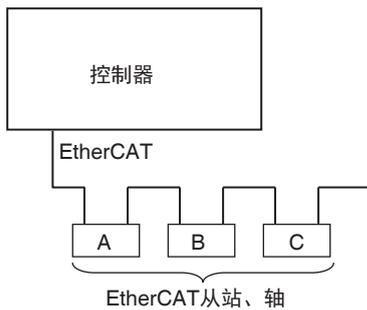
## A-8-7 EtherCAT 从站或轴的有效 / 无效切换应用示例

下面以具体的应用为例，介绍 EtherCAT 从站或轴的有效 / 无效切换使用方法。

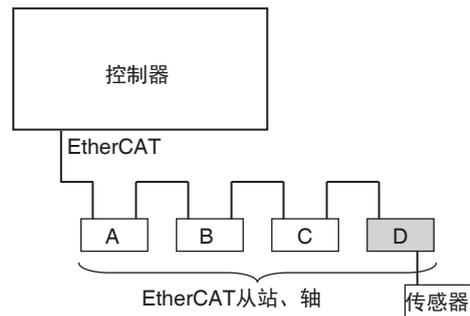
### 应用 1 对 EtherCAT 从站构成和轴构成不同的装置进行统一管理

NY 系列控制器的 EtherCAT 从站或轴构成在装置 1 和装置 2 中不同，具体如下图所示。用 Sysmac Studio 上的一个项目对其进行统一管理。

装置1的EtherCAT从站、轴构成



装置2的EtherCAT从站、轴构成



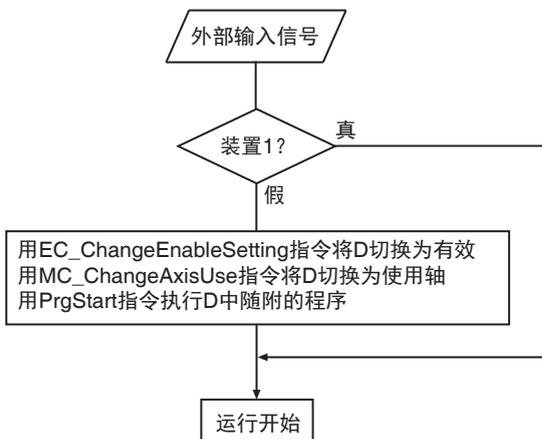
在 Sysmac Studio 的项目中，对图中的 A、B、C、D 4 个 EtherCAT 从站或轴创建 EtherCAT 从站构成。然后，如下在 Sysmac Studio 上根据装置 1 设定各 EtherCAT 从站的有效 / 无效、轴使用的轴 / 未使用轴及随附程序开始运行时的启动 / 停止状态。

EtherCAT 从站	安装到 EtherCAT 网络	有效 / 无效	轴使用	随附程序
A、B、C	有	有效	使用轴	运行开始时、启动
D	无	无效	未使用轴	运行开始时、停止

对应装置 2 时，可利用指令如下切换 EtherCAT 从站的有效 / 无效、轴使用的轴 / 未使用轴及随附程序的启动 / 停止设定。

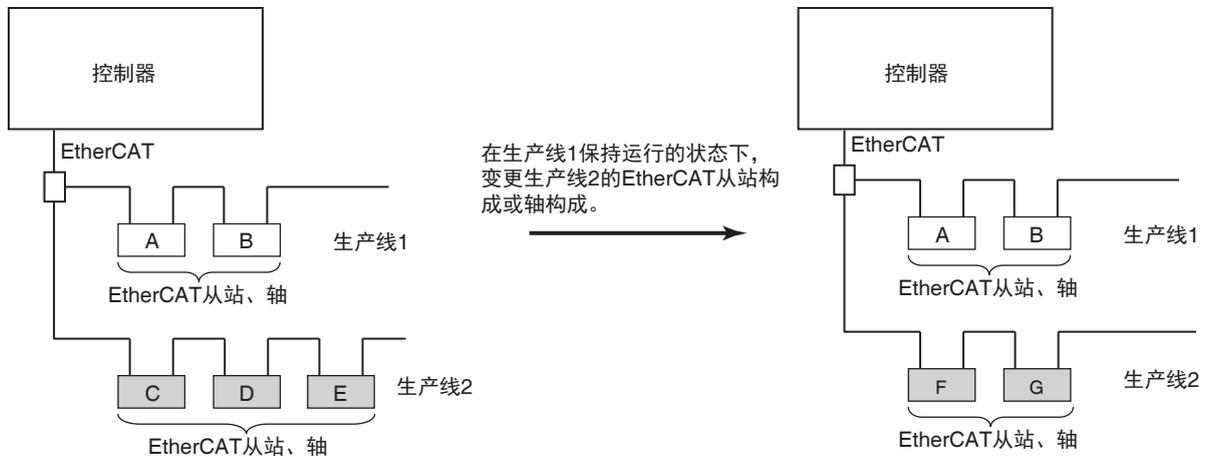
EtherCAT 从站	安装到 EtherCAT 网络	有效 / 无效	轴使用	随附程序
A、B、C	有	有效	使用轴	执行
D	有	有效	使用轴	执行

用户程序的算法如下图所示。假设从控制器外部输入表示运行装置为装置 1 还是装置 2 的信号。



## 应用 2 在运行过程中变更 EtherCAT 从站构成和轴构成

在下图生产线 1 保持运行的状态下，变更生产线 2 的 EtherCAT 从站构成或轴构成。



在 Sysmac Studio 的项目中，对图中的 A ~ G 7 个 EtherCAT 从站或轴创建 EtherCAT 从站构成。使用 Sysmac Studio，事先设定构成变更前 A ~ G EtherCAT 从站的有效 / 无效、轴使用的使用轴 / 未使用轴及随附程序开始运行时的启动 / 停止状态。

EtherCAT 从站	安装到 EtherCAT 网络	有效 / 无效	轴使用	随附程序
A、B	有	有效	使用轴	运行开始时、启动
C、E、D	有	有效	使用轴	运行开始时、启动
F、G	无	无效	未使用轴	运行开始时、停止

将使用的 EtherCAT 从站或轴从 C、D、E 变更为 F、G 的步骤如下。

- 1** 停止生产线 2。
- 2** 使用 MC\_ChangeAxisUse 指令，将 C、D、E 的 [ 轴使用 ] 设为未使用轴。
- 3** 使用 EC\_ChangeEnableSetting 指令，将 C、D、E 的 EtherCAT 从站设定为无效。
- 4** 使用 PrgStop 指令，停止 C、D、E 随附的程序。
- 5** 将 EtherCAT 从站 C、D、E 从生产线 2 上排除。
- 6** 将 EtherCAT 从站 F、G 安装到生产线 2 上。
- 7** 使用 EC\_ChangeEnableSetting 指令，将 F、G 的 EtherCAT 从站设定为有效。
- 8** 使用 MC\_ChangeAxisUse 指令，将 F、G 的 [ 轴使用 ] 设为使用轴。
- 9** 使用 PrgStart 指令，执行 F、G 随附的程序。
- 10** 再次启动生产线 2。

执行上述操作后，A～G EtherCAT 从站的有效 / 无效、轴使用的轴 / 未使用轴及随附程序的启动 / 停止设定变化如下。

EtherCAT 从站	安装到 EtherCAT 网络	有效 / 无效	轴使用	随附程序
A、B	有	有效	使用轴	执行
C、E、D	无	无效	未使用轴	停止
F、G	有	有效	使用轴	执行



#### 使用注意事项

要停止程序时，请在程序停止前，将附带该程序的 EtherCAT 从站或轴设为无效。

## A-9 用户程序的大小相关的限制

受控制器内存储器容量的制约，用户程序的大小有限制。若超出限制，运行时会发生异常。下面按照不同项目介绍 1 台控制器上可创建的用户程序大小限制。

用户程序或变量大小的近似值可使用 Sysmac Studio 的存储器显示功能确认。

创建用户程序时请注意限制。但是，标示的限制值仅供参考。考虑到将来用户程序可能会扩展，创建用户程序时，建议留出一定的余量。



### 使用注意事项

使用在线编辑时，即使用户程序的大小在限制值以内，也可能发生异常。这是因为通过在线编辑变更用户程序后，控制器的存储器中可能还有保留的数据。发生异常时，若将控制器设为程序模式，再次向控制器传送用户程序，有时可以解除异常。

### A-9-1 用户程序的各部件限制

下面介绍用户程序的各部件限制。有限制的部件如下所示。

- POU
- 变量
- 数据类型定义
- 常数（文本）

### POU 的限制

POU 的限制包括 POU 定义的限制和 POU 实例的限制。

#### ● POU 定义的限制

POU 的定义有以下限制。

限制项目	NY5□2-1□□□
程序数的最大值	500
（FB 定义数 + FUN 定义数 + 梯形图程序的段落数）的最大值	3,000
（FB 定义和 FUN 定义的输入 / 输出 / 输入输出变量的总个数）的最大值	64

#### ● POU 实例的限制

POU 的实例有以下限制。

限制项目	NY5□2-1□□□
POU 实例数的最大值	24,000

POU 实例的计数方法请参考 □□「POU 实例的数量」（P.A-106）。

## 变量的限制

变量的限制包括变量使用量的限制和变量定义的限制。

### ● 变量使用量的限制

变量的使用量有以下限制。

限制项目	NY5□2-1□□□
无保持属性的变量总大小 [MB] 的最大值 <sup>*1</sup>	64
有保持属性的变量总大小 [MB] 的最大值 <sup>*1</sup>	4
无保持属性的变量数 <sup>*2</sup> 最大值	180,000
有保持属性的变量数 <sup>*3</sup> 最大值	40,000
网络变量的数量最大值	40,000

\*1. 每个变量的数据大小因数据类型不同而异。各数据类型的大小请参考□□「6-3-5 数据类型」(P.6-30)。

\*2. 计数方法请参考□□「无保持属性的变量数」(P.A-106)。

\*3. 计数方法请参考□□「有保持属性的变量数」(P.A-107)。

### ● 变量定义的限制

变量定义有以下限制。

限制项目	NY5□2-1□□□
1 个排列的要素数最大值	65,535
排列的维数最大值	3
1 个排列的大小 [MB] 最大值	8
排列后缀 (要素编号) 值的最大值	65,535
字符串型变量的大小 [字节] <sup>*1</sup> 最大值	1,986

\*1. 包含结束 NULL 字符。因此，大小为 1986 字节的字符串中，半角英文及数字为 1985 字符。

## 数据类型定义的限制

数据类型定义有以下限制。

限制项目	NY5□2-1□□□
数据类型定义数 <sup>*1</sup> 的最大值	4,000
每个结构体型定义的层级数最大值	8
每个结构体型定义的成员数最大值	2,048
每个结构体型变量的大小 [MB] 最大值	8
每个联合体型定义的成员数最大值	4
每个列举型定义的列举值数最大值	2,048

\*1. 计数方法请参考□□「数据类型定义的数量」(P.A-107)。

## 常数 (文本) 的限制

常数 (文本) 有以下限制。

限制项目	NY5□2-1□□□
字符串型常数 (文本) 的大小 [字节] 最大值	1,985

## A-9-2 用户程序的各部件数量

表示 POU 实例、无保持属性的变量、有保持属性的变量及数据类型定义的计数方法。但是，这里记载的内容仅供参考。因 NY 系列控制器的单元版本不同，部件的计数方法略有不同。因此，请务必在 Sysmac Studio 上确认用户程序的大小是否合适。

### POU 实例的数量

POU 实例的计数方法如下。

#### ● POU 实例的计数对象

POU 实例的计数对象如下。

- 程序
- FB 实例（包括用户创建和指令）
- FUN（包括用户创建和指令）

#### ● POU 实例的计数注意

计算 POU 实例的数量时，请注意以下几点。

- 使用 n 个与同一 FB 定义对应的 FB 实例时，计为 n 个。
- 在同一任务中使用同一 FUN 时，即使 FUN 有多个，也算作 1 个。
- 在不同任务中使用同一 FUN 时，每个任务计为 1 个。

### 无保持属性的变量数

无保持属性的变量计数方法如下。

#### ● 无保持属性的变量计数对象

无保持属性的变量计数对象如下。

- 无保持属性的全局变量
- 程序及 FB 实例（包括用户创建和指令）的无保持属性的本地变量

#### ● 无保持属性的变量计数注意

计算无保持属性变量的数量时，请注意以下几点。

- 排列始终计为 1 个，与要素数无关。
- FB 实例计为 1 个。但是，FB 实例包括用户创建和指令。
- 将 FB 实例作为要素的排列始终计为 1 个，与要素数无关。但是，该 FB 中使用的无保持属性的变量数按照排列的要素数计算。

## 有保持属性的变量数

有保持属性的变量计数方法如下。

### ● 有保持属性的变量计数对象

有保持属性的变量计数对象如下。

- 有保持属性的全局变量
- 程序及 FB 实例（包括用户创建和指令）的有保持属性的本地变量

### ● 有保持属性的变量计数注意

计算有保持属性变量的数量时，请注意以下几点。

- 排列始终计为 1 个，与要素数无关。
- 将 FB 实例作为要素的排列不计数。但是，该 FB 中使用的有保持属性的变量数按照排列的要素数计算。

## 数据类型定义的数量

数据类型定义的计数方法如下。

### ● 数据类型定义的计数对象

数据类型定义的计数对象如下。

- 用户创建的结构体型定义
- 用户创建的联合体型定义
- 用户创建的列举型定义

## A-10 NY 系列的版本相关信息

下面介绍 NY 系列控制器和 Sysmac Studio 的版本对应及各单元版本的支持功能。

NY 系列控制器的单元版本和 Sysmac Studio 的版本有对应关系。一般情况下请组合使用对应的版本。

### 单元版本和 Sysmac Studio 的对应版本

NY 系列控制器的单元版本和 Sysmac Studio 的对应版本如下表所示。

NY 系列控制器的单元版本	Sysmac Studio 的对应版本
Ver.1.12	Ver.1.17

### 单元版本和 Sysmac Studio 的版本不匹配时的规格

NY 系列控制器的单元版本和 Sysmac Studio 的版本不匹配时的规格如下。

- **Sysmac Studio 的版本为 Ver.1.16 以下时**

Ver.1.16 以下的 Sysmac Studio 中，不可使用 NY 系列控制器。



# 索引



## 索引

- A**
- \_AlarmFlag (用户异常状态) ..... 53, 57, 24, 48
  - AT (分配目标) 指定 ..... 56
  - 安全设定 ..... 5
- B**
- \_BackupBusy (备份相关功能执行中标志) ..... 26, 52
  - BOOL ..... 31, 44
  - BOOTP 服务器连接异常 ..... 38, 73
  - BYTE ..... 31, 44
  - 保持 ..... 57
  - 保持条件 ..... 62
  - 备份 ..... 3
  - 备份相关功能执行中标志 ..... 26, 52
  - 本地变量 ..... 28
    - 内部变量 ..... 28
    - 输出变量 ..... 28
    - 输入变量 ..... 28
    - 输入输出变量 ..... 28
    - 外部变量 ..... 28
  - 本地变量表 ..... 10, 18
  - 编程语言 ..... 80
  - 变更了保持变量的数据类型时变量的值 ..... 65
  - 变更 POU 名称或新建 POU 等时保持变量的值 ..... 63
  - 变量
    - 种类 ..... 27
    - 概要 ..... 27
  - 变量的种类
    - 系统定义变量 ..... 10
    - 用户定义变量 ..... 10
    - 准用户定义变量 ..... 10
  - 变量的存储器确保方法 ..... 82
  - 变量的任务间排他控制设定 ..... 10
  - 变量的属性 ..... 29
  - 变量访问时间 ..... 8
  - 变量访问时间设定 ..... 48
  - 变量和 I/O 的访问 ..... 13
  - 变量名称 ..... 56
  - 变量名称等程序相关名称的限制 ..... 73
  - 边缘 ..... 61
  - 标签解决异常 ..... 39, 74
  - 标签数据链接服务 ..... 25
  - 标签数据链接设定异常 ..... 39, 74
  - 标签数据链接通信开始开关 ..... 43, 78
  - 标签数据链接通信停止开关 ..... 43, 78
  - 标签数据链接通信异常 ..... 39, 74
  - 标签数据链接通信中 ..... 41, 76
  - 布尔型 ..... 31, 44
- C**
- \_Card1Access (SD 存储卡访问中标志) ..... 11, 24, 49
  - \_Card1BkupCmd (SD 存储卡备份指示) ..... 25, 49
  - \_Card1BkupSta (SD 存储卡备份状态) ..... 25, 50
  - \_Card1Deteriorated (SD 存储卡使用寿命警告标志)
    - ..... 11, 24, 49
  - \_Card1Err (SD 存储卡错误标志) ..... 11, 24, 49
  - \_Card1PowerFail (SD 存储卡访问中断电标志)
    - ..... 11, 24, 49
  - \_Card1Protect (SD 存储卡写保护标志) ..... 11, 24, 48
  - \_Card1Ready (SD 存储卡可使用标志) ..... 11, 24, 48
  - \_Card1VefySta (SD 存储卡核对状态) ..... 26, 51
  - CJ 单元构成 ..... 5
  - CIP 通信异常 ..... 37, 70
  - CPU 单元的写入保护功能 ..... 23
  - CPU 单元名称 ..... 24
  - CPU 单元名称功能 ..... 24
  - CPU 单元内部温度超限标志 ..... 29, 56
  - \_CurrentTime (系统时刻) ..... 4, 22, 46
  - CX-Designer 的变量表登录方法 ..... 94
  - 参数 ..... 10, 18
  - 参照任务 ..... 40
  - 操作权限 ..... 21
  - 操作权限的认证功能 ..... 21
  - 常量 ..... 60
  - 常数 ..... 75
    - 布尔型 ..... 75
    - 持续时间型 ..... 76
    - 列举型 ..... 79
    - 日期时刻型 ..... 77
    - 日期型 ..... 77
    - 时刻型 ..... 77
    - 实型数据 ..... 76
    - 位列型 ..... 75
    - 整数 ..... 76
    - 字符串型 ..... 78
  - 程序 ..... 8
  - 程序的初始状态 ..... 9
  - 程序的分配设定 ..... 9
  - 程序的运行开始时初始状态 ..... 37
  - 程序的执行顺序 ..... 37
  - 程序模式 ..... 25
  - 程序启动时 1 个周期 ON 标志 ..... 8, 28, 55
  - 程序执行顺序 ..... 9
  - 持续时间型 ..... 31, 44
  - 触发跟踪 ..... 35
  - 初始设定
    - EtherCAT 主机功能模块的初始设定 ..... 14
    - EtherNet/IP 功能模块的初始设定 ..... 15
    - 运动控制功能模块的初始设定 ..... 12
  - 初始值 ..... 58
  - 创建用户程序时的注意事项 ..... 142
  - 从站初始化异常 ..... 32, 63
  - 从站节点地址重复异常 ..... 32, 63
  - 从站异常 ..... 32, 61
  - 从站异常表 ..... 32, 62
  - 从站应用程序异常 ..... 32, 64

存储器全部清除 ..... 2

## D

DATE ..... 31, 44  
 DATE\_AND\_TIME ..... 32, 44  
 DINT ..... 31  
 DNS 服务器连接异常 ..... 39, 75  
 DNS 设定异常 ..... 38, 73  
 DWORD ..... 31  
 单纯名称 ..... 134  
 单元版本 ..... 28, 55  
 当前值变更 ..... 31  
 登录从站表 ..... 34, 65  
 登录目标节点信息 ..... 41, 76  
 电池电压低标志 ..... 29, 56  
 动作模式 ..... 25  
 动作模式的确认 ..... 26  
 动作模式切换 ..... 26  
 动作模式切换时变量值的保持 ..... 25  
 动作设定 ..... 5  
 动作设定标签 ..... 4  
 动作选项 ..... 125, 126  
 断电保持失败标志 ..... 27, 53  
 断电发生次数 ..... 27, 52  
 多个开关同时 ON 异常 ..... 39, 75

## E

\_EC1\_InDataInvalid (输入数据 1 无效) ..... 34  
 \_EC\_CommErrTbl (通信异常从站表) ..... 32, 64  
 \_EC\_CycleExceeded (EtherCAT 通信周期设定异常) ..... 33, 64  
 \_EC\_DisableSlavTbl (无效设定从站表) ..... 34, 66  
 \_EC\_DisconnSlavTbl (脱离指令中的从站表) ..... 34, 66  
 \_EC\_EntrySlavTbl (网络加入从站表) ..... 34, 65  
 \_EC\_ErrSta (内置 EtherCAT 异常) ..... 32, 61  
 \_EC\_InDataInvalid (输入数据无效) ..... 34, 67  
 \_EC\_InData1Invalid (输入数据 1 无效) ..... 67  
 \_EC\_LanHwErr (通信控制器异常) ..... 32, 62  
 \_EC\_LinkOffErr (链接断开异常) ..... 32, 62  
 \_EC\_LinkStatus (链接状态) ..... 34, 66  
 \_EC\_MacAdrErr (MAC 地址异常) ..... 32, 62  
 \_EC\_MBXSlavTbl (可进行信息通信的从站表) ..... 34, 65  
 \_EC\_MsgErr (EtherCAT 消息异常) ..... 32, 64  
 \_EC\_MstrErr (主机异常) ..... 32, 61  
 \_EC\_NetCfgCmpErr (网络配置核对异常) ..... 32, 62  
 \_EC\_NetCfgErr (网络配置信息异常) ..... 32, 62  
 \_EC\_NetTopologyErr (网络配置异常) ..... 32, 63  
 \_EC\_PDActive (过程数据通信状态) ..... 34, 66  
 \_EC\_PDCommErr (过程数据通信异常) ..... 32, 63  
 \_EC\_PDSendErr (过程数据发送异常) ..... 32, 63  
 \_EC\_PDSlavTbl (过程数据通信中从站表) ..... 34, 65  
 \_EC\_PDTimeoutErr (过程数据接收超时异常) ..... 32, 63  
 \_EC\_PktMonStop (信息包监视停止) ..... 34, 66  
 \_EC\_PktSaving (信息包数据文件保存中) ..... 34, 67  
 \_EC\_PortErr (通信端口异常) ..... 32, 61  
 \_EC\_RegSlavTbl (登录从站表) ..... 34, 65  
 \_EC\_SlavAdrDupErr (从站节点地址重复异常) ..... 32, 63

\_EC\_SlavAppErr (从站应用程序异常) ..... 32, 64  
 \_EC\_SlavEmergErr (紧急消息检测) ..... 32, 64  
 \_EC\_SlavErr (从站异常) ..... 32, 61  
 \_EC\_SlavErrTbl (从站异常表) ..... 32, 62  
 \_EC\_SlavInitErr (从站初始化异常) ..... 32, 63  
 \_EC\_StatisticsLogBusy (诊断统计信息日志执行状态标志) ..... 35, 68  
 \_EC\_StatisticsLogErr (诊断统计信息日志异常结束标志) ..... 35  
 \_EC\_StatisticsLogCycleSec (诊断统计信息日志保存周期) ..... 35, 68  
 \_EC\_StatisticsLogEnable (诊断统计信息日志许可) ... 35  
 \_EC\_StatisticsLogErr (诊断统计信息日志异常结束标志) ..... 68  
 \_EIP1\_BootpErr (通信端口 1 BOOTP 服务器连接异常) ..... 38, 73  
 \_EIP1\_EtnCfgErr (通信端口 1 Ethernet 基本设定异常) ..... 38, 71  
 \_EIP1\_EtnOnlineSta (内部通信端口 1 在线) ..... 76  
 \_EIP1\_EtnOnlineSta (通信端口 1 在线) ..... 41, 76  
 \_EIP1\_IPAdrCfgErr (通信端口 1 IP 地址设定异常) ..... 38, 72  
 \_EIP1\_IPAdrDupErr (内部通信端口 1 IP 地址重复异常) ..... 38  
 \_EIP1\_IPAdrDupErr (通信端口 1 IP 地址重复异常) ..... 38, 72  
 \_EIP1\_LanHwErr (通信端口 1 通信控制器异常) ..... 37, 71  
 \_EIP1\_MacAdrErr (通信端口 1 MAC 地址异常) .. 37, 71  
 \_EIP1\_PortErr (通信端口 1 异常) ..... 36, 69  
 \_EIP\_BootpErr (BOOTP 服务器连接异常) ..... 38, 73  
 \_EIP\_CipErr (CIP 通信异常) ..... 37, 70  
 \_EIP\_DNSCfgErr (DNS 设定异常) ..... 38, 73  
 \_EIP\_DNSSrvErr (DNS 服务器连接异常) ..... 39, 75  
 \_EIP\_ErrSta (内置 EtherNet/IP 异常) ..... 36, 69  
 \_EIP\_EstbTargetSta (正常目标节点信息) ..... 41, 77  
 \_EIP\_EtnCfgErr (Ethernet 基本设定异常) ..... 37, 71  
 \_EIP\_EtnOnlineSta (在线) ..... 41, 75  
 \_EIP\_IdentityErr (Identity 信息不正确) ..... 38, 74  
 \_EIPIn1\_EtnOnlineSta (内部通信端口 1 在线) ..... 41  
 \_EIPIn1\_IPAdrCfgErr (内部通信端口 1 IP 地址设定异常) ..... 38, 72  
 \_EIPIn1\_IPAdrDupErr (内部通信端口 1 IP 地址重复异常) ..... 73  
 \_EIPIn1\_PortErr (内部通信端口 1 异常) ..... 37, 70  
 \_EIP\_IPAdrCfgErr (IP 地址设定异常) ..... 38, 72  
 \_EIP\_IPAdrDupErr (IP 地址重复异常) ..... 38, 72  
 \_EIP\_IPRTblErr (IP 路由表异常) ..... 38, 73  
 \_EIP\_LanHwErr (通信控制器异常) ..... 37, 71  
 \_EIP\_MacAdrErr (MAC 地址异常) ..... 37, 70  
 \_EIP\_MultiSwONErr (多个开关同时 ON 异常) ... 39, 75  
 \_EIP\_NTPResult (NTP 的动作信息) ..... 42  
 \_EIP\_NTPResult.ExecNormal (NTP 的动作结果) ..... 42, 78  
 \_EIP\_NTPResult.ExecTime (NTP 的最终动作时间) ..... 42, 78  
 \_EIP\_NTPSrvErr (NTP 服务器连接异常) ..... 39, 75  
 \_EIP\_PortErr (通信端口异常) ..... 36, 69  
 \_EIP\_RegTargetSta (登录目标节点信息) ..... 41, 76

_EIP_TagAdrErr (标签解决异常)	39, 74
_EIP_TargetNodeErr (目标节点异常信息)	42, 77
_EIP_TargetPLCErr (目标 PLC 异常信息)	42, 77
_EIP_TargetPLCModeSta (目标 PLC 动作模式)	42, 77
_EIP_TcpAppCfgErr (TCP/IP 设定异常)	39
_EIP_TcpAppCfgErr (TCP 应用程序设定异常)	75
_EIP_TcpAppErr (TCP 应用通信异常)	37, 70
_EIP_TDLinkAllRunSta (所有标签数据链接通信中)	41, 76
_EIP_TDLinkCfgErr (标签数据链接设定异常)	39, 74
_EIP_TDLinkErr (标签数据链接通信异常)	39, 74
_EIP_TDLinkOpnErr (建立标签数据链接失败)	39, 74
_EIP_TDLinkRunSta (标签数据链接通信中)	41, 76
_EIP_TDLinkStartCmd (标签数据链接通信开始开关)	43, 78
_EIP_TDLinkStopCmd (标签数据链接通信停止开关)	43, 78
EN	19
ENO	20
_ErrSta (控制器异常状态)	50, 24, 48
EtherCAT 从站或轴的有效 / 无效切换	97
EtherCAT 通信周期设定异常	33, 64
EtherCAT 网络配置	5
EtherCAT 消息异常	32, 64
EtherCAT 主机功能模块的初始设定	14
EtherNet/IP 功能模块的初始设定	15
Ethernet 基本设定异常	37, 71

## F

FB 名称或指令名称	10
FTP 服务器功能	10
FTP 客户端通信指令	10
FUN 定义名称或指令名称	17
返回值	20
范围指定 ((,))	54
风扇转速降低标志	29, 56

## G

各种状态下的动作	24
更新任务	40
功能块	
从外部参照变量	16
内部变量的规格	12
启动条件	15
ST 语言的调用表述	11
实例	13
功能块 (FB)	
参数省略时的动作	23
参数异常时的动作	24
EN、ENO 的使用 / 省略可否	23
功能模块	
EtherCAT 主机功能模块	4
EtherNet/IP 功能模块	4
PLC 功能模块	4
运动控制功能模块	4
共通异常状态	31, 59

固定周期任务 (执行优先级 16、17、18) 的设定项目	39
固定周期任务 (执行优先级 16、17、18)	14
关机处理等待时间设定	5
关机请求标志	27, 53
规格	
功能规格	6
性能规格	3
规格概要	7
过程数据发送异常	32, 63
过程数据接收超时异常	32, 63
过程数据通信异常	32, 63
过程数据通信中从站表	34, 65
过程数据通信状态	34, 66
功能块 (FB) 构成	9
功能块 (FB) 详情	9

## H

_HardwareRevision (硬件修订版本)	55
函数 (FUN)	
参数省略时的动作	23
参数异常时的动作	24
EN、ENO 的使用 / 省略可否	23
函数 (FUN)	
构成	17
函数 (FUN)	
内部变量的规格	19
启动条件	22
ST 语言的表述	18
函数 (FUN)	
详情	17
核对	3
恢复	3

## I

I/O 端口	5
I/O 端口名称	6
I/O 刷新周期超限	51
Identity 信息不正确	38, 74
INT	31
IP 地址设定异常	38, 72
IP 地址重复异常	38, 72
IP 路由表异常	38, 73

## J

基本构成	5
基本数据类型	30
建立标签数据链接失败	39, 74
接点	81
结构体成员的偏置指定	42
结构体类型	40
紧急消息检测	32, 64
进位标志	8, 126, 28, 54

**K**

可进行信息通信的从站表	34, 65
可在线连接的构成	4
控制器的状态	22
控制器内的数据	21
控制器设定	4
控制器事件	46, 51
控制器异常状态	50, 24, 48
库	138

**L**

LD	80
LINT	31
LREAL	31
LWORD	31
类别	47
离线调试	3
联合体型	45
联机 ST	85
链接断开异常	32, 62
连接线 [ 连接线 ]	81
链接状态	34, 66
连续跟踪	35
列举型 (ENUM)	48

**M**

MAC 地址异常	32, 37, 62, 70
_MC_AX (轴变量)	31, 60
_MC_AX_ErrSta (轴异常状态)	31, 59
_MC_COM (通用变量)	31, 60
_MC_ComErrSta (共通异常状态)	31, 59
MC 调试	9
_MC_ErrSta (运动控制功能模块的异常状态)	31, 59
_MC_GRP (轴组变量)	31, 60
_MC_GRP_ErrSta (轴组异常状态)	31, 59
名称空间	133
名称空间的使用声明	135
模拟程序	6
模拟功能	3
模拟器的启动和停止	3
模拟速度	7
默认的 CAST	117
目标节点异常信息	42, 77
目标 PLC 动作模式	42, 77
目标 PLC 异常信息	42, 77
母线	80

**N**

NTP 的动作结果	42, 78
NTP 的动作信息	42
NTP 的最终动作时间	42, 78
NTP 服务器连接异常	39, 75
内部变量	19
内部通信端口 1 IP 地址设定异常	38, 72
内部通信端口 1 IP 地址重复异常	38, 73

内部通信端口 1 异常	37, 70
内部通信端口 1 在线	41, 76
内置 EtherCAT 异常	32, 61
内置 EtherNet/IP 异常	36, 69

**O**

_OSErrorState (OS 异常状态标志)	27, 54
_OSHalted (OS 停止中标志)	27, 53
_OSRunning (OS 运行中标志)	27, 53
OS 停止中标志	27, 53
OS 异常状态标志	27, 54
OS 运行中标志	27, 53

**P**

P_CY (进位标志)	8, 126, 28, 54
P_First_Run (程序启动时 1 个周期 ON 标志)	8, 28, 55
P_First_RunMode (运行开始时 1 个周期 ON 标志)	8, 28, 54
_PLC_ErrSta (PLC 功能模块异常状态)	30, 58
PLC 功能模块	4
PLC 功能模块异常状态	30, 58
_PLC_TraceSta.IsComplete (追踪完成时 ON 标志)	40, 30, 57
_PLC_TraceSta.IsStart (追踪执行中 ON 标志)	40, 30, 57
_PLC_TraceSta.IsTrigger (追踪触发监视标志)	40, 30, 57
_PLC_TraceSta.ParamErr (追踪参数异常标志)	40, 30, 57
P_Off (始终 OFF 标志)	28, 54
P_On (始终 ON 标志)	28, 54
POU	
程序	6
功能块 (FB)	6
函数 (FUN)	6
POU (程序构成单位)	5
POU 的限制事项	25
_PowerOnCount (断电发生次数)	27, 52
_PowerOnHour (通电时间)	27, 52
P_PRGER (指令错误发生标志)	8, 128, 131, 28, 55
排列指定 (ARRAY OF)	50
派生数据类型	40

**Q**

嵌套的层级	26
强制值刷新	27
全局变量	28
确保任务间变量值的同时性	40

**R**

REAL	31, 44
_RequestShutdown (关机请求标志)	27, 53
_RetainFail (断电保持失败标志)	27, 53
_<任务名称>_Active (任务执行中标志)	6, 23, 46

\_ <任务名称> \_ExceedCount (任务周期超出次数)  
 ..... 6, 50, 23, 47  
 \_ <任务名称> \_Exceeded (任务周期超出标志)  
 ..... 6, 50, 54, 23, 47  
 \_ <任务名称> \_ExecCount (任务执行次数)  
 ..... 6, 54, 23, 47  
 \_ <任务名称> \_LastExecTime (上次任务执行时间)  
 ..... 6, 23, 46  
 \_ <任务名称> \_MaxExecTime (任务执行时间最大值)  
 ..... 6, 23, 46  
 \_ <任务名称> \_MinExecTime (任务执行时间最小值)  
 ..... 6, 23, 46  
 任务  
   程序的分配 ..... 37  
   分配到 I/O 刷新中 ..... 31  
   NY 系列的规格 ..... 8  
   执行优先级 ..... 9  
 任务间排他控制设定 ..... 47  
 任务间排他控制指令 ..... 43  
 任务类型 ..... 7, 13  
 任务名称 ..... 7, 8  
 任务设计示例 ..... 57  
 任务设置 ..... 7  
 任务相关的系统定义变量 ..... 6  
 任务相关的指令 ..... 6  
 任务执行超时 ..... 51  
 任务执行超时时间 ..... 8  
 任务执行次数 ..... 6, 54, 23, 47  
 任务执行时间 ..... 55  
 任务执行时间的监视 ..... 53  
 任务执行时间的确认 ..... 56  
 任务执行时间最大值 ..... 6, 23, 46  
 任务执行时间最小值 ..... 6, 23, 46  
 任务执行中标志 ..... 6, 23, 46  
 任务执行状态的监视 ..... 52  
 任务执行状态监视 ..... 10  
 任务中可分配的 POU ..... 37  
 任务周期超出标志 ..... 6, 50, 54, 23, 47  
 任务周期超出次数 ..... 6, 50, 23, 47  
 任务周期超限 ..... 50  
 任务周期超限检测 ..... 8  
 日期时刻型 ..... 32, 44  
 日期型 ..... 31, 44  
 软件的动作 ..... 5  
 软件构成 ..... 4

## S

SD 存储卡  
   功能一览 ..... 5  
 SD 存储卡备份禁止功能 ..... 19  
 SD 存储卡备份指示 ..... 25, 49  
 SD 存储卡备份状态 ..... 25, 50  
 SD 存储卡错误标志 ..... 11, 24, 49  
 SD 存储卡的备份功能 ..... 10  
 SD 存储卡访问中标志 ..... 11, 24, 49  
 SD 存储卡访问中断电标志 ..... 11, 24, 49  
 SD 存储卡核对状态 ..... 26, 51  
 SD 存储卡可使用标志 ..... 11, 24, 48

SD 存储卡使用寿命警告标志 ..... 11, 24, 49  
 SD 存储卡写保护标志 ..... 11, 24, 48  
 SD 存储卡  
   指令 ..... 9  
 \_SelfTest\_HighTemperature (CPU 单元内部温度超限标志)  
 ..... 29, 56  
 \_SelfTest\_LowBattery (电池电压低标志) ..... 29, 56  
 \_SelfTest\_LowFanRevolution (风扇转速降低标志)  
 ..... 29, 56  
 \_SelfTest\_UPSSigna (UPS 信号检测标志) ..... 27  
 \_SelfTest\_UPSSignal (UPS 信号检测标志) ..... 53  
 SINT ..... 31, 44  
 STRING ..... 32, 44  
 ST 语言 ..... 86  
   CASE ..... 99  
   代入 ..... 93  
   EXIT ..... 110  
   FB 调用 ..... 111  
   FOR ..... 102  
   FUN 调用 ..... 114  
   各运算符的操作对象中可使用的数据类型 ..... 92  
   IF (1 个条件) ..... 95  
   IF (多个条件) ..... 97  
   REPEAT ..... 108  
   RETURN ..... 94  
   WHILE ..... 106  
   语句的关键词 ..... 89  
   运算符 ..... 90  
 Sysmac Studio 备份文件的导入 / 导出功能 ..... 24  
 Sysmac Studio 变量 / 存储器的备份功能 ..... 25  
 Sysmac Studio 控制器备份功能 ..... 20  
 上次任务执行时间 ..... 6, 23, 46  
 设备变量 ..... 11, 6  
 设定控制 I/O 的任务 ..... 8  
 事件代码 ..... 47  
 事件发生源 ..... 46  
 事件任务 ..... 15  
   ActEventTask 指令 ..... 15  
   变量的条件式一致 ..... 17  
   设定项目 ..... 39  
 事件日志功能 ..... 45  
 事件日志设定 ..... 5  
 事件设定表 ..... 54  
 事件重要程度 ..... 48  
 事件重要程度变更功能 ..... 58  
 时刻型 ..... 31, 44  
 实例 ..... 13  
 实例名称 ..... 10  
 实型数据 ..... 31, 44  
 时钟功能 ..... 2  
 始终 OFF 标志 ..... 28, 54  
 始终 ON 标志 ..... 28, 54  
 输出变量 ..... 19  
 数据保护功能 ..... 19  
 数据的保持 / 非保持属性一览 ..... 79  
 数据类型 ..... 30  
 数据类型的转换 ..... 39  
 数据追踪 ..... 35  
 数据追踪的动作 ..... 37

数据追踪的规格	36
输入变量	19
输入输出变量	19
输入输出响应时间	58
输入数据 1 无效	34, 67
输入数据无效	34, 67
顺序控制的原理	18
顺序控制和运动控制的同步	20
顺序控制和运动控制的原理	16
算法	10, 17
所有标签数据链接通信中	41, 76

## T

TCP/IP 设定异常	39
TCP 应用程序设定异常	75
TCP 应用通信异常	37, 70
TIME	31, 44
TIME_OF_DAY	31, 44
调试程序	6
梯形图语言	80
梯形图中 FB、FUN 的连接	82
条件标志	126
通电时间	27, 52
通过当前值变更, 变更向外部的输出状态时的注意事项	32
通信端口 1 BOOTP 服务器连接异常	38, 73
通信端口 1 Ethernet 基本设定异常	38, 71
通信端口 1 IP 地址设定异常	38, 72
通信端口 1 IP 地址重复异常	38, 72
通信端口 1 MAC 地址异常	37, 71
通信端口 1 通信控制器异常	37, 71
通信端口 1 异常	36, 69
通信端口 1 在线	41, 76
通信端口异常	32, 36, 61, 69
通信控制器异常	32, 37, 62, 71
通信异常从站表	32, 64
通用变量	31, 60
凸轮数据变量	11
脱离指令中的从站表	34, 66

## U

UDINT	31
UINT	31
ULINT	31
_UnitVersion (单元版本)	28, 55
UPS 信号检测标志	27, 53
USINT	31

## W

WORD	31
外部变量	19
完全限定名称	134
网络公开	61
网络加入从站表	34, 65
网络配置	6
网络配置核对异常	32, 62

网络配置信息异常	32, 62
网络配置异常	32, 63
微分监控	40
位列型	31, 44
位列型、实型数据、字符串型的数据格式	35
文本	75
无法在线连接的构成	5
无效设定从站表	34, 66
无用户程序复原信息传送功能	17

## X

系统定义变量	12, 27
系统定义变量的个别规格	45
系统定义变量一览	22
系统服务	25
系统时刻	4, 22, 46
线圈	81
项目文件整体的保护功能	18
信息包监视停止	34, 66
信息包数据文件保存中	34, 67
序列 ID 功能	26
虚拟 SD 存储卡	
文件访问的排他控制	12
虚拟 SD 存储卡设定	5

## Y

异常状态	44
硬件修订版本	55
用户程序执行用 ID	14
用户定义变量	27
种类	28
用户事件	46, 52
用户异常状态	53, 57, 24, 48
与 EtherCAT 从站的访问	13
语法错误	130
运动控制的原理	19
运动控制功能模块的初始设定	12
运动控制功能模块的异常状态	31, 59
运动控制系统变量	12
运动控制周期超限	51
运行开始时 1 个周期 ON 标志	8, 28, 54
运行模式	25
运行模式改写	6

## Z

在程序中使用强制值时的注意事项	30
在线	41, 75
在线编辑	33
诊断统计信息日志保存周期	35, 68
诊断统计信息日志许可	35
诊断统计信息日志异常结束标志	35, 68
诊断统计信息日志执行状态标志	35, 68
正常目标节点信息	41, 77
整数型	31, 44
支持软件	5, 6
指令	124

指令错误 .....	127
指令错误发生标志 .....	8, 128, 131, 28, 55
轴 .....	11
轴变量 .....	11, 31, 60
轴及轴变量的创建和使用方法 .....	13
周期 / 执行条件 .....	7
轴异常状态 .....	31, 59
轴组变量 .....	11, 31, 60
轴组异常状态 .....	31, 59
主固定周期任务 .....	13
设定项目 .....	38
主机异常 .....	32, 61
主控 .....	143
追踪参数异常标志 .....	40, 30, 57
追踪触发监视标志 .....	40, 30, 57
追踪完成时 ON 标志 .....	40, 30, 57
追踪执行中 ON 标志 .....	40, 30, 57
准用户定义变量 .....	11, 27
字符串型 .....	32, 44

## 承诺事项

承蒙对欧姆龙株式会社(以下简称“本公司”)产品的一贯厚爱和支持,藉此机会再次深表谢意。

如果未特别约定,无论贵司从何处购买的产品,都将适用本承诺事项中记载的事项。

请在充分了解这些注意事项基础上订购。

### 1. 定义

本承诺事项中的术语定义如下。

- (1)“本公司产品”:是指“本公司”的FA系统机器、通用控制器、传感器、电子/结构部件。
- (2)“产品目录等”:是指与“本公司产品”有关的欧姆龙综合产品目录、FA系统设备综合产品目录、安全组件综合产品目录、电子/机构部件综合产品目录以及其他产品目录、规格书、使用说明书、操作指南等,包括以电子数据方式提供的资料。
- (3)“使用条件等”:是指在“产品目录等”资料中记载的“本公司产品”的使用条件、额定值、性能、运行环境、操作使用方法、使用时的注意事项、禁止事项以及其他事项。
- (4)“客户用途”:是指客户使用“本公司产品”的方法,包括将“本公司产品”组装或运用到客户生产的部件、电子电路板、机器、设备或系统等产品中。
- (5)“适用性等”:是指在“客户用途”中“本公司产品”的(a)适用性、(b)动作、(c)不侵害第三方知识产权、(d)法规法令的遵守以及(e)满足各种规格标准。

### 2. 关于记载事项的的注意事项

对“产品目录等”中的记载内容,请理解如下要点。

- (1)额定值及性能值是在单项试验中分别在各种条件下获得的值,并不构成对各额定值及性能值的综合条件下获得值的承诺。
- (2)提供的参考数据仅作为参考,并非可在该范围内一直正常运行的保证。
- (3)应用示例仅作参考,不构成对“适用性等”的保证。
- (4)如果因技术改进等原因,“本公司”可能会停止“本公司产品”的生产或变更“本公司产品”的规格。

### 3. 使用时的注意事项

选用及使用本公司产品时请理解如下要点。

- (1)除了额定值、性能指标外,使用时还必须遵守“使用条件等”。
- (2)客户应事先确认“适用性等”,进而再判断是否选用“本公司产品”。“本公司”对“适用性等”不做任何保证。
- (3)对于“本公司产品”在客户的整个系统中的设计用途,客户应负责事先确认是否已进行了适当配电、安装等事项。
- (4)使用“本公司产品”时,客户必须采取如下措施:(i)相对额定值及性能指标,必须在留有余量的前提下使用“本公司产品”,并采用冗余设计等安全设计(ii)所采用的安全设计必须确保即使“本公司产品”发生故障时也可将“客户用途”中的危险降到最小程度、(iii)构建随时提示使用者危险的完整安全体系、(iv)针对“本公司产品”及“客户用途”定期实施各项维护保养。
- (5)因DDoS攻击(分布式DoS攻击)、计算机病毒以及其他技术性有害程序、非法侵入,即使导致“本公司产品”、所安装软件、或者所有的计算机器材、计算机程序、网络、数据库受到感染,对于由此而引起的直接或间接损失、损害以及其他费用,“本公司”将不承担任何责任。  
对于(i)杀毒保护、(ii)数据输入输出、(iii)丢失数据的恢复、(iv)防止“本公司产品”或者所安装软件感染计算机病毒、(v)防止对“本公司产品”的非法侵入,请客户自行负责采取充分措施。
- (6)“本公司产品”是作为应用于一般工业产品的通用产品而设计生产的。如果客户将“本公司产品”用于以下所列用途,则本公司对产品不作任何保证。但“本公司”已表明可用于特殊用途,或已与客户有特殊约定时,另行处理。
  - (a)必须具备很高安全性的用途(例:核能控制设备、燃烧设备、航空/宇宙设备、铁路设备、升降设备、娱乐设备、医疗设备、安全装置、其他可能危及生命及人身安全的用途)
  - (b)必须具备很高可靠性的用途(例:燃气、自来水、电力等供应系统、24小时连续运行系统、结算系统、以及其他处理权利、财产的用途等)
  - (c)具有苛刻条件或严酷环境的用途(例:安装在室外的设备、会受到化学污染的设备、会受到电磁波影响的设备、会受到振动或冲击的设备等)
  - (d)“产品目录等”资料中未记载的条件或环境下的用途
- (7)除了不适用于上述3.(6)(a)至(d)中记载的用途外,“本产品目录等资料中记载的产品”也不适用于汽车(含二轮车,以下同)。请勿配置到汽车上使用。关于汽车配置用产品,请咨询本公司销售人员。

### 4. 保修条件

“本公司产品”的保修条件如下。

- (1)保修期限 自购买之日起1年。(但是,“产品目录等”资料中有明确说明时除外。)
- (2)保修内容 对于发生故障的“本公司产品”,由“本公司”判断并可选择以下其中之一方式进行保修。
  - (a)在本公司的维修保养服务点对发生故障的“本公司产品”进行免费修理(但是对于电子、结构部件不提供维修服务。)
  - (b)对发生故障的“本公司产品”免费提供同等数量的替代品
- (3)当故障因以下任何一种情形引起时,不属于保修的范围。
  - (a)将“本公司产品”用于原本设计用途以外的用途
  - (b)超过“使用条件等”范围的使用
  - (c)违反本注意事项“3.使用时的注意事项”的使用
  - (d)非因“本公司”进行的改装、修理导致故障时
  - (e)非因“本公司”出品的软件导致故障时
  - (f)“本公司”生产时的科学、技术水平无法预见的原因
  - (g)除上述情形外的其它原因,如“本公司”或“本公司产品”以外的原因(包括天灾等不可抗力)

### 5. 责任限制

本承诺事项中记载的保修是关于“本公司产品”的全部保证。对于因“本公司产品”而发生的其他损害,“本公司”及“本公司产品”的经销商不负任何责任。

### 6. 出口管理

客户若将“本公司产品”或技术资料出口或向境外提供时,请遵守中国及各国关于安全保障进出口管理方面的法律、法规。否则,“本公司”有权不予提供“本公司产品”或技术资料。

IC320GC-zh

202304

注:规格如有变更,恕不另行通知。请以最新产品说明书为准。

欧姆龙自动化(中国)有限公司

<http://www.fa.omron.com.cn> 咨询热线:400-820-4535