



三菱电机FA整合控制器



MELSEC MX Controller



MELSEC MX控制器
编程技术指南


- MXR300-16
- MXR300-32
- MXR300-64
- MXR500-128
- MXR500-256
- SW1DND-GXW3-J

安全注意事项

(使用之前请务必阅读)

在使用本产品之前，应仔细阅读本手册以及本手册中介绍的关联手册，同时在充分注意安全的前提下正确地操作。在“安全注意事项”中，安全注意事项被分为“警告”和“注意”两个等级。

 警告	表示错误操作可能造成危险后果，导致死亡或重伤事故。
 注意	表示错误操作可能造成危险后果，导致中度伤害、轻伤及设备损失。

此外，即使注意这一级别的事项根据情况不同也有可能引发严重后果。

两级注意事项记载的都是重要内容，必须遵照执行。

请妥善保管本手册以备需要时阅读，并将本手册交给最终用户。

关于产品的应用

(1) 在使用三菱电机可编程控制器时，应该符合以下条件：即使在可编程控制器设备出现问题或故障时也不会导致重大事故，并且应在设备外部系统地配备能应付任何问题或故障的备用设备及失效安全功能。

(2) 三菱电机可编程控制器是以一般工业用途等为对象设计和制造的通用产品。

因此，三菱电机可编程控制器不应用于以下设备・系统等特殊用途。如果用于以下特殊用途，对于三菱电机可编程控制器的质量、性能、安全等所有相关责任（包括但不限于债务未履行责任、瑕疵担保责任、质量保证责任、违法行为责任、制造物责任），三菱电机将不负责。

- 面向各电力公司的核电站以及其它发电厂等对公众有较大影响的用途。
- 用于各铁路公司或公用设施目的等有特殊质量保证体系要求的用途。
- 航空航天、医疗、铁路、焚烧・燃料装置、载人移动设备、载人运输装置、娱乐设备、安全设备等预计对人身财产有较大影响的用途。

然而，对于上述应用，如果在限定于具体用途，无需特殊质量（超出一般规格的质量等）要求的条件下，经过三菱电机的判断也可以使用三菱电机可编程控制器，详细情况请与当地三菱电机代表机构协商。

(3) 因拒绝服务攻击（DoS攻击）、非法访问、计算机病毒以及其他网络攻击引发的可编程控制器与系统方面的各种问题，三菱电机不承担责任。

前言

在此非常感谢贵方购买了三菱电机可编程控制器产品。

本手册是用于让用户了解使用MELSEC MX控制器(MX-R型)的下述对象模块时的有效编程技术相关内容的手册。

在使用之前应熟读本手册及关联手册，在充分了解控制器的功能・性能的基础上正确地使用本产品。

此外，将本手册中介绍的程序示例应用于实际系统的情况下，应充分验证对象系统中不存在控制方面的问题。应将本手册交给最终用户。

对象模块

MXR300-16、MXR300-32、MXR300-64、MXR500-128、MXR500-256

目录

安全注意事项	1
关于产品的应用	2
前言	2
关联手册	6
术语	7
总称/简称	7
第1章 概要	8
第2章 提高工程的可读性	10
2.1 通过FB和结构体进行部件化	10
概要	10
执行步骤	11
2.2 FB的数组化	16
概要	16
执行步骤	16
2.3 运动轴分组	20
概要	20
执行步骤	20
2.4 划分全局标签	24
概要	24
执行步骤	24
第3章 加速程序转换	27
3.1 划分程序文件	28
概要	28
执行步骤	30
3.2 程序的FB化	37
概要	37
执行步骤	38
3.3 缩短包含控制语法的程序	42
概要	42
执行步骤	43
3.4 更改全局标签(结构体)的设置	45
概要	45
执行步骤	46
3.5 减少对数组同一元素的连续引用	50
概要	50
执行步骤	50
第4章 缩短扫描时间	53
4.1 BMOV指令替换为多个指令	54
概要	54
执行步骤	54
4.2 减少上升沿执行指令	59
概要	59
执行步骤	59

4.3	减少模块访问指令	61
	概要	61
	执行步骤	62
4.4	应用初始执行类型程序	64
	概要	64
	执行步骤	64
4.5	省略自变量和直接引用	70
	概要	70
	执行步骤	70
4.6	划分含有多条指令的梯形图块	74
	概要	74
	执行步骤	74
4.7	应用标签初始值	77
	概要	77
	执行步骤	77
第5章 优化运动控制		79
5.1	使用通信周期混合提高特定轴速度	79
	概要	79
	执行步骤	79
5.2	同步用户程序和运动运算	84
	概要	84
	执行步骤	84
第6章 标签效率		91
6.1	减少全局标签	91
	概要	91
	执行步骤	91
6.2	删除未使用标签	93
	概要	93
	执行步骤	93
第7章 高速CC-Link IE TSN通信		95
7.1	缩短刷新时间	96
	概要	96
	执行步骤	97
7.2	均衡设备站的台数	103
	概要	103
	执行步骤	103
7.3	设置发送上限大小	104
	概要	104
	执行步骤	104
7.4	减少设备站台数	106
	概要	106
	执行步骤	106
7.5	删除链接软元件设置	110
	概要	110
	执行步骤	110
7.6	减少链接软元件设置点数	115
	概要	115

执行步骤	115
7.7 统一网络同步通信设置	120
概要	120
执行步骤	120
7.8 设置循环传送分散	123
概要	123
执行步骤	123
修订记录	127
资讯与服务	128
商标	128

关联手册

手册名称	内容
MELSEC MX控制器 编程技术指南 [BCN-89999-0837] (本手册)	记载了提高MELSEC MX控制器工程的可读性、加速程序转换、缩短扫描时间、优化运动控制、标签效率以及高速CC-Link IE TSN通信的编程技术。
MELSEC MX控制器 (MX-R型) 用户手册 [SH-082642CHN]	记载了控制器运行的步骤、规格、软元件、存储器、功能、参数、故障排除等有关内容。
MELSEC MX控制器 (MX-R型) 编程手册 [SH-082645CHN]	记载了程序语言的规格、控制器的指令、通用函数/FB、运动控制FB的规格的有关内容。
GX Works3 操作手册 [SH-081271CHN]	对GX Works3的系统配置、参数设置、在线功能的操作方法等有关内容进行说明。

术语

在本手册中，除非特别标明，否则将使用下述术语进行说明。

术语	内容
全局标签	工程内创建多个程序数据时，对于所有程序数据皆为有效的标签。 全局标签中，有GX Works3自动生成的模块固有标签(模块标签)和对于任意指定的软元件可以创建的标签。
设备站	主站以外的站(本地站、远程站)。
程序块	配置程序的程序部件的集合。
主站	是控制网络整体的站。可以与全部的站进行循环传送及瞬时传送。 1个网络中仅存在1个。
运动控制FB	运动控制FB相关的FB。名称从MC_或MCv_开始。
模块标签	是各模块将特有定义的存储器(输入输出信号及缓冲存储器)以，任意字符串表示的标签。 对于使用的模块，GX Works3将自动生成标签，且该标签可作为全局标签使用。
标签	指在输入输出数据或内部处理中指定任意字符串的变量。
远程站	对位单位的输入输出信号和字单位的输入输出数据进行循环传送的站。也可以进行瞬时传送。
链接软元件	是CC-Link IE TSN功能部或网络模块内部具有的软元件(RX、RY、RWr、RWw、LB、LW)。
本地站	是与主站及其它本地站进行循环传送及瞬时传送的站。
虚拟驱动轴	可以虚拟生成指令的虚拟轴。
轴	进行运动控制的对象。
轴变量	包括轴相关的参数及数据，AXIS_*型变量的实例。
实编码器轴	是从与设备连接的编码器的当前位置生成指令位置的轴。用于单轴同步控制的主轴。
实驱动轴	与对应CiA402驱动配置文件的csp/csv/cst模式(依次指令)的CC-Link IE TSN对应设备关联的轴。
CC-Link IE TSN	是采用扩展标准以太网规范的“TSN(Time-Sensitive Networking)”，可以同时处理确保实时性的控制和其他开放式网络信息的开放式网络。
CC-Link IE TSN Class	CC-Link协会根据CC-Link IE TSN对应的设备及交换集线器的功能・性能进行的等级区分。 关于CC-Link IE TSN Class，请参阅CC-Link协会发行的CC-Link IE TSN Installation Manual(BAP-C3007ENG-001)。
FB	FB是功能块(Function Block)的简称，将在顺控程序中重复使用的梯形图块部件化以便在顺控程序中使用。 这样可以提高程序开发效率，减少程序错误，提高程序质量。
FB实例	粘贴到顺控程序中的FB。
GX Works3	是MELSEC可编程控制器软件包的产品名。
RWr	链接软元件的远程寄存器。设备站以16位(1字)单位向主站输入。(本地站有部分不同。)
RWw	链接软元件的远程寄存器。主站以16位(1字)单位向设备站输出。(本地站有部分不同。)
RX	链接软元件的远程输入。设备站以位单位向主站输入。(本地站有部分不同。)
RY	链接软元件的远程输出。主站以位单位向设备站输出。(本地站有部分不同。)

总称/简称

本参考文献中，除特别标明外，皆使用下述所示的总称和简称进行说明。

术语	内容
工程工具	是进行控制器及伺服放大器的设置、编程、调试及维护用的工具。
控制器	MXR300-16、MXR300-32、MXR300-64、MXR500-128、MXR500-256的总称。

1 概要

本手册介绍了提高MELSEC MX控制器(以下简称控制器)工程和程序的结构体和速度、优化运动控制、标签效率以及高速CC-Link IE TSN通信等技术。

关于控制器的基本参数设置和编程, 请参阅下述手册。

📖 MELSEC MX控制器(MX-R型)用户手册

📖 MELSEC MX控制器(MX-R型)编程手册

此外, 本手册通过下述版本的GX Works3进行说明。

- GX Works3 Ver. 1.115V

技术和效果

各技术预期的效果如下所示。

技术		效果				参照	
		可读性/ 可移植性 提高	转换时间缩短	扫描时间缩短	同步性提高		
程序结构	通过FB和结构体进行部件化	○	—	—	—	☞ 10页 提高工程的可读性	
	FB的数组化	○	—	—	—		
	运动轴分组	○	—	—	—		
	划分全局标签	○	○	—	—		
加速程序转换	划分程序文件	○	○	—	—	☞ 27页 加速程序转换	
	程序的FB化	○	○	—	—		
	缩短包含控制语法的程序	○	○	—	—		
	更改全局标签(结构体)的设置	—	○	—	—		
	减少对数组同一元素的连续引用	—	○	○	—		
缩短扫描时间	BMOV指令替换为多个指令	—	—	○	—	☞ 53页 缩短扫描时间	
	减少上升沿执行指令	—	—	○	—		
	减少模块访问指令	—	—	○	—		
	应用初始执行类型程序	—	—	○	—		
	省略自变量和直接引用	—	—	○	—		
	划分含有多条指令的梯形图块	○	—	○	—		
	应用标签初始值	○	—	○	—		
优化运动控制	使用通信周期混合提高特定轴速度	—	—	○	—	☞ 79页 优化运动控制	
	同步用户程序和运动运算	—	—	—	○		
标签效率	删除全局标签	—	○	—	—	☞ 91页 标签效率	
	删除未使用标签	—	○	—	—		
高速CC-Link IE TSN通信	缩短刷新时间	缩短刷新时间	—	—	○	☞ 95页 高速CC-Link IE TSN通信	
	缩短通信周期间隔/ 循环传送时间	均衡设备站的台数	—	—	○		—
		设置发送上限大小					
		减少设备站台数					
		删除链接软元件设置					
		减少链接软元件设置点数					
		统一网络同步通信设置					
设置循环传送分散							

2 提高工程的可读性

将程序分为较小的处理单元(部件)建立层次结构并统一工程的基本配置和规则，可以提高工程的可读性。提高工程的可读性可以在工程更改时更容易识别修改部分，同时也能使工程创建者以外的人更容易理解。

本章介绍提高工程可读性的技术。

技术一览

本章介绍的技术一览如下所示。

项目	内容	参照
通过FB和结构体进行部件化	使用FB和结构体将各程序块中通用的程序部件化。	☞ 10页 通过FB和结构体进行部件化
FB的数组化	将同时执行的FB数组化，并替换为与FOR语句组合的FB。	☞ 16页 FB的数组化
运动轴分组	将轴号按照实驱动轴、实编码器轴和虚拟驱动轴等轴类型进行分组。	☞ 20页 运动轴分组
划分全局标签	将全局标签分组为有意义的单元，并移动至不同全局标签编辑器。	☞ 24页 划分全局标签

2.1 通过FB和结构体进行部件化

概要

通过将各程序块中通用的FB及结构体部件化，可减少程序的记述量，创建易读的程序。

将程序部件化还有下述好处。

提高程序的可移植性和再利用性

将程序部件化可以使处理或流程单元程序的移植和再利用变得更加容易。

提高程序的维护性

如果将程序部件化并在多个处理或流程中使用可提高程序的维护性，因为即使处理有添加或更改，只修改部件化的结构体或FB即可，无需分别修改多个程序。

执行步骤

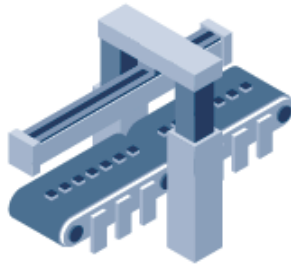
以下述组装装置、检查装置、包装装置为例，说明创建运输的通用部件并在各装置的程序中使用的步骤。

运输的通用部件使用运输工件的“MC_MoveRelative”（相对值定位）和进行停止的“MC_Stop”（强制停止）的运动控制FB。

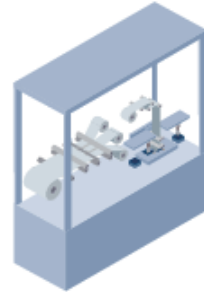
组装装置



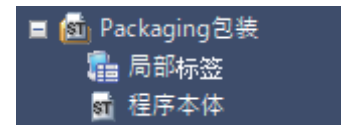
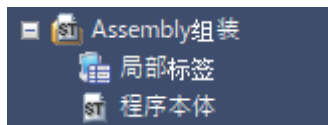
检查装置



包装装置



各装置的程序



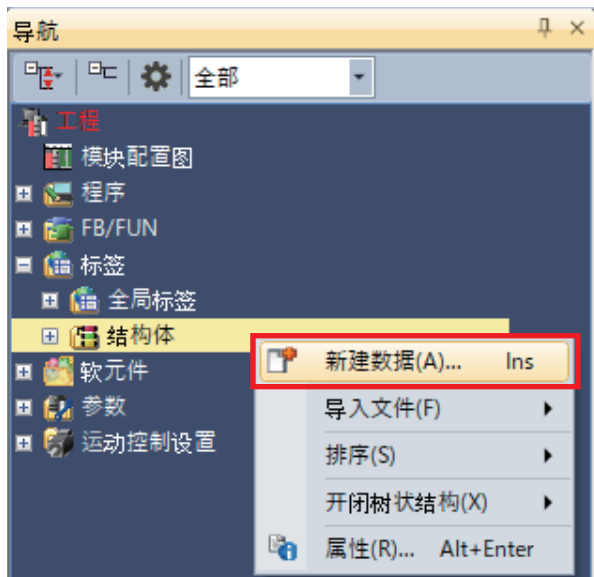
通用部分(运输)
•MC_MoveRelative
•MC_Stop



操作步骤

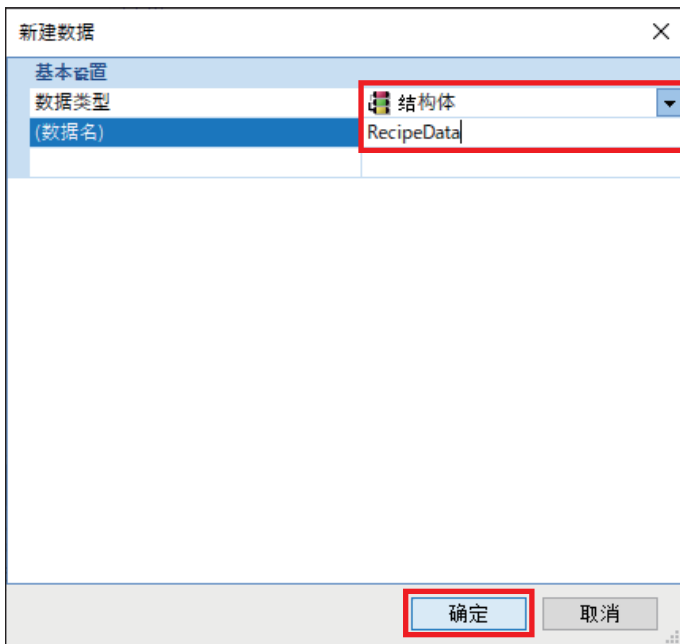
1. 新建一个结构体，定义各流程的运输及停止的配方数据。

🔗 导航窗口⇒[标签]⇒右击[结构体]⇒[新建数据]



2. 设置各项目，点击[确定]按钮。

- 数据类型：结构体
- (数据名)：RecipeData



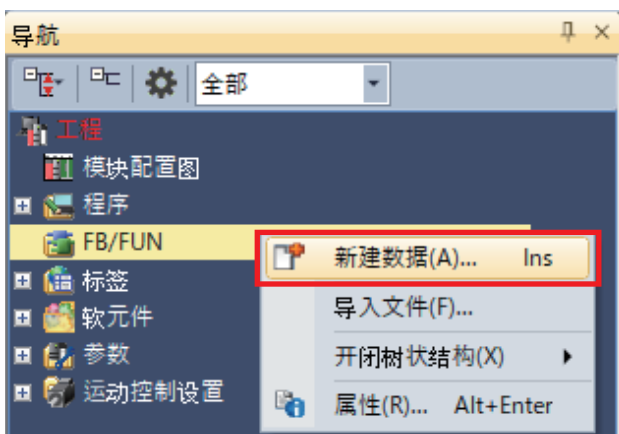
3. 为结构体“RecipeData”的构件定义以下“MC_MoveRelative”及“MC_Stop”的通用标签。

No.	标签名	数据类型	类	初始值	常数
1	Distance	双精度实数	...		
2	Velocity	双精度实数	...		
3	Acceleration	双精度实数	...		
4	Deceleration	双精度实数	...		
5	Jerk	双精度实数	...		

No.	标签名	数据类型
1	Distance	双精度实数
2	Velocity	双精度实数
3	Acceleration	双精度实数
4	Deceleration	双精度实数
5	Jerk	双精度实数

4. 新建通用部件的FB。

👉 导航窗口⇒右击[FB/FUN]⇒[新建数据]



5. 设置各项目，点击[确定]按钮。

- 数据类型：FB
- (数据名)：Transport
- 程序语言：ST

6. 打开FB“Transport”的局部标签，定义运输所需的以下FB实例及标签。

No.	标签名	数据类型	类	初始值	常数
1	MC_MoveRelative_1	MC_MoveRelative	VAR		
2	MC_Stop_1	MC_Stop	VAR		
3	Axis_Transport	AXIS_REF	VAR_INPUT		
4	Start_Transport	位	VAR_INPUT		
5	Stop_Transport	位	VAR_INPUT		
6	RecipeData_Transport	RecipeData	VAR_PUBLIC		
7	RecipeData_Stop	RecipeData	VAR_PUBLIC		

No.	标签名	数据类型	类	备注
1	MC_MoveRelative_1	MC_MoveRelative	VAR	运动控制FB“MC_MoveRelative”的FB实例。
2	MC_Stop_1	MC_Stop	VAR	运动控制FB“MC_Stop”的FB实例。
3	Axis_Transport	AXIS_REF	VAR_INPUT	使用结构体“AXIS_REF”作为数据类型。
4	Start_Transport	位	VAR_INPUT	—
5	Stop_Transport	位	VAR_INPUT	—
6	RecipeData_Transport	RecipeData	VAR_PUBLIC	使用步骤1~3中创建的结构体“RecipeData”作为数据类型。
7	RecipeData_Stop	RecipeData	VAR_PUBLIC	

7. 打开FB“Transport”的程序本体，记述执行运动控制FB“MC_MoveRelative”和“MC_Stop”的下述程序。

```

ST Transport [FB] [ST] * x
1 //运输工件
2 MC_MoveRelative_1.Axis := Axis_Transport;
3 MC_MoveRelative_1(
4     Execute:= Start_Transport ,
5     Distance:= RecipeData_Transport.Distance ,
6     Velocity:= RecipeData_Transport.Velocity ,
7     Acceleration:= RecipeData_Transport.Acceleration ,
8     Deceleration:= RecipeData_Transport.Deceleration ,
9     Jerk:= RecipeData_Transport.Jerk
10 );
11
12 //停止
13 MC_Stop_1.Axis := Axis_Transport;
14 MC_Stop_1(
15     Execute:= Stop_Transport ,
16     Deceleration:= RecipeData_Stop.Deceleration ,
17     Jerk:= RecipeData_Stop.Jerk
18 );

```

行数	处理内容
第1~3行	为运动控制FB“MC_MoveRelative”和“MC_Stop”的自变量“Axis”指定标签“Axis_Transport”。
第5~13行	执行运动控制FB“MC_MoveRelative”。 为自变量“Distance”、“Velocity”、“Acceleration”、“Deceleration”和“Jerk”指定标签“RecipeData_Transport”的各构件。
第15~20行	执行运动控制FB“MC_Stop”。 为自变量“Deceleration”和“Jerk”指定标签“RecipeData_Stop”的各构件。

8. 打开组装装置程序块(本示例中为“Assembly组装”)的局部标签，添加在步骤4~7中创建的FB“Transport”的FB实例。

- 标签名: Transport
- 数据类型: Transport (FB“Transport”)
- 类: VAR

标签名	数据类型	类	初始值	常数
1 Transport	Transport	VAR		

9. 打开各装置的程序本体，使用通用部件FB“Transport”创建任意的运输程序。

• 组装装置的程序

```

st] Assembly组装 [PRG] [ST] 28字节 ×
1 //运输处理
2 //初始设置
3 Transport.Axis_Transport := Axis0001.AxisRef;
4 Transport.RecipeData_Transport.Distance := 10000.0;
5 Transport.RecipeData_Transport.Acceleration := 2000.0;
6 Transport.RecipeData_Transport.Deceleration := 2000.0;
7 Transport.RecipeData_Transport.Jerk := 300.0;
8
9 //运输开始
10 IF G_Process = G_Process_Assembly THEN
11   IF NOT G_Stop_Assembly THEN
12     Transport.Start_Transport := TRUE ;
13   ELSE
14     Transport.Start_Transport := FALSE ;
15     Transport.Stop_Transport := TRUE ;
16   END_IF;
17 END_IF;
18 Transport();
19
20 //后续组装的独立处理

```

• 检查装置的程序

```

st] Inspection检查 [PRG] [ST] 28字节 ×
1 //运输处理
2 //初始设置
3 Transport.Axis_Transport := Axis0001.AxisRef;
4 Transport.RecipeData_Transport.Distance := 20000.0;
5 Transport.RecipeData_Transport.Acceleration := 2000.0;
6 Transport.RecipeData_Transport.Deceleration := 2000.0;
7 Transport.RecipeData_Transport.Jerk := 300.0;
8
9 //运输开始
10 IF G_Process = G_Process_Inspection THEN
11   IF NOT G_Stop_Inspection THEN
12     Transport.Start_Transport := TRUE ;
13   ELSE
14     Transport.Start_Transport := FALSE ;
15     Transport.Stop_Transport := TRUE ;
16   END_IF;
17 END_IF;
18 Transport();
19
20 //后续检查的独立处理

```

• 包装装置的程序

```

st] Packaging包装 [PRG] [ST] 28字节 ×
1 //运输处理
2 //初始设置
3 Transport.Axis_Transport := Axis0001.AxisRef;
4 Transport.RecipeData_Transport.Distance := 30000.0;
5 Transport.RecipeData_Transport.Acceleration := 2000.0;
6 Transport.RecipeData_Transport.Deceleration := 2000.0;
7 Transport.RecipeData_Transport.Jerk := 300.0;
8
9 //运输开始
10 IF G_Process = G_Process_Packaging THEN
11   IF NOT G_Stop_Packaging THEN
12     Transport.Start_Transport := TRUE ;
13   ELSE
14     Transport.Start_Transport := FALSE ;
15     Transport.Stop_Transport := TRUE ;
16   END_IF;
17 END_IF;
18 Transport();
19
20 //后续包装的独立处理

```

2.2 FB的数组化

概要

通过同时执行多次的FB数组化并使用FOR语句重复执行，可减少程序的记述量，从而创建易读的程序。

将FB数组化还有下述好处。

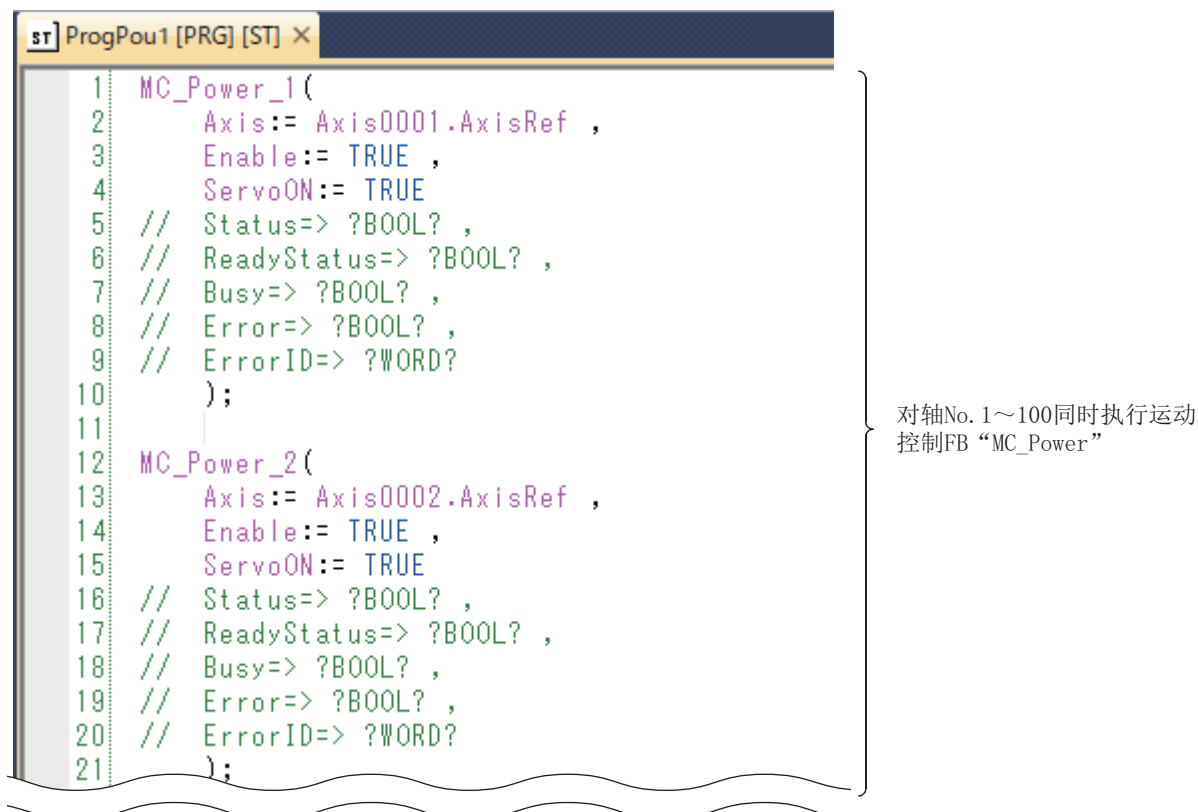
提高程序的可移植性

减少程序的记述量也可减少移植程序时的程序量，从而提高程序的可移植性。

执行步骤

以下述运动程序为例，说明将对多个轴同时执行的运动控制FB“MC_Power”数组化的步骤。

- ST程序



```
1 MC_Power_1(  
2   Axis:= Axis0001.AxisRef ,  
3   Enable:= TRUE ,  
4   ServoON:= TRUE  
5 // Status=> ?BOOL? ,  
6 // ReadyStatus=> ?BOOL? ,  
7 // Busy=> ?BOOL? ,  
8 // Error=> ?BOOL? ,  
9 // ErrorID=> ?WORD?  
10 );  
11 |  
12 MC_Power_2(  
13   Axis:= Axis0002.AxisRef ,  
14   Enable:= TRUE ,  
15   ServoON:= TRUE  
16 // Status=> ?BOOL? ,  
17 // ReadyStatus=> ?BOOL? ,  
18 // Busy=> ?BOOL? ,  
19 // Error=> ?BOOL? ,  
20 // ErrorID=> ?WORD?  
21 );
```

对轴No. 1~100同时执行运动控制FB“MC_Power”

- 局部标签



	标签名	数据类型	类	初始值	常数
1	MC_Power_1	MC_Power	VAR		
2	MC_Power_2	MC_Power	VAR		
3	MC_Power_3	MC_Power	VAR		
4	MC_Power_4	MC_Power	VAR		

“MC_Power_1” ~ “MC_Power_100”的FB实例

操作步骤

1. 将数据化的FB实例设为1维的数组。

打开局部标签，在FB实例“MC_Power_1”的数据类型中添加“(1..100)”。此外，根据数组化更改标签名。

- 标签名: MC_Power_Axis1_100
- 数据类型: MC_Power(1..100)

标签名	数据类型	类
1 MC_Power_Axis1_100	MC_Power(1..100)	VAR
2 MC_Power_2	MC_Power	VAR

要点

- 将数组元素数是从执行FB轴No. 的起始到末尾设置的。
- 也可以点击“数据类型”列的[...]按钮在显示的“数据类型选择”画面中更改标签的数据类型。在“数组元素”栏中勾选“数组”，设置元素数。

2. 新建一个局部标签(显示执行运动控制FB的轴No. 的标签)用于FOR语句的重复变量。

- 标签名: w_AxisNo
- 数据类型: 字[有符号]
- 类: VAR

标签名	数据类型	类
1 MC Power Axis1 100	MC Power(1..100)	VAR
2 w_AxisNo	字[有符号]	VAR
3 MC_Power_2	MC_Power	VAR

3. 打开程序，在FB实例“MC_Power_Axis1_100”中添加数组元素“w_AxisNo”。
此外将数组元素“w_AxisNo”从1逐一增加至100，并添加以下FOR语句执行轴No. 1~100的FB。

```
FOR w_AxisNo := 1 TO 100 BY 1 DO  
MC_Power_Axis1_100[w_AxisNo](  
  执行语句;  
);  
END_FOR;
```

```
1 FOR w_AxisNo := 1 TO 100 BY 1 DO  
2   MC_Power_Axis1_100[w_AxisNo](  
3     Axis := Axis0001.AxisRef ,  
4     Enable := TRUE ,  
5     ServoON := TRUE  
6     // Status=> ?BOOL? ,  
7     // ReadyStatus=> ?BOOL? ,  
8     // Busy=> ?BOOL? ,  
9     // Error=> ?BOOL? ,  
10    // ErrorID=> ?WORD?  
11  );  
12 END_FOR;
```

对轴No. 1~100重复执行
FB “MC_Power_Axis1_100”

要点

关于FOR语句的详细内容，请参阅下述手册。
 MELSEC MX控制器(MX-R型)编程手册

4. 使用轴变量AxisName.AxisRef.AxisNo(轴No.)添加为FB实例“MC_Power_Axis1_100”的输入变量“Axis”直接指定标签“w_AxisNo”(INT型)的下述语句。

```
MC_Power_Axis1_100[w_AxisNo].Axis.AxisNo := INT_TO_WORD(w_AxisNo);
```

```
1 FOR w_AxisNo := 1 TO 100 BY 1 DO  
2   MC_Power_Axis1_100[w_AxisNo].Axis.AxisNo := INT_TO_WORD(w_AxisNo);  
3   MC_Power_Axis1_100[w_AxisNo](  
4     Axis := Axis0001.AxisRef ,  
5     Enable := TRUE ,  
6     ServoON := TRUE  
7     // Status=> ?BOOL? ,  
8     // ReadyStatus=> ?BOOL? ,  
9     // Busy=> ?BOOL? ,  
10    // Error=> ?BOOL? ,  
11    // ErrorID=> ?WORD?  
12  );  
13 END_FOR;
```

要点

轴变量AxisName.AxisRef.AxisNo(轴No.)的数据类型为WORD(UINT)。因此，为AxisName.AxisRef.AxisNo指定WORD(UINT)型的数据。

5. 注释掉或删除FB执行语句中设置输入变量“Axis”的语句(示例中的第4行)。
此外,从程序和局部标签中删除不需要的FB实例(“MC_Power_2”~“MC_Power_100”)。

• ST程序

```

1 FOR w_AxisNo := 1 TO 100 BY 1 DO
2   MC_Power_Axis1_100[w_AxisNo].Axis.AxisNo := INT_TO_WORD(w_AxisNo);
3   MC_Power_Axis1_100[w_AxisNo](
4     // Axis:= Axis0001.AxisRef ,
5     Enable:= TRUE ,
6     ServoON:= TRUE
7     // Status=> ?BOOL? ,
8     // ReadyStatus=> ?BOOL? ,
9     // Busy=> ?BOOL? ,
10    // Error=> ?BOOL? ,
11    // ErrorID=> ?WORD?
12  );
13 END_FOR;
14
15
16
17
18
19
20
21
22

```

注释掉

删除“MC_Power_2”~
“MC_Power_100”的FB实例

• 局部标签

标签名	数据类型	类	初始值	常数
1 MC_Power_Axis1_100	MC_Power(1..100)	VAR	...	
2 w_AxisNo	字[有符号]	VAR	...	
3				

删除“MC_Power_2”~
“MC_Power_100”的FB实例

要点

记述FB时,可以省略不使用或不更改初始值的输入输出变量。

2.3 运动轴分组

概要

使用实驱动轴、实编码器轴和虚拟驱动轴等多个轴类型时，可通过将轴按照轴类别和装置进行分组，或确定各轴类型的轴编号范围和起始轴编号明确程序中使用的轴。

对运动轴进行分组还有下述好处。

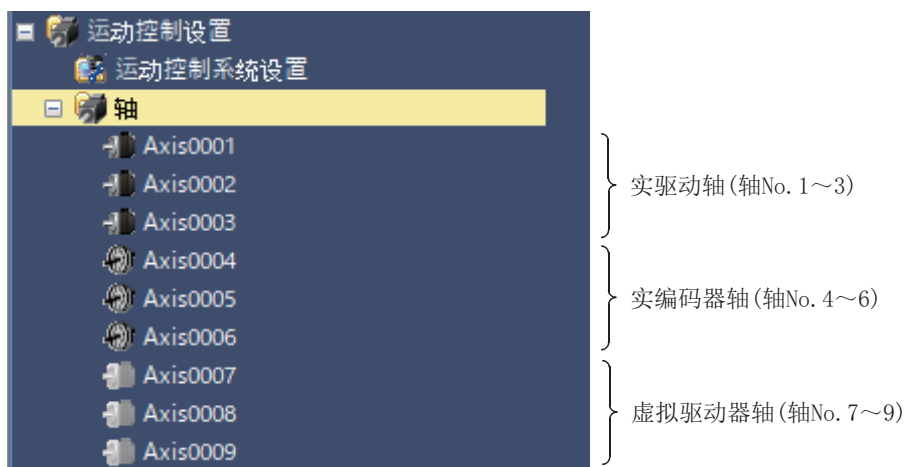
提高工程的维护性

根据轴类型和装置分组轴的文件夹，可以更轻松地进行管理。此外，通过确定各轴类型的轴编号范围和起始轴编号，可以减少因设备改造而添加/删除轴时对程序的修改，从而提高维护性。

执行步骤

以下述程序和轴编号的规则为例，说明按照轴类型对运动轴进行分组的步骤。

- 工程



- 轴编号的规则

轴类型	轴编号
实驱动轴	1~999
实编码器轴	1000~1999
虚拟驱动轴	2000~2999

操作步骤

1. 新建文件夹用于对轴进行分组。

🔗 导航窗口⇒[运动控制设置]⇒右击[轴]⇒[新建文件夹]

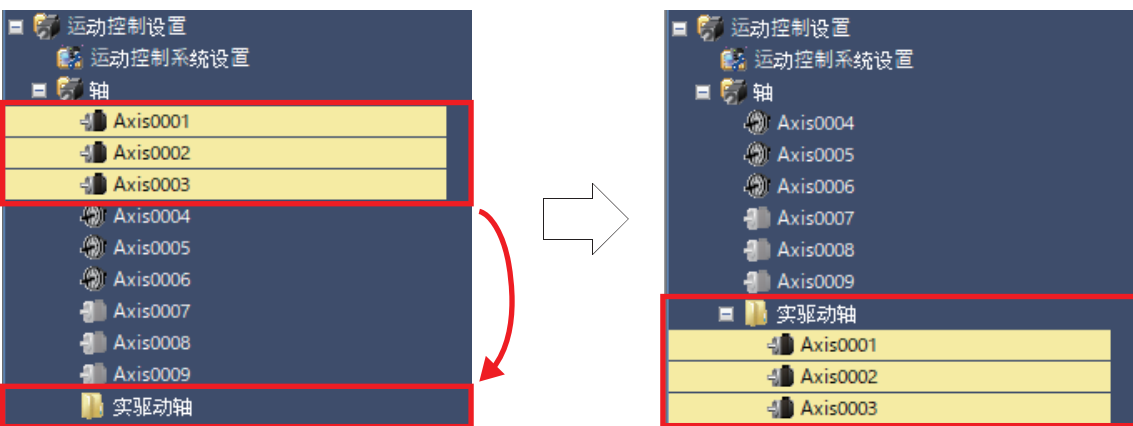


2. 将文件夹名设置为任意的名称。

- 文件夹名：实驱动轴



3. 选择存储的轴“Axis0001”～“Axis0003”，拖放到步骤2中创建的文件夹中。



4. 重复步骤1~3，将实编码器轴和虚拟驱动轴以相同方式进行分组。



5. 打开轴参数设置。

☞ 导航窗口⇒[运动控制设置]⇒[轴]⇒双击任意轴



6. 为各轴类型设置轴编号。

☞ 要设置的轴类型⇒[轴信息]⇒[轴号]

•实际驱动器轴：轴No. 1~3

项目	Axis0001	Axis0002	Axis0003
轴信息	设置 轴信息		
轴号	1	2	3
基本参数	设置 基本参数		
站地址设置			
轴类型设置	0:实际驱动器轴	0:实际驱动器轴	0:实际驱动器轴
轴仿真器启用	0:禁用	0:禁用	0:禁用
控制周期设置	0:自动设置	0:自动设置	0:自动设置

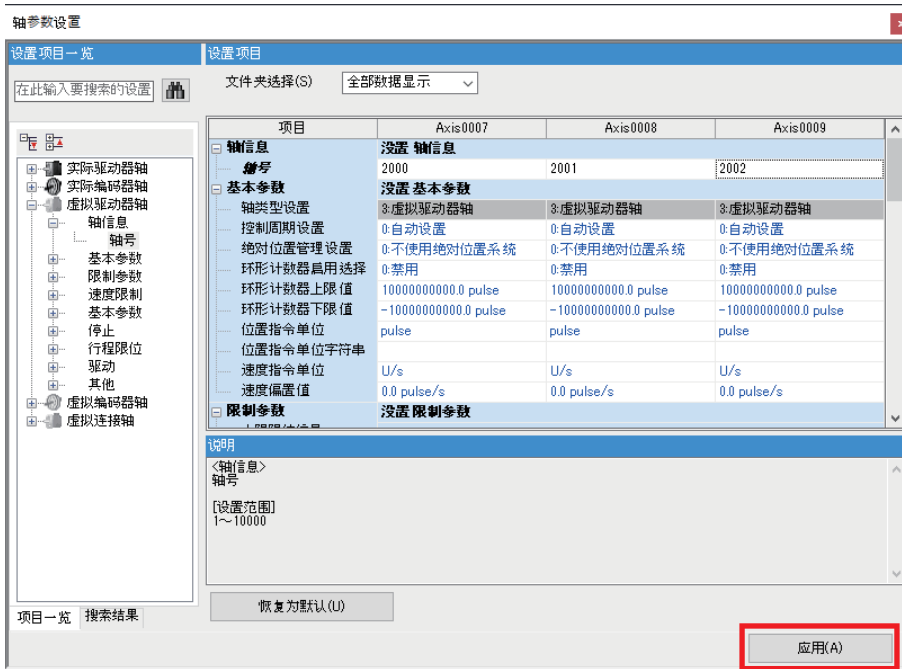
•实际编码器轴：轴No. 1000~1002

项目	Axis0004	Axis0005	Axis0006
轴信息	设置 轴信息		
轴号	1000	1001	1002
基本参数	设置 基本参数		
站地址设置			
轴类型设置	2:实际编码器轴	2:实际编码器轴	2:实际编码器轴
实际编码器轴类型设置	1:经由驱动器模块	1:经由驱动器模块	1:经由驱动器模块
控制周期设置	0:自动设置	0:自动设置	0:自动设置

•虚拟驱动器轴：轴No. 2000~2002

项目	Axis0007	Axis0008	Axis0009
轴信息	设置 轴信息		
轴号	2000	2001	2002
基本参数	设置 基本参数		
轴类型设置	3:虚拟驱动器轴	3:虚拟驱动器轴	3:虚拟驱动器轴
控制周期设置	0:自动设置	0:自动设置	0:自动设置
绝对位置管理设置	0:不使用绝对位置系统	0:不使用绝对位置系统	0:不使用绝对位置系统
环形计数器启用选择	0:禁用	0:禁用	0:禁用

7. 点击[应用]按钮应用设置。



注意事项

如果未为各轴设置站地址(IP地址)，则应用时会发生警告。

如果发生警告，请从各轴的参数设置的[基本参数]⇒[站地址设置]中设置站地址。

2.4 划分全局标签

概要

通过将全局标签分为流程和装置等有意义的单位，并移动至不同的全局标签编辑器，可以明确各单元的标签。划分全局标签还有下述好处。

提高全局标签的可移植性

通过划分全局标签，可以将全局标签移植到装置和流程等每个有意义的单元。

缩短程序转换时间

划分全局标签可将程序的转换范围局部化，从而降低程序转换时的影响。因此，可以缩短程序转换时间。

提高全局标签的编辑操作性

将全局标签分隔为装置或流程等有意义的单元划分，便于在编辑时发现和修改相应的全局标签。

执行步骤

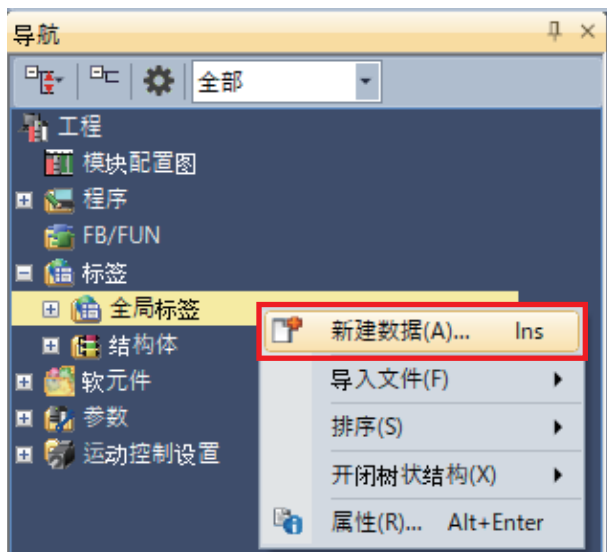
以下述全局标签为例，说明划分为装置1使用的全局标签和装置2使用的全局标签的步骤。

标签名	数据类型	类	分配(软元件/标签)	初始值	常数
1 CalcInPou1_1	字[有符号]	VAR_GLOBAL			
2 CalcInPou1_2	字[有符号]	VAR_GLOBAL			
3 CalcInPou1_3	字[有符号]	VAR_GLOBAL			
4 CalcInPou1_4	字[有符号]	VAR_GLOBAL			
5 SignallnPou1_1	位	VAR_GLOBAL			
6 SignallnPou1_2	位	VAR_GLOBAL			
7 SignallnPou1_3	位	VAR_GLOBAL			
8 SignallnPou1_4	位	VAR_GLOBAL			
9 CalcInPou2_1	字[有符号]	VAR_GLOBAL			
10 CalcInPou2_2	字[有符号]	VAR_GLOBAL			
11 CalcInPou2_3	字[有符号]	VAR_GLOBAL			
12 CalcInPou2_4	字[有符号]	VAR_GLOBAL			
13 SignallnPou2_1	位	VAR_GLOBAL			
14 SignallnPou2_2	位	VAR_GLOBAL			
15 SignallnPou2_3	位	VAR_GLOBAL			
16 SignallnPou2_4	位	VAR_GLOBAL			
17 timerInPou1_1	定时器	VAR_GLOBAL			
18 timerInPou1_2	定时器	VAR_GLOBAL			
19 timerInPou1_3	定时器	VAR_GLOBAL			
20 timerInPou1_4	定时器	VAR_GLOBAL			

操作步骤

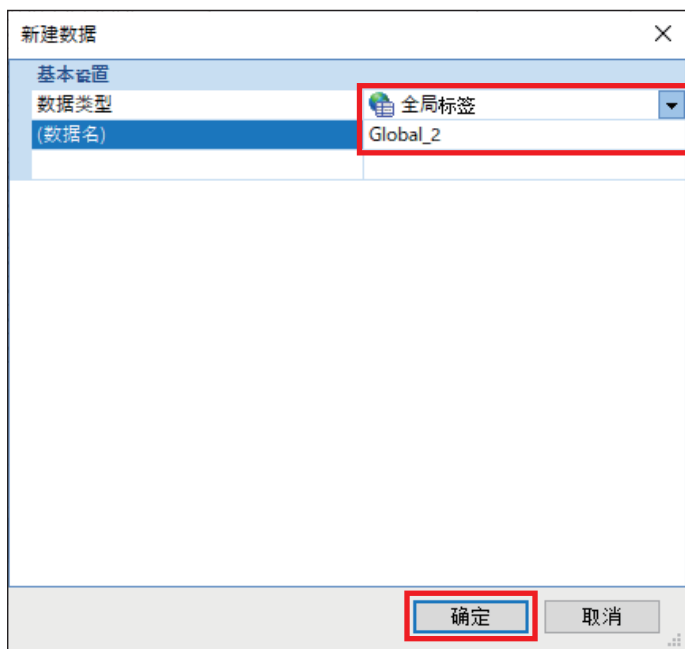
1. 新建一个全局标签作为划分的移动目标。

☞ 导航窗口⇒[标签]⇒右击[全局标签]⇒[新建数据]



2. 设置各项目，点击[确定]按钮。

- 数据类型：全局标签
- (数据名)：Global_2




3. 打开划分前的全局标签，选择并剪切要划分的全局标签(本示例中为No. 9~16)。点击编辑器的表的行编号，选择整行后剪切。



行号	标签名	数据类型	类	分配(软件/标签)	初始值	常数
1	CalcInPou1_1	字[有符号]	VAR_GLOBAL			
2	CalcInPou1_2	字[有符号]	VAR_GLOBAL			
3	CalcInPou1_3	字[有符号]	VAR_GLOBAL			
4	CalcInPou1_4	字[有符号]	VAR_GLOBAL			
5	SignallnPou1_1	位	VAR_GLOBAL			
6	SignallnPou1_2	位	VAR_GLOBAL			
7	SignallnPou1_3	位	VAR_GLOBAL			
8	SignallnPou1_4	位	VAR_GLOBAL			
9	CalcInPou2_1	字[有符号]	VAR_GLOBAL			
10	CalcInPou2_2	字[有符号]	VAR_GLOBAL			
11	CalcInPou2_3	字[有符号]	VAR_GLOBAL			
12	CalcInPou2_4	字[有符号]	VAR_GLOBAL			
13	SignallnPou2_1	位	VAR_GLOBAL			
14	SignallnPou2_2	位	VAR_GLOBAL			
15	SignallnPou2_3	位	VAR_GLOBAL			
16	SignallnPou2_4	位	VAR_GLOBAL			
17	timerInPou1_1	定时器	VAR_GLOBAL			
18	timerInPou1_2	定时器	VAR_GLOBAL			
19	timerInPou1_3	定时器	VAR_GLOBAL			
20	timerInPou1_4	定时器	VAR_GLOBAL			

- 要点**
- 如果未选择整行，则无法复制标签编辑器上隐藏的内容。
 - 建议将1个全局标签编辑器中的标签数最多划分5000个。

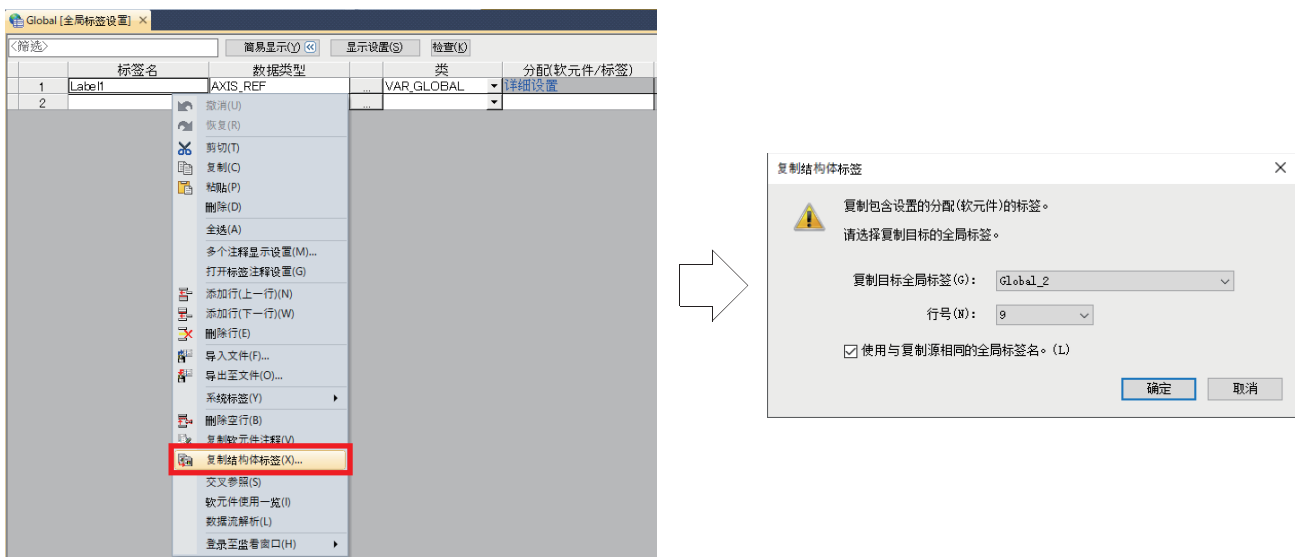
4. 打开在步骤1、2中创建的全局标签，粘贴剪切的全局标签。



行号	标签名	数据类型	类	分配(软件/标签)	初始值	常数
1	CalcInPou2_1	字[有符号]	VAR_GLOBAL			
2	CalcInPou2_2	字[有符号]	VAR_GLOBAL			
3	CalcInPou2_3	字[有符号]	VAR_GLOBAL			
4	CalcInPou2_4	字[有符号]	VAR_GLOBAL			
5	SignallnPou2_1	位	VAR_GLOBAL			
6	SignallnPou2_2	位	VAR_GLOBAL			
7	SignallnPou2_3	位	VAR_GLOBAL			
8	SignallnPou2_4	位	VAR_GLOBAL			

注意事项

- 如果移动至划分目标的全局标签的数据类型为结构体，则在该执行步骤中不会复制分配软件的内容。选择并右击相应的全局标签⇒选择[复制结构体标签]，复制标签并划分全局标签。(复制后，请删除划分前的全局标签。)



复制结构体标签

复制包含设置的分配(软件)的标签。
请选择复制目标的全局标签。

复制目标全局标签(G): Global_2

行号(N): 9

使用与复制源相同的全局标签名。(L)

确定 取消

3 加速程序转换

通过划分程序或使用FB进行部件化可将程序的转换范围局部化，从而加速程序转换。
本章介绍加速程序转换的技术。

技术一览

本章介绍的技术一览如下所示。

项目	内容	参照
划分程序文件	将程序文件、程序块划分为有意义的单元，并移动至不同的程序文件。	☞ 28页 划分程序文件
程序的FB化	将在多个位置类似的程序或子程序等以指针或有意义的单元分隔的程序FB化。	☞ 37页 程序的FB化
缩短包含控制语法的程序	将子程序和控制语法中的大型程序替换为FB。 或者，将程序移动至语法外。	☞ 42页 缩短包含控制语法的程序
更改全局标签(结构体)的设置	从结构体中删除不需要的软元件分配。 此外，应修改为不会将分配了软元件的结构体直接传递至FB的自变量，而是仅指定需要的结构体构建。	☞ 45页 更改全局标签(结构体)的设置
减少对数组同一元素的连续引用	如果同一数组元素被多次访问，则进行修改使其只被访问一次。	☞ 50页 减少对数组同一元素的连续引用

3.1 划分程序文件

概要

通过将程序文件、程序块划分为装置、流程和目的等有意义的单元，使程序结构化和标准化。这样可以使程序转换的对象局部化，从而缩短程序转换时间。

通过参数设置划分的程序文件的执行顺序，可以按照划分前的顺序操作程序。

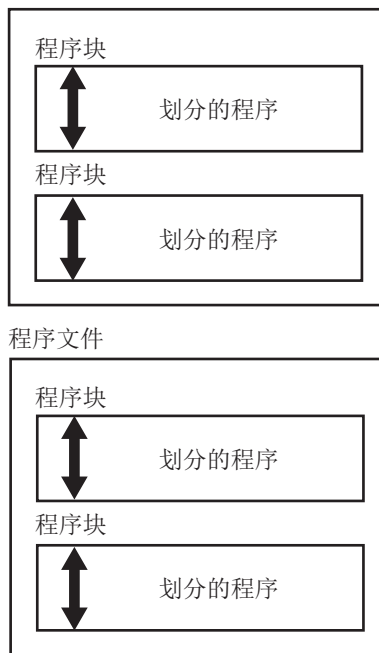
【技术应用前】

程序文件



【技术应用后】

程序文件



划分程序文件还有下述好处。

提高程序的可读性

明确了装置的处理和各流程单元的处理，可以清楚地看到程序的处理流程和结构。

提高程序的可移植性和再利用性

可以创建持有装置处理或各流程单元的程序本体和局部标签的程序文件，从而更容易移植和再利用处理或流程单元的程序。这样也提高了程序的开发效率。

程序步数和转换速度的关联

程序步数和转换速度的关联示例如下所示。
划分程序文件或程序块时，请以此为标准。

程序步数*1*2	转换速度(秒)*3*4		
	无划分 (1个程序文件中存在1个程序块)	在1个程序文件中将程序块分为4个	在2个程序文件中分别将程序块分为2个
1000	4.2	4.2	4.2
10000	11.2	10.2	9.2
100000	125.2	83.1	79.7

*1 这些值是简单程序中的测量值，该程序中只有与步数相等的触点和线圈。

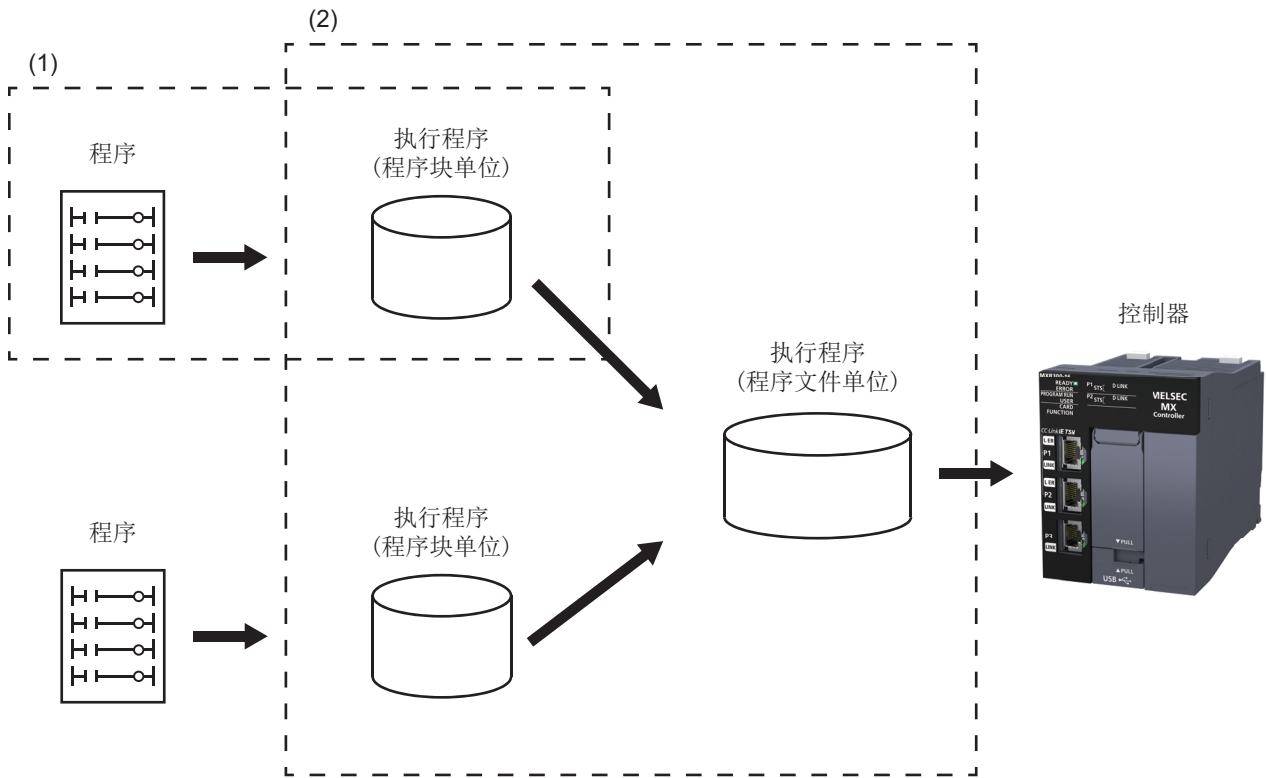
*2 转换速度会因工程构成或程序内容等各种原因变动。请注意，转换可能需要一些时间，特别是对于使用指针的程序。

*3 在搭载Intel® Core™ i7-10610U(1.80GHz)的计算机上的测量值。

*4 在执行全部转换后显示的对话框中选择“转换后创建执行程序”时的转换速度。

程序转换的流程和该技术的关联

转换程序的流程和该技术的关联如下所示。



(1) 根据记述的程序，生成以程序块为单位的执行程序

(2) 将以程序块为单位的执行程序合并为以程序文件为单位的程序

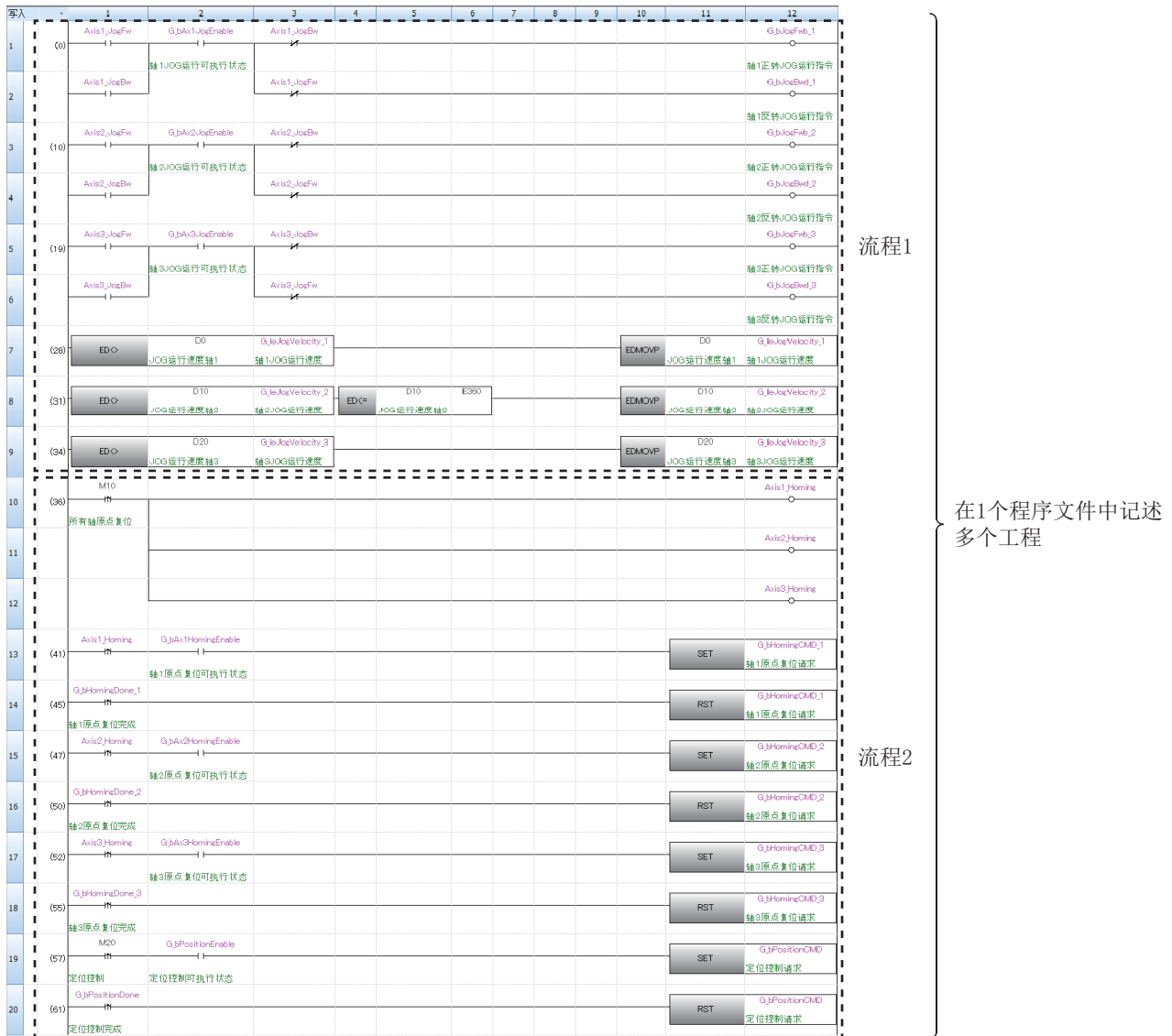
如果将程序块分为多个程序块，可以在流程(1)中并行进行执行程序的生成，从而缩短“全部转换•转换•转换+RUN中写入”的时间。

此外，如果将程序文件分为多个文件，在流程(2)中写入可编程控制器的文件本身的大小会更小，从而缩短仅转换更改部分的“转换•转换+RUN中写入”的时间。

执行步骤

以下述程序为例说明按照流程划分程序文件的步骤。

• 程序



• 局部标签

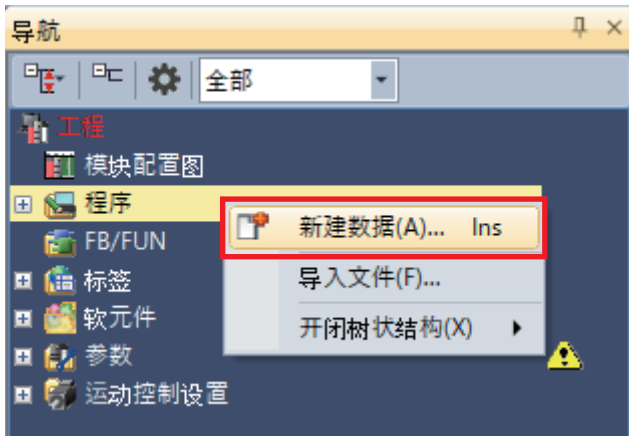
标签名	数据类型	类	初始值	常数
1 Axis1_JogFw	位	VAR		
2 Axis1_JogBw	位	VAR		
3 Axis2_JogFw	位	VAR		
4 Axis2_JogBw	位	VAR		
5 Axis3_JogFw	位	VAR		
6 Axis3_JogBw	位	VAR		
7 Axis1_Homing	位	VAR		
8 Axis2_Homing	位	VAR		
9 Axis3_Homing	位	VAR		

在流程1中使用 (rows 1-6)
 在流程2中使用 (rows 7-9)

操作步骤

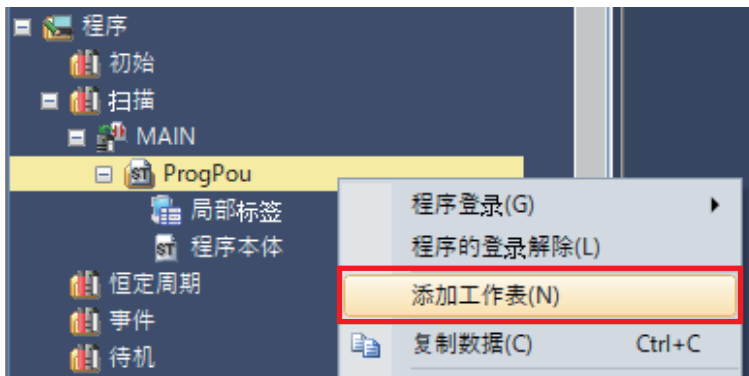
1. 新建一个程序文件作为划分的程序的移动目标。

☞ 导航窗口⇒右击[程序]⇒[新建数据]



要点

- 每个程序文件的程序量越少，程序转换时间就越短。因此，建议将划分的程序移动到新的程序文件中。(即使移动到已有的程序文件中，操作也不会受影响)
- ST程序或FBD程序也可以在同一POU中添加并划分新的工作表。这样可以提高程序的可读性/可移植性。(没有缩短程序转换时间的效果)



2. 设置各项目，点击[确定]按钮。

- 数据类型：程序文件
- (数据名)：原点复位
- 执行类型：扫描

新建数据

基本设置

数据类型 程序文件

(数据名) 原点复位

详细设置

程序文件

执行类型 扫描

确定 取消

3. 按照与步骤1相同的步骤打开“新建数据”画面，新建程序块。设置各项目，点击[确定]按钮。

新建数据

基本设置

数据类型 程序块

(数据名) 原点复位_轴1~3

详细设置

程序配置

程序语言 梯形图

程序文件

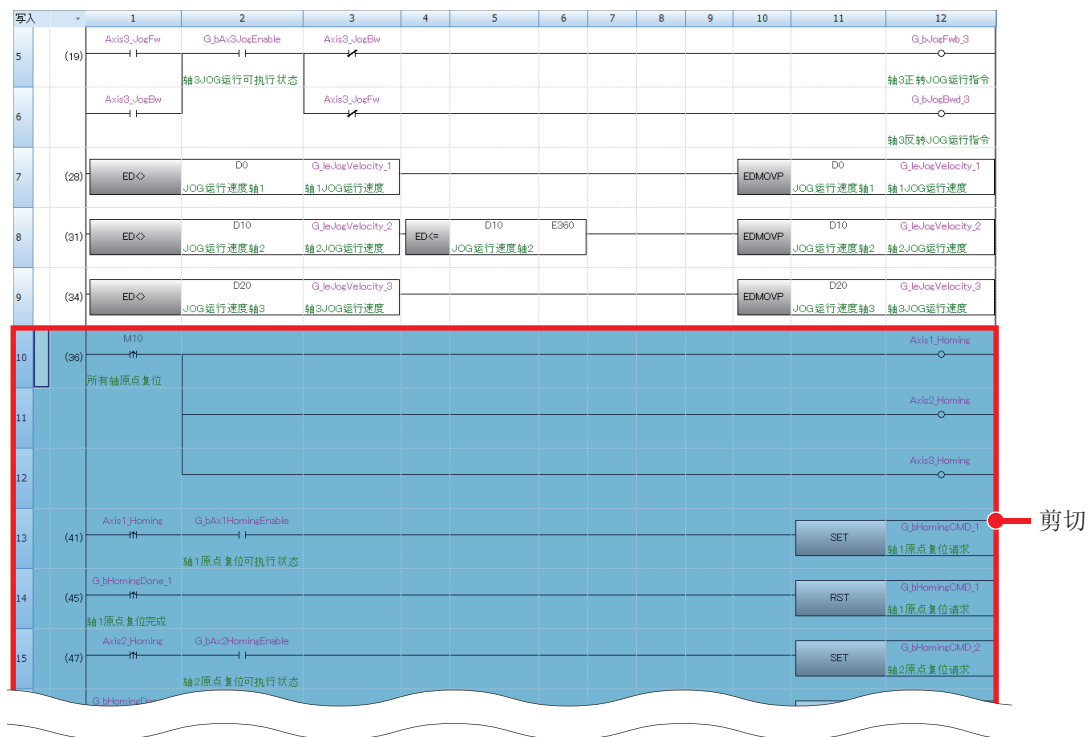
执行类型 扫描

添加目标的程序文件 原点复位

确定 取消

项目	设置内容	备注
数据类型	程序块	—
(数据名)	原点复位_轴1~3	—
程序语言	梯形图	—
执行类型	扫描	设置与步骤2中创建的程序文件相同的执行类型。
添加目标的程序文件	原点复位	设置在步骤2中创建的程序文件。

4. 打开要划分的程序文件的程序本体决定要划分的位置(示例中为流程2的起始), 然后选择并剪切后续的所有程序。



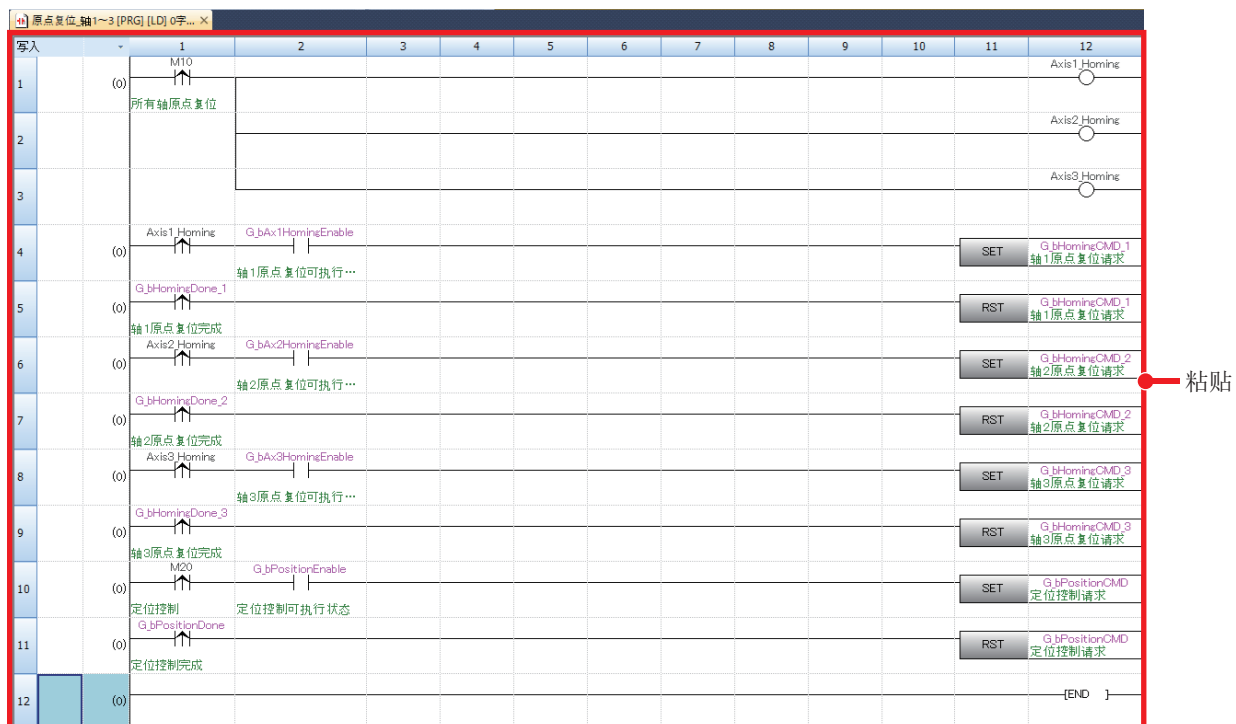
要点

划分程序, 使划分前的程序使用的局部标签和局部软元件只在划分后的任一程序中使用。如果这样划分较困难, 建议将程序FB化。

关于程序的FB化, 请参阅下述章节。

37页 程序的FB化

5. 打开在步骤3中创建的程序块的程序本体, 粘贴剪切的程序。



要点

如果有执行顺序比划分后的程序块还要靠后的程序块时, 则将相应的程序块移动到新的程序文件中。

6. 打开划分源程序文件的局部标签，剪切划分目标的程序中使用的局部标签(本示例中为No. 7~9)。点击编辑器的表的行编号，选择整行后剪切。



	标签名	数据类型		类	初始值	常数
1	Axis1_JbgFw	位	...	VAR		
2	Axis1_JbgBw	位	...	VAR		
3	Axis2_JbgFw	位	...	VAR		
4	Axis2_JbgBw	位	...	VAR		
5	Axis3_JbgFw	位	...	VAR		
6	Axis3_hgBw	位	...	VAR		
7	Axis1_Homing	位	...	VAR		
8	Axis2_Homing	位	...	VAR		
9	Axis3_Homing	位	...	VAR		

要点

如果未选择整行，则无法复制标签编辑器上隐藏的内容。

7. 打开在步骤3中创建的程序块的局部标签，粘贴剪切的局部标签。



	标签名	数据类型		类	初始值	常数
1	Axis1_Homing	位	...	VAR		
2	Axis2_Homing	位	...	VAR		
3	Axis3_Homing	位	...	VAR		

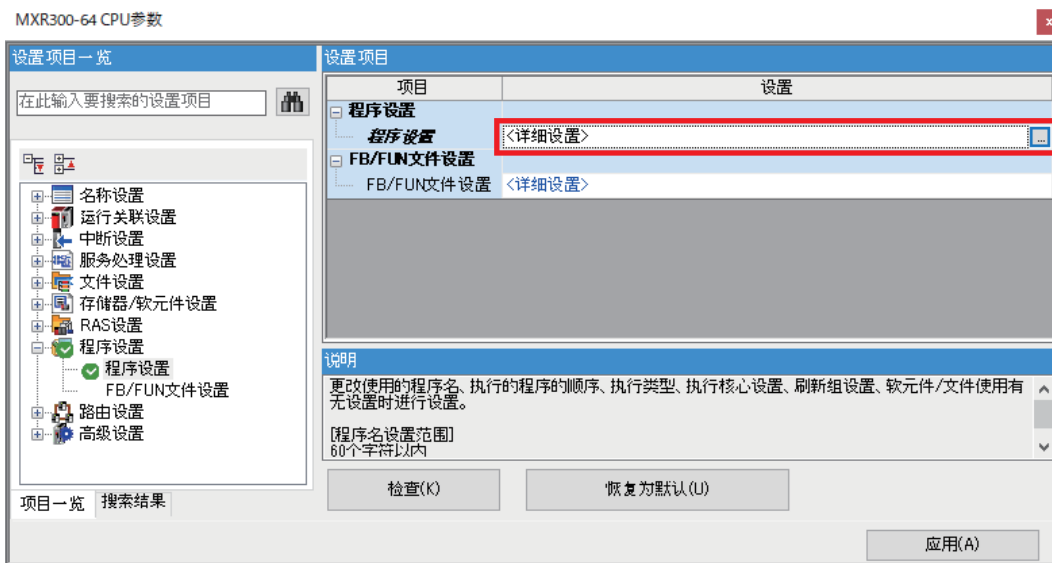
8. 打开CPU参数。

双击导航窗口⇒[参数]⇒机型名⇒[CPU参数]

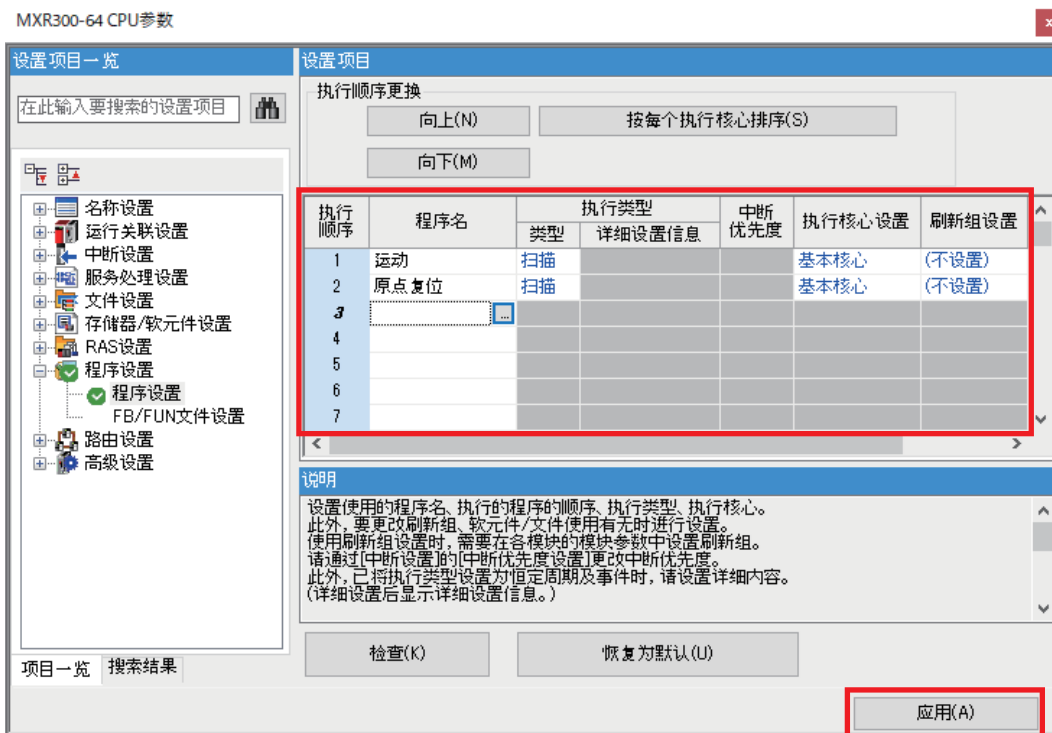


9. 要设置程序的执行顺序，请打开程序设置。


双击[程序设置]⇒[程序设置]的<详细设置>

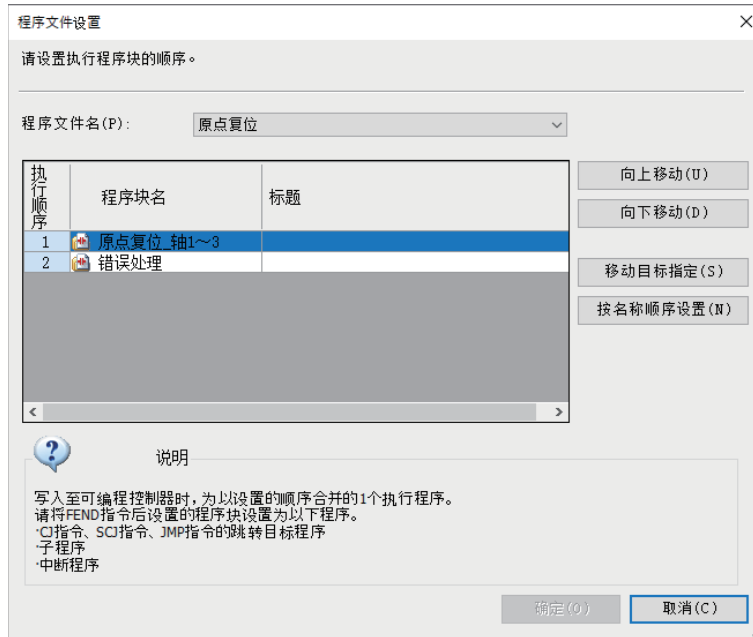


10. 按照执行顺序排列程序，点击[应用]按钮应用设置。



如果更改程序块的执行顺序，可以在“程序文件设置”画面中设置程序块的执行顺序。
关于详细内容，请参阅下述手册。

 GX Works3 操作手册



注意事项

如果划分程序文件导致使用指针的指令的跳转源与跳转目标在不同程序文件中，或配置子程序、控制语法等元素不存在同一个程序中，会发生转换出错。

3.2 程序的FB化

概要

将多个位置的类似程序或以指针分隔的子程序等，按照有意义的单元分隔的程序FB化，可将程序转换对象的范围局部化，从而降低程序转换时的影响。

此外，在使用指针的程序中，通过将指针替换为FB可将程序的转换对象结构化，从而缩短程序转换时间。

在梯形图程序中，可以使用现有梯形图创建FB。从FB化之前的程序的所在位置调用生成的FB，并将需要的软元件和标签作为自变量传送，即可实现与FB化之前相同的程序操作。

将程序FB化还有下述好处。

提高程序的可读性

通过将程序FB化，可使其成为仅输入/输出“箱”(FB)的简单程序，从而创建易读的程序。

提高程序的可移植性和再利用性

通过将标准的程序FB化，移植程序时无需复制程序和修改软元件。此外，如果将FB化的程序保存为库，则可以在各工程中使用通用的库，从而提高程序的再利用性。


提高程序质量

通过将程序部件化为FB并再利用，即使开发人员不同也可以使用相同的FB进行通用处理，从而可以开发出质量统一的程序，不受程序开发人员的技术水平影响。

要点

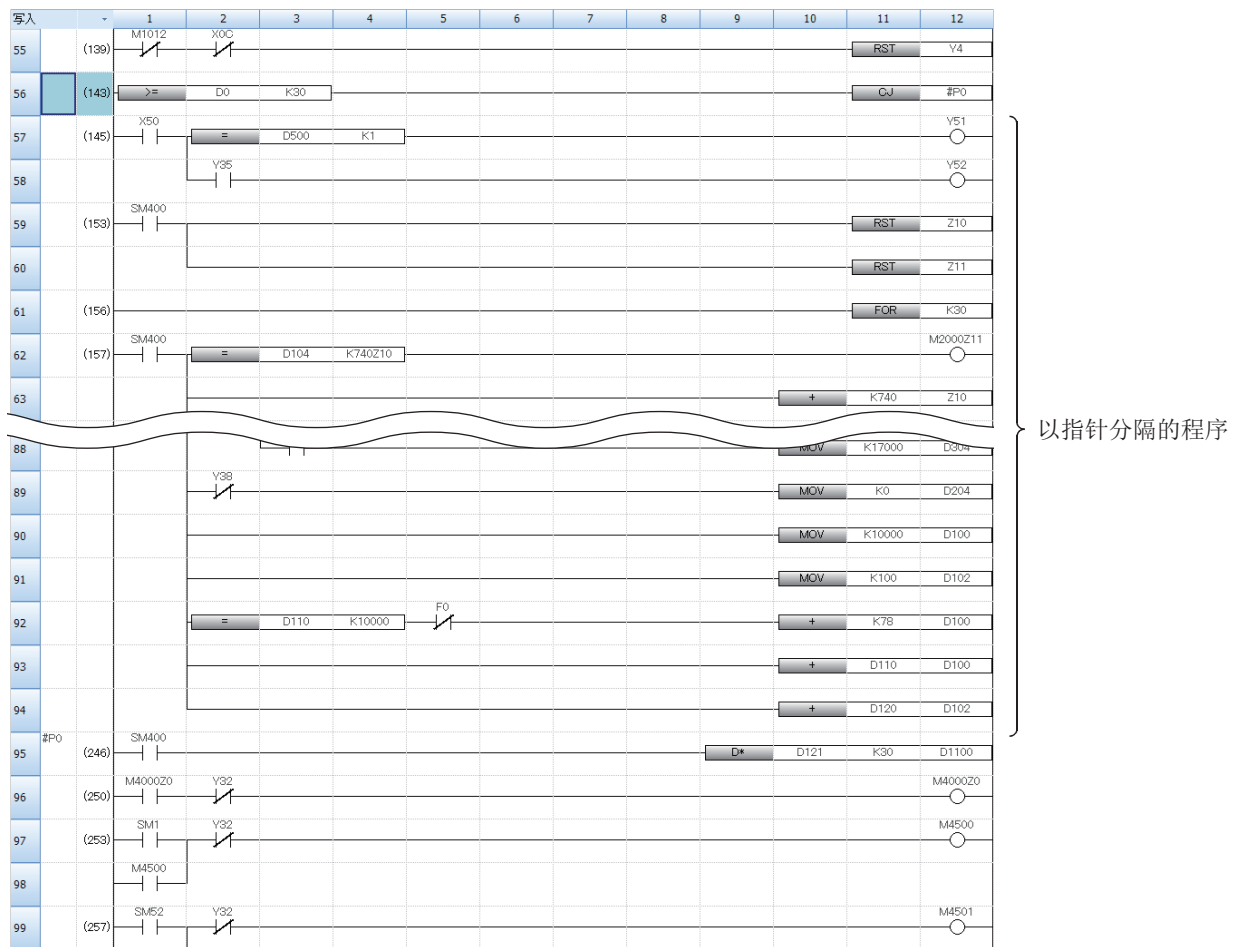
如果是梯形图程序以外的程序(ST、FBD/LD)，可以通过剪切要FB化的程序和局部标签并粘贴至新FB的方法创建FB。

关于新建FB的内容，请参阅下述手册。

 GX Works3 操作手册

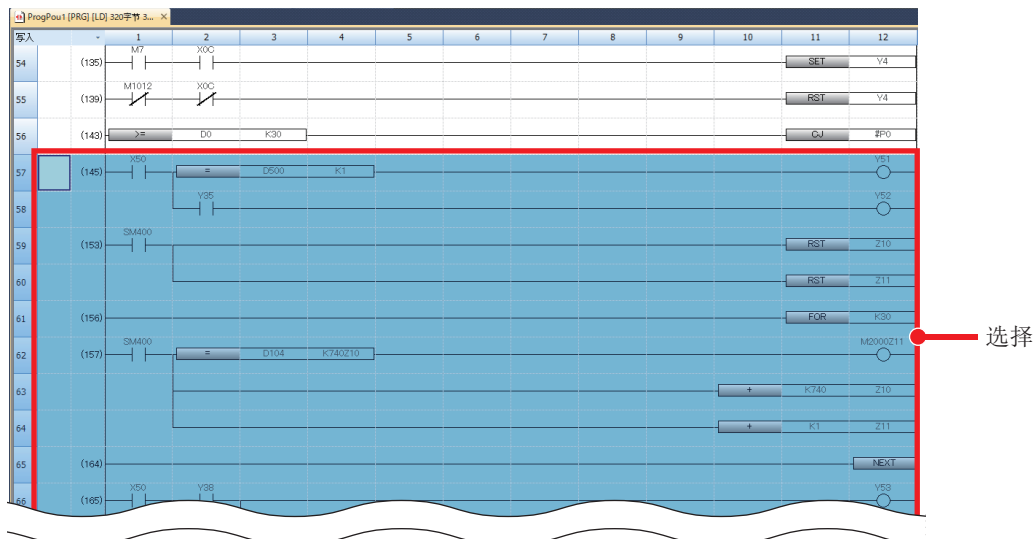
执行步骤

以下述梯形图程序为例说明以指针分隔的程序FB化的步骤。



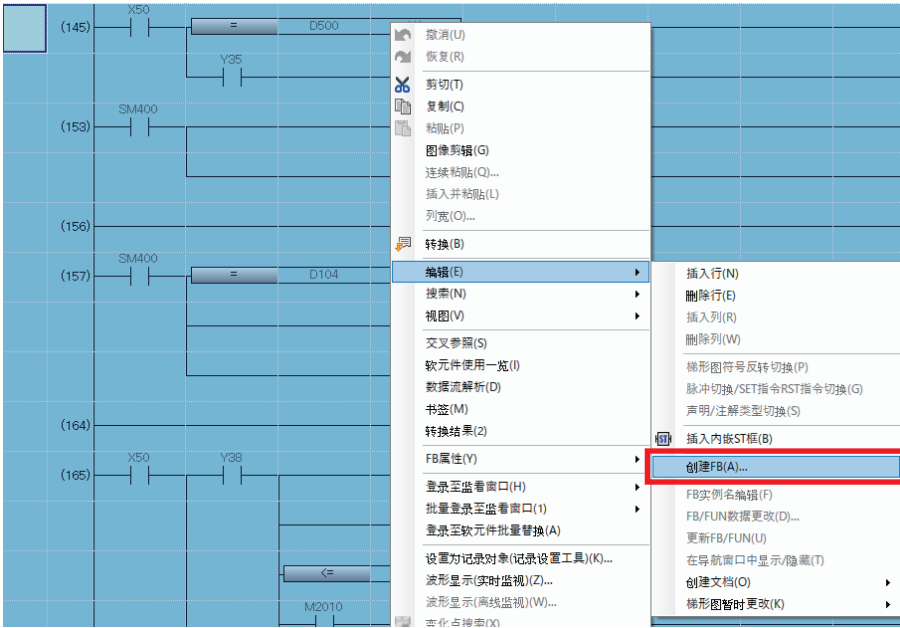
操作步骤

1. 打开要FB化的程序，然后选择要FB化的范围。



2. 从所选范围生成FB。

右击⇒[编辑]⇒[创建FB]

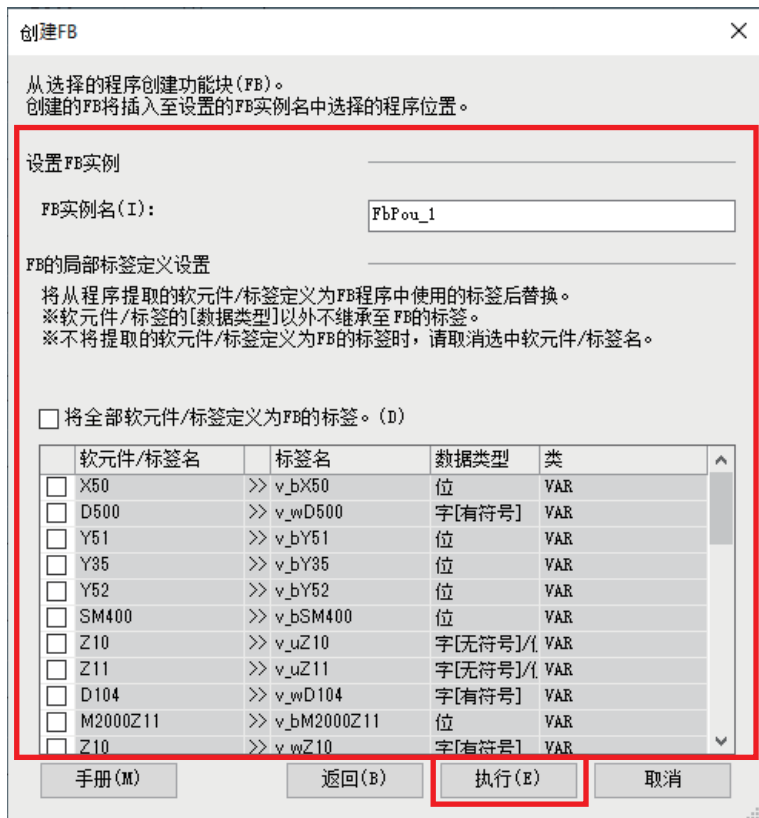


3. 根据程序内容设置要生成FB的各项目，点击[确定]按钮。

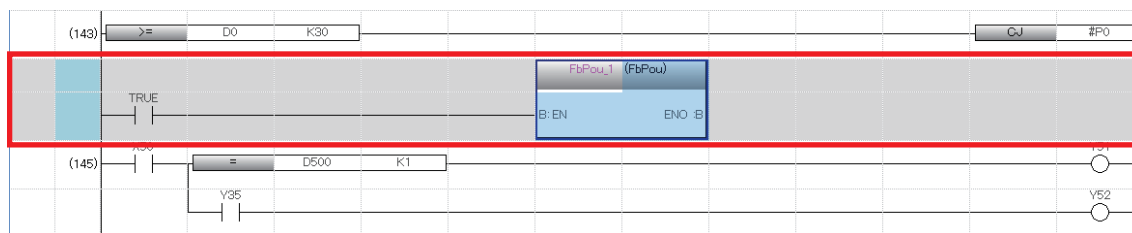
本示例中，将“使用EN/ENO”设置为“是”。



4. 设置FB的实例名和局部标签定义，点击[执行]按钮。
 在示例中，取消勾选“将全部软元件/标签定义为FB的标签。”。



在FB化范围的起始位置插入创建的FB。



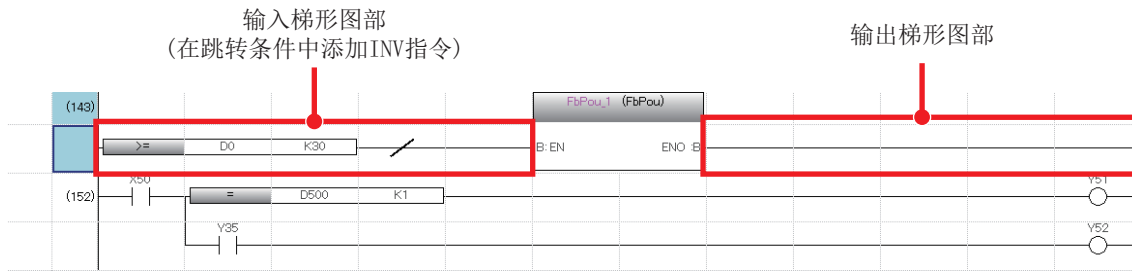
注意事项

设置FB的局部标签定义时，请注意以下事项。

- 局部软元件不能设置为FB的自变量。请在“FB的局部标签定义设置”中取消勾选软元件/软元件名。
 - 如果在FB程序内外使用相同的局部软元件/局部标签，请将局部软元件/局部标签定义为FB的标签(自变量)。
- 关于FB的局部标签定义设置的详细内容，请参阅下述手册。

GX Works3 操作手册

5. 根据FB的输入标签和输出标签创建输入/输出梯形图部分，修改为与FB化前相同的程序。
该示例中，在指针的跳转条件中添加了INV指令并修改为FB的执行条件。



注意事项

- 如果程序的FB化导致使用指针的指令的跳转源和跳转目标不存在于同一个程序中，或配置子程序、控制语法等元素不存在于同一个程序中，会发生转换出错。
- 如果通过FB的自变量传递的数据量变多，可能会延长扫描时间。

3.3 缩短包含控制语法的程序

概要

在梯形图程序中的FOR~NEXT指令及子程序、ST程序中的IF语句及FOR语句等控制语法中有大型程序时，可以通过将程序FB化或将不需要在控制语法中记载的程序移动至语法外来缩短程序。这样可以减少程序量，从而缩短程序转换时间。

每种程序可缩短的控制语法如下所示。

类型	控制语法
梯形图程序	<ul style="list-style-type: none">• FOR~NEXT指令• 子程序等，由指针分隔的梯形图块组
ST程序	<ul style="list-style-type: none">• IF语句• CASE语句• FOR语句• WHILE语句• REPEAT语句
FBD/LD程序	<ul style="list-style-type: none">• FOR~NEXT指令• 以跳转标签分隔的梯形图组

缩短程序还有下述好处。

提高程序的可读性

程序量的减少可以提高程序的可读性。

要点

关于程序FB化的概要和好处，请参阅下述章节。

☞ 37页 概要

控制语法中的程序步数和转换速度的关联

控制语法中的程序步数和转换速度的关联示例如下所示。

移动至程序语法外时，请以此为标准。

程序步数*1*2	转换速度(秒)*3*4	
	控制语法中的程序*5	控制语法外的程序
1000	5.6	4.2
10000	19.7	11.2
100000	447.5	125.2

*1 这些值是简单程序中的测量值，该程序中只有与步数相等的触点和线圈。

*2 转换速度会因工程构成或程序内容等各种原因变动。请注意，转换可能需要一些时间，特别是对于使用指针的程序。

*3 在搭载Intel® Core™ i7-10610U(1.80GHz)的计算机上的测量值。

*4 在执行全部转换后显示的对话框中选择“转换后创建执行程序”时的转换速度。

*5 这些值是程序中的测量值，该程序是在FOR~NEXT指令中记述的。

执行步骤

缩短程序的步骤如下所示。

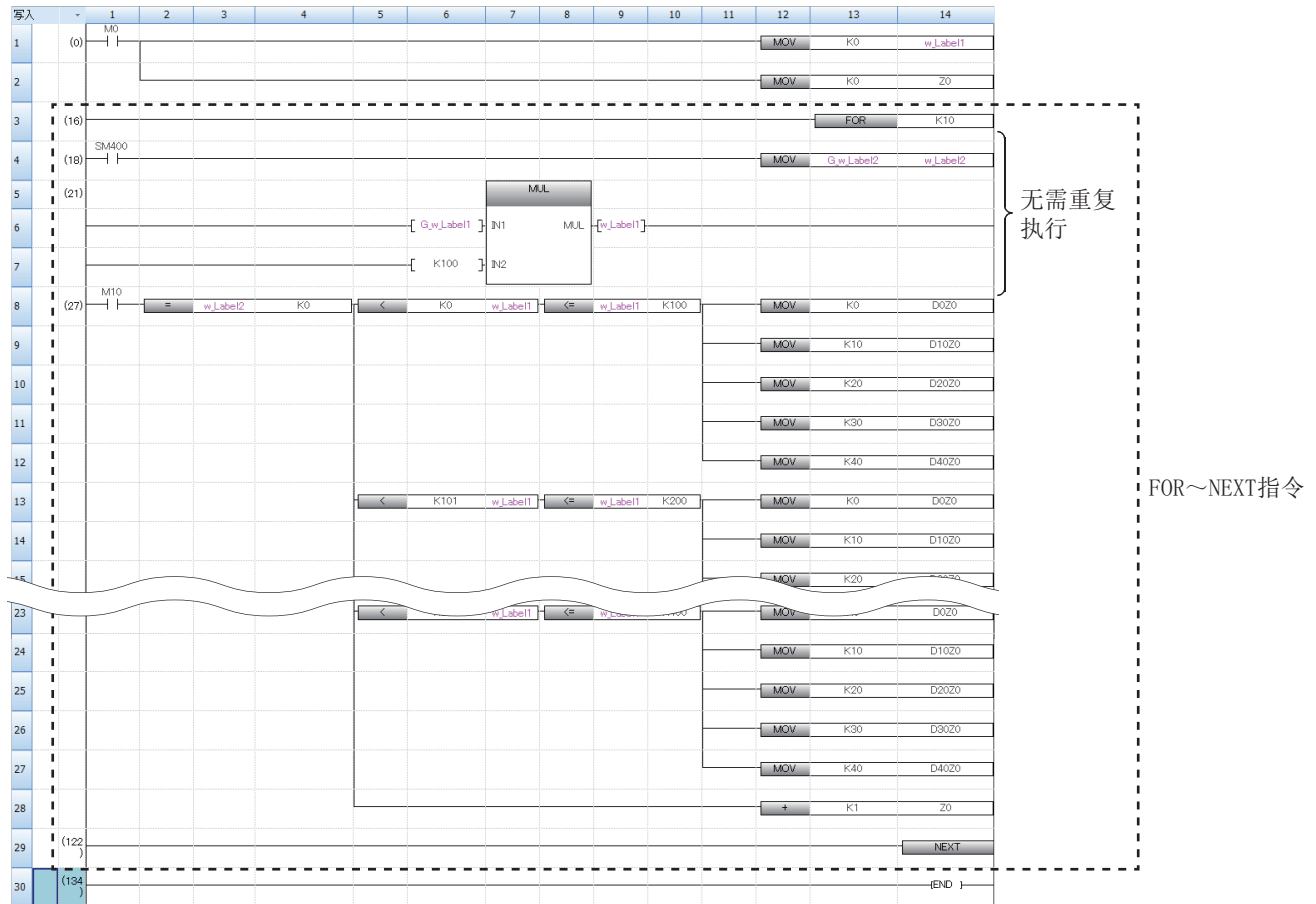
将控制语法中的程序FB化

关于将控制语法中的程序FB化，请参阅下述步骤。

☞ 38页 执行步骤

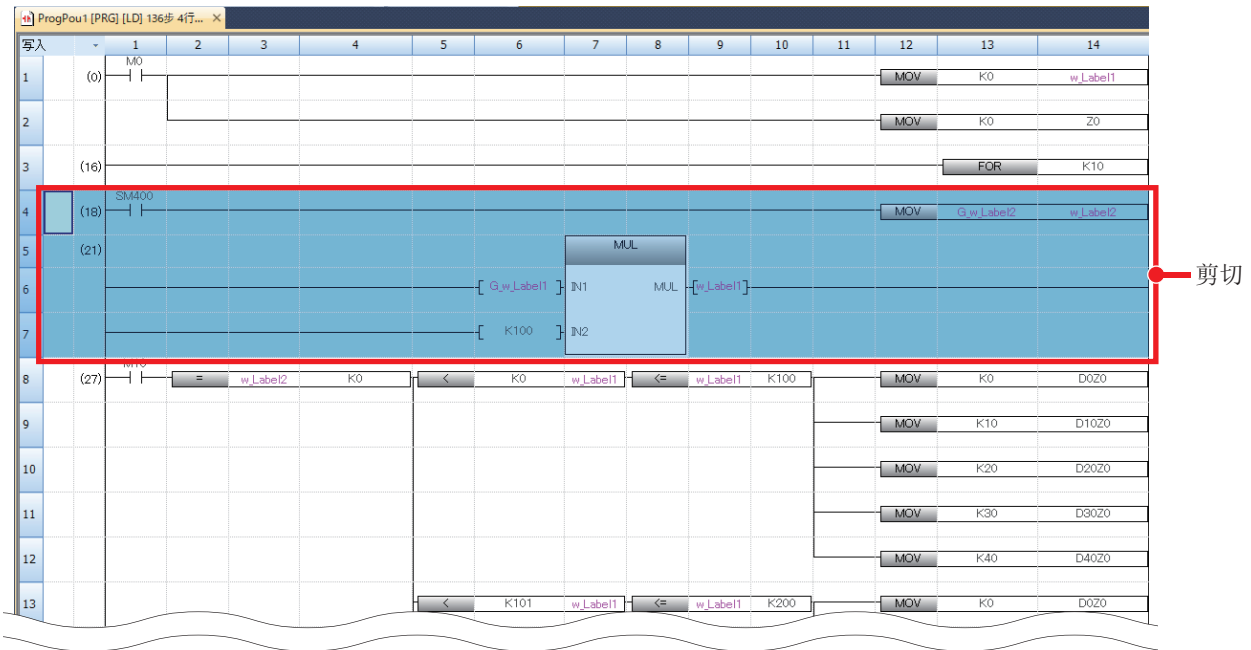
移动控制语法中不需要的程序

以下述程序为例说明将在FOR~NEXT指令的程序中不需要重复执行的程序移动到FOR~NEXT指令外的步骤。

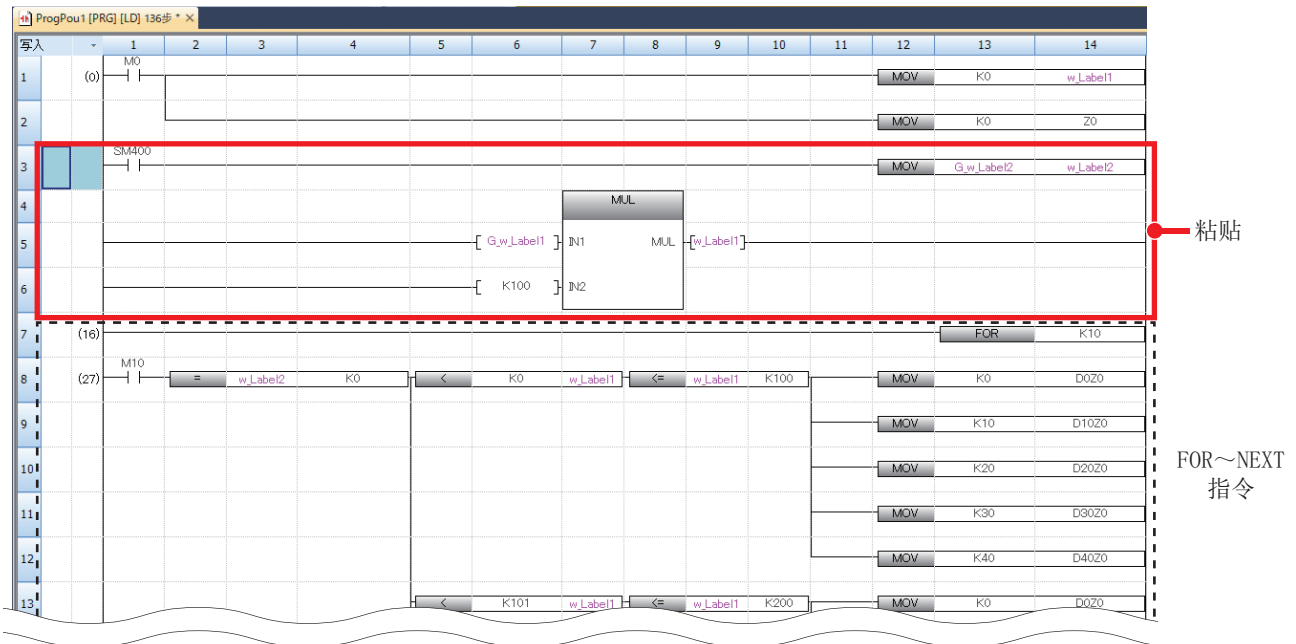


操作步骤

1. 打开程序，选择并剪切FOR~NEXT指令中不需要重复执行的程序。



2. 将剪切的程序粘贴到控制语法外。按照执行顺序粘贴到正确位置。



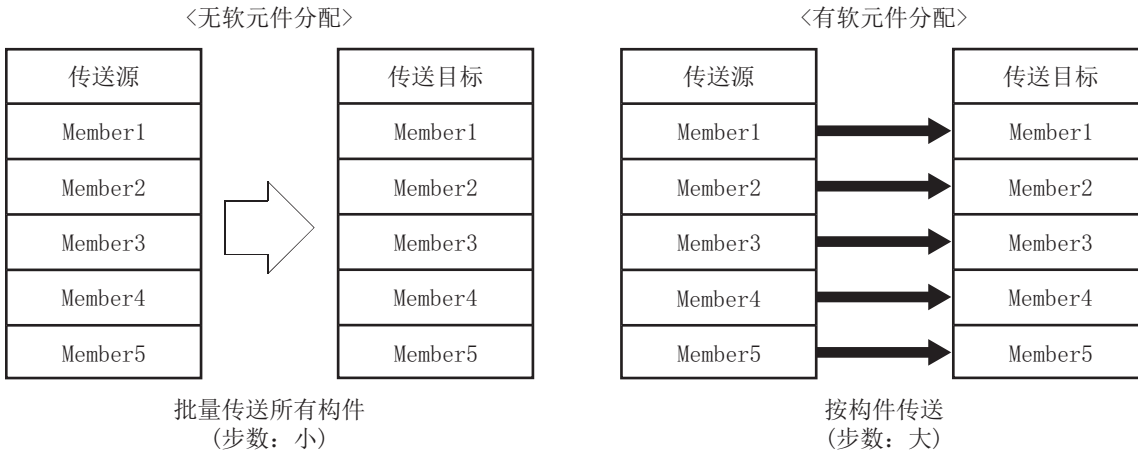
3.4 更改全局标签(结构体)的设置

概要

在以下自变量或代入语句中使用分配了软元件的结构体时，将会为构造体的每个构件传送给，因此内部会生成许多传送指令。

- 子程序型FB的自变量
- MOVE指令的自变量
- ST程序或FBD程序的代入语句

将无软元件分配的结构体和有软元件分配的结构体用作MOVE指令的自变量时的传送示例如下所示。



通过删除结构体的实例不需要的软元件分配，可以控制传送指令的生成。此外，子程序型FB的自变量只需将需要的结构体构件作为自变量即可控制传送指令的生成。减少在内部生成的指令可以缩短程序转换时间。

修改结构体还有下述好处。

缩小程序大小

修改结构体可以控制内部传送指令的生成，从而缩小程序大小。

执行步骤

以下述程序为例说明删除分配给结构体“Struct1”的实例“StrInst_WithDeviceAssign”的软元件的步骤。此外，还说明修改使用输入自变量为“StrInst_WithDeviceAssign”的子程序型FB“FBPou_1”的程序的步骤。

- 结构体“Struct1”

	标签名	数据类型	类	初始值	常数
1	member_st1	Struct2	...		
2	member2	字[有符号](0.15)	...		
3	member3	字符串(32)(0.15)	...		
4	member4	位(0.15)	...		

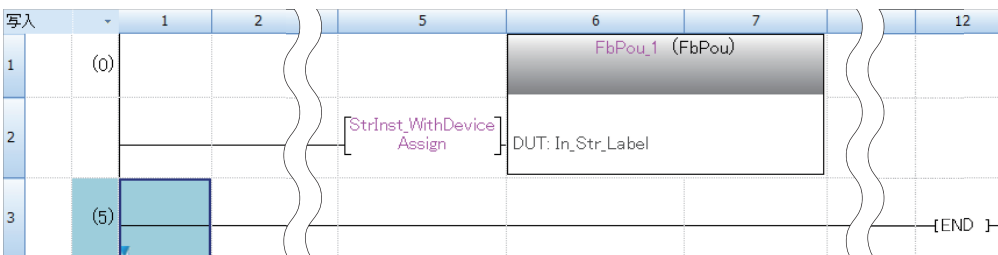
- 结构体实例“StrInst_WithDeviceAssign”（全局标签）

	标签名	数据类型	类	分配(软元件/标签)	初始值	常数
1	StrInst_WithDeviceAssign	Struct1	VAR_GLOBAL	详细设置		
2						

标签名	数据类型	软元件
member_st1	Struct2	
member2	字[有符号](0.15)	
member3	字符串(32)(0.15)	
member4	位(0.15)	D677.0

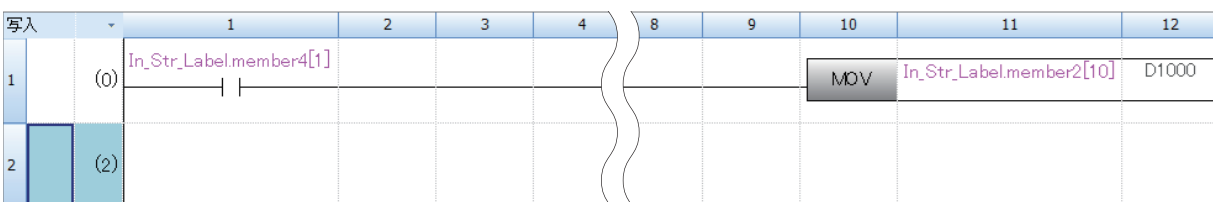
将软元件 (D677.0) 分配给构件“member4”

- 使用子程序型FB“FBPou_1”的程序



在FB的输入自变量中指定结构体的实例

- 子程序型FB“FBPou_1”的程序



- 子程序型FB“FBPou_1”的局部标签

	标签名	数据类型	类	初始值	常数
1	In_Str_Label	Struct1	VAR_INPUT		
2					

删除结构体实例不需要的软元件分配

删除结构体实例“StrInst_WithDeviceAssign”不需要的软元件分配的步骤如下所示。

操作步骤

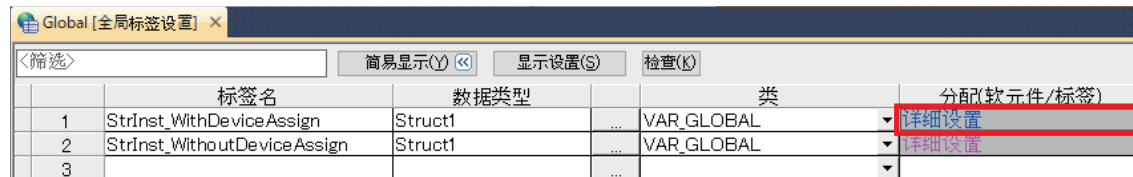
1. 打开登录了结构体实例“StrInst_WithDeviceAssign”的全局标签，点击[显示设置]按钮。



标签名	数据类型	类	分配软元件/标签	初始值	常数
1 StrInst_WithDeviceAssign	Struct1	VAR_GLOBAL	详细设置		
2					


要点

分配了软元件的结构体实例，“分配(软元件/标签)”列的“详细设置”会显示为蓝色。



标签名	数据类型	类	分配软元件/标签
1 StrInst_WithDeviceAssign	Struct1	VAR_GLOBAL	详细设置
2 StrInst_WithoutDeviceAssign	Struct1	VAR_GLOBAL	详细设置
3			

2. [扩展显示设置]⇒将[显示区域]更改为“始终显示”，点击[确定]按钮。



显示区域	显示/隐藏
列项目	
标签名	显示
数据类型	显示
类	显示
分配(软元件/标签)	显示
初始值	显示
常数	显示
系统标签的关联	显示
系统标签名	显示
属性	显示
备注	显示
外部设备的访问	显示
多个注释	...
扩展显示设置	
显示区域	始终显示
系统标签设置	
显示区域	隐藏

确定 取消

- 选择分配了软元件的结构体的实例，从扩展显示的“软元件”列中删除member4的分配软元件“D677.0”。

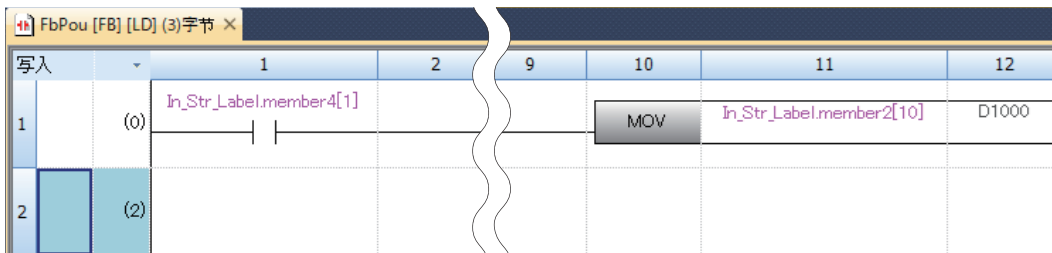


修改子程序型FB的自变量

将子程序型FB“FBPou_1”的输入自变量“StrInst_WithDeviceAssign”修改为仅限需要的结构体构件的步骤如下所示。

操作步骤

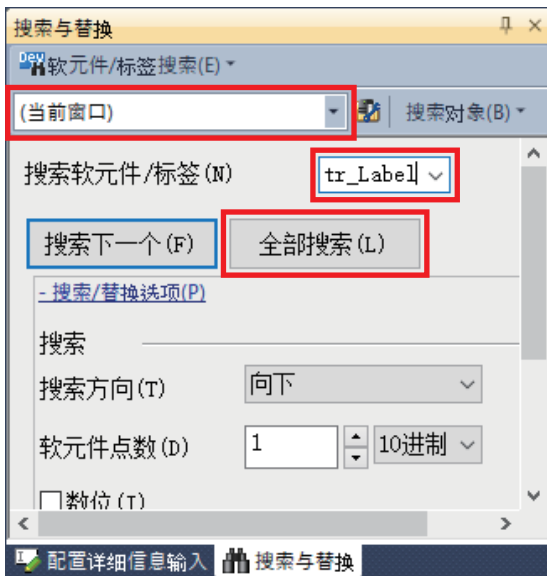
- 打开子程序型FB“FBPou_1”的程序。



- 搜索在FB“FBPou_1”中使用的结构体实例“In_Str_Label”。

在搜索与替换窗口如下设置，点击[全部搜索]按钮。

- 搜索范围：(当前窗口)
- 搜索软元件/标签：In_Str_Label



3. 将在搜索结果窗口显示使用结构体实例“`In_Str_Label`”的对象一览。

对象一览	路径	位置
<code>In_Str_Label.member4[1]</code>	3.2\FB\FUN\FBFILE\FbPou\程序本体	0步
<code>In_Str_Label.member2[10]</code>	3.2\FB\FUN\FBFILE\FbPou\程序本体	1步

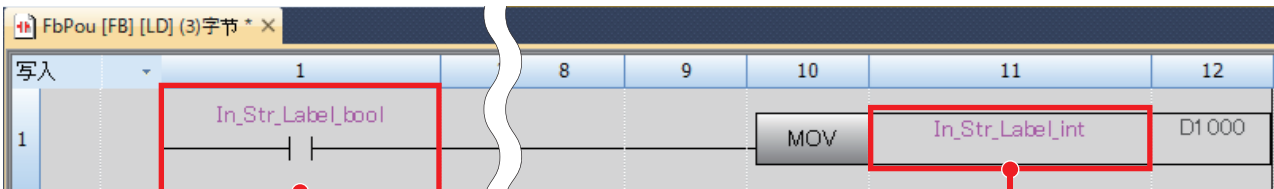
4. 打开FB“`FBPou_1`”的局部标签，创建以下标签代替在步骤3中搜索的结构体构件。

数据类型和类与结构体构件和结构体实例匹配。

- 代替`In_Str_Label.member4[1]`的标签：`In_Str_Label_bool` (数据类型：位，类：`VAR_INPUT`)
- 代替`In_Str_Label.member2[10]`的标签：`In_Str_Label_int` (数据类型：字[有符号]，类：`VAR_INPUT`)

标签名	数据类型	类	初始值	常数
1 <code>In_Str_Label</code>	Struct1	...	<code>VAR_INPUT</code>	
2 <code>In_Str_Label_bool</code>	位	...	<code>VAR_INPUT</code>	
3 <code>In_Str_Label_int</code>	字[有符号]	...	<code>VAR_INPUT</code>	

5. 打开FB“`FBPou_1`”的程序，将在步骤3中搜索的结构体构件替换为在步骤4中创建的标签。

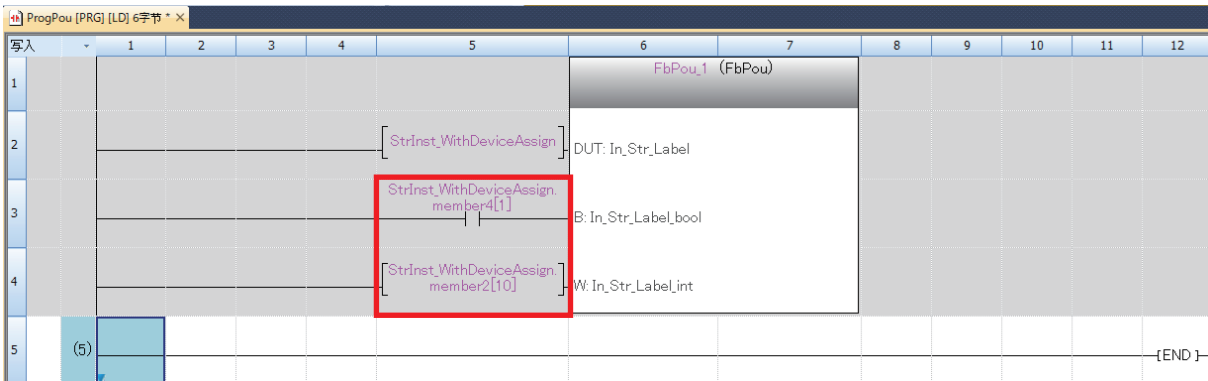


将`In_Str_Label.member4[1]`
替换为`In_Str_Label_bool`

将`In_Str_Label.member2[10]`
替换为`In_Str_Label_int`

6. 打开使用FB“`FBPou_1`”的程序，为各自变量设置结构体实例“`StrInst_WithDeviceAssign`”的构件。

将自变量“`In_Str_Label_bool`”设置为“`StrInst_WithDeviceAssign.member4[1]`”，将自变量“`In_Str_Label_int`”设置为“`StrInst_WithDeviceAssign.member2[10]`”。



7. 打开FB“`FBPou_1`”的局部标签，删除不需要的结构体实例“`In_Str_Label`”。

标签名	数据类型	类	初始值	常数
1 <code>In_Str_Label</code>	Struct1	...	<code>VAR_INPUT</code>	
2 <code>In_Str_Label_bool</code>	位	...	<code>VAR_INPUT</code>	
3 <code>In_Str_Label_int</code>	字[有符号]	...	<code>VAR_INPUT</code>	

删除

3.5 减少对数组同一元素的连续引用

概要

如果动态指定数组元素(指定软元件或标签作为数组的下标),则执行程序时会生成多个指令来访问存储器。因此,如果有多个数组访问同一元素,将其复制到其他存储器中一次就可以减少执行程序时访问存储器的指令输出次数。这样可以减少内部生成的指令,从而缩短程序转换时间和扫描时间。

执行步骤

以下述程序为例说明减少对同一数组元素的连续引用的步骤。

- 程序

```
st ProgPou1 [PRG] [ST] ×
1 IF (targetStLabel[D1, D2] > 5) THEN
2   destLabel1 := targetStLabel[D1, D2] ;
3   destLabel2 := targetStLabel[D1, D2] + K10 ;
4   destLabel3 := targetStLabel[D1, D2] * K2 ;
5   destLabel4 := targetStLabel[D1, D2] + targetStLabel[D10, D20];
6 // ELSE
7 // ?执行语句? ;
8 END_IF;
9
```

} 多次访问targetStLabel[D1, D2]

- 局部标签

标签名	数据类型	类	初始值	常数
1 targetStLabel	字[有符号](0.2,0.2)	VAR		
2 destLabel1	字[有符号]	VAR		
3 destLabel2	字[有符号]	VAR		
4 destLabel3	字[有符号]	VAR		
5 destLabel4	字[有符号]	VAR		

操作步骤

1. 打开相应的程序,确定替换对象(连续引用同一数组元素的位置)。

该示例中,将在程序中使用了5次的“targetStLabel[D1, D2]”作为替换对象。

```
st ProgPou1 [PRG] [ST] ×
1 IF (targetStLabel[D1, D2] > 5) THEN
2   destLabel1 := targetStLabel[D1, D2] ;
3   destLabel2 := targetStLabel[D1, D2] + K10 ;
4   destLabel3 := targetStLabel[D1, D2] * K2 ;
5   destLabel4 := targetStLabel[D1, D2] + targetStLabel[D10, D20];
6 // ELSE
7 // ?执行语句? ;
8 END_IF;
9
```

2. 打开相应程序的局部标签，添加局部标签“tempLabel”代替访问数组元素的标签。

- 标签名: tempLabel
- 数据类型: 字[有符号]
- 类: VAR

标签名	数据类型	类	初始值	常数
1 targetStLabel	字[有符号](0.2,0.2)	VAR		
2 destLabel1	字[有符号]	VAR		
3 destLabel2	字[有符号]	VAR		
4 destLabel3	字[有符号]	VAR		
5 destLabel4	字[有符号]	VAR		
6 tempLabel	字[有符号]	VAR		

3. 打开相应的程序，在程序的第1行添加语句，将“targetStLabel[D1, D2]”保存到在步骤2中创建的局部标签“tempLabel”中。

- 保存语句: tempLabel := targetStLabel[D1, D2];

```

1 tempLabel := targetStLabel[D1, D2];
2 IF (targetStLabel[D1, D2] > 5) THEN
3     destLabel1 := targetStLabel[D1, D2] ;
4     destLabel2 := targetStLabel[D1, D2] + K10 ;
5     destLabel3 := targetStLabel[D1, D2] * K2 ;
6     destLabel4 := targetStLabel[D1, D2] + targetStLabel[D10, D20];
7     // ELSE
8     // ?执行语句? ;
9 END_IF;

```

4. 将“targetStLabel[D1, D2]”替换为在步骤2中创建的局部标签“tempLabel”。

在搜索与替换窗口如下设置，点击[替换]按钮。（重复执行直到替换所有相应的标签）

- 搜索/替换方法: 字符串替换
- 搜索范围: (当前窗口)
- 搜索字符串: targetStLabel[D1, D2]
- 替换字符串: tempLabel

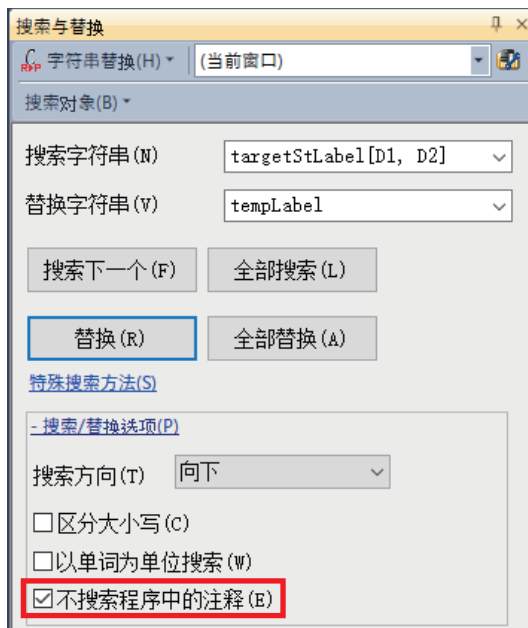


替换完成后将变为如下所示的程序。

```
st ProgPou1 [PRG] [ST] 13字节 * x
1 tempLabel := targetStLabel[D1, D2];
2 IF tempLabel > 5) THEN
3     destLabel1 := tempLabel ;
4     destLabel2 := tempLabel + K10 ;
5     destLabel3 := tempLabel * K2 ;
6     destLabel4 := tempLabel + targetStLabel[D10, D20];
7 // ELSE
8 // ?执行语句? ;
9 END_IF;
```

要点

- 可以通过[全部替换]按钮批量替换，但建议逐一确认并替换。
- 如果不替换程序中的注释，请在搜索/替换选项中勾选“不搜索程序中的注释”。



注意事项

- 在数组的下标不是常数时，该技术有效。
- 根据程序的内容可能无法替换访问数组元素的标签，例如动态指定数组元素使用的软元件或标签在程序运行期间被改写等。请确认程序的内容并替换。

“targetStLabel[D0, D1]”中D1的值在程序运行期间被改写

```
st ProgPou2 [PRG] [ST] * x
1 IF (targetStLabel[D1, D2] > 5) THEN
2     destLabel1 := targetStLabel[D1, D2] ;
3     destLabel2 := targetStLabel[D1, D2] + K10 ;
4     D1 := 0 ;
5     destLabel3 := targetStLabel[D1, D2] * K2 ;
6     destLabel4 := targetStLabel[D1, D2] + targetStLabel[D10, D20];
7 // ELSE
8 // ?执行语句? ;
9 END_IF;
```

4 缩短扫描时间

通过记述程序和设置参数缩短扫描时间。通过缩短扫描时间，可以缩短程序的处理时间。
本章介绍缩短扫描时间的技术。

技术一览

本章介绍的技术一览如下所示。

项目	内容	参照
BMOV指令替换为多个指令	将BMOV指令替换为使用FOR~NEXT指令的多个MOV指令。	☞ 54页 BMOV指令替换为多个指令
减少上升沿执行指令	减少使用上升沿执行指令(仅在OFF→ON上升沿时执行的指令)。	☞ 59页 减少上升沿执行指令
减少模块访问指令	将直接访问智能模块的缓冲存储器的位置更改为智能模块的自动刷新或链接刷新。	☞ 61页 减少模块访问指令
应用初始执行类型程序	在ST程序中，将只执行1次的处理移动到初始执行类型程序中。	☞ 64页 应用初始执行类型程序
省略自变量和直接引用	在ST程序中，省略或删除对运动控制FB的输入输出自变量的代入处理和不需要分配。	☞ 70页 省略自变量和直接引用
划分含有多条指令的梯形图块	在梯形图程序中，将1个梯形图块中含有的指令数减少到100条指令或以下。	☞ 74页 划分含有多条指令的梯形图块
应用标签初始值	使用标签初始值为不更改值的标签和FB设置初始值。	☞ 77页 应用标签初始值

扫描时间的确认方法

扫描时间值可以通过下述特殊寄存器确认。

如果要确认执行各技术时的扫描时间，请使用下述设备。

编号	名称	内容	详细内容	设置
SD518	初始扫描时间	初始扫描时间(ms单位)	初始扫描时间存储到SD518、SD519中。(计测以1μs单位进行) SD518: 存储ms的位(存储范围: 0~65535) SD519: 存储μs的位(存储范围: 0~999) STOP→RUN时将被清零。	每次进行END处理时在系统侧设置
SD519		初始扫描时间(μs单位)		
SD520	当前扫描时间	当前扫描时间(ms单位)	当前扫描时间存储到SD520、SD521中。(计测以1μs单位进行) SD520: 存储ms的位(存储范围: 0~65535) SD521: 存储μs的位(存储范围: 0~999) STOP时将被清零。	
SD521		当前扫描时间(μs单位)		
SD522	最短扫描时间	最短扫描时间(ms单位)	去除初始执行程序的扫描时间后的扫描时间的最小值将被存储到SD522、SD523中。(计测以1μs单位进行) SD522: 存储ms的位(存储范围: 0~65535) SD523: 存储μs的位(存储范围: 0~999) STOP→RUN时将被清零。	
SD523		最短扫描时间(μs单位)		
SD524	最长扫描时间	最长扫描时间(ms单位)	去除初始执行程序的扫描时间后的扫描时间的最大值将被存储到SD524、SD525中。(计测以1μs单位进行) SD524: 存储ms的位(存储范围: 0~65535) SD525: 存储μs的位(存储范围: 0~999) STOP→RUN时将被清零。	
SD525		最长扫描时间(μs单位)		

此外，梯形图程序可以使用扫描时间测定功能。

关于扫描时间测定功能的详细内容，请参阅下述手册。

📖 GX Works3 操作手册

4.1 BMOV指令替换为多个指令

概要

由于MOV指令的执行速度比BMOV指令块，将BMOV指令替换为MOV指令可以缩短扫描时间。
替换BMOV指令有两种方法，替换为多个MOV指令方法或通过FOR~NEXT指令多次执行MOV指令。

MOV指令和BMOV指令的性能值

MXR500-256中MOV指令(INC指令)和BMOV指令的性能值的示例如下所示。
将BMOV指令替换为MOV指令时，请以此为标准。

指令	执行速度(μ秒)
MOV指令	0.00050
INC指令	0.00361
BMOV指令(传送数1点时)	0.076
BMOV指令(传送数1024点时)	0.191

注意事项

性能值根据机型有所不同。

执行步骤

以下述程序为例说明将BMOV指令替换为MOV指令的步骤。

- 程序

写入	1	2	3	4	5	6	7	8	9	10	11	12
1	(o) label trigger								BMOV	D1	destLabel	K5
2	(o)											[END]

- 局部标签

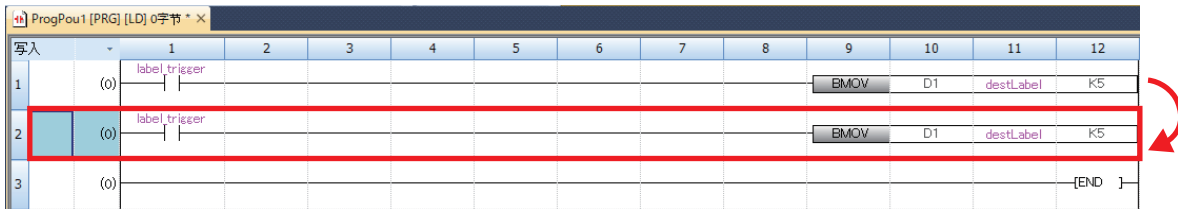
标签名	数据类型	类	初始值	常数
1 label_trigger	位	VAR		
2 destLabel	字[有符号](0.4)	VAR		

替换为多个MOV指令

将BMOV指令替换为多个MOV指令的步骤如下所示。

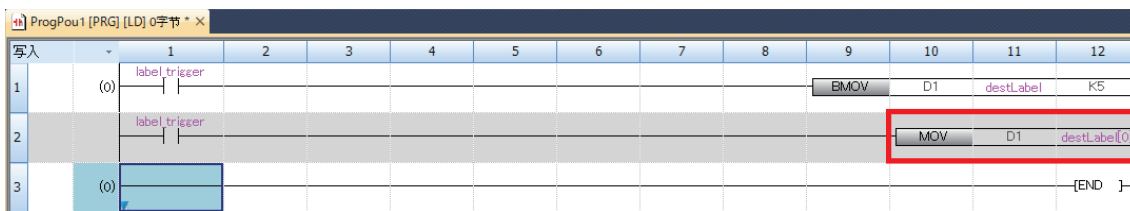
操作步骤

1. 打开使用BMOV指令的程序，复制BMOV指令行并粘贴至下一行。



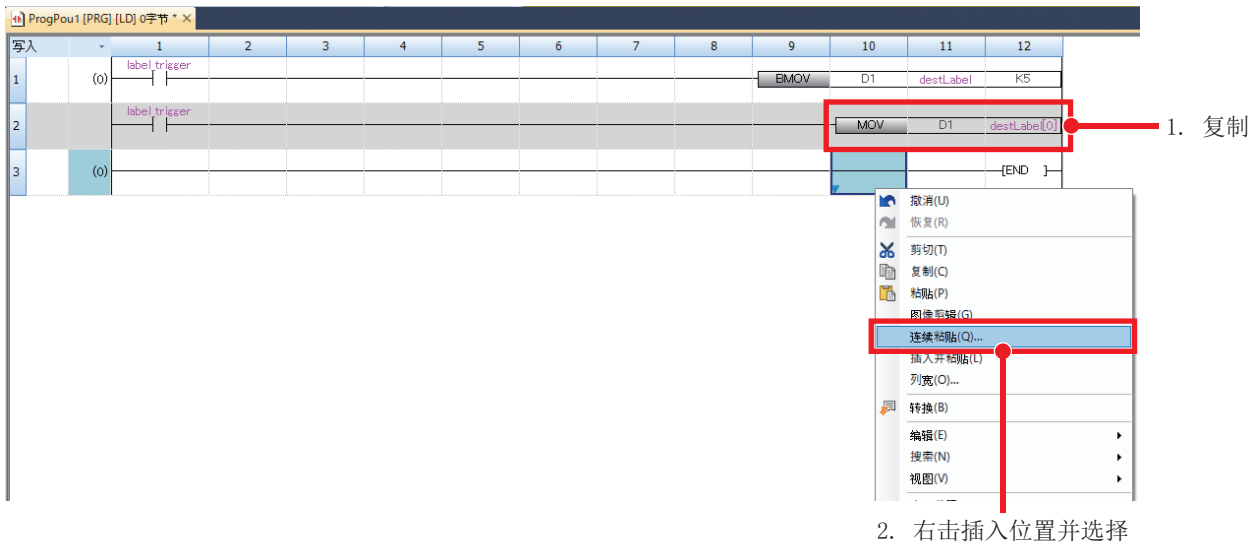
2. 将在步骤1中粘贴的BMOV指令更改为MOV指令。
更改自变量，确保“D1”的值传送到局部标签“destLabel[0]”。

• MOV D1 destLabel[0]



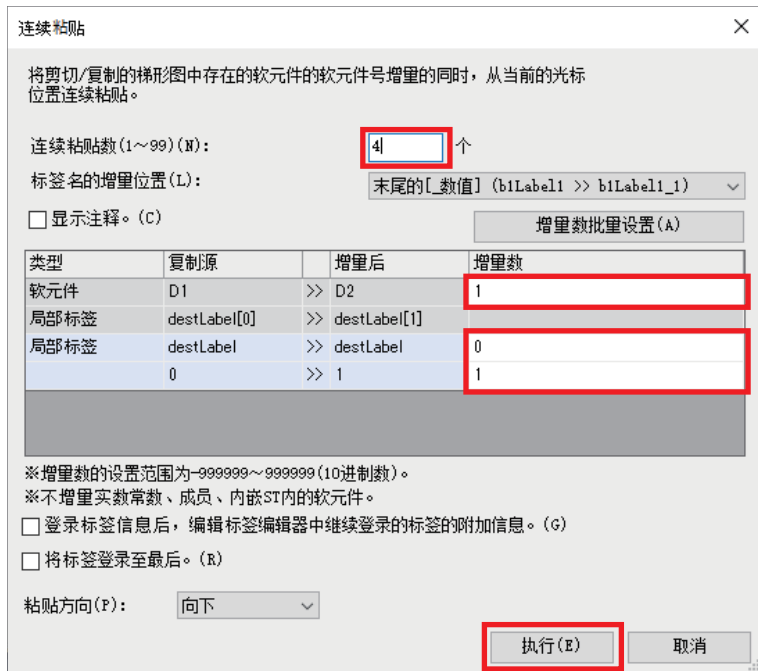
3. 复制在步骤2中修改的MOV指令，并在要插入下一个MOV指令的位置进行连续粘贴。

在要插入MOV指令位置右击⇒[连续粘贴]

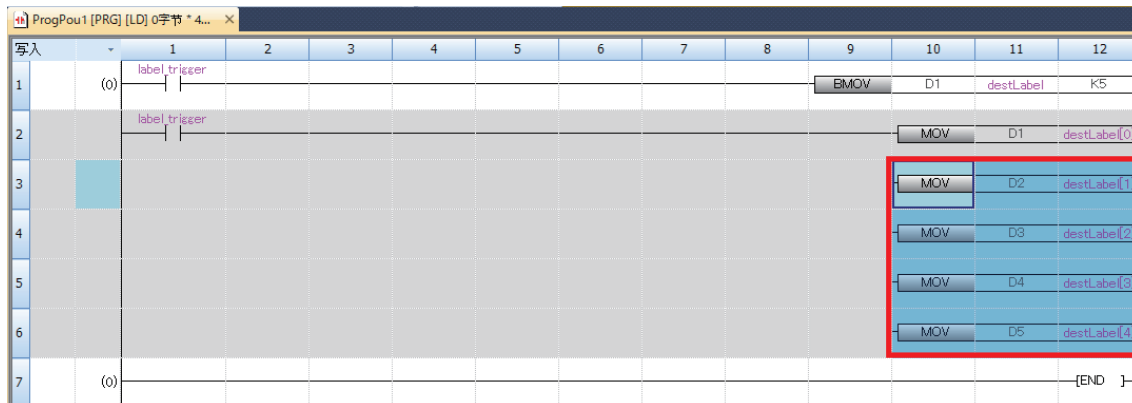


4. 设置连续粘贴数和增量数，点击[执行]按钮。

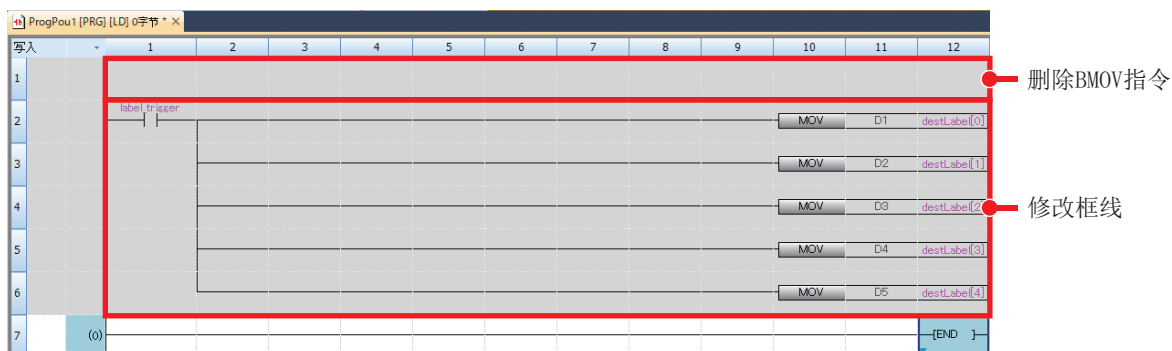
- 连续粘贴数：4个
- 软元件“D1”增量数：1
- 局部标签“destLabel”的数组元素的增量数：1



进行连续粘贴。



5. 修改框线将各MOV指令与在步骤1中添加的行的触点连接，并删除不需要的BMOV指令的行。



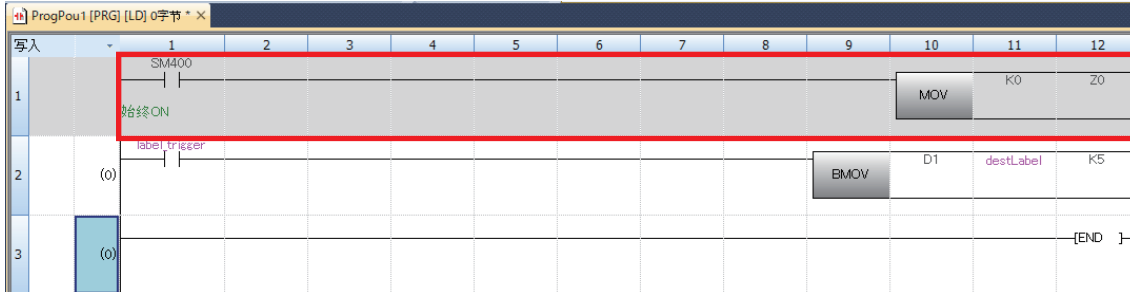
替换为FOR~NEXT指令和MOV指令

通过FOR~NEXT指令将BMOV指令替换为多次执行MOV指令的程序的步骤如下所示。

操作步骤

1. 打开使用了BMOV指令的程序，在第1行插入新行，并添加初始化变址寄存器“Z0”的梯形图。使用特殊继电器SM400(始终ON)作为触点，并通过下述MOV指令将K0传送至变址寄存器“Z0”。

- MOV K0 Z0



注意事项

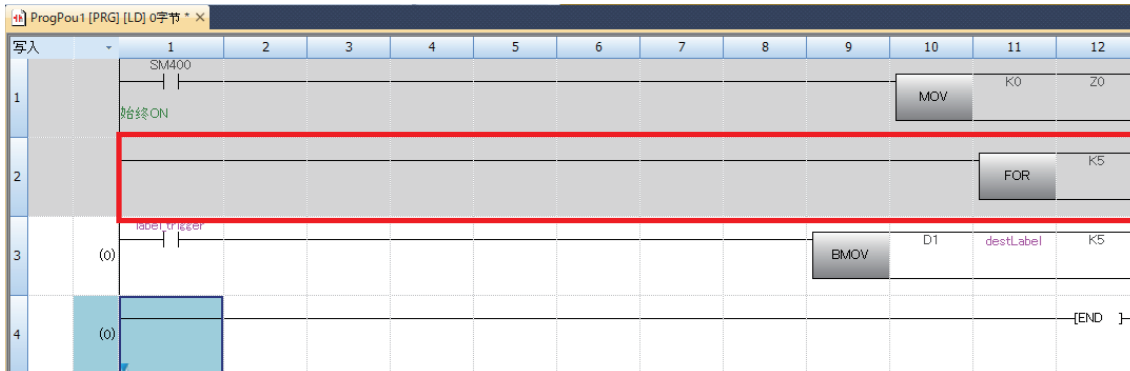
变址寄存器用于间接指定软元件编号。如果在程序的其他位置使用变址寄存器，请确保变址寄存器编号不重复。关于变址寄存器的详细内容，请参阅下述手册。

📖 MELSEC MX控制器(MX-R型)编程手册

2. 在第2行插入新行并添加FOR语句。

将自变量设置为“K5”，确保重复MOV指令的执行次数(传送次数)。

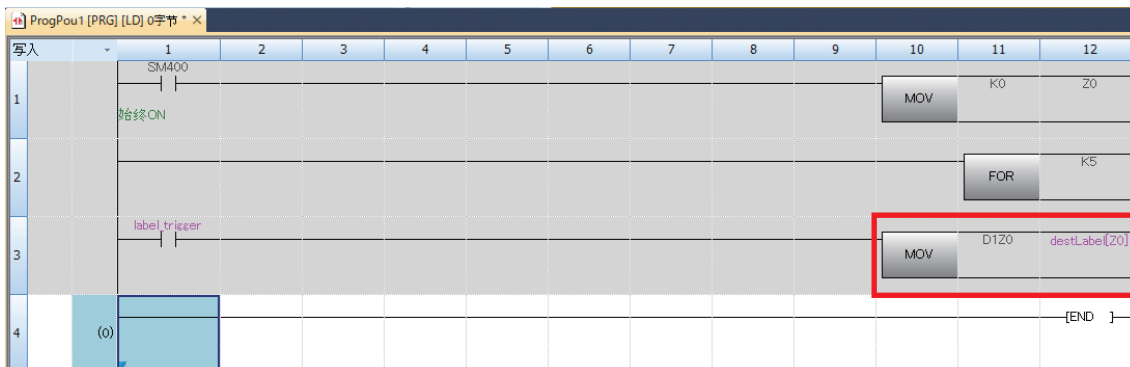
- FOR K5



3. 将BMOV指令更改为MOV指令。

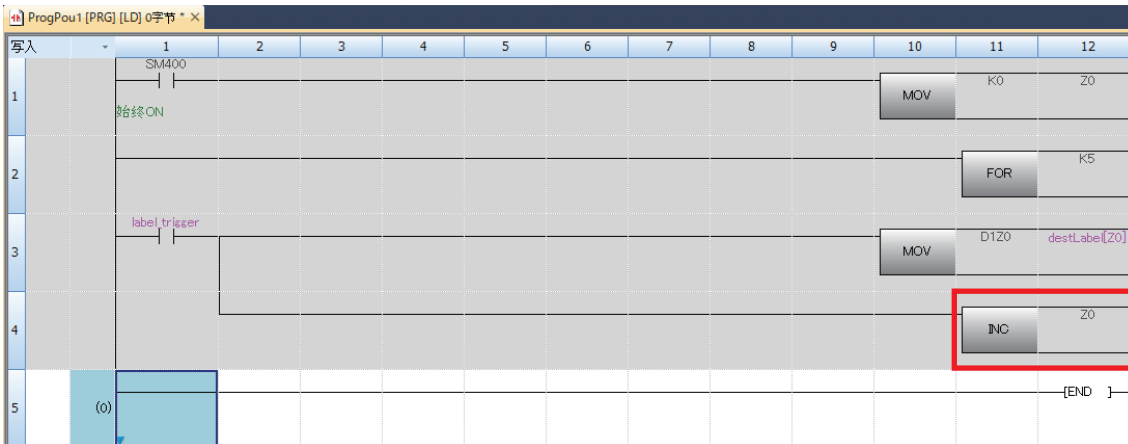
使用在步骤1中初始化的变址寄存器更改自变量，将“D1Z0”的值传送至局部标签“destLabel[Z0]”。

- MOV D1Z0 destLabel[Z0]

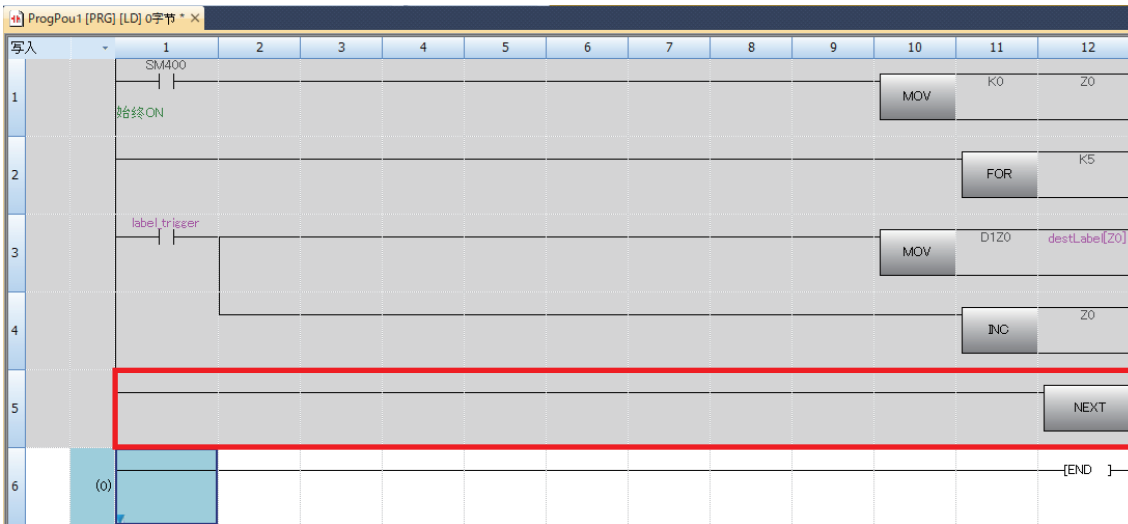


4. 在MOV指令的下一行添加以下INC指令，增加1个变址寄存器。
将添加的INC指令与MOV指令的触点连接。

- INC Z0



5. 在INC指令的下一行添加NEXT语句。



注意事项

- 如果将BMOV指令替换为多个MOV指令，可能会降低程序的可读性。
- 如果BMOV指令的传送数较多，即使替换为MOV指令也可能不会缩短扫描时间。替换前请计算并确认MOV指令和BMOV指令的执行速度。关于执行速度，请参照下述章节。

☞ 54页 MOV指令和BMOV指令的性能值

4.2 减少上升沿执行指令

概要

由于上升沿执行指令(指令符号的句尾有“P”，且仅在OFF→ON的上升沿时执行的指令)的处理性能比ON时执行指令慢，因此减少上升沿执行指令可以缩短扫描时间。

减少上升沿指令有以下方法。

- 将不需要的上升沿执行指令替换为ON时执行指令
- 如果使用多个上升沿执行指令，则将触点更改为上升沿执行指令，并将其他上升沿执行指令更改为ON时执行指令

上升沿执行指令的性能值

如果执行条件为上升沿指令时，则会在ON时执行指令的处理时间上增加额外的时间。(例：MOV_P的处理时间 = MOV的处理时间 + 额外时间)

各机型的额外时间如下所示。

替换或更改上升沿指令时，请以此为标准。

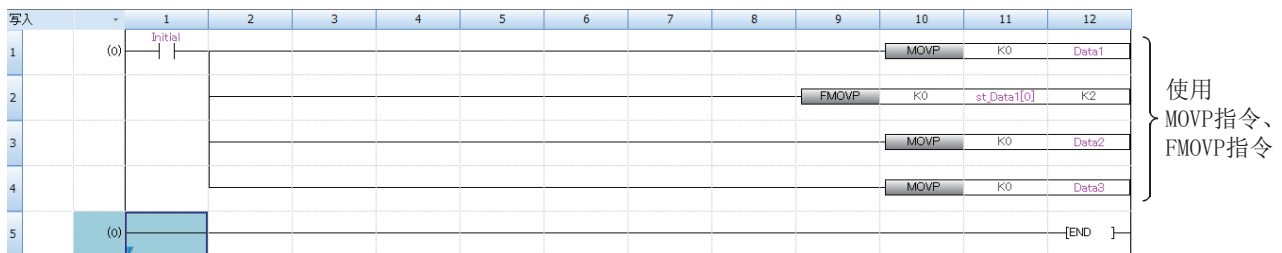
机型	额外时间(n秒)
MXR300-16、MXR300-32、MXR300-64	28.62
MXR500-128、MXR500-256	20.81

注意事项

性能值根据机型有所不同。

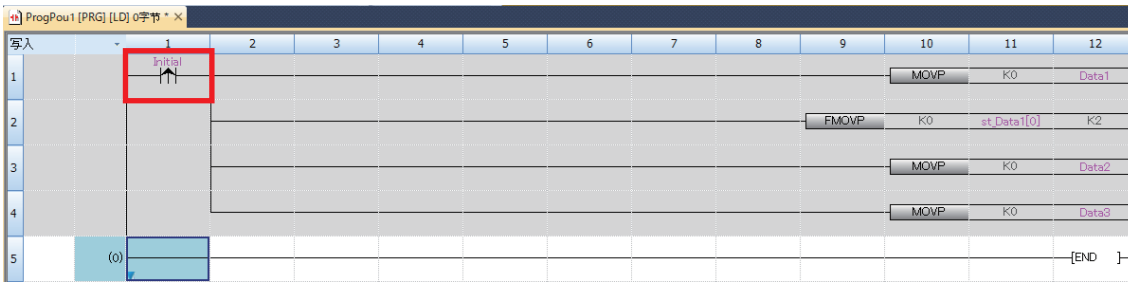
执行步骤

以下述程序为例说明减少上升沿执行指令“MOV_P”和“FMOV_P”的步骤。



操作步骤

1. 打开程序，将连接多个上升沿执行指令的触点更改为上升沿执行指令。
该示例中，将常开触点运算开始指令(LD)更改为上升沿脉冲运算开始指令(LDP)。



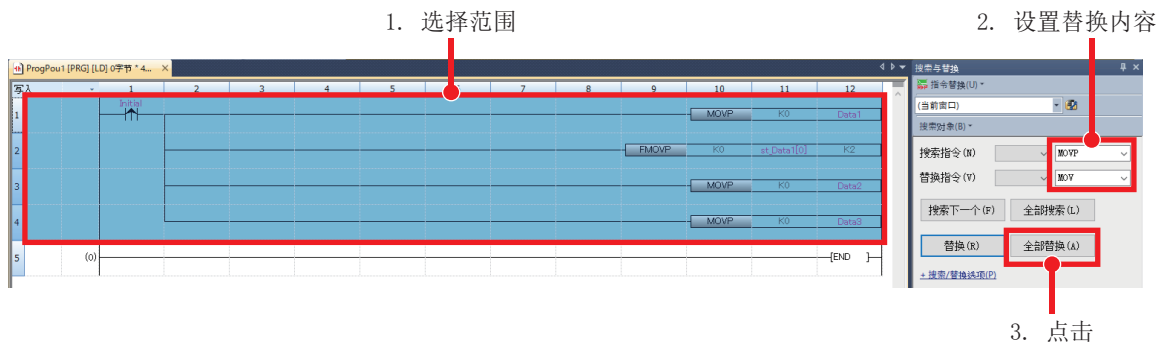
2. 将步骤1中与触点连接的上升沿执行指令更改为ON时执行指令。

- MOV指令更改为MOV指令
- FMOV指令更改为FMOV指令



要点

如果有多个相同指令，可以在“搜索与替换”窗口的“指令替换”中批量替换。
选择替换范围后，设置搜索指令和替换指令，点击[全部替换]按钮。



4.3 减少模块访问指令

概要

访问模块缓冲存储器时，可以通过合并使用FROM指令和TO指令等模块访问指令对缓冲存储器的直接访问，或更改为使用智能模块的自动刷新功能或链接刷新功能进行访问来缩短扫描时间。


模块访问指令和链接刷新的性能值

MXR500-128、MXR500-256的模块访问指令、自动刷新功能及链接刷新功能的性能值示例如下所示。

替换模块访问指令时，请以此为标准。

项目	每个字的最小执行性能值 (n秒)		
	传送数：1字	传送数：1024字	传送数：32768字
FROM指令	5470	40.42	—
TO指令	3740	23.43	—
自动刷新(读取)*1	10730	30.45	20.32
自动刷新(写入)*1	7360	17.17	10.22
链接刷新(CC-Link IE TSN模块)*1	52012	62.78	13.58

*1 使用MX控制器手册记载的计算公式计算。关于详细内容，请参阅下述手册。

 MELSEC MX控制器 (MX-R型) 用户手册

注意事项

性能值根据机型有所不同。

执行步骤

减少模块访问指令的步骤如下所示。

FROM指令和TO指令的聚合

以下述从起始I/O为0x0040的智能功能模块的缓冲存储器读取2000点(2000字)的值至D软元件的程序为例，说明将20个FROM指令聚合为1个FROM指令的步骤。

写入	1	2	3	4	5	6	7	8	9	10	11	12
1	(0)							FROM	U4	K49152	D0	K100
2								FROM	U4	K49152	D100	K100
3								FROM	U4	K49152	D200	K100
4								FROM	U4	K49152	D300	K100
5								FROM	U4	K49152	D400	K100
6								FROM	U4	K49152	D500	K100
7								FROM	U4	K49152	D600	K100
8								FROM	U4	K49152	D700	K100
9								FROM	U4	K49152	D800	K100
10								FROM	U4	K49152	D900	K100
11								FROM	U4	K49152	D1000	K100
12								FROM	U4	K49152	D1100	K100
13								FROM	U4	K49152	D1200	K100
14								FROM	U4	K49152	D1300	K100
15								FROM	U4	K49152	D1400	K100
16								FROM	U4	K49152	D1500	K100
17								FROM	U4	K49152	D1600	K100
18								FROM	U4	K49152	D1700	K100
19								FROM	U4	K49152	D1800	K100
20								FROM	U4	K49152	D1900	K100
21								FROM	U4	K49152	D2000	K100

用20个FROM指令读取2000字的数据，每次读取100字

操作步骤

1. 打开相应的程序文件，将第1行FROM指令的读取数据数更改为K2000(2000字)。

写入	1	2	3	4	5	6	7	8	9	10	11	12
1	(0)							FROM	U4	K49152	D0	K2000
2								FROM	U4	K49152	D100	K100
3								FROM	U4	K49152	D200	K100
4								FROM	U4	K49152	D300	K100
5								FROM	U4	K49152	D400	K100
6								FROM	U4	K49152	D500	K100
7								FROM	U4	K49152	D600	K100

修改为K2000

2. 删除第2行以后的FROM指令。

这样可以将2000字数据依次读取100字的20个FROM指令聚合为1个批量读取2000字数据的FROM指令。

写入	1	2	3	4	5	6	7	8	9	10	11	12
1	(0)							FROM	U4	K49152	D0	K2000
2												
3												
4												
5												
6												
7												
8												
9												
10												
11												
12												
13												
14												
15												
16												
17												
18												
19												
20												
21												

删除

替换为自动刷新功能或链接刷新功能

将模块访问指令替换为自动刷新功能或链接刷新功能。

关于自动刷新功能和链接刷新功能的使用方法，请参阅各模块的手册。

4.4 应用初始执行类型程序

概要

扫描执行类型程序，每次扫描执行1次程序。

如果在扫描执行类型的ST程序(运动控制程序)中有可编程控制器RUN后只需执行1次的处理，则可通过将其移动至初始执行类型ST程序来缩短扫描时间。

执行步骤

以下述ST程序为例，说明在程序中将在可编程控制器RUN后只需执行1次的处理移动至初始执行类型程序的步骤。

- ST程序

```
1 | WdRollWidth := 300.0; //表单长度[mm]
2 | WdSpeedMax := 600.0; //最大速度[r/min]
3 | WdMaxDia := 120.0; //卷径最大值[mm]
4 | MotorInertia := 0.0777e-4; //惯性[kg·m^2]
5 |
6 | G_MCv_AllPower_1(
7 | // Axis:= ?AXIS_REF? ,
8 | Enable:= TRUE ,
9 | ServoON:= TRUE
10 | // Busy=> ?BOOL? ,
11 | // Error=> ?BOOL? ,
12 | // ErrorID=> ?WORD?
13 | );
14 |
```

- 局部标签

标签名	数据类型	类	初始值	常数
1 WdRollWidth	单精度实数	VAR		
2 WdSpeedMax	单精度实数	VAR		
3 WdMaxDia	单精度实数	VAR		
4 MotorInertia	单精度实数	VAR		
5 MCv_Jog.1	MCv_Jog	VAR		

操作步骤

1. 新建一个初始执行类型的程序文件。

☞ 导航窗口⇒[程序]⇒右击[初始]⇒[新建数据]



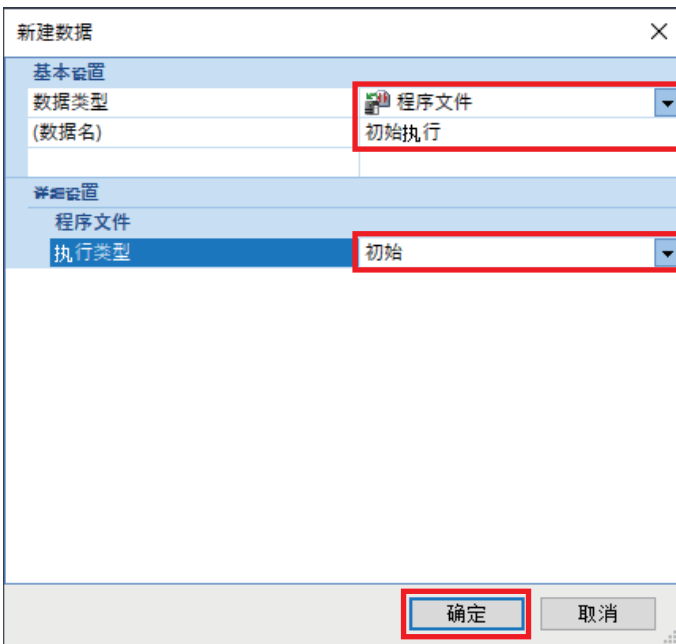
要点

每个程序文件的程序量越少，程序转换时间就越短。

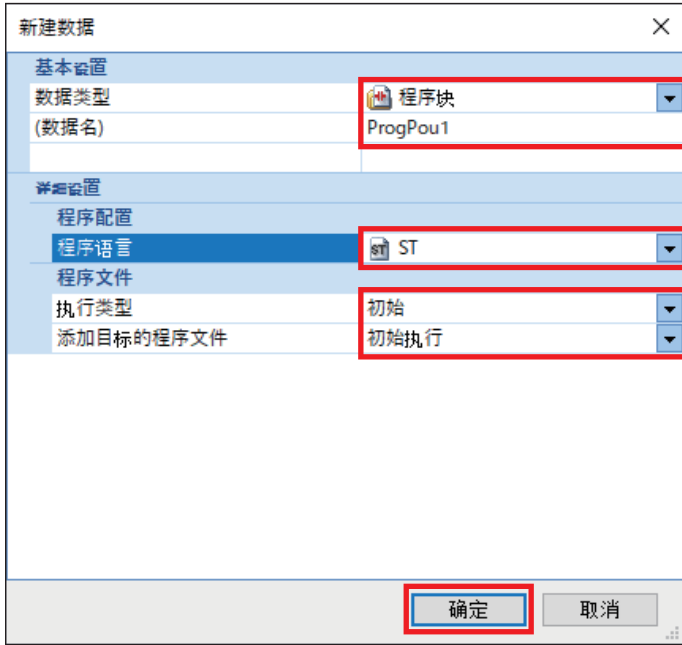
因此，建议将可编程控制器RUN后只执行1次的处理移动到新的初始执行类型的程序文件中。（即使移动到现有的初始执行类型的程序文件中，操作也不会受影响）

2. 设置各项目，点击[确定]按钮。

- 数据类型：程序文件
- (数据名)：初始执行
- 执行类型：初始

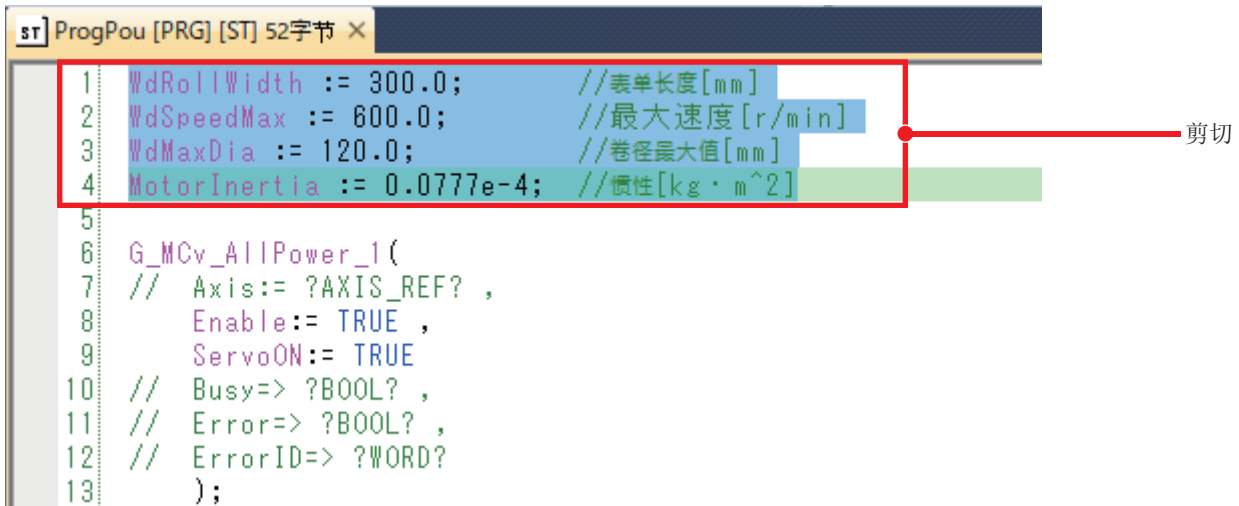


3. 按照与步骤1相同的步骤打开“新建数据”画面，新建程序块。
设置各项目，点击[确定]按钮。

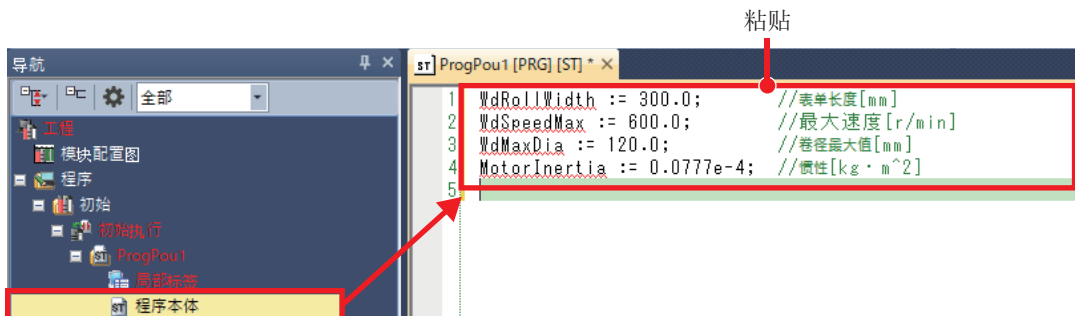


项目	设置内容	备注
数据类型	程序块	—
(数据名)	ProgPou1	—
程序语言	ST	—
执行类型	初始	设置与步骤2中创建的程序文件相同的执行类型。
添加目标的程序文件	初始执行	设置在步骤2中创建的程序文件。

4. 打开移动前的ST程序，剪切第1~4行设置各标签的初始值的处理。



5. 打开在步骤3中创建的初始执行类型程序，粘贴在步骤4中剪切的处理。



6. 打开移动前的ST程序文件的局部标签，剪切在初始执行类型程序中使用的局部标签(该示例中为No. 1~4)。点击编辑器的表的行编号，选择整行后剪切。

	标签名	数据类型	类	初始值	常数
1	WdRollWidth	单精度实数	VAR		
2	WdSpeedMax	单精度实数	VAR		
3	WdMaxDia	单精度实数	VAR		
4	MotorInertia	单精度实数	VAR		
5	MCv_Jbg_1	MCv_Jbg	VAR		

要点

如果未选择整行，则无法复制标签编辑器上隐藏的内容。

7. 打开全局标签，粘贴在步骤6中剪切的全局标签，设置类。

	标签名	数据类型	类	分配软元件/标签	初始值	常数
1	WdRollWidth	单精度实数	VAR_GLOBAL			
2	WdSpeedMax	单精度实数	VAR_GLOBAL			
3	WdMaxDia	单精度实数	VAR_GLOBAL			
4	MotorInertia	单精度实数	VAR_GLOBAL			

要点

本示例中，局部标签粘贴在全局标签中，但不一定是全局标签。请确认相应标签的引用状况，选择要粘贴到局部标签还是全局标签。

8. 打开初始执行类型程序，写入将FB实例“G_MCv_AllPower_1”的输入变量“Enable”和“ServoON”指定为“TRUE”的处理。

- G_MCv_AllPower_1.Enable := TRUE;
- G_MCv_AllPower_1.ServoON := TRUE;

```

1 WdRollWidth := 300.0; // 表单长度[mm]
2 WdSpeedMax := 600.0; // 最大速度[r/min]
3 WdMaxDia := 120.0; // 卷径最大值[mm]
4 MotorInertia := 0.0777e-4; // 惯性[kg·m^2]
5
6 G_MCv_AllPower_1.Enable := TRUE ;
7 G_MCv_AllPower_1.ServoON := TRUE ;
8

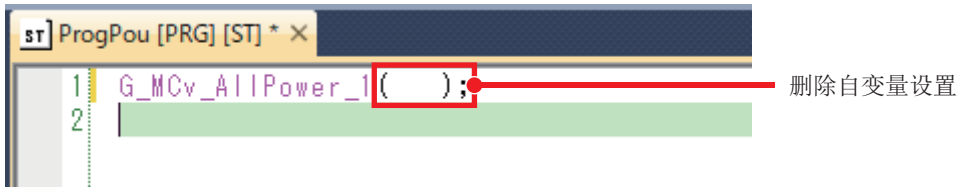
```

要点

- 由于运动控制FB的实例在移动前的程序和初始执行类型程序中都会使用，因此必须是全局标签。如果是局部标签，请更改为全局标签。
- 在运动控制FB的实例名的后面输入“.”(句号)会显示标签名的候补，选择候补即可自动输入标签名。如果不显示候补，则可以从菜单的[工具]⇒[选项]⇒“编辑”⇒“指令/软元件/标签候补显示”⇒“运行设置”⇒“预测并显示指令/软元件/标签名”设置。关于详细内容，请参阅下述手册。

📖 GX Works3 操作手册

9. 打开移动前的ST程序，删除FB实例“G_McV_AllPower_1”的自变量设置。



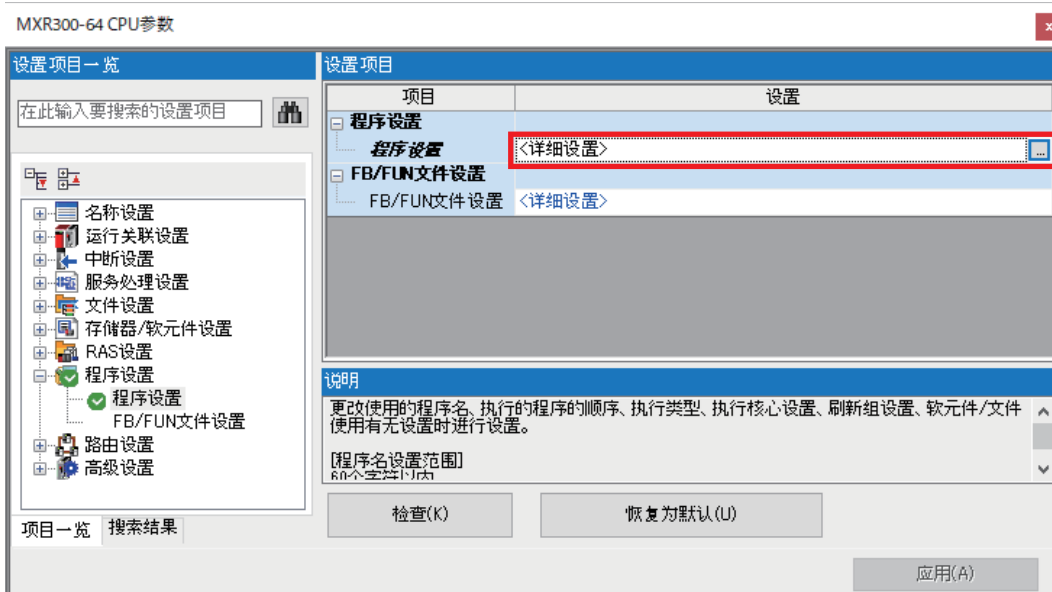
10. 打开CPU参数。

双击导航窗口⇒[参数]⇒机型名⇒[CPU参数]

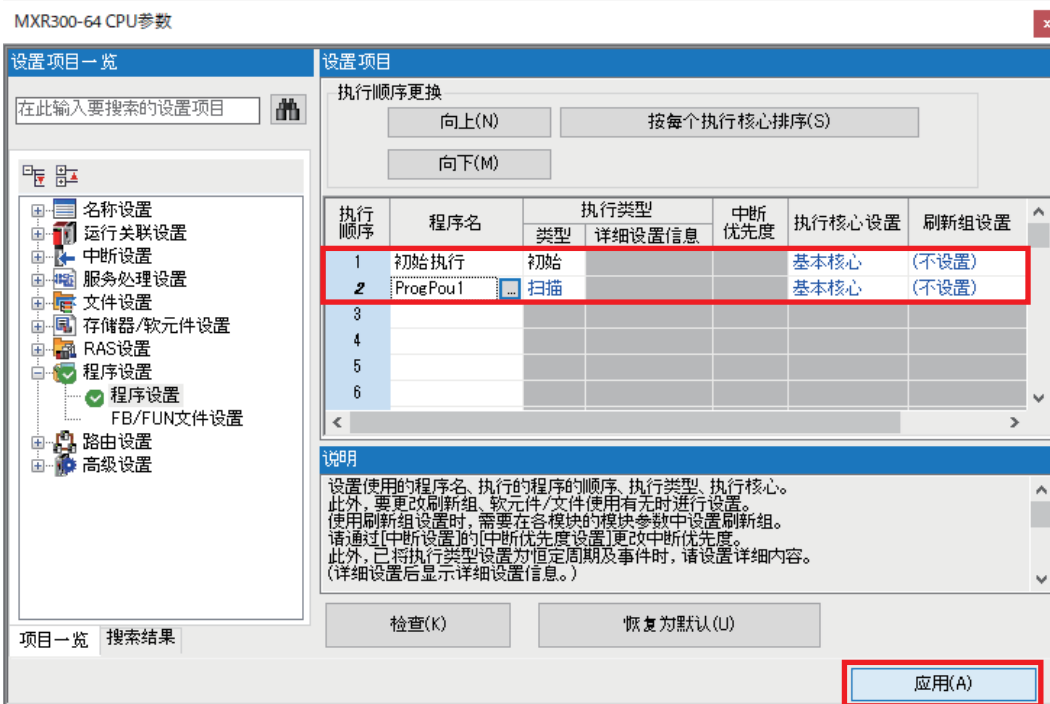


11. 要设置程序的执行顺序，请打开程序设置。

双击[程序设置]⇒[程序设置]的<详细设置>



12. 重新排列程序，确保首先执行初始执行类型程序，并点击[应用]按钮应用设置。



4.5 省略自变量和直接引用

概要

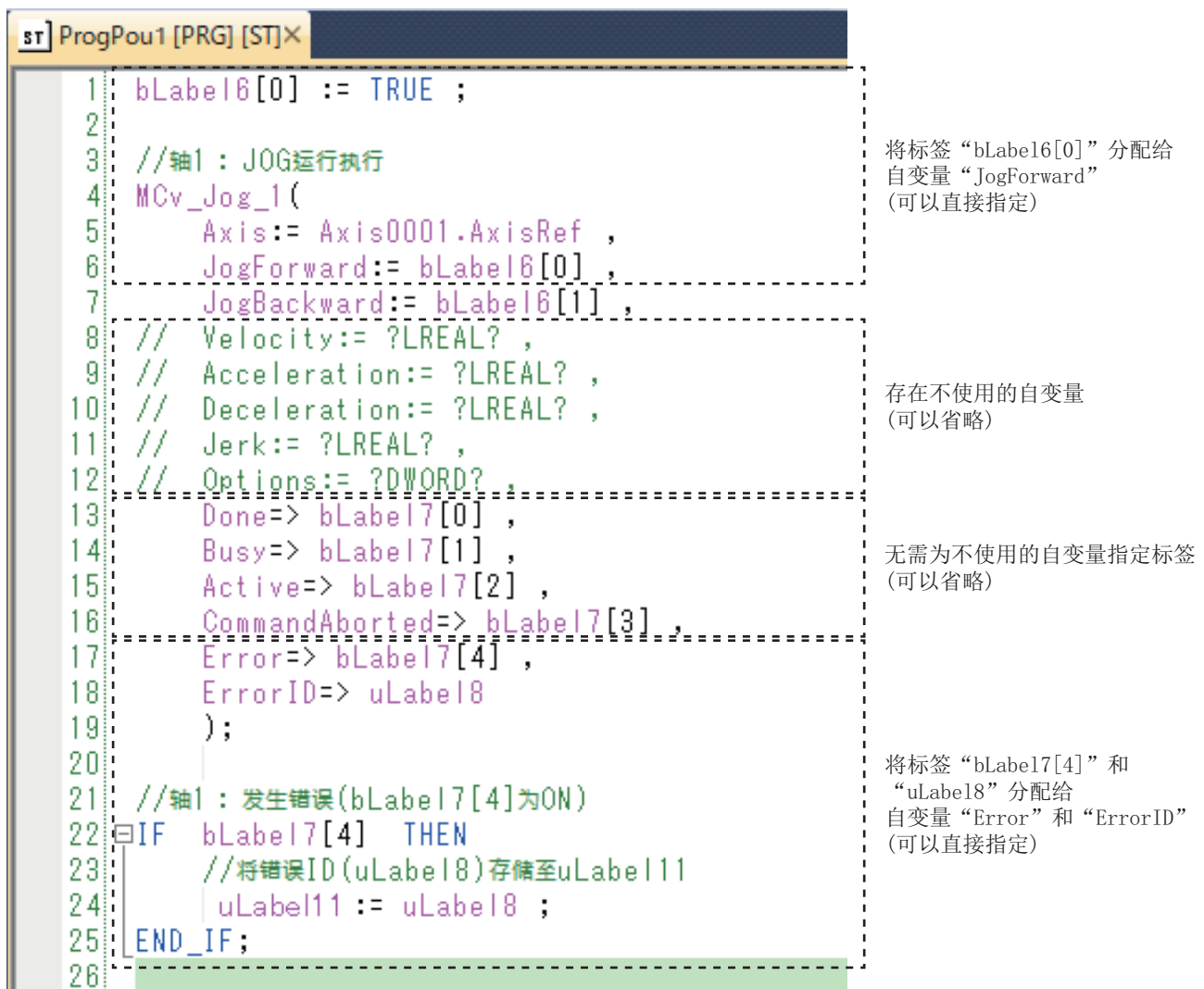
ST程序(运动控制程序)中的运动控制FB持有许多自变量，如果给输入输出自变量分配其他标签，容易发生代入/引用处理的负载。

因此不使用以下的其他标签，可以减少代入/引用处理的生成，从而缩短扫描时间。

- 不使用标签，直接向输入输出自变量输入值
- 省略不需要的FB自变量的值/标签的分配
- 在IF语句的条件判定中直接引用FB的标签
- 将FB的值代入其他标签时直接代入FB自变量

执行步骤

以下述程序为例说明修改运动控制FB的实例“MCv_Jog_1”的输入输出自变量设置的步骤。



```
1 | bLabel16[0] := TRUE ;
2 |
3 | //轴1 : JOG运行执行
4 | MCv_Jog_1(
5 |     Axis:= Axis0001.AxisRef ,
6 |     JogForward:= bLabel16[0] ,
7 |     JogBackward:= bLabel16[1] ,
8 |     // Velocity:= ?LREAL? ,
9 |     // Acceleration:= ?LREAL? ,
10 |    // Deceleration:= ?LREAL? ,
11 |    // Jerk:= ?LREAL? ,
12 |    // Options:= ?DWORD? ,
13 |    Done=> bLabel17[0] ,
14 |    Busy=> bLabel17[1] ,
15 |    Active=> bLabel17[2] ,
16 |    CommandAborted=> bLabel17[3] ,
17 |    Error=> bLabel17[4] ,
18 |    ErrorID=> uLabel18
19 | );
20 |
21 | //轴1 : 发生错误(bLabel17[4]为ON)
22 | IF bLabel17[4] THEN
23 |     //将错误ID(uLabel18)存储至uLabel11
24 |     uLabel11 := uLabel18 ;
25 | END_IF;
26 |
```

将标签“bLabel16[0]”分配给自变量“JogForward”
(可以直接指定)

存在不使用的自变量
(可以省略)

无需为不使用的自变量指定标签
(可以省略)

将标签“bLabel17[4]”和“uLabel18”分配给自变量“Error”和“ErrorID”
(可以直接指定)

操作步骤

1. 不使用标签“bLabel6[0]”，直接将FB实例“MCv_Jog_1”的输入变量“JogForward”指定为“TRUE”。将第1行的标签“bLabel6[0]”更改为“MCv_Jog_1.JogForward”。此外，注释掉或删除“MCv_Jog_1”的自变量“JogForward”的设置。

```

st] ProgPou1 [PRG] [ST] ×
1  MCv_Jog_1.JogForward := TRUE ;
2
3  //轴1 : JOG运行执行
4  MCv_Jog_1(
5      Axis:= Axis0001.AxisRef ,
6      // JogForward:= bLabel6[0] ,
7      JogBackward:= bLabel6[1] ,
8      // Velocity:= ?LREAL? ,
9      // Acceleration:= ?LREAL? ,
10     // Deceleration:= ?LREAL? ,
11     // Jerk:= ?LREAL? ,
12     // Options:= ?DWORD? ,
13     Done=> bLabel7[0] ,
14     Busy=> bLabel7[1] ,
15     Active=> bLabel7[2] ,
16     CommandAborted=> bLabel7[3] ,
17     Error=> bLabel7[4] ,
18     ErrorID=> uLabel8
19 );

```

更改为“MCv_Jog_1.JogForward”

注释掉

要点

在运动控制FB的实例名的后面输入“.”(句号)会显示标签名的候补，选择候补即可自动输入标签名。

如果不显示候补，则可以从菜单的[工具]⇒[选项]⇒“编辑”⇒“指令/软元件/标签候补显示”⇒“运行设置”⇒“预测并显示指令/软元件/标签名”设置。关于详细内容，请参阅下述手册。

📖 GX Works3 操作手册

2. 注释掉或删除第8~16行未使用的自变量设置。

```

st] ProgPou [PRG] [ST] * ×
1  MCv_Jog_1.JogForward := TRUE ;
2
3  //轴1 : JOG运行执行
4  MCv_Jog_1(
5      Axis:= Axis0001.AxisRef ,
6      // JogForward:= bLabel6[0] ,
7      JogBackward:= bLabel6[1] ,
8
9
10
11
12
13     // Done=> bLabel7[0] ,
14     // Busy=> bLabel7[1] ,
15     // Active=> bLabel7[2] ,
16     // CommandAborted=> bLabel7[3] ,
17     Error=> bLabel7[4] ,
18     ErrorID=> uLabel8
19 );

```

删除或注释掉

可以省略不使用或不更改初始值的FB的输入输出变量。

3. 不使用标签“bLabel7[4]”，直接指定FB实例“MCv_Jog_1”的输入变量“Error”。
 将第16、17行的标签“bLabel7[4]”更改为“MCv_Jog_1.Error”。此外，注释掉或删除“MCv_Jog_1”的自变量“Error”的设置。

```

1  MCv_Jog_1.JogForward := TRUE ;
2
3  //轴1 : JOG运行执行
4  MCv_Jog_1(
5      Axis:= Axis0001.AxisRef ,
6      // JogForward:= bLabel6[0] ,
7      JogBackward:= bLabel6[1] ,
8      // Done=> bLabel7[0] ,
9      // Busy=> bLabel7[1] ,
10     // Active=> bLabel7[2] ,
11     // CommandAborted=> bLabel7[3] ,
12     // Error=> bLabel7[4] ,
13     ErrorID=> uLabel8
14 );
15
16 //轴1 : 发生错误(MCv_Jog_1.Error为ON)
17 IF MCv_Jog_1.Error THEN
18     //将错误ID(uLabel8)存储至uLabel11
19     uLabel11 := uLabel8 ;
20 END_IF;
21
    
```

删除或注释掉

将“bLabel7[4]”更改为“MCv_Jog_1.Error”

4. 不使用标签“uLabel8”，直接指定FB实例“MCv_Jog_1”的输入变量“ErrorID”。
 将第18、19行的标签“uLabel8”更改为“MCv_Jog_1.ErrorID”。此外，注释掉或删除“MCv_Jog_1”的自变量“ErrorID”的设置。

```

1  MCv_Jog_1.JogForward := TRUE ;
2
3  //轴1 : JOG运行执行
4  MCv_Jog_1(
5      Axis:= Axis0001.AxisRef ,
6      // JogForward:= bLabel6[0] ,
7      JogBackward:= bLabel6[1] ,
8      // Done=> bLabel7[0] ,
9      // Busy=> bLabel7[1] ,
10     // Active=> bLabel7[2] ,
11     // CommandAborted=> bLabel7[3] ,
12     // Error=> bLabel7[4] ,
13     // ErrorID=> uLabel8
14 );
15
16 //轴1 : 发生错误(MCv_Jog_1.Error为ON)
17 IF MCv_Jog_1.Error THEN
18     //将错误ID(MCv_Jog_1.ErrorID)存储至uLabel11
19     uLabel11 := MCv_Jog_1.ErrorID ;
20 END_IF;
21
    
```

删除或注释掉

将“uLabel8”更改为“MCv_Jog_1.ErrorID”

5. 根据在步骤1~4中注释掉和删除内容修改程序，确保语句没有错误。
在本示例中，删除FB自变量设置多余的逗号。

```
st] ProgPou [PRG] [ST] 52字节 * ×
1  MCv_Jog_1.JogForward := TRUE ;
2
3  //轴1 : JOG运行执行
4  MCv_Jog_1(
5      Axis:= Axis0001.AxisRef ,
6      // JogForward:= bLabel6[0] ,
7      JogBackward:= bLabel6[1] ,
8      // Done=> bLabel7[0] ,
9      // Busy=> bLabel7[1] ,
10     // Active=> bLabel7[2] ,
11     // CommandAborted=> bLabel7[3] ,
12     // Error=> bLabel7[4] ,
13     // ErrorID=> uLabel8
14     );
15
16 //轴1 : 发生错误(MCv_Jog_1.Error为ON)
17 IF MCv_Jog_1.Error THEN
18     //将错误ID(MCv_Jog_1.ErrorID)存储至uLabel11
19     uLabel11 := MCv_Jog_1.ErrorID ;
20 END_IF;
21
```

删除逗号

要点

如果因直接指定输入输出自变量或删除不需要的标签分配而产生整个项目中不使用的标签，请从标签编辑器中删除不需要的标签。

4.6 划分含有多条指令的梯形图块

概要

如果1个梯形图块中有100多个指令，可以将指令划分为多个梯形图块并将1个梯形图块中含有的指令数设为100个或更少，从而缩短指令处理时间。这样可以缩短扫描时间。

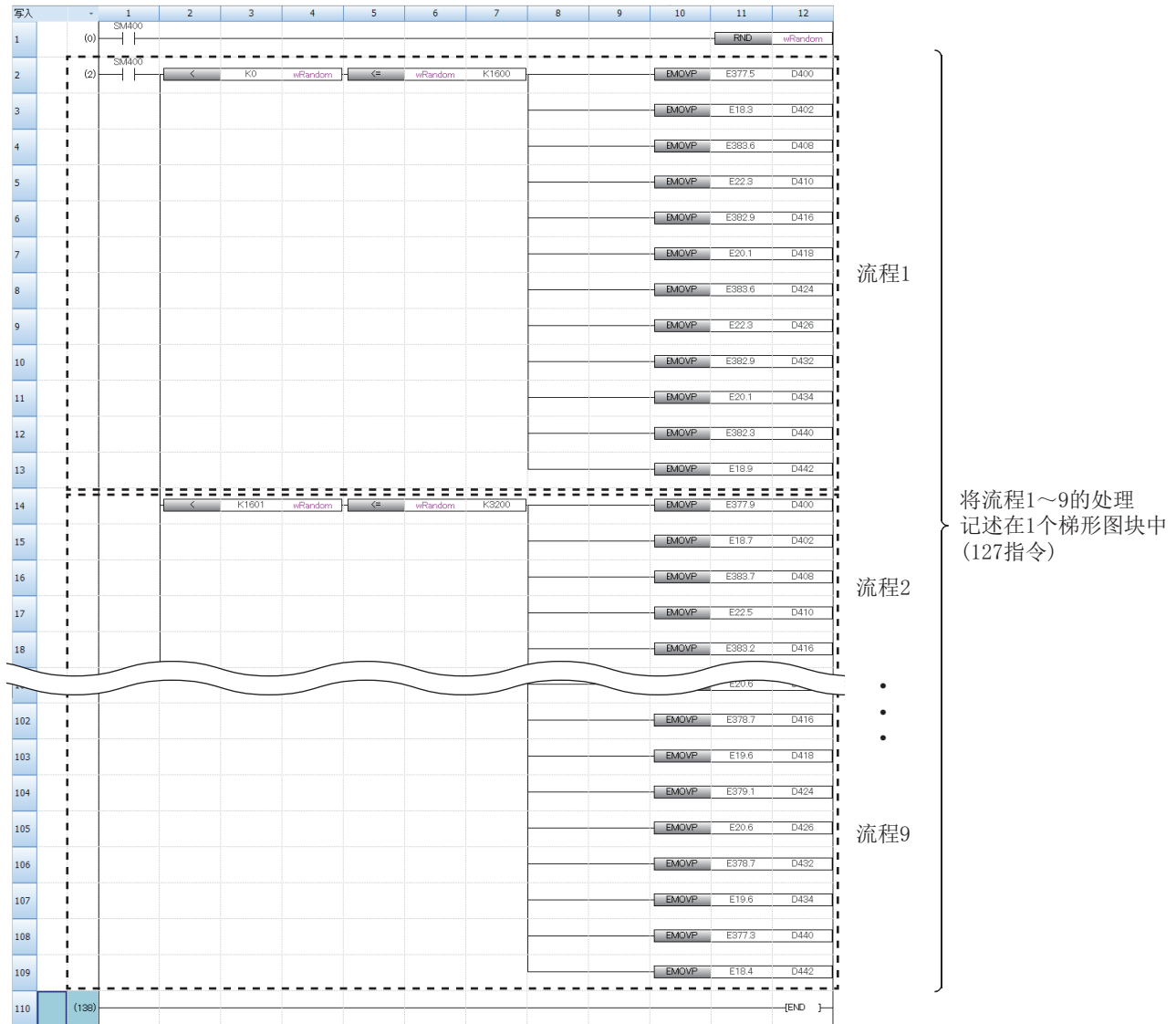
划分梯形图块还有下述好处。

提高程序的可读性

将梯形图块划分为多个，可以提高程序的可读性。

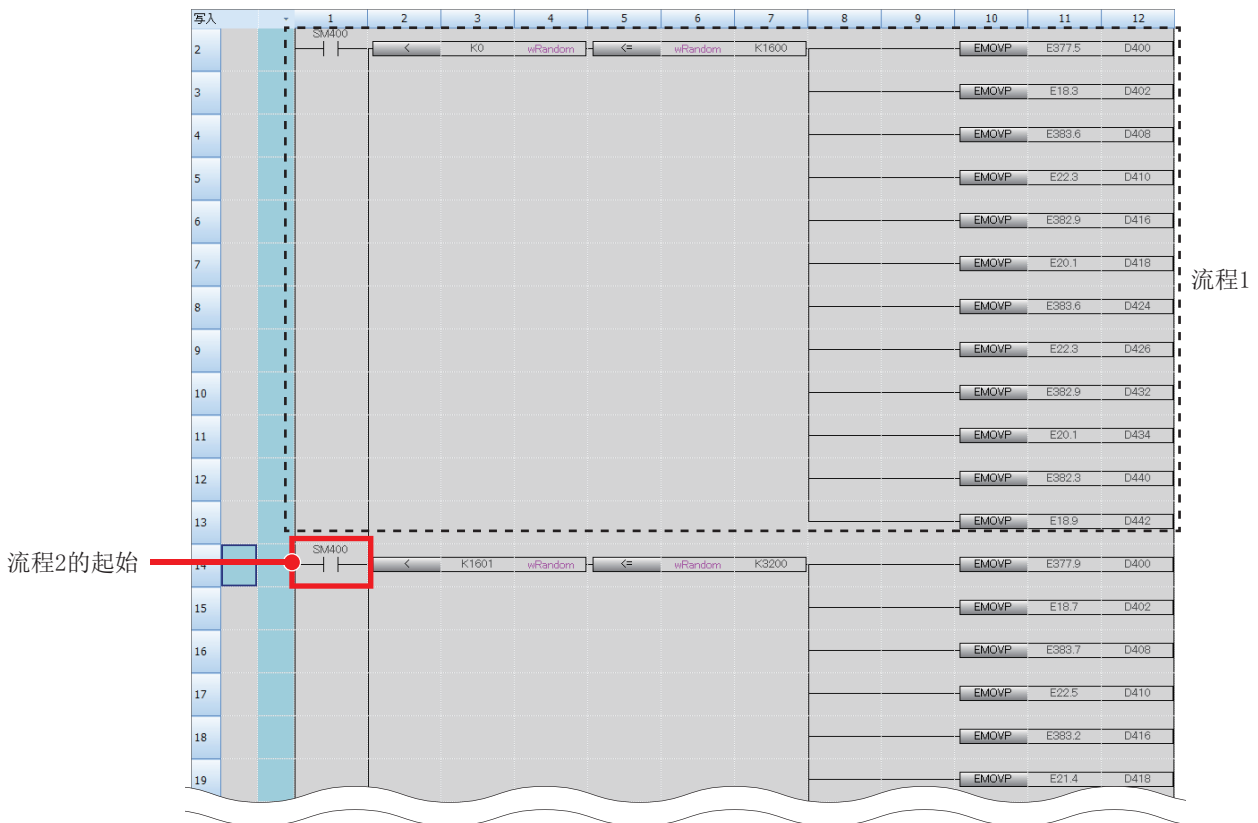
执行步骤

以下述程序为例，说明将包含流程1~9指令(127指令)的梯形图块按照流程划分的步骤。

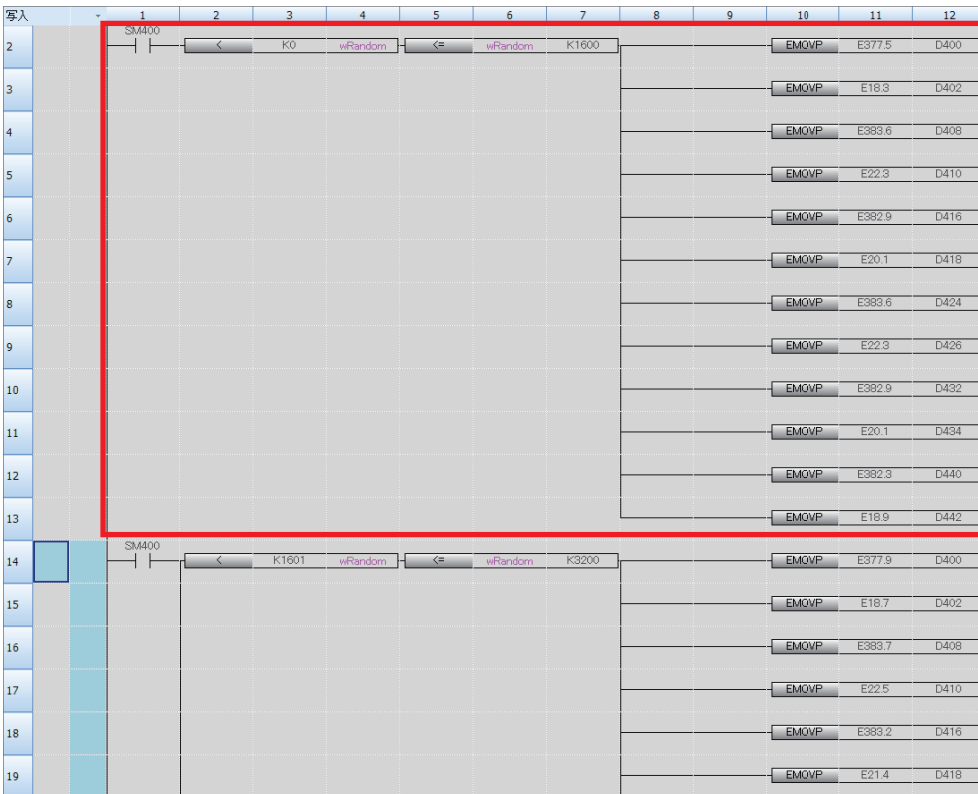


操作步骤

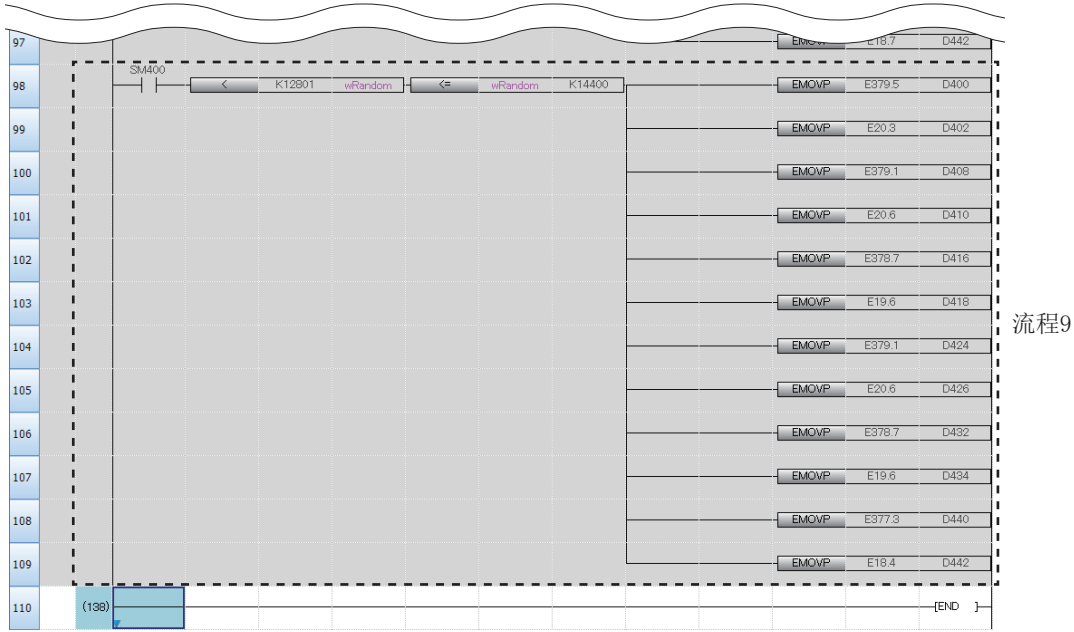
1. 打开程序，在划分位置(流程2的起始)添加与流程1的第1行相同的触点“SM400”。



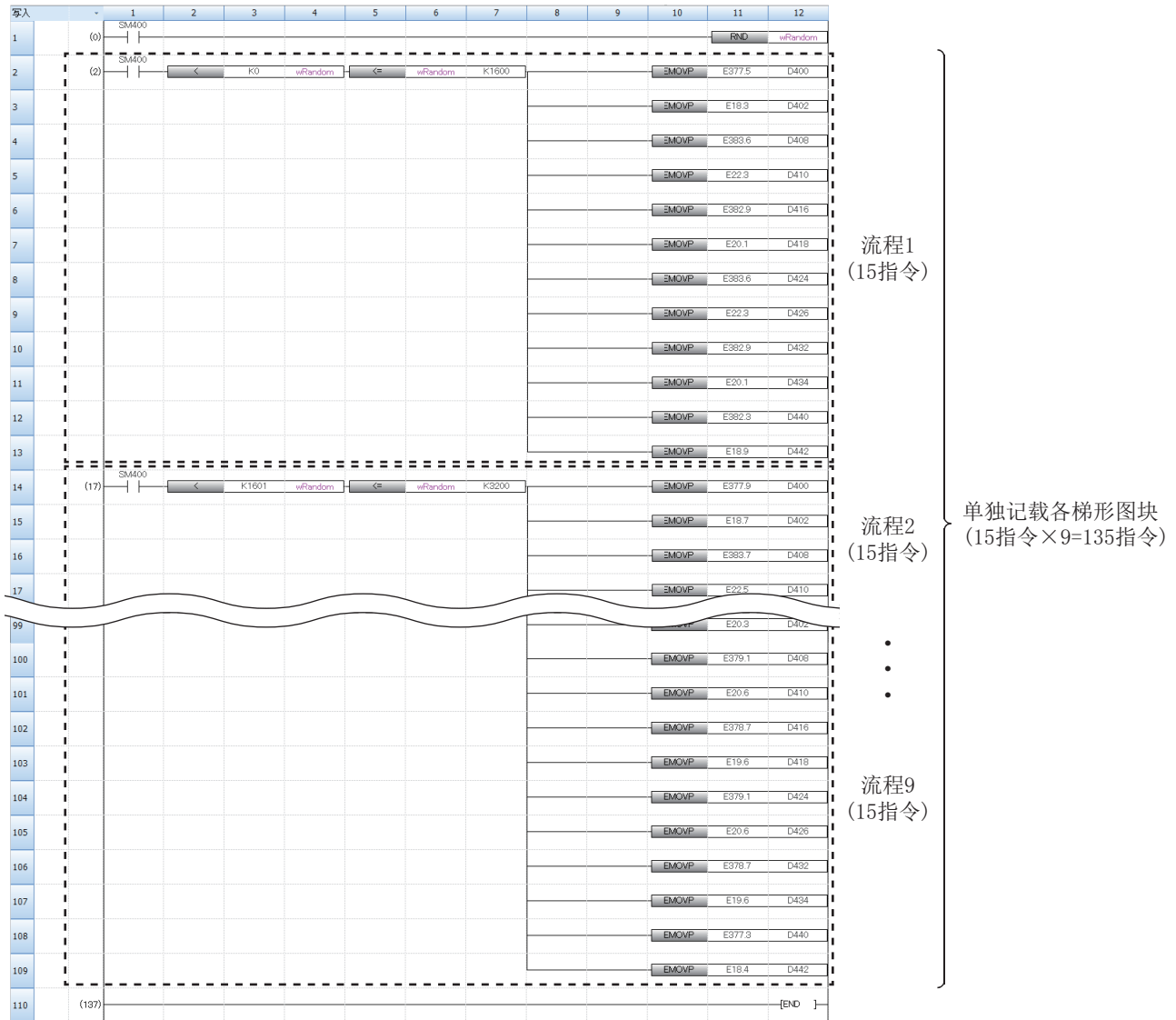
2. 修改流程1的触点的框线。



3. 按照与步骤1、2相同的步骤按流程划分梯形图块至流程9。



步骤1~3是1流程(15个指令)×9块的程序。



4.7 应用标签初始值

概要

对于在程序中不需要不断更改值的标签和FB的初始值，无需在程序中指定初始值，通过标签编辑器设置初始值即可缩短扫描时间。

应用标签初始值还有下述好处。

提高程序的可读性

可以省略在程序中指定初始值，从而提高程序的可读性。

执行步骤

以下述初始执行类型程序为例，说明从标签编辑器设置程序中指定的全局标签和FB的初始值的步骤。

- 初始执行类型程序

```
st ProgPou1 [PRG] [ST] x
1 //Label initialize
2 Label1 := 1;
3 Label2 := 1;
4
5 //FB initialize
6 Move1.Velocity := 1000;
7 Move1.Acceleration := 100;
8 Move1.Deceleration := 100;
```

指定Label1、Label2的初始值

指定FB“Move1”的初始值

- 全局标签

标签名	数据类型	类	分配软件/标签	初始值	常数
1 Label1	字[无符号]/位串[16位]	VAR_GLOBAL			
2 Label2	字[无符号]/位串[16位]	VAR_GLOBAL			
3 Move1	MC_MoveAbsolute	VAR_GLOBAL			

操作步骤

1. 打开全局标签编辑器，将“Label1”、“Label2”设置为与初始执行类型程序中指定的初始值相同的内容。

- Label1: 1
- Label2: 1

标签名	数据类型	类	分配软件/标签	初始值	常数
1 Label1	字[无符号]/位串[16位]	VAR_GLOBAL		1	
2 Label2	字[无符号]/位串[16位]	VAR_GLOBAL		1	
3 Move1	MC_MoveAbsolute	VAR_GLOBAL			

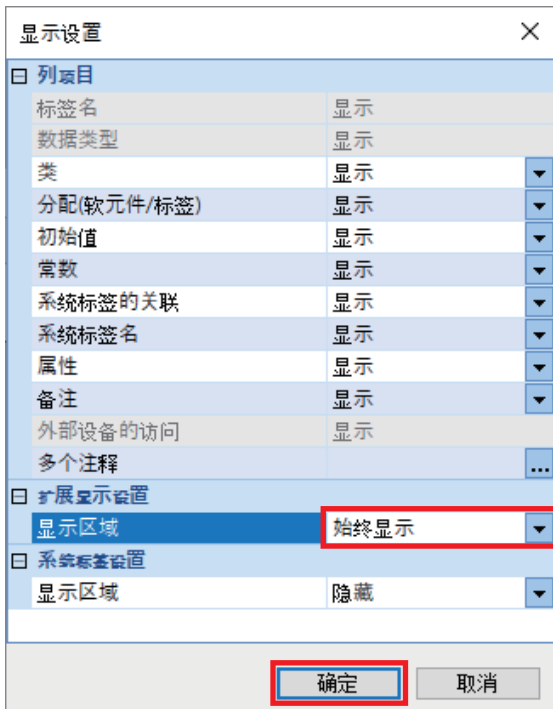
要点

如果不显示“初始值”列，点击[详细设置]按钮即可显示。

2. 点击[显示设置]按钮。

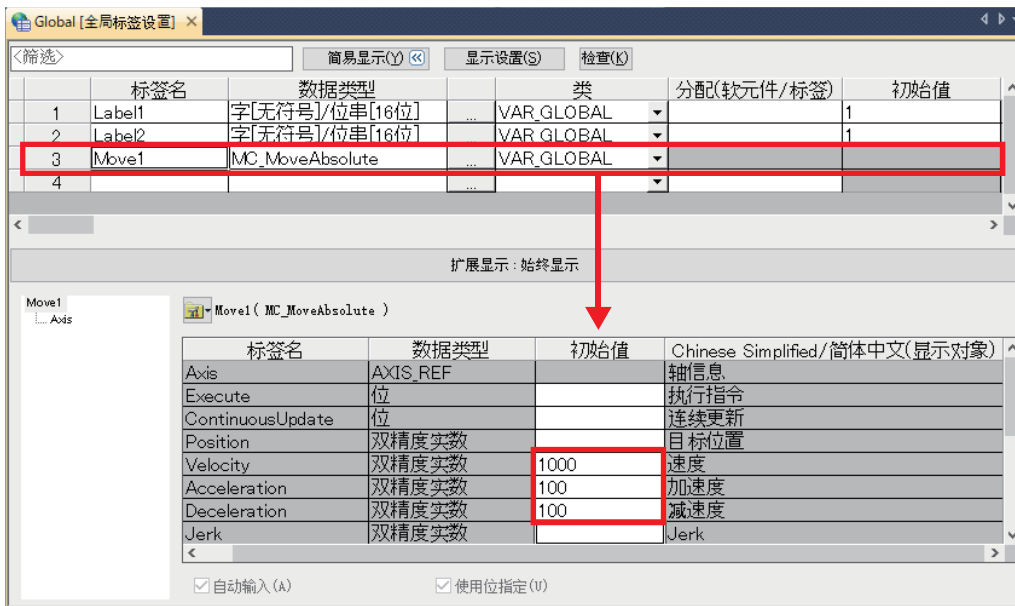
标签名	数据类型	类	分配软件/标签	初始值	常数
1 Label1	字[无符号]/位串[16位]	VAR_GLOBAL		1	
2 Label2	字[无符号]/位串[16位]	VAR_GLOBAL		1	
3 Move1	MC_MoveAbsolute	VAR_GLOBAL			

3. [扩展显示设置]⇒将[显示区域]更改为“始终显示”，点击[确定]按钮。

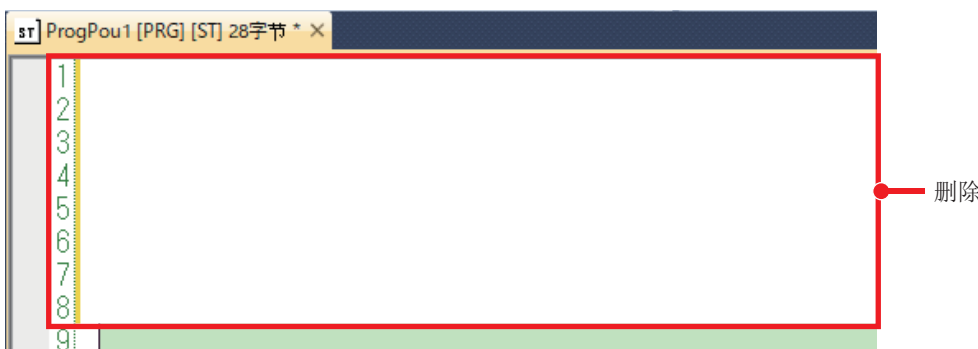


4. 选择全局标签“Move1”，将扩展显示的“初始值”列设置为与初始执行类型程序中指定的初始值相同的内容。

- Velocity: 1000
- Acceleration: 100
- Deceleration: 100



5. 打开指定初始值的初始执行类型程序，删除指定初始值的处理。



5 优化运动控制

通过加速多轴控制中特定的轴或同步用户程序和运动控制来优化运动控制。
本章介绍优化运动控制的技术。

技术一览

本章介绍的技术一览如下所示。

项目	内容	参照
使用通信周期混合提高特定轴速度	使用通信周期混合功能，只对多轴控制中需要高速控制的轴加速。	☞ 79页 使用通信周期混合提高特定轴速度
同步用户程序和运动运算	同步多核中用户程序和运动运算的周期，并同步变量的更新时机。	☞ 84页 同步用户程序和运动运算

5.1 使用通信周期混合提高特定轴速度

概要

多轴控制的运动程序中存在部分需要高速控制的轴时，如果使用通信周期混合功能为每个轴设置周期，则可以用较短的运算周期控制需要高速控制的轴。此外，在多轴配置中，可以调整整体的控制负载。

要点

该技术与高速CC-Link IE TSN技术结合使用时效果更好。关于高速CC-Link IE TSN，请参阅下述内容。

☞ 95页 高速CC-Link IE TSN通信

执行步骤

以下述轴配置为例，显示使用通信周期混合功能仅对特定轴加速的步骤。

台数	型号	站号	站类型	RX设置 点数	RX设置 点数	RW设置 点数	RW设置 点数	参数自动设置	PDO映射设置	IP地址	子网掩码	默认网关	保留/出错 无效站	网络同步通 信设置	通信周期 设置
0	本站	0	主站							192.168.4.253					
1	MR-J5-G	1	远程站			24	20	☐ <详细设置>	<详细设置>	192.168.4.1			无设置	同步	基本周期
2	MR-J5-G	2	远程站			24	20	☐ <详细设置>	<详细设置>	192.168.4.2			无设置	同步	基本周期
3	MR-J5-G	3	远程站			24	20	☐ <详细设置>	<详细设置>	192.168.4.3			无设置	同步	基本周期
4	MR-J5-G	4	远程站			24	20	☐ <详细设置>	<详细设置>	192.168.4.4			无设置	同步	基本周期
5	MR-J5-G	5	远程站			24	20	☐ <详细设置>	<详细设置>	192.168.4.5			无设置	同步	基本周期
6	MR-J5-G	6	远程站			24	20	☐ <详细设置>	<详细设置>	192.168.4.6			无设置	同步	基本周期
7	MR-J5-G	7	远程站			24	20	☐ <详细设置>	<详细设置>	192.168.4.7			无设置	同步	基本周期
8	MR-J5-G	8	远程站			24	20	☐ <详细设置>	<详细设置>	192.168.4.8			无设置	同步	基本周期
9	MR-J5-G	9	远程站			24	20	☐ <详细设置>	<详细设置>	192.168.4.9			无设置	同步	基本周期
10	MR-J5-G	10	远程站			24	20	☐ <详细设置>	<详细设置>	192.168.4.10			无设置	同步	基本周期
11	MR-J5-G	11	远程站			24	20	☐ <详细设置>	<详细设置>	192.168.4.11			无设置	同步	基本周期
41	MR-J5-G	41	远程站			24	20	☐ <详细设置>	<详细设置>	192.168.4.41			无设置	同步	基本周期
42	MR-J5-G	42	远程站			24	20	☐ <详细设置>	<详细设置>	192.168.4.42			无设置	同步	基本周期
43	MR-J5-G	43	远程站			24	20	☐ <详细设置>	<详细设置>	192.168.4.43			无设置	同步	基本周期
44	MR-J5-G	44	远程站			24	20	☐ <详细设置>	<详细设置>	192.168.4.44			无设置	同步	基本周期
45	MR-J5-G	45	远程站			24	20	☐ <详细设置>	<详细设置>	192.168.4.45			无设置	同步	基本周期
46	MR-J5-G	46	远程站			24	20	☐ <详细设置>	<详细设置>	192.168.4.46			无设置	同步	基本周期
47	MR-J5-G	47	远程站			24	20	☐ <详细设置>	<详细设置>	192.168.4.47			无设置	同步	基本周期
48	MR-J5-G	48	远程站			24	20	☐ <详细设置>	<详细设置>	192.168.4.48			无设置	同步	基本周期
49	MR-J5-G	49	远程站			24	20	☐ <详细设置>	<详细设置>	192.168.4.49			无设置	同步	基本周期
50	MR-J5-G	50	远程站			24	20	☐ <详细设置>	<详细设置>	192.168.4.50			无设置	同步	基本周期

通信周期为“基本周期”的
MR-J5-G 50台

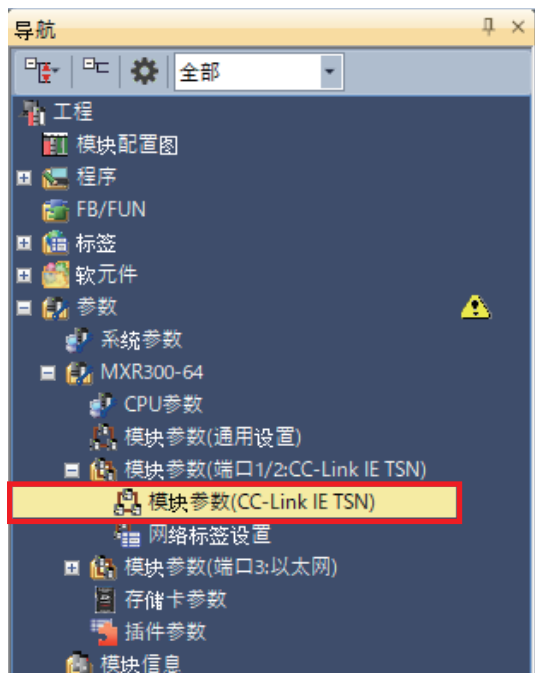
在示例中，MR-J5-G的通信周期设置如下。

站号	通信周期
1~5	基本周期
6~20	中速
21~50	低速

操作步骤

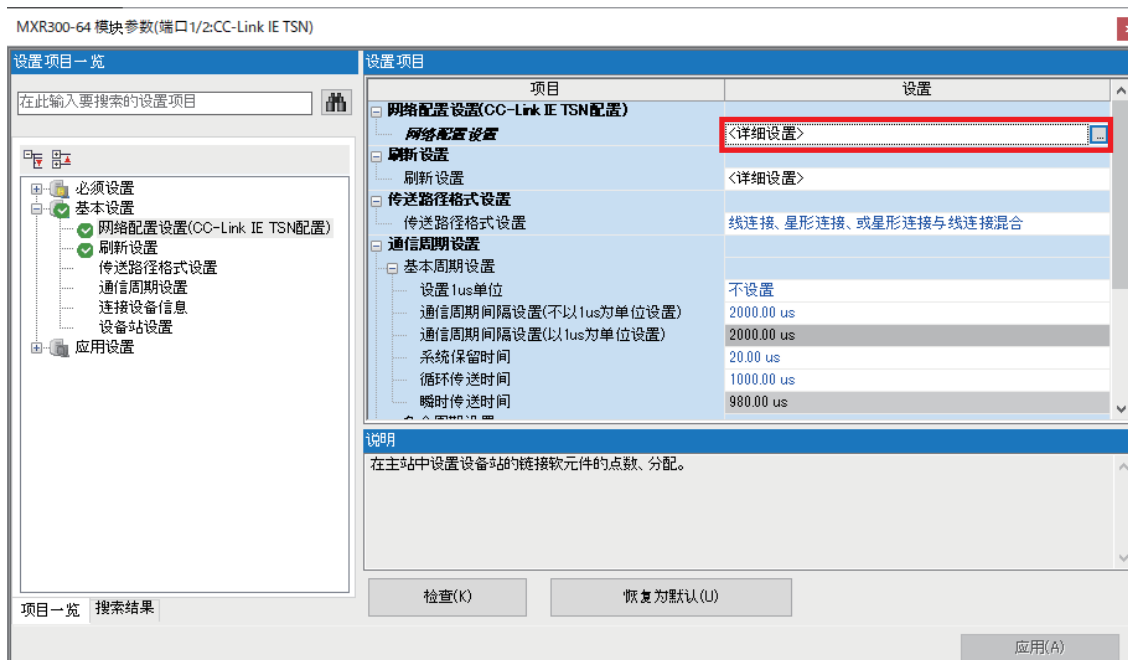
1. 打开模块参数(CC-Link IE TSN)。

双击导航窗口⇒[参数]⇒机型名⇒[模块参数(端口1/2:CC-Link IE TSN)]⇒[模块参数(CC-Link IE TSN)]

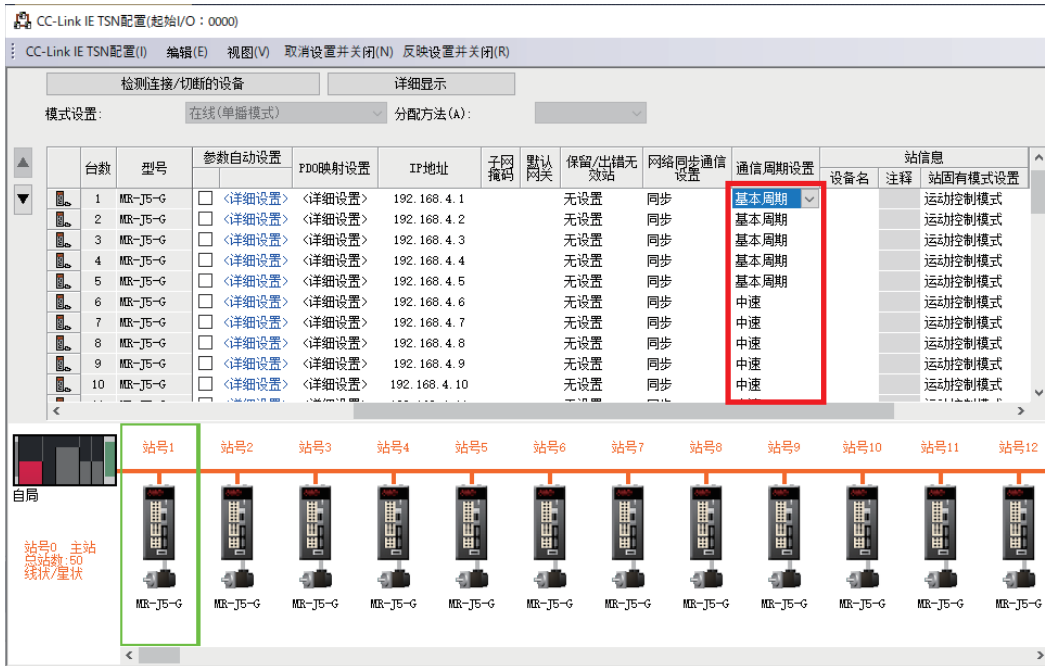


2. 在基本设置中设置网络配置设置。

双击[基本设置]⇒[网络配置设置(CC-Link IE TSN配置)]⇒[网络配置设置]的<详细设置>



3. 在“CC-Link IE TSN配置”画面的“通信周期设置”列中设置各轴的通信周期。



站号	通信周期
1~5	基本周期
6~20	中速
21~50	低速

要点

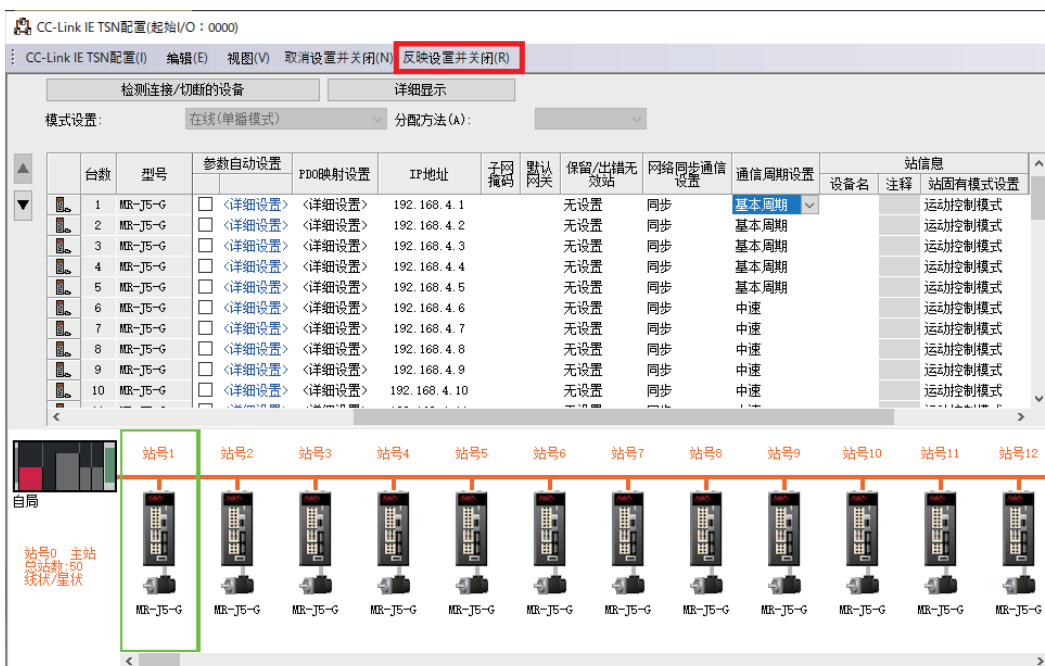
关于基本周期的恒定周期间隔及中速和低速的倍率，可以在“模块参数(CC-Link IE TSN)”的[基本设置]⇒[通信周期设置]中设置。

注意事项

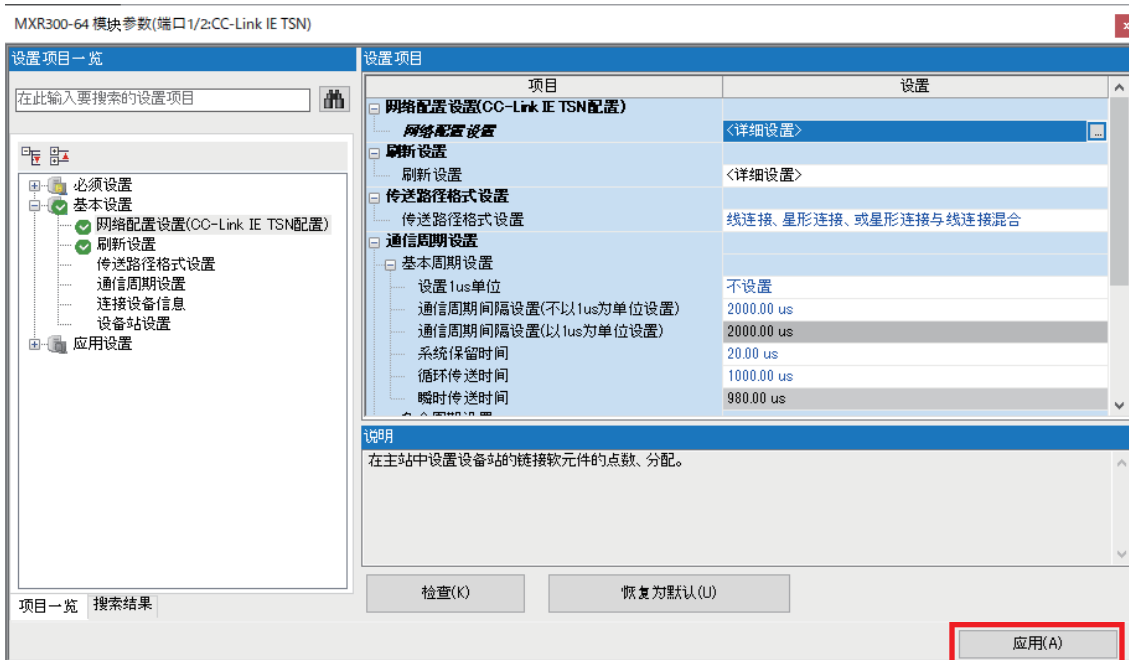
根据主站及远程站的通信速度，可能无法设置特定的通信周期。关于主站及远程站的通信速度设置，请参阅下述手册。

📖 MELSEC MX控制器(MX-R型)用户手册

4. 点击[反映设置并关闭]按钮。

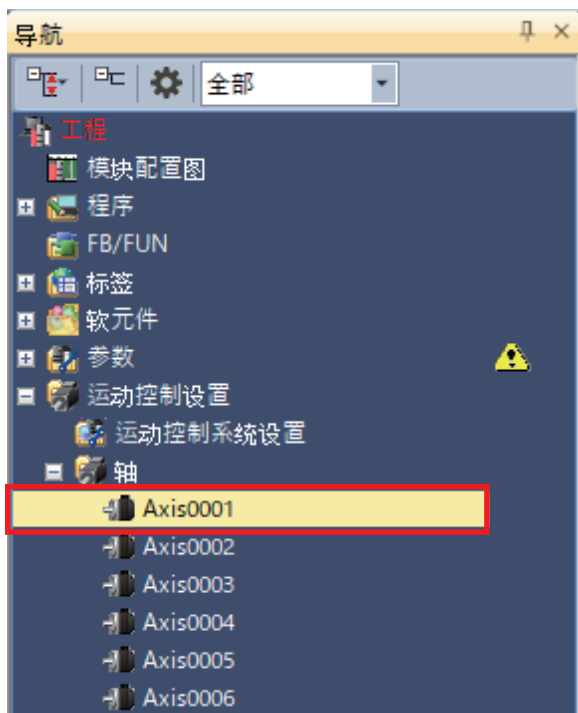


5. 点击[应用]按钮应用设置。



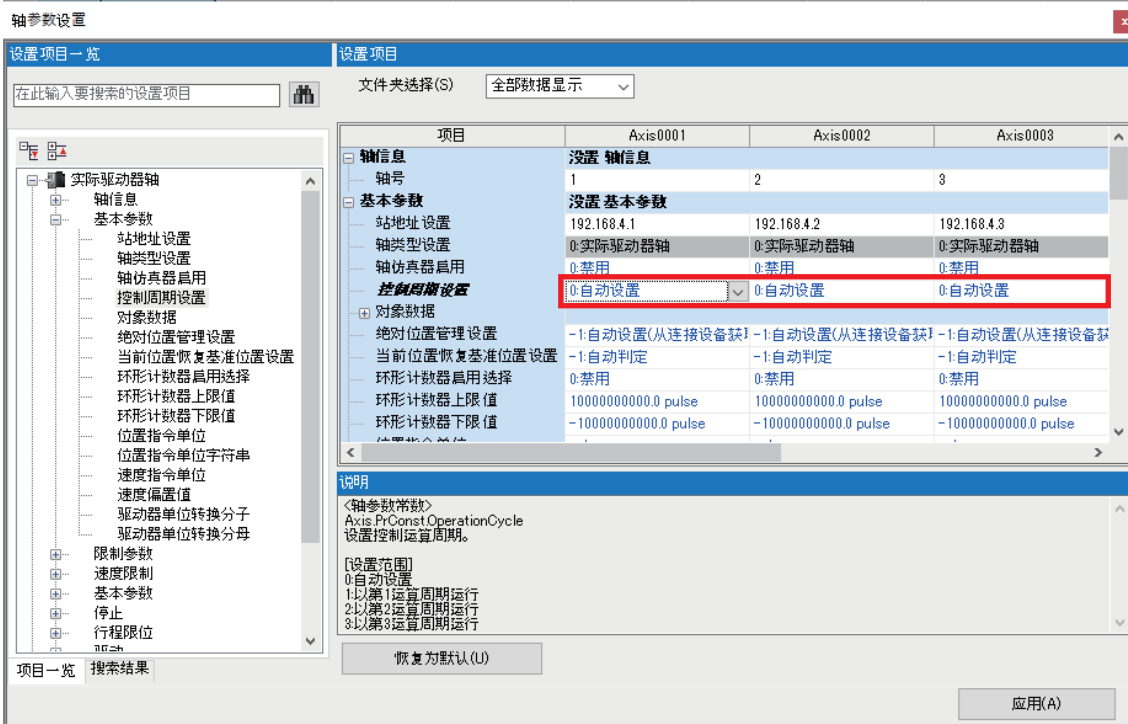
6. 设置各轴的参数。

🖱️ 导航窗口⇒[运动控制设置]⇒[轴]⇒双击任意轴

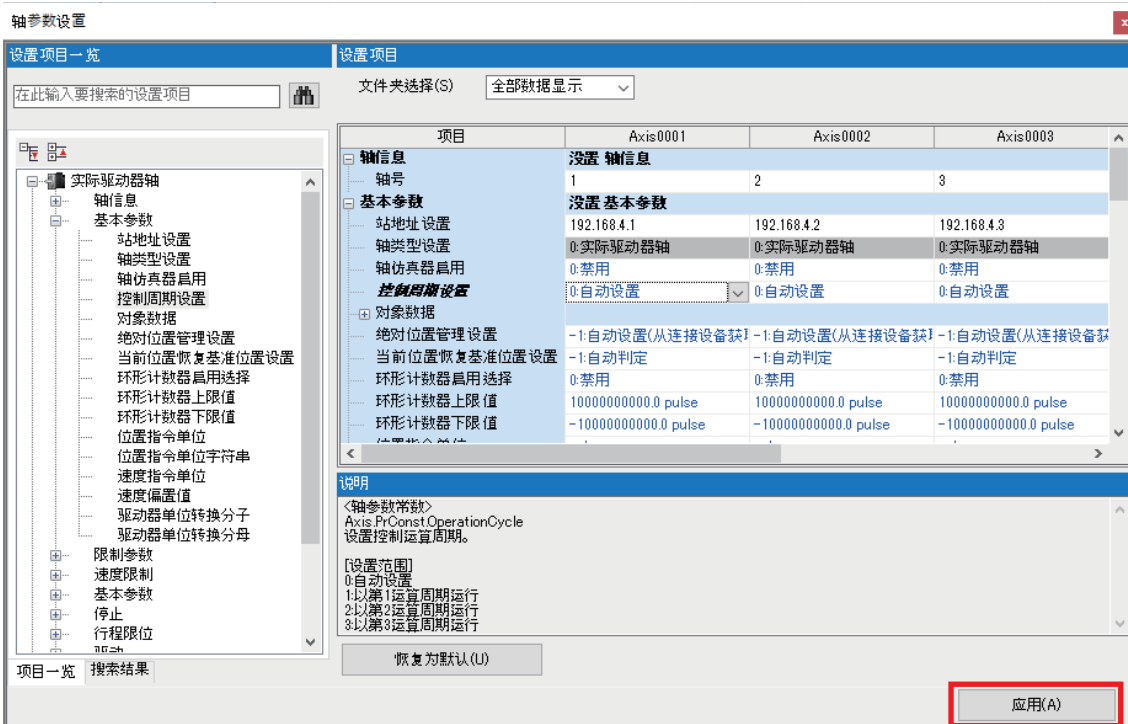


7. 将各轴的控制周期设置设置为“0:自动设置”（默认）。

设置的轴类型⇒[基本参数]⇒[控制周期设置]



8. 如果在步骤7更改了设置，请点击[应用]按钮应用设置。



5

注意事项

对不同周期的轴进行同步控制时会有限制。关于详细内容，请参阅下述手册。

MELSEC MX控制器(MX-R型)用户手册

5.2 同步用户程序和运动运算

概要

控制器在多核上并行执行用户程序和运动运算。因此，如果从执行周期与运动运算周期不同的用户程序访问运动运算中使用的变量(AxisName.Md.SetPosition等运动功能部分更新的变量)，则变量的更新时机可能会不固定。

如果变量的更新时机非常重要，应创建与运动运算的周期同步的程序，同步用户程序和运动运算使用的变量。

与运动运算的周期同步的程序

程序的执行周期以网络通信周期的起始为起点，按照程序→运动运算的执行顺序始终可与运动运算周期保持同步。

与运动运算的周期同步的网络通信周期、程序的组合如下所示。

运动运算周期	同步网络通信周期	同步程序		
		执行类型	触发类型	执行周期
第1运算周期	基本周期	事件	网络通信周期同步	基本周期
第2运算周期	中速周期			中速周期
第3运算周期	低速周期			低速周期

执行步骤

以运动运算中使用的变量“AxisName.Md.SetPosition”（指令当前位置）、“AxisName.Cd.VelocityOverride”（速度超驰系数）为例，将这些变量的更新时机与程序同步的步骤如下所示。

操作步骤

1. 打开任意的全局标签编辑器，定义用于保存变量“AxisName.Md.SetPosition”、“AxisName.Cd.VelocityOverride”的全局标签(以下简称临时标签)。

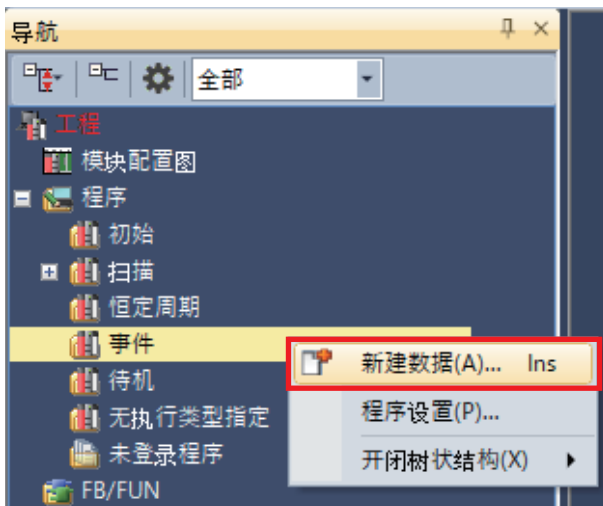
该步骤定义了以下临时标签。

No.	标签名	数据类型	类
1	temp_NormalSpeed_Read_Axis0001_SetPosition	双精度实数	VAR_GLOBAL
2	temp_NormalSpeed_Write_Axis0001_VelocityOverride	双精度实数	VAR_GLOBAL
3	temp_LowSpeed_Read_Axis0001_SetPosition	双精度实数	VAR_GLOBAL
4	temp_LowSpeed_Write_Axis0001_VelocityOverride	双精度实数	VAR_GLOBAL

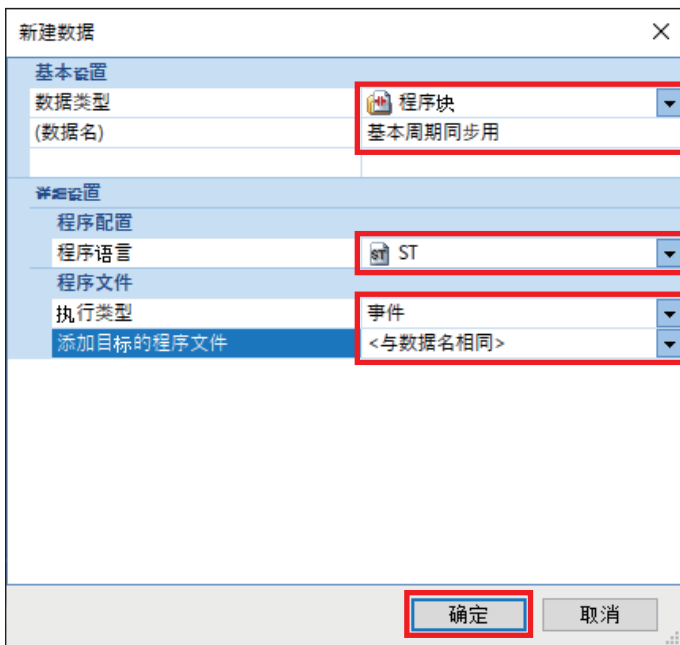
No.	标签名	数据类型	类	备注
1	temp_NormalSpeed_Read_Axis0001_SetPosition	双精度实数	VAR_GLOBAL	是用于访问中速周期的指令当前位置临时标签。
2	temp_NormalSpeed_Write_Axis0001_VelocityOverride	双精度实数	VAR_GLOBAL	是用于访问中速周期的速度超驰临时标签。
3	temp_LowSpeed_Read_Axis0001_SetPosition	双精度实数	VAR_GLOBAL	是用于访问低速周期的指令当前位置临时标签。
4	temp_LowSpeed_Write_Axis0001_VelocityOverride	双精度实数	VAR_GLOBAL	是用于访问低速周期的速度超驰临时标签。

2. 新建事件执行类型的程序块，使运动运算与基本周期同步。

🔍 导航窗口⇒[程序]⇒右击[事件]⇒[新建数据]



3. 设置各项目，点击[确定]按钮。

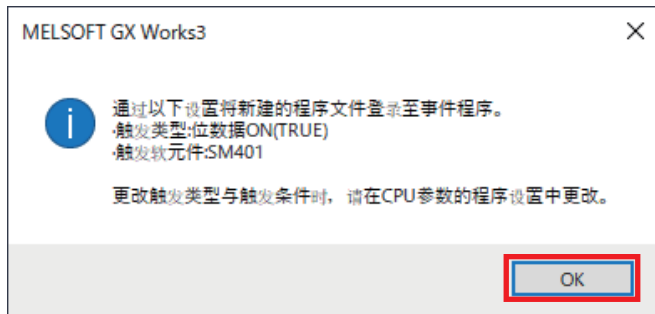


项目	设置内容
数据类型	程序块
(数据名)	基本周期同步用
程序语言	ST
执行类型	事件
添加目标的程序文件	<与数据名相同>

要点 🔍

新建程序文件和程序块时，要使程序文件与程序块名相同，在创建程序块时将“添加目标的程序文件”设置为“<与数据名相同>”即可同时创建同名的程序文件。这样就可以省略新建程序文件的步骤。

4. 点击[确定]按钮。



5. 打开在步骤3创建的程序块的程序本体，记述以下将变量与基本周期同步的处理。

```

1 //基本周期 读写用保存处理
2
3 //保存至中速(周期时间:基本周期的4倍)周期的临时标签
4 IF MotionSystem.Md.SystemBaseCycle_Counter MOD 4 = 1 THEN
5     //读取
6     temp_NormalSpeed_Read_Axis0001_SetPosition := Axis0001.Md.SetPosition;
7     //写入
8     Axis0001.Cd.VelocityOverride := temp_NormalSpeed_Write_Axis0001_VelocityOverride;
9 END_IF;
10
11 //保存至低速(周期时间:基本周期的16倍)周期的临时标签
12 IF MotionSystem.Md.SystemBaseCycle_Counter MOD 16 = 1 THEN
13     //读取
14     temp_LowSpeed_Read_Axis0001_SetPosition := Axis0001.Md.SetPosition;
15     //写入
16     Axis0001.Cd.VelocityOverride := temp_LowSpeed_Write_Axis0001_VelocityOverride;
17 END_IF;

```

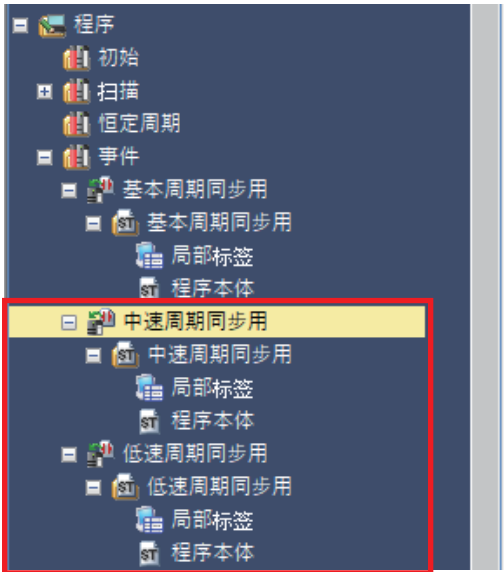
行数	处理内容
第3~9行	<p>从中速周期(基本周期的4倍)的程序访问第1个运算周期控制的轴的轴标签时，以系统基本周期计数器*1除以4的余数为1的周期执行保存和恢复处理。</p> <ul style="list-style-type: none"> • 读取时：变量“AxisName.Md.SetPosition”的值保存在临时标签中。 • 写入时：将临时标签的值恢复为变量“AxisName.Cd.VelocityOverride”。
第11~17行	<p>从低速周期(基本周期的16倍)的程序访问第1个运算周期控制的轴的轴标签时，以系统基本周期计数器*1除以16的余数为1的周期执行保存和恢复处理。</p> <ul style="list-style-type: none"> • 读取时：变量“AxisName.Md.SetPosition”的值保存在临时标签中。 • 写入时：将临时标签的值恢复为变量“AxisName.Cd.VelocityOverride”。

*1 系统基本周期计数器(System.Md.SystemBaseCycle_Counter)是以网络的基本周期间隔递增1的计数器，当周期超过各运动运算周期的最小公倍数时返回1。通过该计数器可以确认各运动运算周期处理的执行时机。

关于详细内容，请参阅下述手册。

MELSEC MX控制器(MX-R型)用户手册

6. 参照步骤2~4，以相同方式创建与中速周期和低速周期同步的事件执行类型程序块。



7. 打开在步骤6创建的与中速周期和低速周期同步的事件执行类型程序块，记述以下处理，根据变量“AxisName.Md.SetPosition”的值设置变量“AxisName.Cd.VelocityOverride”的值。

- 与中速周期同步的事件执行类型程序块

```

[ST] 中速周期同步用 [PRG] [ST] * X
1 //在第1运算周期中运行的轴Axis0001的Md.SetPosition为1000.0以下
2 //将第1运算周期中运行的轴Axis0001的Cd.VelocityOverride设置为10.0
3 IF temp_NormalSpeed_Read_Axis0001_SetPosition <= 1000.0 THEN
4     temp_NormalSpeed_Write_Axis0001_VelocityOverride := 10.0;
5 END_IF;
  
```

- 与低速周期同步的事件执行类型程序块

```

[ST] 低速周期同步用 [PRG] [ST] * X
1 //在第1运算周期中运行的轴Axis0001的Md.SetPosition超过1000.0
2 //将第1运算周期中运行的轴Axis0001的Cd.VelocityOverride设置为5.0
3 IF temp_NormalSpeed_Read_Axis0001_SetPosition > 1000.0 THEN
4     temp_NormalSpeed_Write_Axis0001_VelocityOverride := 5.0;
5 END_IF;
  
```

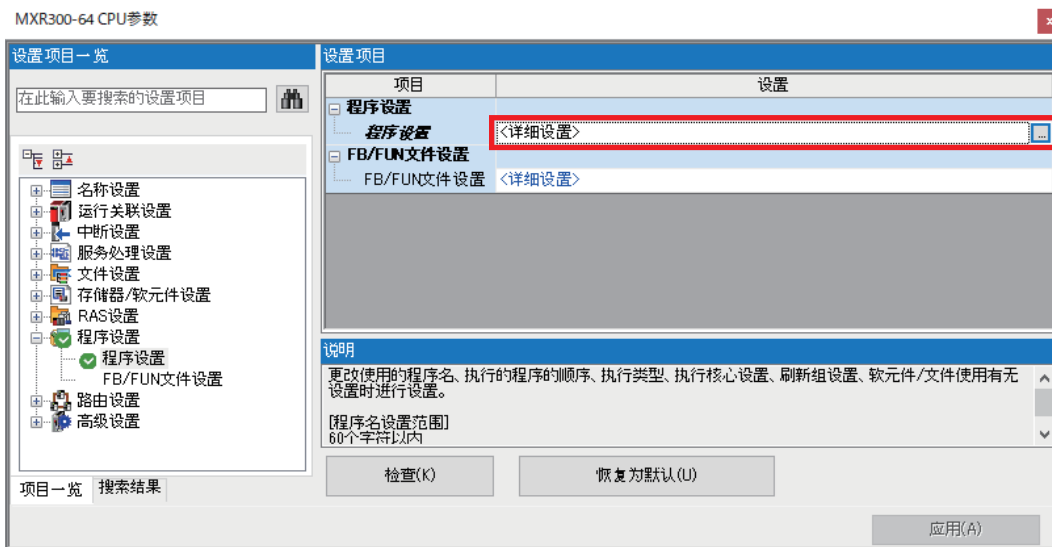
8. 打开CPU参数。

双击导航窗口⇒[参数]⇒机型名⇒[CPU参数]

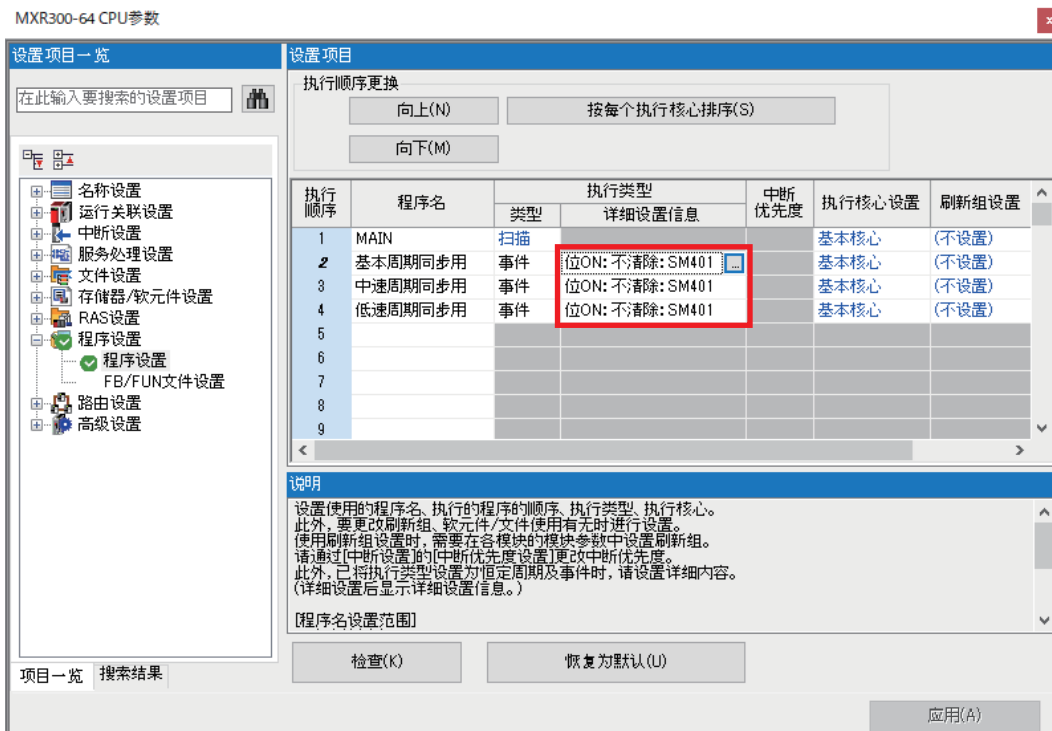


9. 要更改事件执行类型程序的触发设置，打开程序设置。

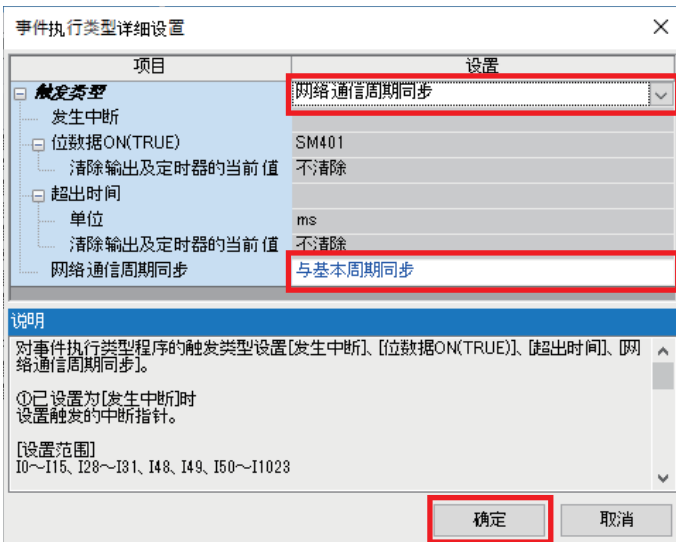
双击[程序设置]⇒[程序设置]的<详细设置>



10. 双击在该步骤创建的事件执行类型程序的“详细设置信息”列。

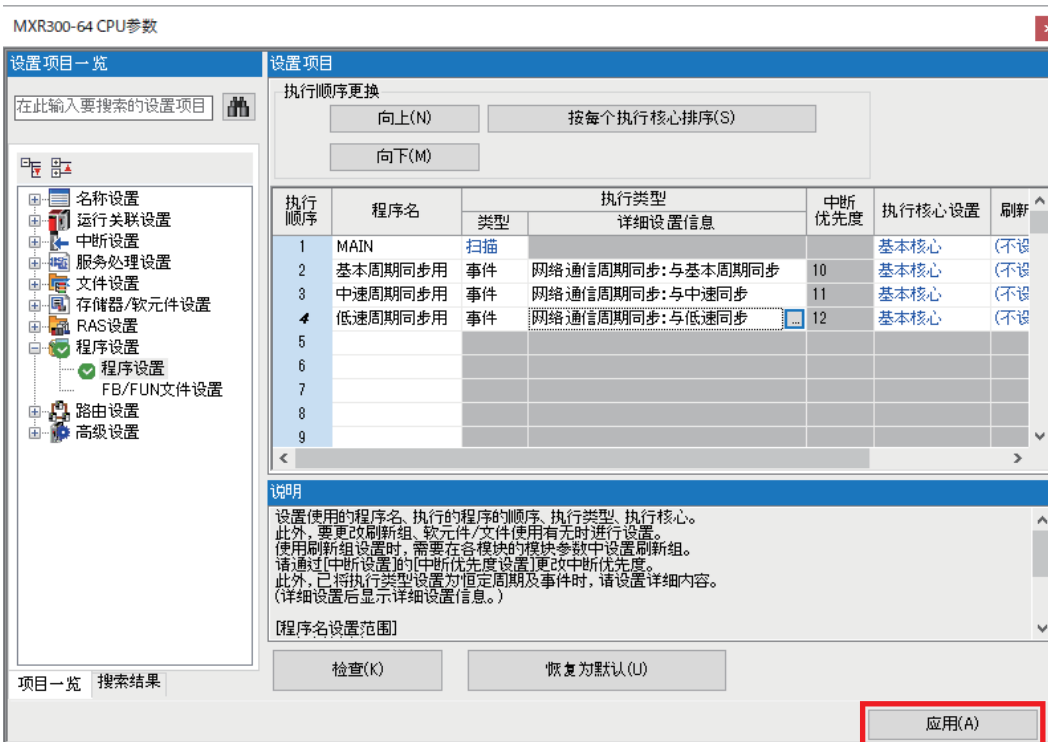


11. 根据程序内容如下设置“触发类型”和“网络通信周期同步”，然后点击[确定]按钮。



程序	触发类型	网络通信周期同步
与基本周期同步的事件执行类型程序块 (该步骤中为“基本周期同步用”)	网络通信周期同步	与基本周期同步
与中速周期同步的事件执行类型程序块 (该步骤中为“中速周期同步用”)		与中速同步
与低速周期同步的事件执行类型程序块 (该步骤中为“低速周期同步用”)		与低速同步

12. 点击[应用]按钮应用设置。



注意事项

- 从优先度高于保存和恢复临时标签的程序的程序访问保存目标的临时标签时，请使用DI指令/EI指令在中断禁止状态下进行保存和恢复处理。

```
st ProgPou1 [PRG] [ST] 24字节 ×
1 //读写用保存处理
2 //中断禁止
3 DI(TRUE);
4 //保存至低速(周期时间:基本周期的16倍)周期的临时标签
5 IF MotionSystem.Md.SystemBaseCycle_Counter MOD 16 = 1 THEN
6 //读取
7 temp_LowSpeed_Read_Axis0001_SetPosition := Axis0001.Md.SetPosition;
8 //写入
9 Axis0001.Cd.VelocityOverride := temp_LowSpeed_Write_Axis0001_VelocityOverride;
10 END IF;
11 //解除中断禁止状态
12 EI(TRUE);
13
```

- 由于该技术为临时标签添加了保存和恢复处理，这可能会导致程序复杂化并延长扫描时间。

6 标签效率

通过将不需要作为全局标签的标签更改为局部标签，或删除不需要的标签，从而提高标签的效率。提高标签效率也会提高程序整体的可读性。

本章介绍提高标签效率的技术。

技术一览

本章介绍的技术一览如下所示。

项目	内容	参照
减少全局标签	将不需要作为全局标签的标签更改为局部标签。	☞ 91页 减少全局标签
删除未使用标签	搜索工程内未使用的标签并批量删除。	☞ 93页 删除未使用标签

6.1 减少全局标签

概要

全局标签是需要多个程序中使用标签或要为标签分配软件元件时使用的标签。可以将其他情况下使用的全局标签更改为局部标签以减少全局标签。

减少全局标签还有下述好处。

降低程序发生故障的风险

可将标签对程序的影响范围局部化，从而降低发生故障的风险。

缩短程序转换时间

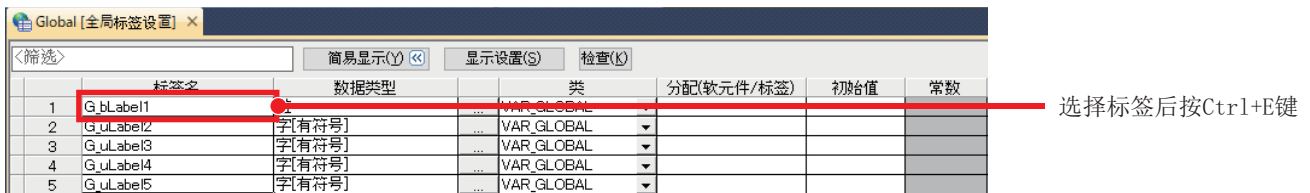
可将标签对程序的影响范围局部化，从而降低程序转换时的影响，缩短程序转换时间。

执行步骤

搜索并删除不必要的全局标签，将其更改为局部标签的步骤如下所示。

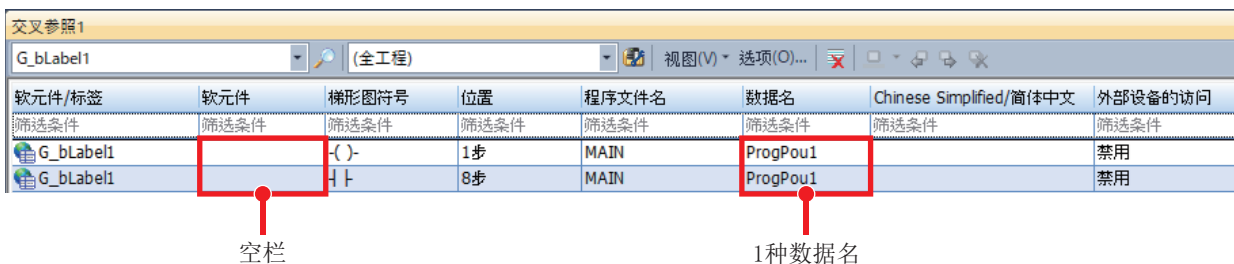
操作步骤

1. 使用交叉参照确认各全局标签的使用状况。
打开全局标签，选择要确认的标签后按Ctrl+E键。



2. 显示交叉参照结果。

可以判断“软件元件”列为空栏且“数据名”列只显示一种数据名的标签不需要全局标签。



要点

双击参照表结果中的全局标签可以确认使用该标签的程序和位置。请在删除全局标签前进行确认。

3. 从全局标签编辑器中剪切要删除的全局标签。

点击编辑器的表的行编号，选择整行后剪切。



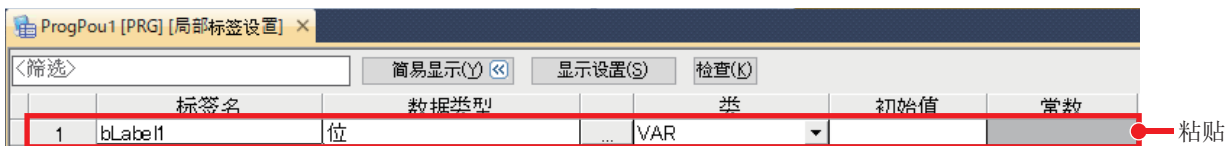
	标签名	数据类型	类	分配(软元件/标签)	初始值	常数
1	G_bLabel1	位	VAR_GLOBAL			
2	G_uLabel2	字[有符号]	VAR_GLOBAL			
3	G_uLabel3	字[有符号]	VAR_GLOBAL			
4	G_uLabel4	字[有符号]	VAR_GLOBAL			
5	G_uLabel5	字[有符号]	VAR_GLOBAL			

要点

如果未选择整行，则无法复制标签编辑器上隐藏的内容。

4. 打开使用要删除的全局标签的程序的局部标签，粘贴步骤3中剪切的标签。


粘贴后，设置任意类。如果已为局部标签和全局标签决定了命名规则，则根据需要更改标签名。



	标签名	数据类型	类	初始值	常数
1	bLabel1	位	VAR		

要点

- 将复制内容从全局标签粘贴至局部标签时，不会粘贴类的设置。请重新设置任意类。
- 程序中使用的标签名将根据步骤1~4的更改自动更改。因此，无需修改程序。如果未变更，可以从菜单的[工具]⇒[选项]⇒“其他编辑器”⇒“标签编辑器通用”⇒“运行设置”⇒“将标签名自动跟踪至程序编辑器”设置。关于详细内容，请参阅下述手册。

 GX Works3 操作手册

5. 对其他全局标签也按相同方式执行步骤1~4。

注意事项

要删除的全局标签(移动至局部标签的标签)无法从外部设备访问或在其他程序中使用。删除全局标签前，请确认使用该标签的程序内容，判断是否可以删除。

6.2 删除未使用标签

概要

GX Works3交叉参照表功能可以搜索并删除程序中未使用的标签，从而提高标签的效率。
删除未使用标签还有下述好处。

增加可使用标签量

删除未使用标签，可以增加可使用的标签容量。

缩短程序的搜索时间

删除未使用标签，在程序中搜索标签时可以减少搜索对象，从而缩短搜索时间。

缩短程序转换时间

删除未使用标签可以减少标签的数据量，从而缩短程序转换时间。

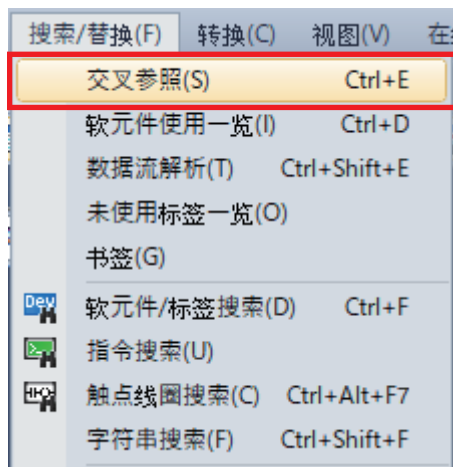
执行步骤

删除未使用标签的步骤如下所示。

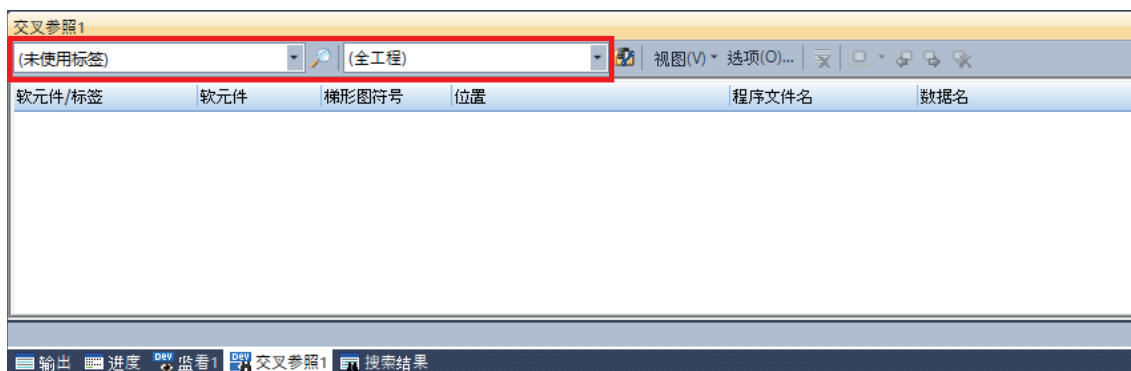
操作步骤

1. 打开搜索未使用标签的交叉参照。

☞ 菜单的[搜索/替换]⇒[交叉参照]



2. 将搜索范围设置为“(未使用标签)”、“(全工程)”，点击[🔍]按钮。



3. 将显示程序中的未使用标签一览。

按住Ctrl键选择多个要删除的标签，并右击选择[删除标签]删除标签。



要点

- 也可以按Ctrl+A键选择所有标签后批量删除，但建议逐一确认内容后再删除。
- 删除标签后可以再次搜索未使用标签，确认未使用标签已被删除。

7 高速CC-Link IE TSN通信

通过修改设备站的链接刷新设置和通信周期设置等缩短链接刷新时间、通信周期间隔及循环传送时间，可以加速CC-Link IE TSN通信。高速CC-Link IE TSN通信可以实现高速控制，从而缩短装置的指挥时间。

本章介绍高速CC-Link IE TSN通信的技术。

技术一览

本章介绍的技术一览如下所示。

项目	内容	参照	
缩短刷新时间	重新分配链接软元件编号，确保刷新设置的软元件范围连续。	☞ 96页 缩短刷新时间	
缩短通信周期间隔/ 循环传送时间	均衡设备站的台数	均衡连接主站各端口的设备站台数。	☞ 103页 均衡设备站的台数
	设置发送上限大小	如果CC-Link IE TSN Class为“Class A”且通信周期设置为“低速”的设备站，则将CC-Link IE TSN Class A组的发送上限大小设置为较小的值。	☞ 104页 设置发送上限大小
	减少设备站台数	如果有不需要的设备站，请减少连接主站的设备站台数。	☞ 106页 减少设备站台数
	删除链接软元件设置	删除各设备站不需要的链接软元件设置。	☞ 110页 删除链接软元件设置
	减少链接软元件设置点数	减少各设备站的链接软元件设置的点数。	☞ 115页 减少链接软元件设置点数
	统一网络同步通信设置	在CC-Link IE TSN Class为“Class B”的设备站中，如果网络同步通信设置在“不同步”与“同步”之间混合，请统一其中一个。	☞ 120页 统一网络同步通信设置
	设置循环传送分散	如果CC-Link IE TSN Class为“Class B”且通信周期设置为“中速”或“低速”的设备站，请设置循环传送分散。	☞ 123页 设置循环传送分散

7

通信周期间隔和循环传送时间的确认方法

通信周期间隔的实测值和循环传送时间的计算值可以从下述缓冲存储器中确认。

请用于确认执行各技术时的通信周期间隔和循环传送时间。

- 通信周期间隔(实测值)：(U3E0\G10200~U3E0\G10201)
- 循环传送时间(计算值)：(U3E0\G10272~U3E0\G10273)

由于通信周期间隔是实测值，因此如果复位控制器，显示的值可能会稍微增加或减少。

7.1 缩短刷新时间

概要

通过“通信周期间隔缩短分配”重新分配网络配置设置中设置的设备站的RX设置/RX设置、RW_r设置/RW_w设置的软元件编号，可以连续设置刷新目标的软元件范围。这样可以减少刷新设置的设置数，从而简化刷新设置。此外，还可以缩短链接刷新时间、通信周期间隔及循环传送时间，实现高速CC-Link IE TSN通信。

通信周期间隔缩短分配

网络配置设置的软元件编号重新分配功能可以将软元件编号连续分配给指定的对象站的链接软元件。

如果选择“通信周期间隔缩短分配”作为软元件编号重新分配功能的分配方法，则会将缩短循环接收处理时间的软元件编号分配给所有站。

在通信周期间隔缩短分配中，将网络配置设置中设置的设备站的RX设置、RY设置、RW_r设置和RW_w设置的软元件编号按以下顺序连续分配。(各设置中的软元件编号按照站号顺序分配)

分配顺序	网络同步通信设置	通信周期设置
1	不同步	基本周期
2	不同步	中速
3	不同步	低速
4	同步	基本周期
5	同步	中速
6	同步	低速

关于通信周期间隔缩短分配的详细内容，请参阅下述手册。

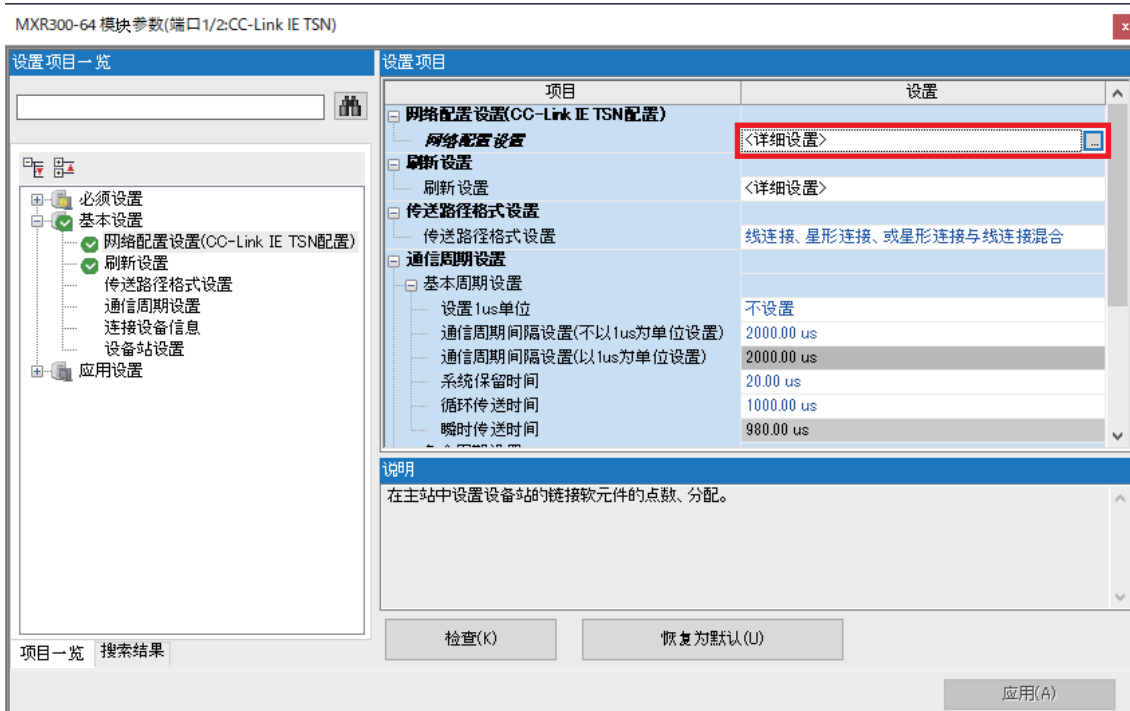
📖 MELSEC MX控制器(MX-R型)用户手册

注意事项

- 如果在已为远程站的链接软元件分配了软元件编号时进行通信周期间隔缩短分配，则现有链接软元件编号的顺序会被更改。
- 存在网络同步通信设置设置为“同步”的设备站时，通信周期间隔缩短分配有效。

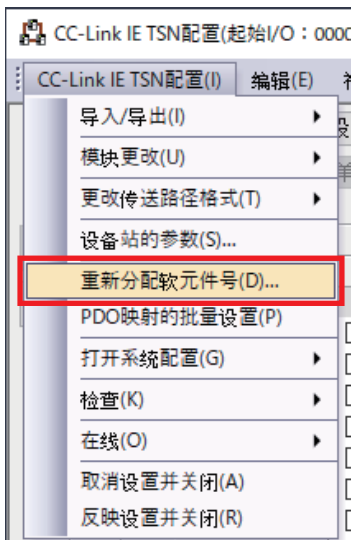
2. 在基本设置中打开网络配置设置。

双击[基本设置]⇒[网络配置设置(CC-Link IE TSN配置)]⇒[网络配置设置]的<详细设置>



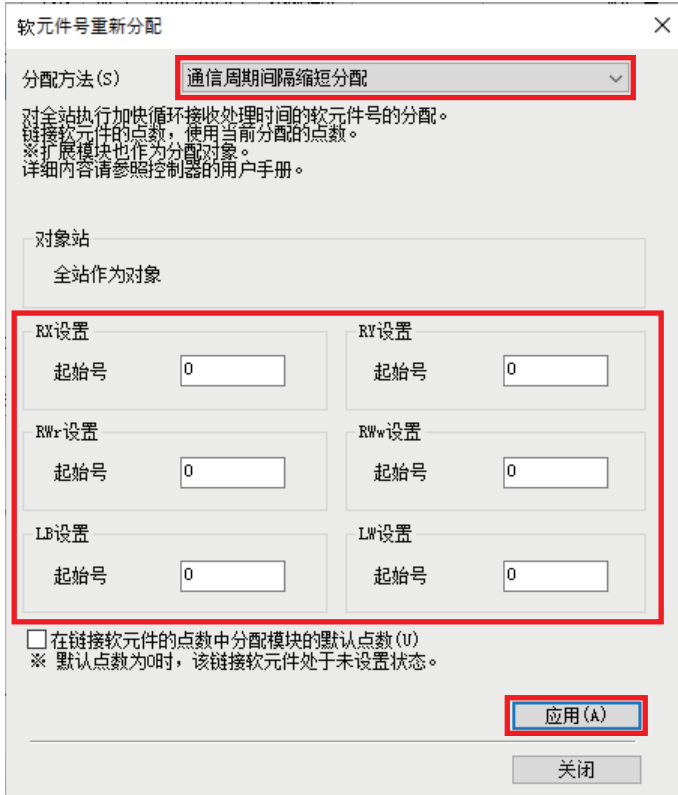
3. 重新分配软元件编号。

点击菜单的[CC-Link IE TSN配置]⇒[重新分配软元件]



4. 为分配方法选择“通信周期间隔缩短分配”，设置各链接软元件的起始编号点击[应用]按钮。

- 各链接软元件编号的起始编号：0

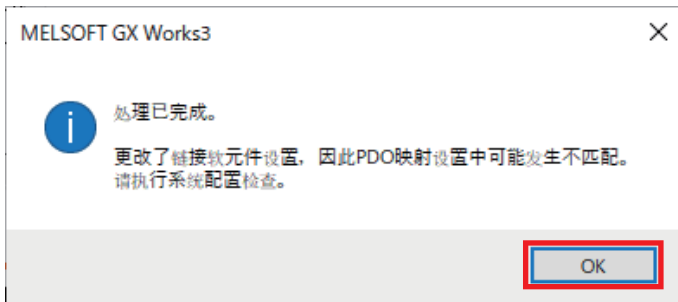


注意事项

如果已为各链接软元件分配了编号，则会更改编号顺序。关于软元件的分配顺序，请参阅下述章节。

☞ 96页 通信周期间隔缩短分配

5. 点击[OK]按钮。



软元件编号被重新分配。

CC-Link IE TSN配置(起始I/O : 0000)

CC-Link IE TSN配置(0) 编辑(E) 视图(V) 取消设置并关闭(N) 反映设置并关闭(R)

检测连接/切除的设备 简易显示

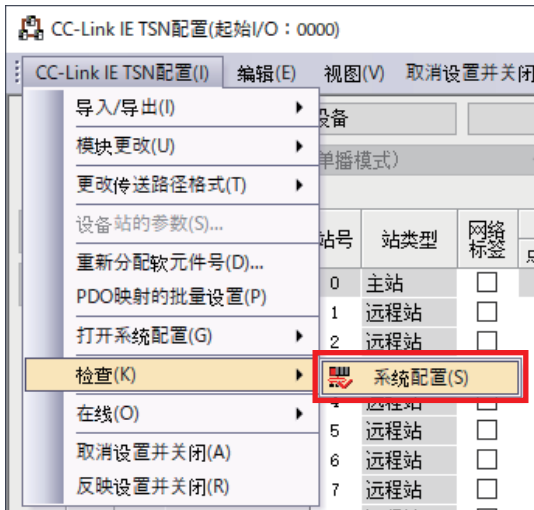
模式设置: 在线(单播模式) 分配方法(A): 点数/起始

台数	型号	站号	站类型	网络标签	RX设置			RY设置			RW-设置			RW+设置			LB设置			LW设置			参数自动设置	默认网关	保留/出缺无效站	网络同步通信设置	通信周期设置	
					点数	起始	结束	点数	起始	结束	点数	起始	结束	点数	起始	结束	点数	起始	结束	点数	起始	结束						点数
0	本站	0	主站																									
1	NZ2GR12A4-16D	1	远程站		16	0050	005F	16	0050	005F	4	0014	0017	4	0014	0017												
2	NZ2GR12A4-16D	2	远程站		16	0030	003F	16	0030	003F	4	000C	000F	4	000C	000F												
3	NZ2GR12A4-16D	3	远程站		16	0060	006F	16	0060	006F	4	0018	001B	4	0018	001B												
4	NZ2GR12A4-16D	4	远程站		16	0070	007F	16	0070	007F	4	001C	001F	4	001C	001F												
5	NZ2GR12A4-16D	5	远程站		16	0020	002F	16	0020	002F	4	0008	000B	4	0008	000B												
6	NZ2GR12A4-16D	6	远程站		16	00A0	00AF	16	00A0	00AF	4	0028	002B	4	0028	002B												
7	NZ2GR12A4-16D	7	远程站		16	0040	004F	16	0040	004F	4	0010	0013	4	0010	0013												
8	NZ2GR12A4-16D	8	远程站		16	0000	000F	16	0000	000F	4	0000	0003	4	0000	0003												
9	NZ2GR12A4-16D	9	远程站		16	0080	008F	16	0080	008F	4	0020	0023	4	0020	0023												
10	NZ2GR12A4-16D	10	远程站		16	0010	001F	16	0010	001F	4	0004	0007	4	0004	0007												
11	NZ2GR12A4-16D	11	远程站		16	0090	009F	16	0090	009F	4	0024	0027	4	0024	0027												
12	NZ2GR12A4-16D	12	远程站		16	00B0	00BF	16	00B0	00BF	4	002C	002F	4	002C	002F												

6. 进行系统配置检查。

发生出错或警告时，请检查配置和设置。

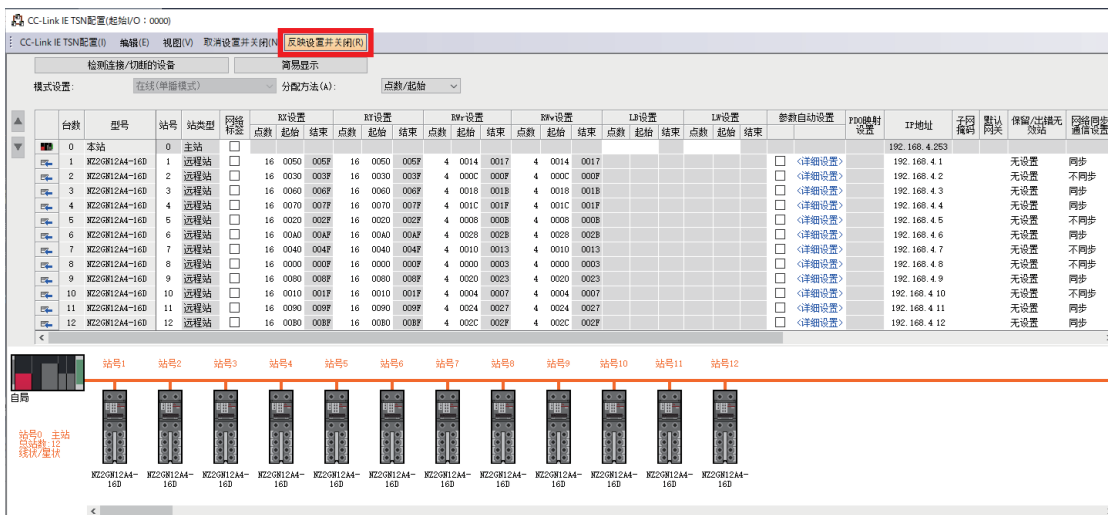
点击菜单的[CC-Link IE TSN配置]⇒[检查]⇒[系统配置]



要点

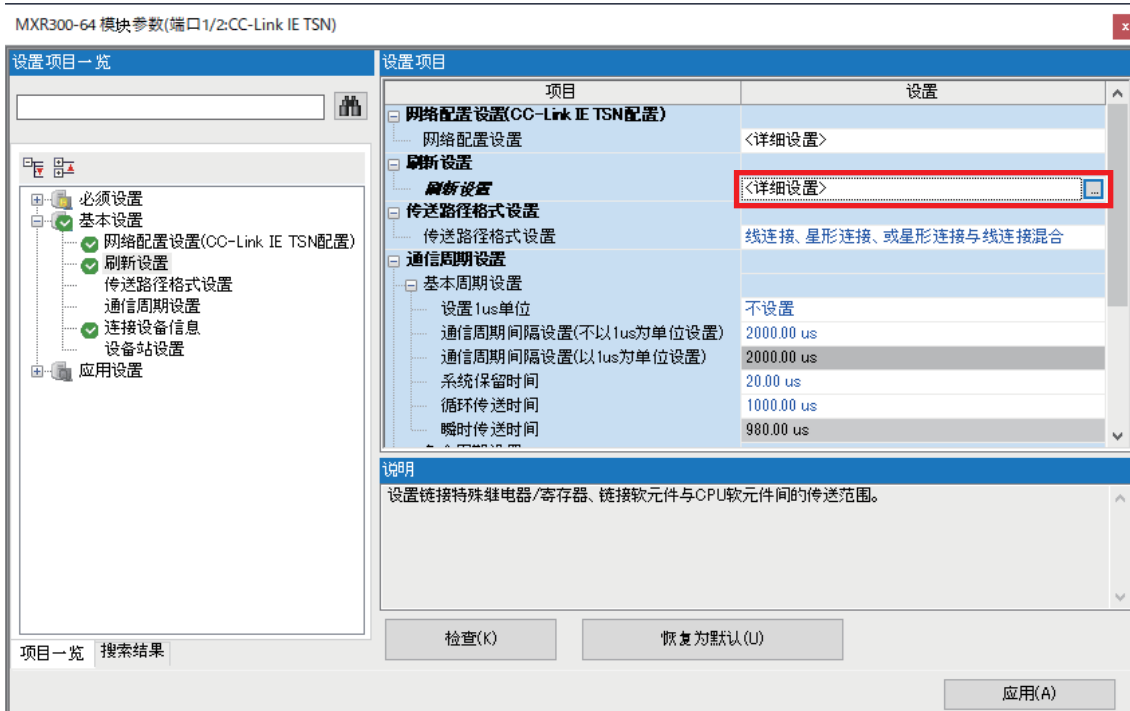
如果发生出错或警告，会显示在“输出”折叠窗口中。

7. 点击[反映设置并关闭]按钮。

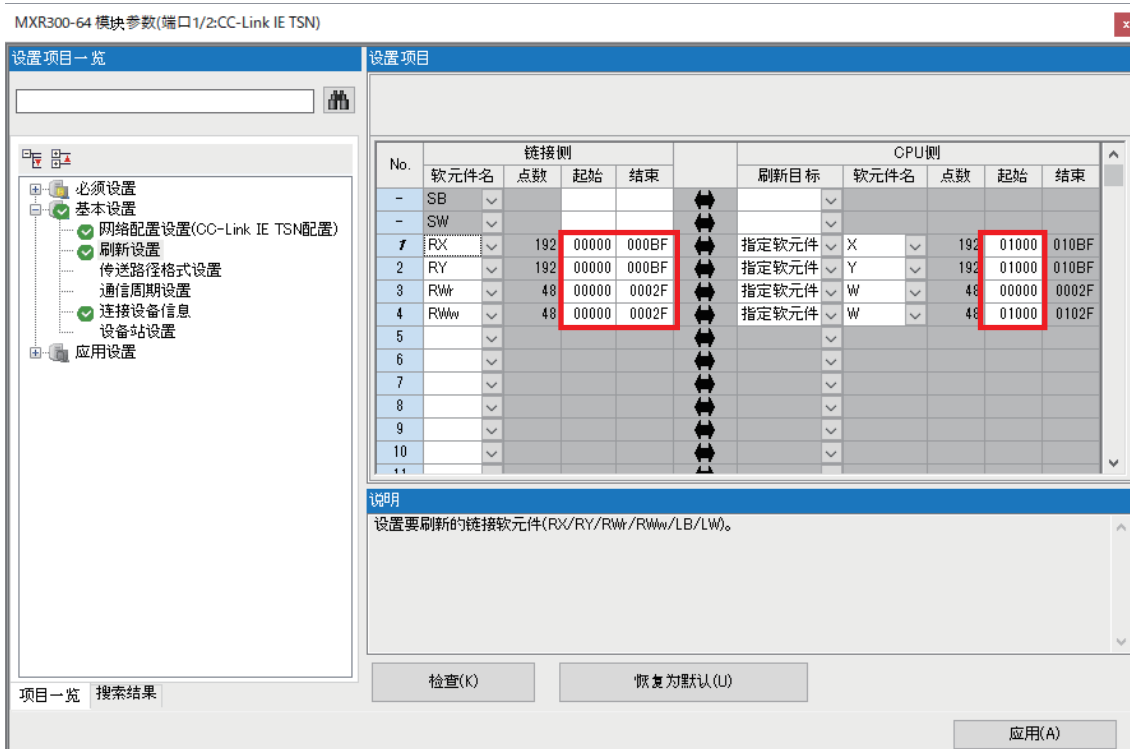


8. 结合刷新设置设置刷新。

双击[基本设置]⇒[刷新设置]⇒[刷新设置]的<详细设置>



9. 通过链接软元件的重新分配，确认链接侧软元件的起始编号和最终编号是否被更改，根据需要进行更改。如果链接侧的设置已更改，请同时检查CPU侧的设置。



10. 点击[应用]按钮应用设置。

MXR300-64 模块参数(端口1/2:CC-Link IE TSN)

设置项目一览

- 必须设置
- 基本设置
- 网络配置设置(CC-Link IE TSN配置)
- 刷新设置
- 传送路径格式设置
- 通信周期设置
- 连接设备信息
- 设备站设置
- 应用设置

设置项目

No.	链接侧					CPU侧				
	软元件名	点数	起始	结束		刷新目标	软元件名	点数	起始	结束
-	SB									
-	SW									
1	RX	192	00000	000BF	指定软元件	X	192	01000	010BF	
2	RY	192	00000	000BF	指定软元件	Y	192	01000	010BF	
3	RWw	48	00000	0002F	指定软元件	W	48	00000	0002F	
4	RWw	48	00000	0002F	指定软元件	W	48	01000	0102F	
5										
6										
7										
8										
9										
10										
11										

说明

设置要刷新的链接软元件(RX/RX/RWw/RWw/LB/LW)。

检查(K)
恢复为默认(L)

应用(A)

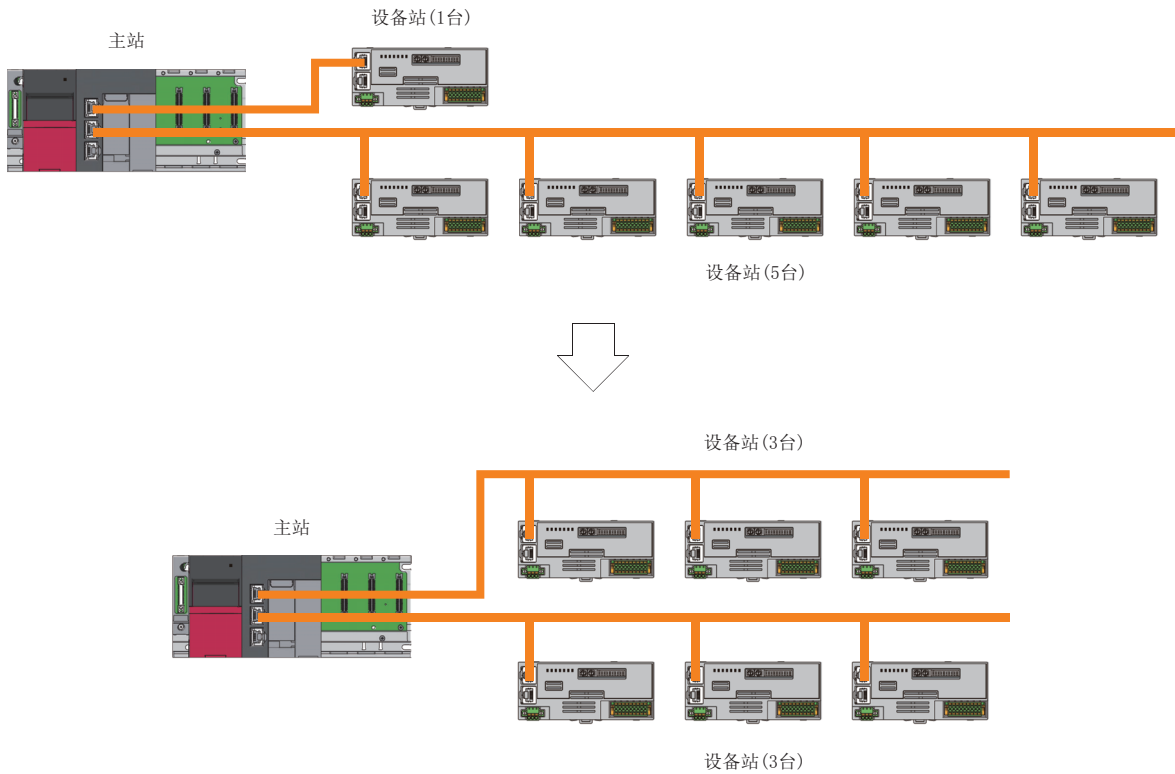
7.2 均衡设备站的台数

概要

通过均衡连接主站各端口的设备站台数，可以缩短通信周期间隔及循环传送时间。

执行步骤

如果各端口连接的设备站台数如下所示不同，则重新连接使其均衡。



注意事项

均衡设备站的台数，可能会对设备站的配置和可实现的系统造成限制。请根据要实现的系统内容判断能否使用该技术。

7.3 设置发送上限大小

概要

如果连接主站的设备站的CC-Link IE TSN Class为“Class A”且通信周期设置为“低速”，可以通过减小CC-Link IE TSN Class A组的发送上限大小缩短通信周期间隔及循环传送时间。

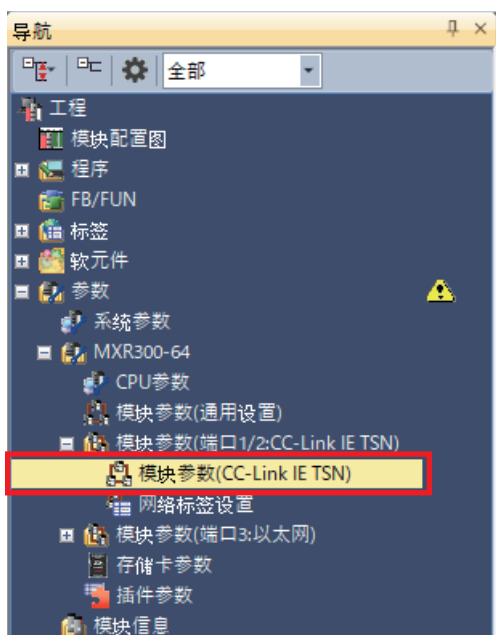
执行步骤

以CC-Link IE TSN Class为“Class A”且通信周期设置为“低速”的设备站连接主站的情况为例，说明设置CC-Link IE TSN Class A组的发送上限大小的步骤。

操作步骤

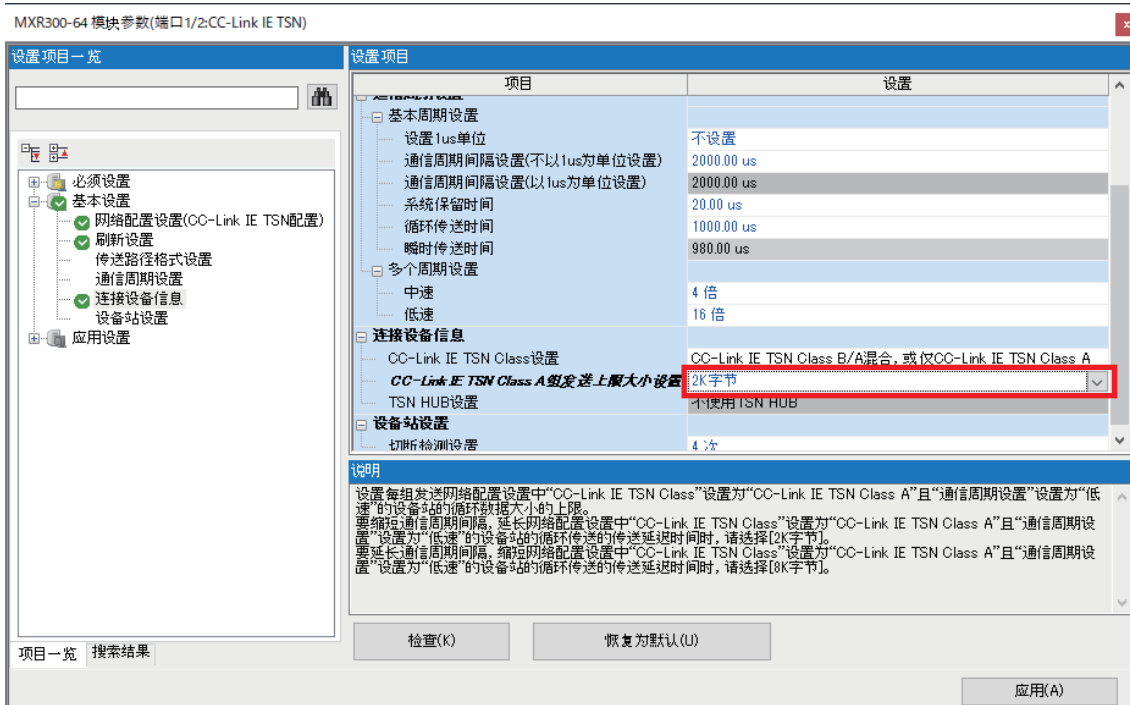
1. 打开模块参数(CC-Link IE TSN)。

① 双击导航窗口⇒[参数]⇒机型名⇒[模块参数(端口1/2:CC-Link IE TSN)]⇒[模块参数(CC-Link IE TSN)]

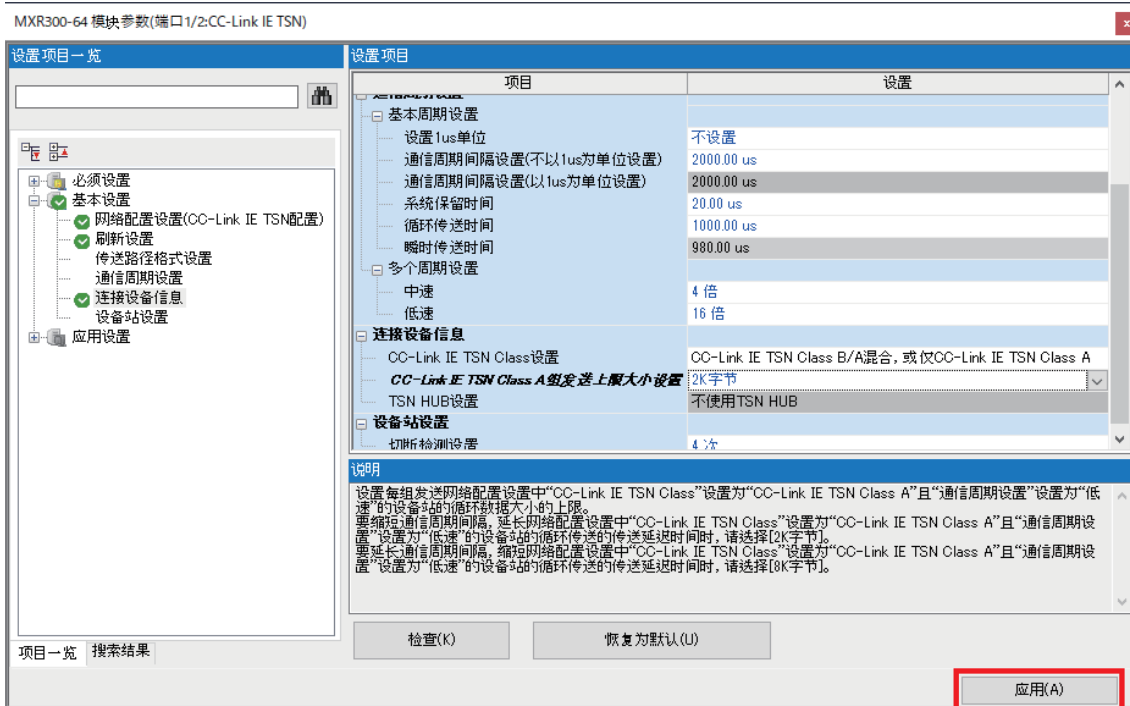


2. 将CC-Link IE TSN Class A组的发送上限大小设置为“2K字节”（默认）。

[基本设置]⇒[连接设备信息]⇒[CC-Link IE TSN Class A组发送上限大小设置]



3. 点击[应用]按钮应用设置。



注意事项

- 设置发送上限大小，可能会对可实现的系统造成限制。请根据要实现的系统内容判断能否使用该技术。
- 当“CC-Link IE TSN Class A组发送上限大小设置”设置为“2K字节”时，在1个通信周期内与1个组进行通信，则CC-Link IE TSN Class为“Class A”且通信周期设置为“低速”的设备站的传送延迟时间会更长。关于详细内容，请参阅下述手册。

📖 MELSEC MX控制器(MX-R型)用户手册

7.4 减少设备站台数

概要

通过减少连接主站的设备站台数，可以缩短通信周期间隔及循环传送时间。

执行步骤

以下述网络配置和刷新设置为例，说明删除设备站台数的步骤。

- 网络配置

台数	型号	站号	站类型	网络标签	RX设置			RY设置			RW设置			RW+设置			LP设置			LP+设置			参数自动设置	FDO映射设置	IP地址
					点数	起始	结束	点数	起始	结束	点数	起始	结束	点数	起始	结束	点数	起始	结束	点数	起始	结束			
0	本站	0	主站																					192.168.4.253	
1	NZ2GR12A4-16D	1	远程站		16	0000	000F	16	0000	000F	4	0000	0003	4	0000	0003									192.168.4.1
2	NZ2GR12A4-16D	2	远程站		16	0010	001F	16	0010	001F	4	0004	0007	4	0004	0007									192.168.4.2
3	NZ2GR12A4-16D	3	远程站		16	0020	002F	16	0020	002F	4	0008	000B	4	0008	000B									192.168.4.3
4	NZ2GR12A4-16D	4	远程站		16	0030	003F	16	0030	003F	4	000C	000F	4	000C	000F									192.168.4.4
5	NZ2GR12A4-16D	5	远程站		16	0040	004F	16	0040	004F	4	0010	0013	4	0010	0013									192.168.4.5
6	NZ2GR12A4-16D	6	远程站		16	0050	005F	16	0050	005F	4	0014	0017	4	0014	0017									192.168.4.6
7	NZ2GR12A4-16D	7	远程站		16	0060	006F	16	0060	006F	4	0018	001B	4	0018	001B									192.168.4.7
8	NZ2GR12A4-16D	8	远程站		16	0070	007F	16	0070	007F	4	001C	001F	4	001C	001F									192.168.4.8
9	NZ2GR12A4-16D	9	远程站		16	0080	008F	16	0080	008F	4	0020	0023	4	0020	0023									192.168.4.9
10	NZ2GR12A4-16D	10	远程站		16	0090	009F	16	0090	009F	4	0024	0027	4	0024	0027									192.168.4.10
11	NZ2GR12A4-16D	11	远程站		16	00A0	00AF	16	00A0	00AF	4	0028	002B	4	0028	002B									192.168.4.11
12	NZ2GR12A4-16D	12	远程站		16	00B0	00BF	16	00B0	00BF	4	002C	002F	4	002C	002F									192.168.4.12

} 站号10~12的设备站未使用

- 刷新设置

No.	链接侧					CPU侧				
	软元件名	点数	起始	结束		刷新目标	软元件名	点数	起始	结束
-	SB				↔					
-	SW				↔					
1	RX	192	00000	000BF	↔	指定软元件	X	192	01000	010BF
2	RY	192	00000	000BF	↔	指定软元件	Y	192	01000	010BF
3	RWw	48	00000	0002F	↔	指定软元件	W	48	00000	0002F
4	RWw	48	00000	0002F	↔	指定软元件	W	48	01000	0102F

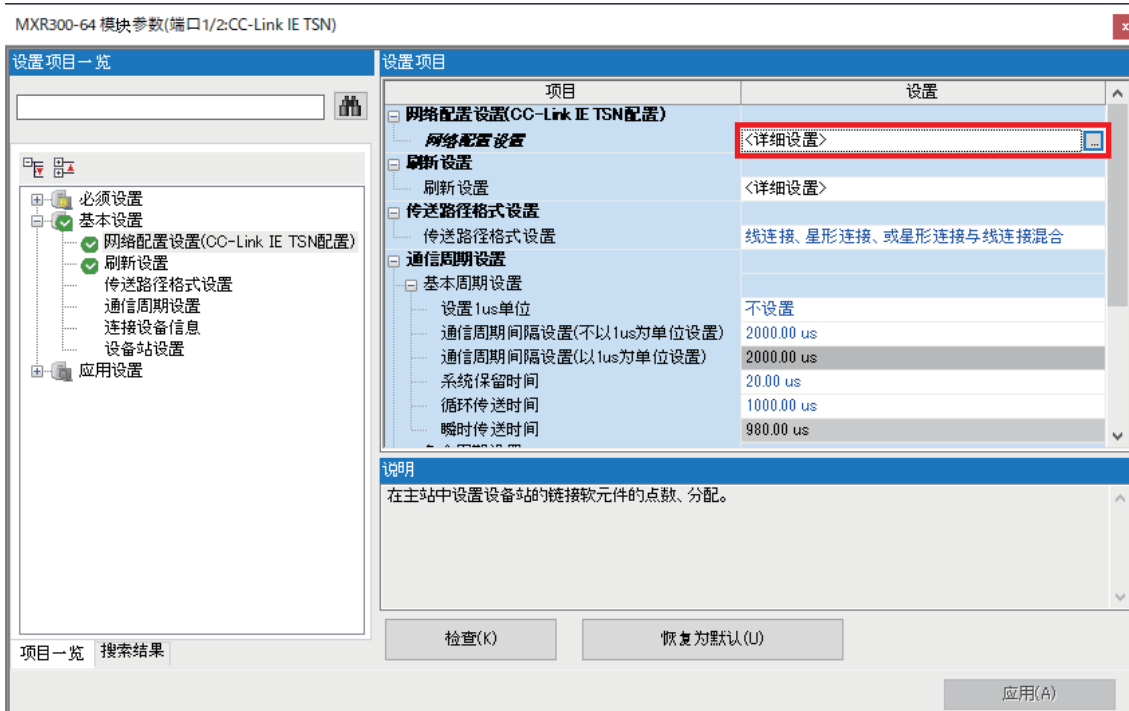
操作步骤

1. 打开模块参数(CC-Link IE TSN)。

双击导航窗口⇒[参数]⇒机型名⇒[模块参数(端口1/2:CC-Link IE TSN)]⇒[模块参数(CC-Link IE TSN)]

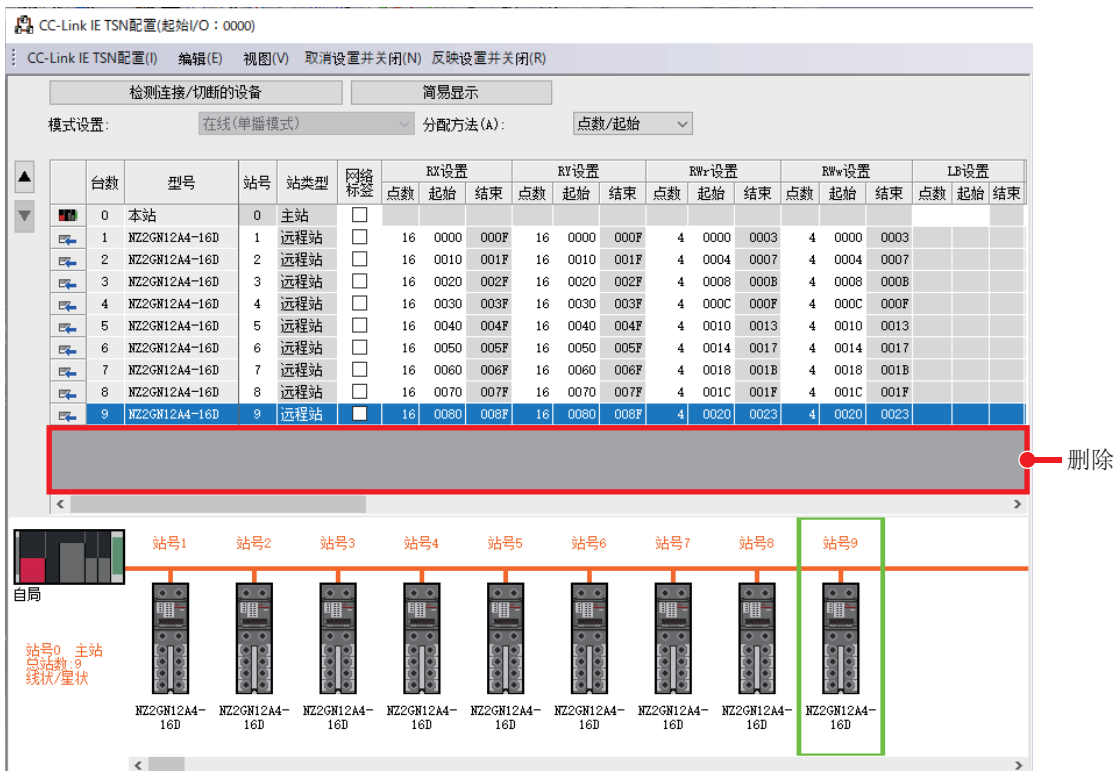
2. 在基本设置中打开网络配置设置。

双击[基本设置]⇒[网络配置设置(CC-Link IE TSN配置)]⇒[网络配置设置]的<详细设置>



3. 删除不使用和不需要的设备站。

- 站号10~12的设备站：删除

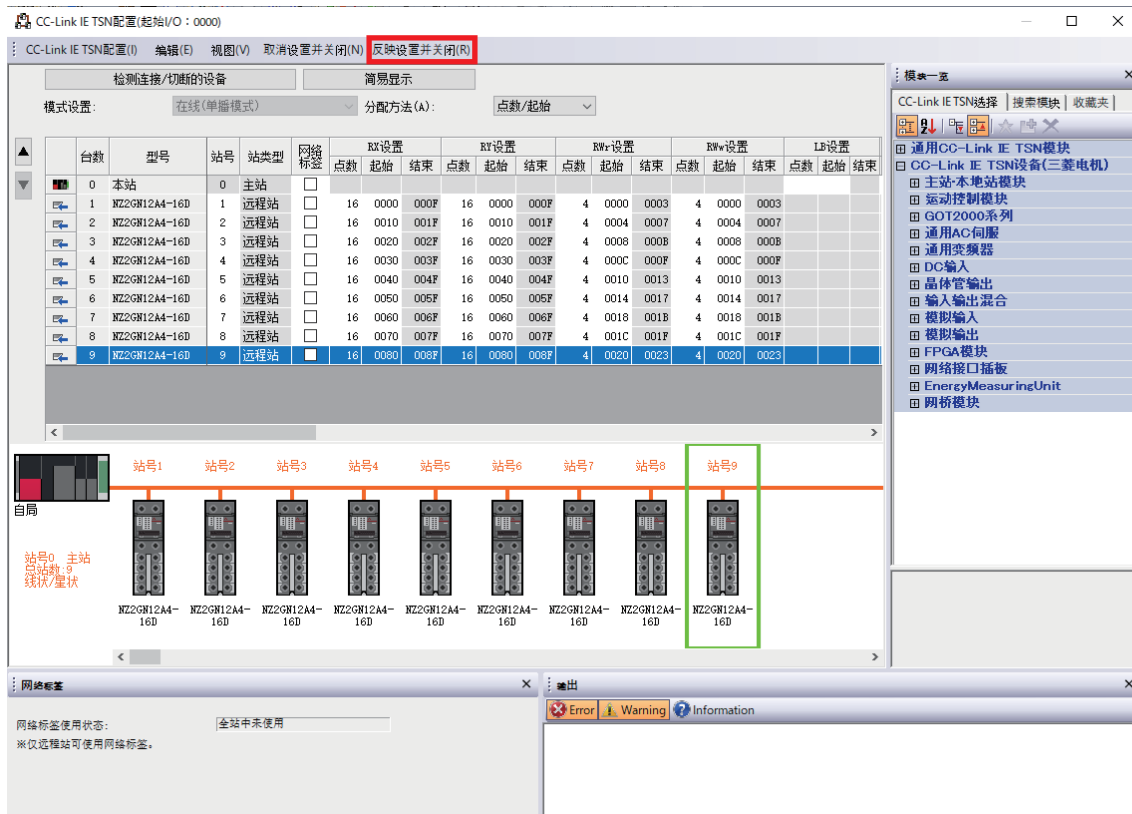


要点

删除不需要的设备站后，重新分配软元件编号可能会缩短链接刷新时间、通信周期间隔和循环传送时间。关于详细内容，请参阅下述章节。

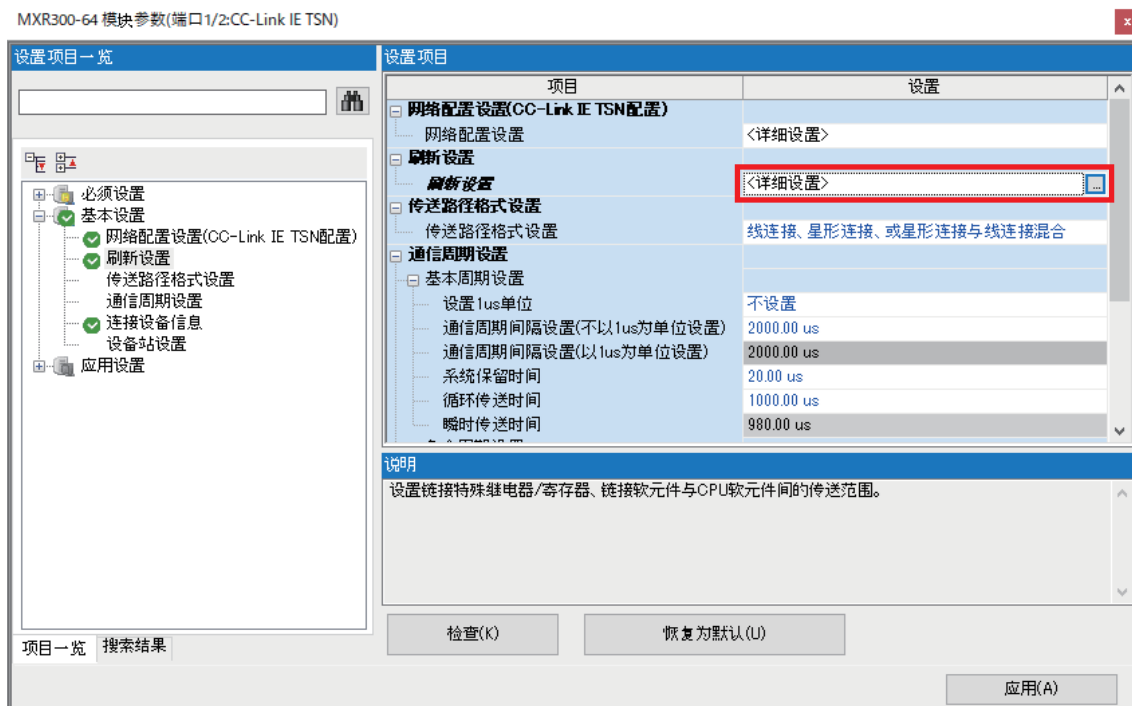
96页 缩短刷新时间

4. 点击[反映设置并关闭]按钮。



5. 结合刷新设置设置刷新。

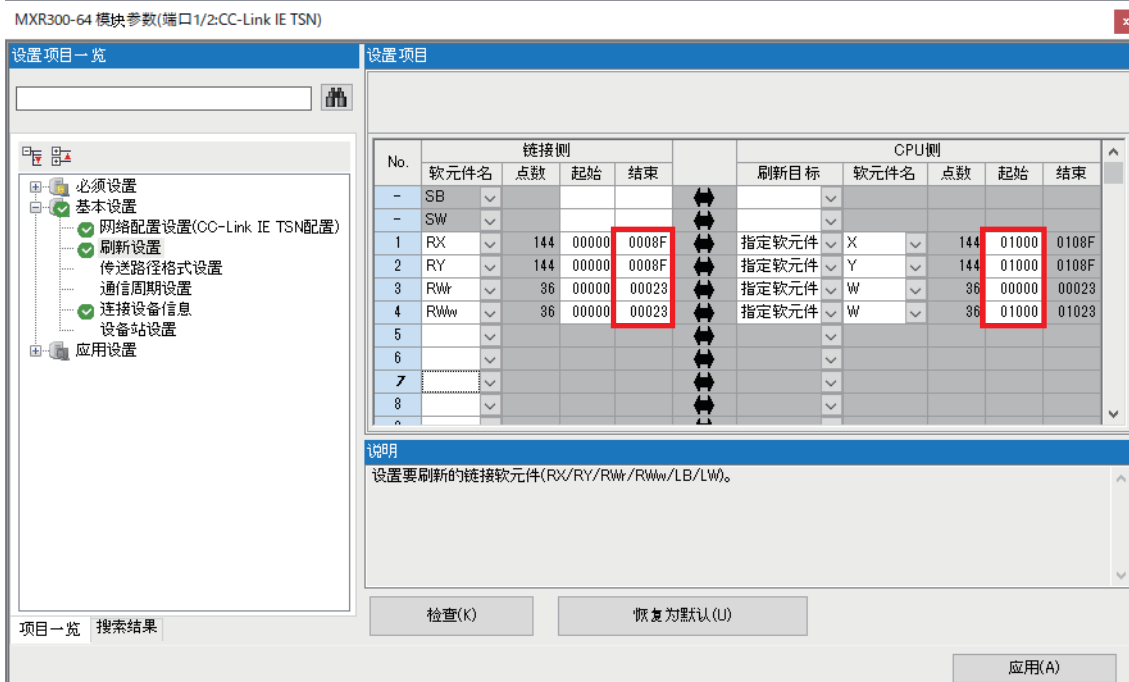
双击[基本设置]⇒[刷新设置]⇒[刷新设置]的<详细设置>



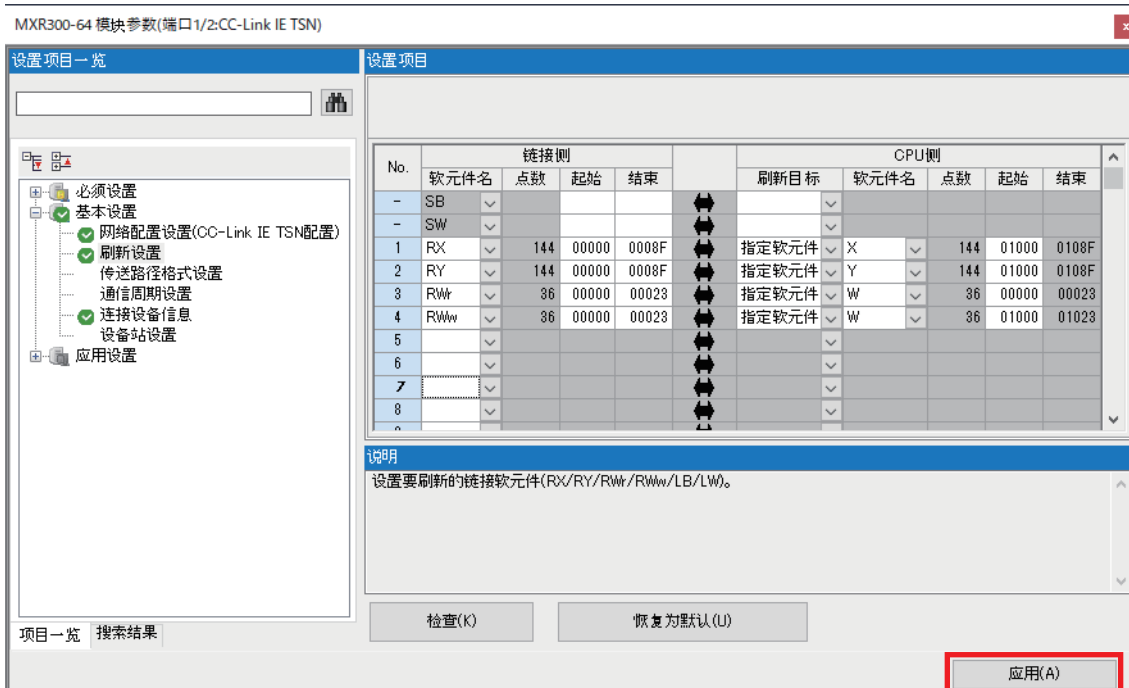
6. 结合步骤3中删除不需要的设备站更改链接侧的软元件的最终编号。

此外，根据链接侧设置的更改情况，也要检查CPU侧的设置。

- RX设置、RY设置的最终编号：0008F
- RWr设置、RWw设置的最终编号：00023



7. 点击[应用]按钮应用设置。



注意事项

- 删除设备站的台数，可能会减少连接的设备站台数，并对可实现的系统造成限制。请根据要实现的系统内容判断能否使用该技术。
- 即使删除CC-Link IE TSN Class为“CC-Link IE TSN Class A”且通信周期设置为“低速”的设备站，对加快CC-Link IE TSN也没有影响。关于各设备站的设置，请在“网络配置设置”画面中确认。

7.5 删除链接软元件设置

概要

删除各设备站不需要的链接软元件设置，可以缩短通信周期间隔及循环传送时间。

执行步骤

以下述网络配置和刷新设置为例，说明删除不需要的链接软元件设置的步骤。

- 网络配置

站号5~12的RX设置
未使用
RWr设置
未使用
站号1~4的RWw设置
未使用

台数	型号	站号	站类型	网络标签	RX设置			RY设置			RWr设置			RWw设置			LS设置	
					点数	起始	结束	点数	起始	结束	点数	起始	结束	点数	起始	结束	点数	起始
0	本站	0	主站															
1	NZ2GN12A4-16D	1	远程站	<input type="checkbox"/>	16	0000	000F	16	0000	000F	4	0000	0003	4	0000	0003		
2	NZ2GN12A4-16D	2	远程站	<input type="checkbox"/>	16	0010	001F	16	0010	001F	4	0004	0007	4	0004	0007		
3	NZ2GN12A4-16D	3	远程站	<input type="checkbox"/>	16	0020	002F	16	0020	002F	4	0008	000B	4	0008	000B		
4	NZ2GN12A4-16D	4	远程站	<input type="checkbox"/>	16	0030	003F	16	0030	003F	4	000C	000F	4	000C	000F		
5	NZ2GN12A4-16D	5	远程站	<input type="checkbox"/>	16	0040	004F	16	0040	004F	4	0010	0013	4	0010	0013		
6	NZ2GN12A4-16D	6	远程站	<input type="checkbox"/>	16	0050	005F	16	0050	005F	4	0014	0017	4	0014	0017		
7	NZ2GN12A4-16D	7	远程站	<input type="checkbox"/>	16	0060	006F	16	0060	006F	4	0018	001B	4	0018	001B		
8	NZ2GN12A4-16D	8	远程站	<input type="checkbox"/>	16	0070	007F	16	0070	007F	4	001C	001F	4	001C	001F		
9	NZ2GN12A4-16D	9	远程站	<input type="checkbox"/>	16	0080	008F	16	0080	008F	4	0020	0023	4	0020	0023		
10	NZ2GN12A4-16D	10	远程站	<input type="checkbox"/>	16	0090	009F	16	0090	009F	4	0024	0027	4	0024	0027		
11	NZ2GN12A4-16D	11	远程站	<input type="checkbox"/>	16	00A0	00AF	16	00A0	00AF	4	0028	002B	4	0028	002B		
12	NZ2GN12A4-16D	12	远程站	<input type="checkbox"/>	16	00B0	00BF	16	00B0	00BF	4	002C	002F	4	002C	002F		

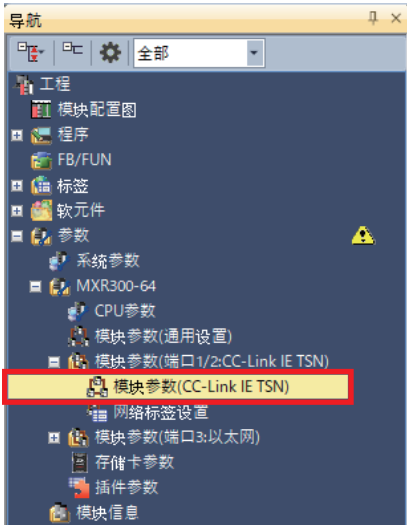
- 刷新设置

No.	链接侧					CPU侧				
	软元件名	点数	起始	结束		刷新目标	软元件名	点数	起始	结束
-	SB				↔					
-	SW				↔					
1	RX	192	00000	000BF	↔	指定软元件	X	192	01000	010BF
2	RY	192	00000	000BF	↔	指定软元件	Y	192	01000	010BF
3	RWr	48	00000	0002F	↔	指定软元件	W	48	00000	0002F
4	RWw	48	00000	0002F	↔	指定软元件	W	48	01000	0102F

操作步骤

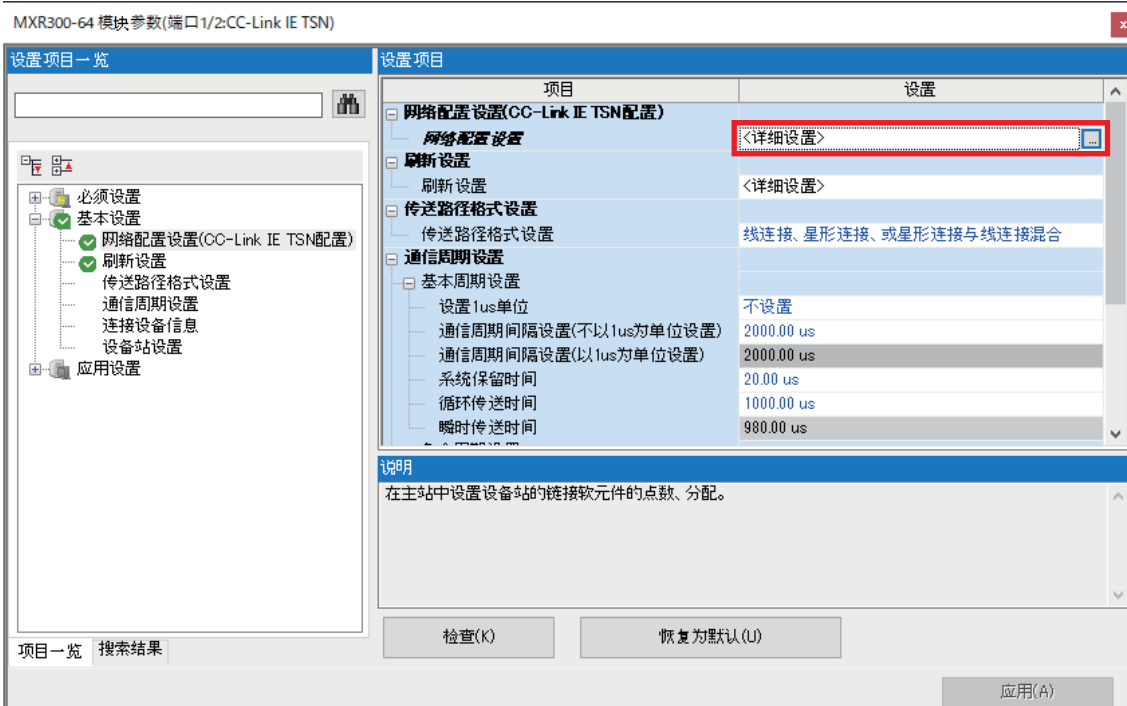
1. 打开模块参数(CC-Link IE TSN)。

双击导航窗口⇒[参数]⇒机型名⇒[模块参数(端口1/2:CC-Link IE TSN)]⇒[模块参数(CC-Link IE TSN)]



2. 在基本设置中打开网络配置设置。

双击[基本设置]⇒[网络配置设置(CC-Link IE TSN配置)]⇒[网络配置设置]的<详细设置>



3. 删除各设备站不需要的RX设置、RY设置、RWr设置和RWw设置(无软件元件分配的状态)。

- 站号5~12的RX设置：删除
- RWr设置的设置：删除
- 站号1~4的RWw设置：删除

台数	型号	站号	站类型	网络标签	RX设置			RY设置			RWr设置			RWw设置			LB设置		LW设置	
					点数	起始	结束	点数	起始	结束	点数	起始	结束	点数	起始	结束	点数	起始	结束	点数
0	本站	0	主站	<input checked="" type="checkbox"/>																
1	NZ2GN12A4-16D	1	远程站	<input type="checkbox"/>	16	0000	000F	16	0000	000F										
2	NZ2GN12A4-16D	2	远程站	<input type="checkbox"/>	16	0010	001F	16	0010	001F										
3	NZ2GN12A4-16D	3	远程站	<input type="checkbox"/>	16	0020	002F	16	0020	002F										
4	NZ2GN12A4-16D	4	远程站	<input type="checkbox"/>	16	0030	003F	16	0030	003F										
5	NZ2GN12A4-16D	5	远程站	<input type="checkbox"/>				16	0040	004F			4	0010	0013					
6	NZ2GN12A4-16D	6	远程站	<input type="checkbox"/>				16	0050	005F			4	0014	0017					
7	NZ2GN12A4-16D	7	远程站	<input type="checkbox"/>				16	0060	006F			4	0018	001B					
8	NZ2GN12A4-16D	8	远程站	<input type="checkbox"/>				16	0070	007F			4	001C	001F					
9	NZ2GN12A4-16D	9	远程站	<input type="checkbox"/>				16	0080	008F			4	0020	0023					
10	NZ2GN12A4-16D	10	远程站	<input type="checkbox"/>				16	0090	009F			4	0024	0027					
11	NZ2GN12A4-16D	11	远程站	<input type="checkbox"/>				16	00A0	00AF			4	0028	002B					
12	NZ2GN12A4-16D	12	远程站	<input type="checkbox"/>				16	00B0	00BF			4	002C	002F					

要点

删除不需要的链接软件元件设置后，重新分配软件元件编号可能会缩短链接刷新时间、通信周期间隔和循环传送时间。

关于详细内容，请参阅下述章节。

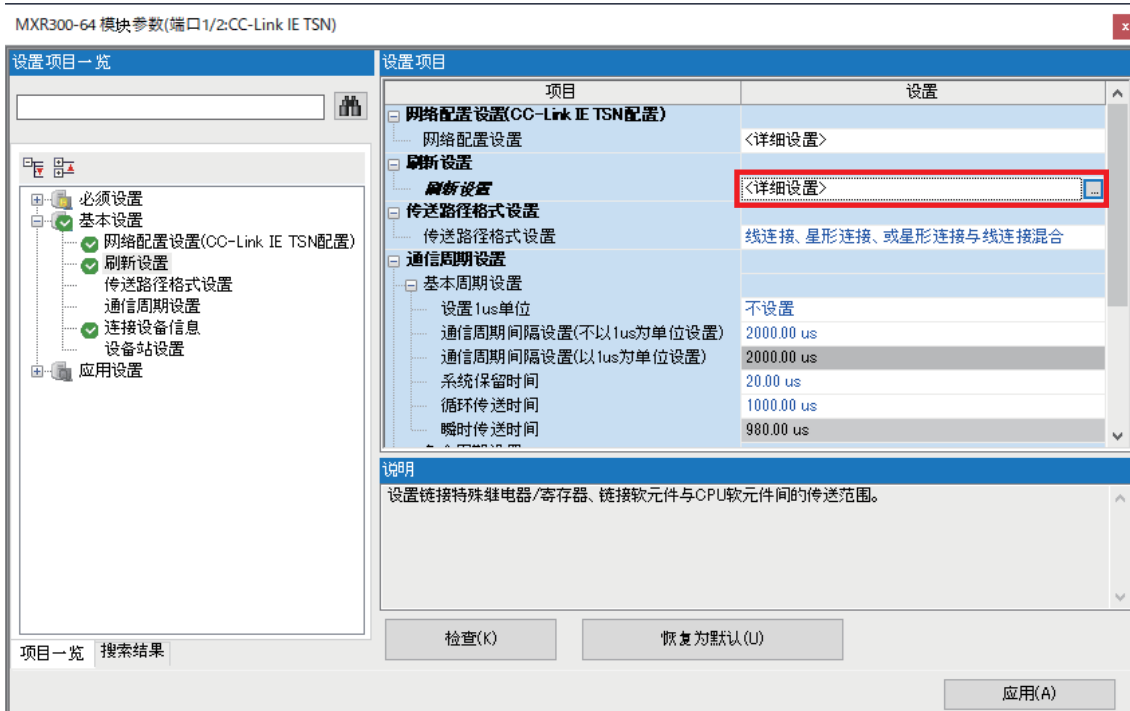
📖 96页 缩短刷新时间

4. 点击[反映设置并关闭]按钮。

台数	型号	站号	站类型	网络标签	RX设置			RY设置			RWr设置			RWw设置			LB设置		LW设置	
					点数	起始	结束	点数	起始	结束	点数	起始	结束	点数	起始	结束	点数	起始	结束	点数
0	本站	0	主站	<input checked="" type="checkbox"/>																
1	NZ2GN12A4-16D	1	远程站	<input type="checkbox"/>	16	0000	000F	16	0000	000F										
2	NZ2GN12A4-16D	2	远程站	<input type="checkbox"/>	16	0010	001F	16	0010	001F										
3	NZ2GN12A4-16D	3	远程站	<input type="checkbox"/>	16	0020	002F	16	0020	002F										
4	NZ2GN12A4-16D	4	远程站	<input type="checkbox"/>	16	0030	003F	16	0030	003F										
5	NZ2GN12A4-16D	5	远程站	<input type="checkbox"/>				16	0040	004F										
6	NZ2GN12A4-16D	6	远程站	<input type="checkbox"/>				16	0050	005F			4	0010	0013					
7	NZ2GN12A4-16D	7	远程站	<input type="checkbox"/>				16	0060	006F			4	0014	0017					
8	NZ2GN12A4-16D	8	远程站	<input type="checkbox"/>				16	0070	007F			4	0018	001B					
9	NZ2GN12A4-16D	9	远程站	<input type="checkbox"/>				16	0080	008F			4	001C	001F					
10	NZ2GN12A4-16D	10	远程站	<input type="checkbox"/>				16	0090	009F			4	0020	0023					
11	NZ2GN12A4-16D	11	远程站	<input type="checkbox"/>				16	00A0	00AF			4	0024	0027					
12	NZ2GN12A4-16D	12	远程站	<input type="checkbox"/>				16	00B0	00BF			4	0028	002B					

5. 结合刷新设置的更改情况，更改刷新设置。

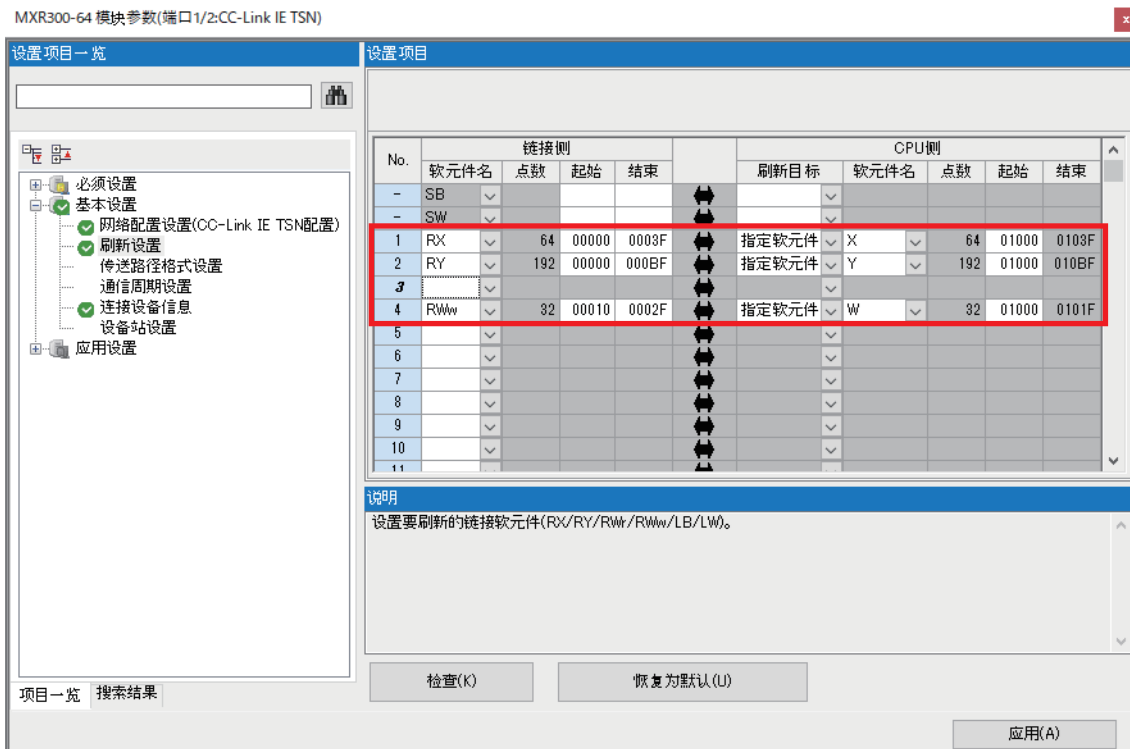
双击[基本设置]⇒[刷新设置]⇒[刷新设置]的<详细设置>



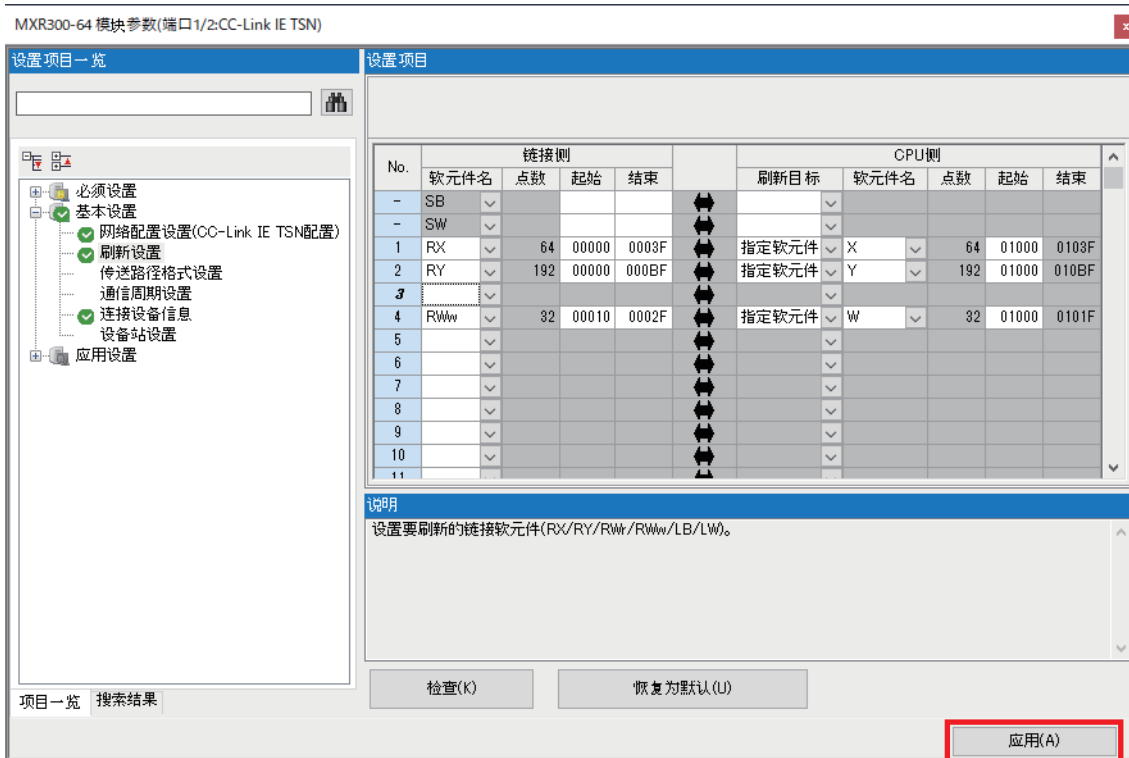
6. 结合步骤3、4中设置的链接软件件编号，更改链接侧软元件的起始编号和最终编号。

此外，根据链接侧设置的更改情况，也要检查CPU侧的设置。

- RX设置的编号范围：00000~0003F
- RWr设置的设置：删除
- RWw设置的编号范围：00010~0002F



7. 点击[应用]按钮应用设置。



注意事项

- 减少链接软件元件设置，可能会对可实现的系统造成限制。请根据要实现的系统内容判断能否使用该技术。
- 即使删除CC-Link IE TSN Class为“CC-Link IE TSN Class A”且通信周期设置为“低速”的设备站，对加快CC-Link IE TSN也没有影响。关于各设备站的设置，请在“网络配置设置”画面中确认。

7.6 减少链接软元件设置点数

概要

大幅减少各设备站的链接软元件设置的点数，可以缩短通信周期间隔及循环传送时间。

执行步骤

以下述网络配置和刷新设置为例，说明减少链接软元件设置点数的步骤。

- 网络配置

可删除到16点 可删除到4点

台数	型号	站号	站类型	网络标志	RX设置			RY设置			RW设置			RWw设置			LW设置			LWw设置			参数自动设置
					点数	起始	结束	点数	起始	结束	点数	起始	结束	点数	起始	结束	点数	起始	结束	点数	起始	结束	
0	本站	0	主站	<input checked="" type="checkbox"/>																			
1	NZ2GN12A4-16D	1	远程站	<input type="checkbox"/>	32	0000	001F	32	0000	001F	8	0000	0007	8	0000	0007							
2	NZ2GN12A4-16D	2	远程站	<input type="checkbox"/>	32	0020	003F	32	0020	003F	8	0008	000F	8	0008	000F							
3	NZ2GN12A4-16D	3	远程站	<input type="checkbox"/>	32	0040	005F	32	0040	005F	8	0010	0017	8	0010	0017							
4	NZ2GN12A4-16D	4	远程站	<input type="checkbox"/>	32	0060	007F	32	0060	007F	8	0018	001F	8	0018	001F							
5	NZ2GN12A4-16D	5	远程站	<input type="checkbox"/>	32	0080	009F	32	0080	009F	8	0020	0027	8	0020	0027							
6	NZ2GN12A4-16D	6	远程站	<input type="checkbox"/>	32	00A0	00BF	32	00A0	00BF	8	0028	002F	8	0028	002F							
7	NZ2GN12A4-16D	7	远程站	<input type="checkbox"/>	32	00C0	00DF	32	00C0	00DF	8	0030	0037	8	0030	0037							
8	NZ2GN12A4-16D	8	远程站	<input type="checkbox"/>	32	00E0	00FF	32	00E0	00FF	8	0038	003F	8	0038	003F							
9	NZ2GN12A4-16D	9	远程站	<input type="checkbox"/>	32	0100	011F	32	0100	011F	8	0040	0047	8	0040	0047							
10	NZ2GN12A4-16D	10	远程站	<input type="checkbox"/>	32	0120	013F	32	0120	013F	8	0048	004F	8	0048	004F							
11	NZ2GN12A4-16D	11	远程站	<input type="checkbox"/>	32	0140	015F	32	0140	015F	8	0050	0057	8	0050	0057							
12	NZ2GN12A4-16D	12	远程站	<input type="checkbox"/>	32	0160	017F	32	0160	017F	8	0058	005F	8	0058	005F							

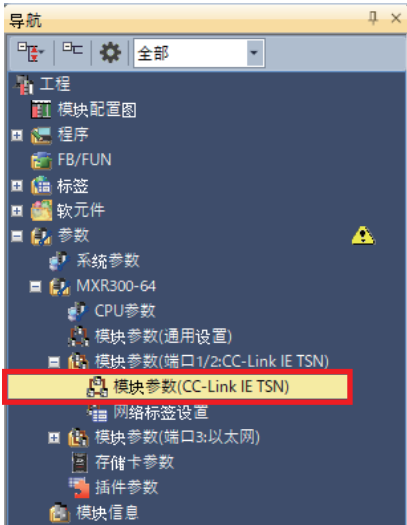
- 刷新设置

No.	链接侧					CPU侧			
	软元件名	点数	起始	结束		刷新目标	软元件名	点数	起始
-	SB								
-	SW								
1	RX	384	00000	0017F		指定软元件	X	384	01000 0117F
2	RY	384	00000	0017F		指定软元件	Y	384	01000 0117F
3	RWw	96	00000	0005F		指定软元件	W	96	00000 0005F
4	RWw	96	00000	0005F		指定软元件	W	96	01000 0105F

操作步骤

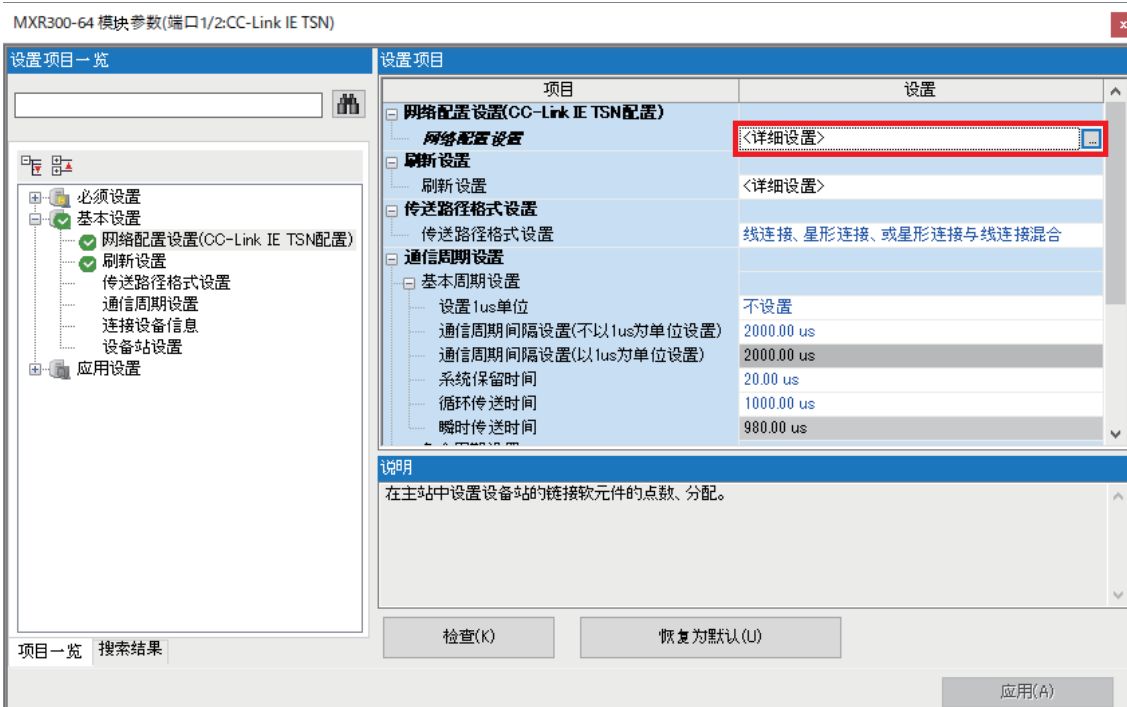
1. 打开模块参数(CC-Link IE TSN)。

双击导航窗口⇒[参数]⇒机型名⇒[模块参数(端口1/2:CC-Link IE TSN)]⇒[模块参数(CC-Link IE TSN)]



2. 在基本设置中打开网络配置设置。

双击[基本设置]⇒[网络配置设置(CC-Link IE TSN配置)]⇒[网络配置设置]的<详细设置>



3. 减少各设备站的RX设置、RY设置、RWr设置和RWw设置的点数，结合删除点数更改起始地址。

- RX设置、RY设置的各点数：16
- RWr设置、RWw设置的各点数：4
- RX设置、RY设置、RWr设置和RWw设置各起始地址：结合点数更改

更改为16点
(同时修改起始地址)

更改为4点
(同时修改起始地址)

要点

减少链接软件设置的点数后，重新分配软件元件编号可能会缩短链接刷新时间、通信周期间隔和循环传送时间。

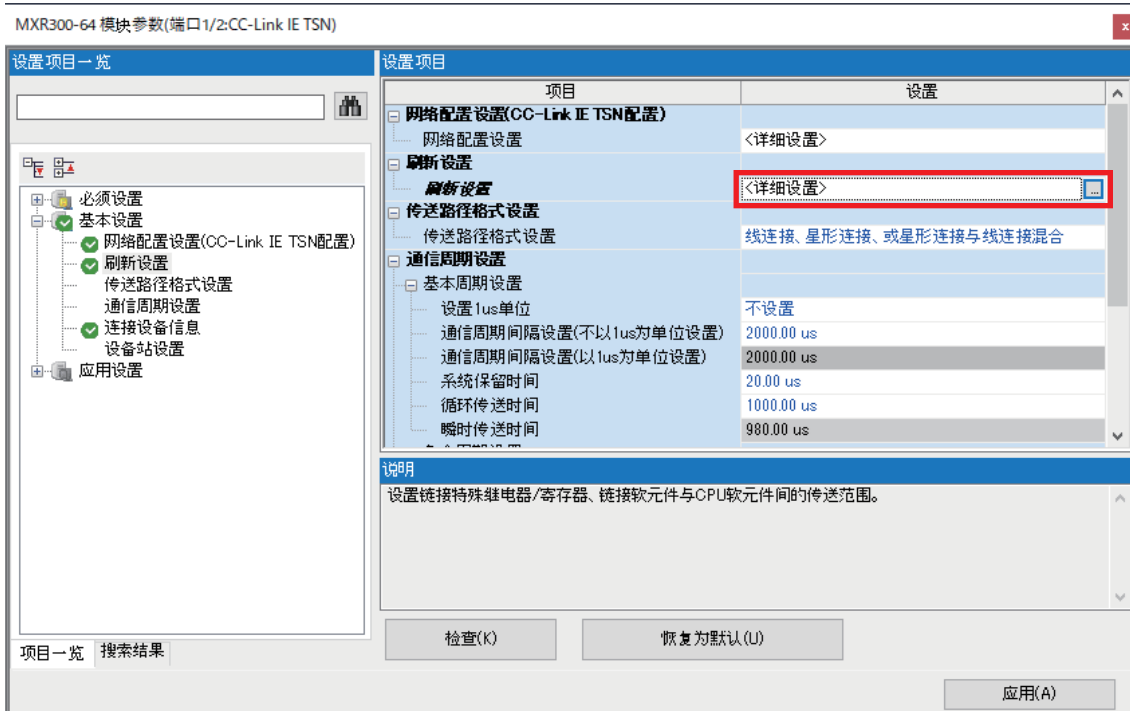
关于详细内容，请参阅下述章节。

☞ 96页 缩短刷新时间

4. 点击[反映设置并关闭]按钮。

5. 结合刷新设置的更改情况，更改刷新设置。

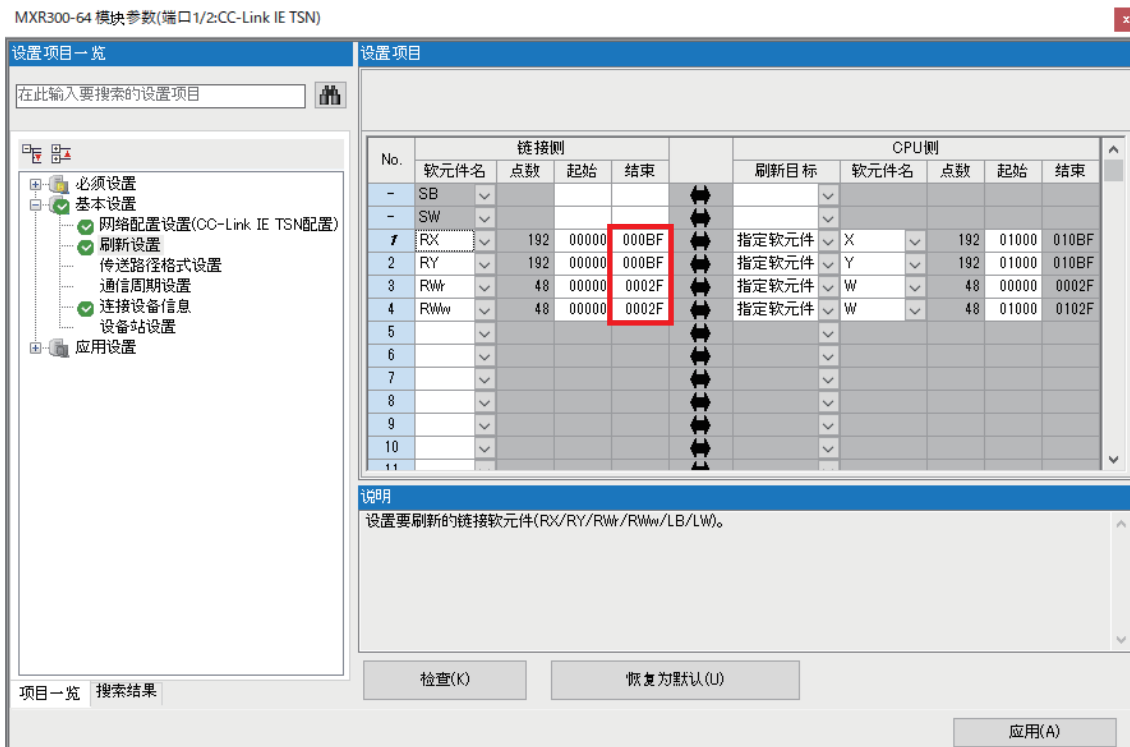
双击[基本设置]⇒[刷新设置]⇒[刷新设置]的<详细设置>



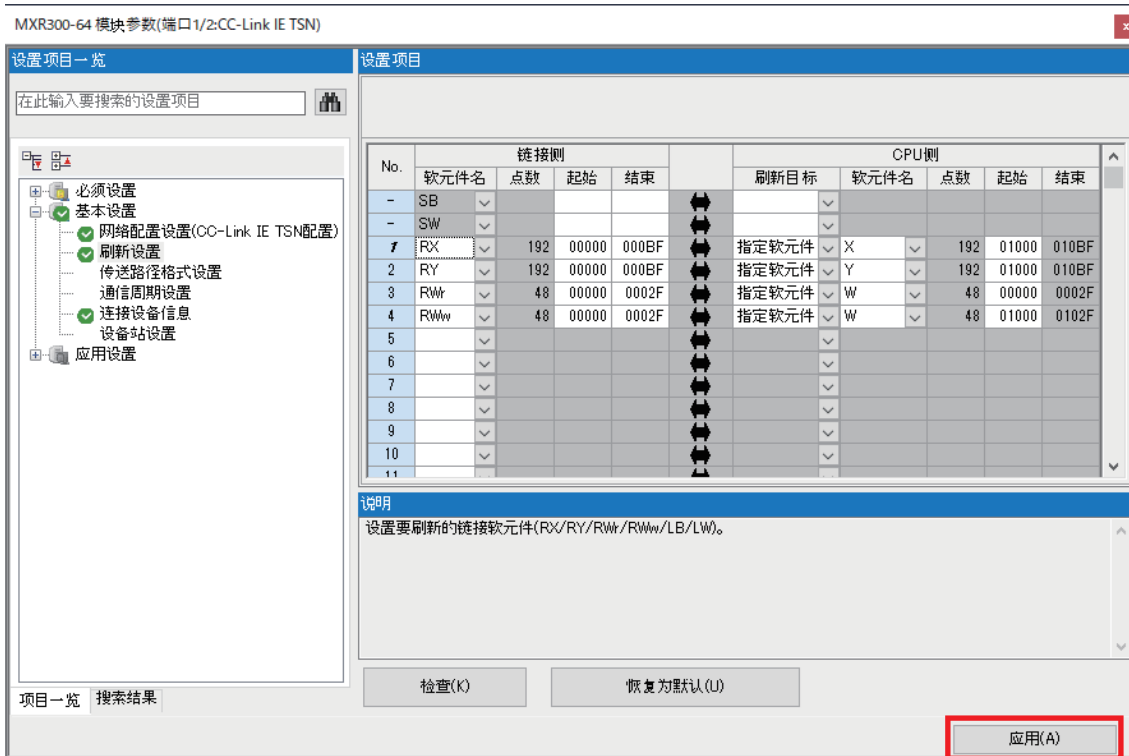
6. 结合步骤3中设置的链接软元件编号，更改链接侧软元件的最终编号。

此外，根据链接侧设置的更改情况，也要检查CPU侧的设置。

- RX设置、RY设置的最终编号：000BF
- RWr设置、RWw设置的最终编号：0002F



7. 点击[应用]按钮应用设置。



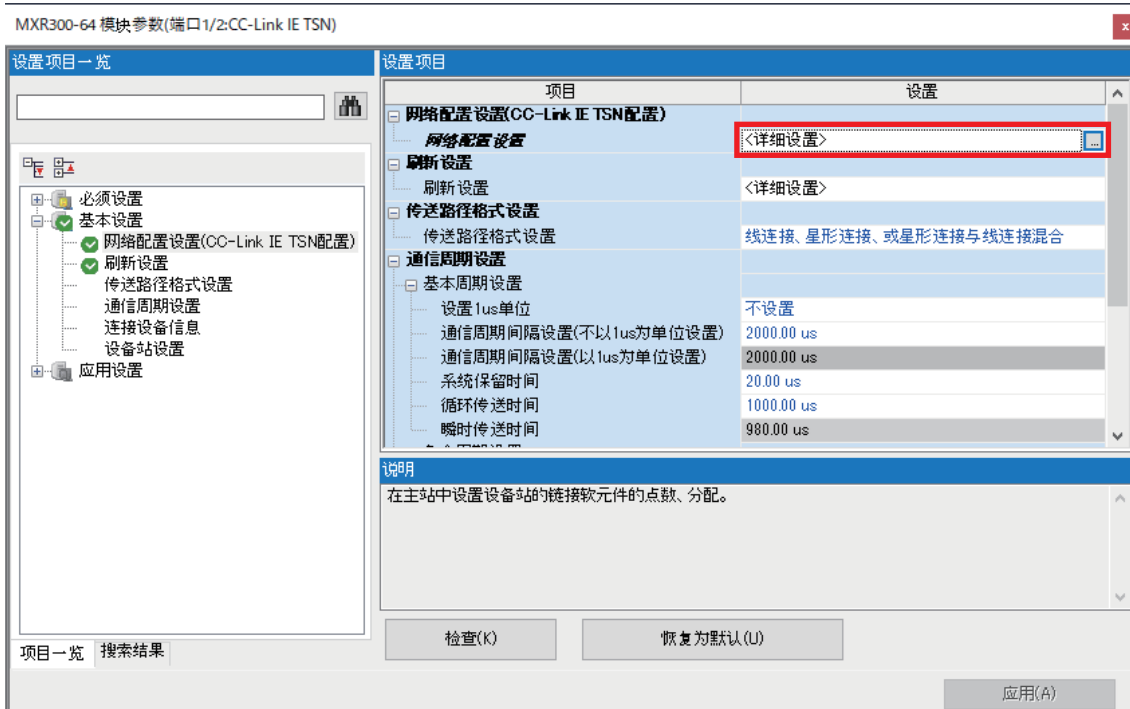
7

注意事项

- 减少链接软件元件设置，可能会对可实现的系统造成限制。请根据要实现的系统内容判断能否使用该技术。
- 即使删除CC-Link IE TSN Class为“CC-Link IE TSN Class A”且通信周期设置为“低速”的设备站，对加快CC-Link IE TSN也没有影响。关于各设备站的设置，请在“网络配置设置”画面中确认。

2. 在基本设置中打开网络配置设置。

双击[基本设置]⇒[网络配置设置(CC-Link IE TSN配置)]⇒[网络配置设置]的<详细设置>



3. 将CC-Link IE TSN Class设置为“Class B”的设备站的“网络同步通信设置”设置为“同步”或“不同步”。

CC-Link IE TSN配置(站址/0: 0000)

检测连接/切断的设备 详细显示

模式设置: 在线(单组模式) 分配方法(A):

台数	型号	站号	站类型	RX设置 点数	TX设置 点数	RW+设置 点数	RW+设置 点数	参数自动设置	FCID映射设置	IP地址	子网掩码	默认网关	保留/出错无效站	网络同步通信设置	通信速度	模式设置	CC-Link IE TSN Class	循环传送分散设置
0	本站	0	主站							192.168.4.253								
1	WD20R12M-16D	1	远端站	16	16	4	4	<详细设置>		192.168.4.1			无设置	同步	基本周期		CC-Link IE TSN Class B	不分散
2	WD20R12M-16D	2	远端站	16	16	4	4	<详细设置>		192.168.4.2			无设置	同步	低速		CC-Link IE TSN Class B	不分散
3	WD20R12M-16D	3	远端站	16	16	4	4	<详细设置>		192.168.4.3			无设置	同步	基本		CC-Link IE TSN Class B	不分散
4	WD20R12M-16D	4	远端站	16	16	4	4	<详细设置>		192.168.4.4			无设置	同步	中速		CC-Link IE TSN Class B	不分散
5	WD20R12M-16D	5	远端站	16	16	4	4	<详细设置>		192.168.4.5			无设置	同步	中速		CC-Link IE TSN Class B	不分散
6	WD20R12M-16D	6	远端站	16	16	4	4	<详细设置>		192.168.4.6			无设置	同步	低速		CC-Link IE TSN Class B	不分散
7	WD20R12M-16D	7	远端站	16	16	4	4	<详细设置>		192.168.4.7			无设置	同步	低速		CC-Link IE TSN Class B	不分散
8	WD20R12M-16D	8	远端站	16	16	4	4	<详细设置>		192.168.4.8			无设置	同步	基本周期		CC-Link IE TSN Class B	不分散
9	WD20R12M-16D	9	远端站	16	16	4	4	<详细设置>		192.168.4.9			无设置	同步	中速		CC-Link IE TSN Class B	不分散
10	WD20R12M-16D	10	远端站	16	16	4	4	<详细设置>		192.168.4.10			无设置	同步	基本		CC-Link IE TSN Class B	不分散
11	WD20R12M-16D	11	远端站	16	16	4	4	<详细设置>		192.168.4.11			无设置	同步	中速		CC-Link IE TSN Class B	不分散
12	WD20R12M-16D	12	远端站	16	16	4	4	<详细设置>		192.168.4.12			无设置	同步	低速		CC-Link IE TSN Class B	不分散

或

CC-Link IE TSN配置(站址/0: 0000)

检测连接/切断的设备 详细显示

模式设置: 在线(单组模式) 分配方法(A):

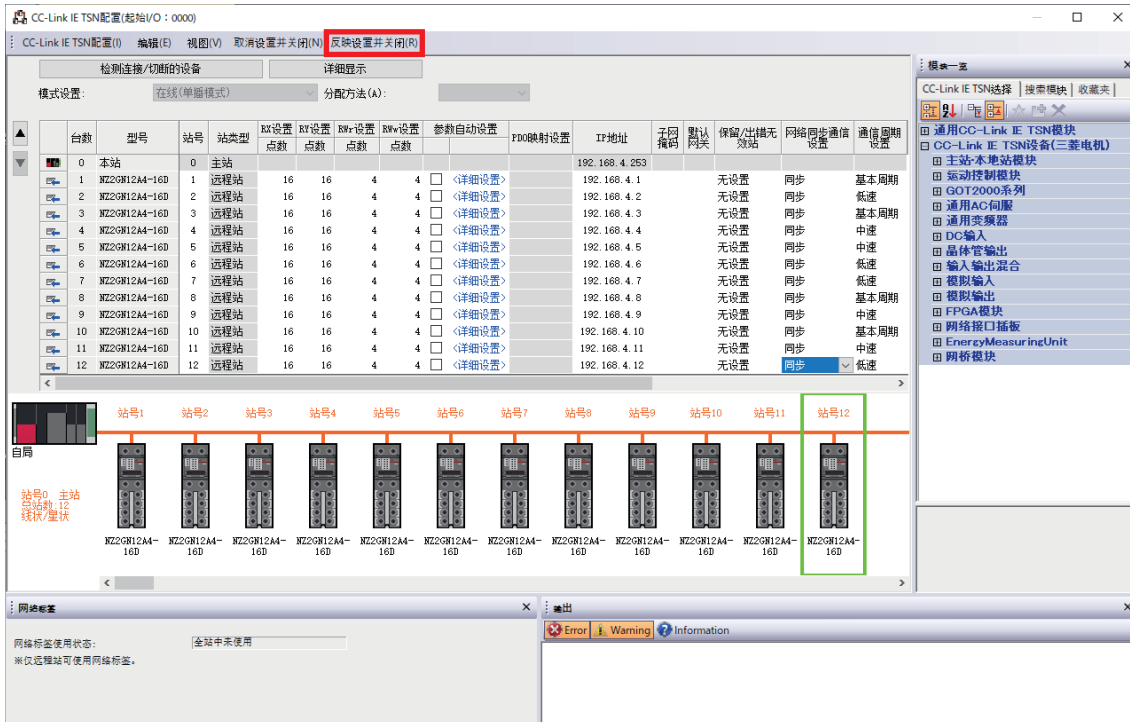
台数	型号	站号	站类型	RX设置 点数	TX设置 点数	RW+设置 点数	RW+设置 点数	参数自动设置	FCID映射设置	IP地址	子网掩码	默认网关	保留/出错无效站	网络同步通信设置	通信速度	模式设置	CC-Link IE TSN Class	循环传送分散设置
0	本站	0	主站							192.168.4.253								
1	WD20R12M-16D	1	远端站	16	16	4	4	<详细设置>		192.168.4.1			无设置	不同步	基本周期		CC-Link IE TSN Class B	不分散
2	WD20R12M-16D	2	远端站	16	16	4	4	<详细设置>		192.168.4.2			无设置	不同步	低速		CC-Link IE TSN Class B	不分散
3	WD20R12M-16D	3	远端站	16	16	4	4	<详细设置>		192.168.4.3			无设置	不同步	基本		CC-Link IE TSN Class B	不分散
4	WD20R12M-16D	4	远端站	16	16	4	4	<详细设置>		192.168.4.4			无设置	不同步	中速		CC-Link IE TSN Class B	不分散
5	WD20R12M-16D	5	远端站	16	16	4	4	<详细设置>		192.168.4.5			无设置	不同步	中速		CC-Link IE TSN Class B	不分散
6	WD20R12M-16D	6	远端站	16	16	4	4	<详细设置>		192.168.4.6			无设置	不同步	低速		CC-Link IE TSN Class B	不分散
7	WD20R12M-16D	7	远端站	16	16	4	4	<详细设置>		192.168.4.7			无设置	不同步	低速		CC-Link IE TSN Class B	不分散
8	WD20R12M-16D	8	远端站	16	16	4	4	<详细设置>		192.168.4.8			无设置	不同步	基本周期		CC-Link IE TSN Class B	不分散
9	WD20R12M-16D	9	远端站	16	16	4	4	<详细设置>		192.168.4.9			无设置	不同步	中速		CC-Link IE TSN Class B	不分散
10	WD20R12M-16D	10	远端站	16	16	4	4	<详细设置>		192.168.4.10			无设置	不同步	基本		CC-Link IE TSN Class B	不分散
11	WD20R12M-16D	11	远端站	16	16	4	4	<详细设置>		192.168.4.11			无设置	不同步	中速		CC-Link IE TSN Class B	不分散
12	WD20R12M-16D	12	远端站	16	16	4	4	<详细设置>		192.168.4.12			无设置	不同步	低速		CC-Link IE TSN Class B	不分散

要点

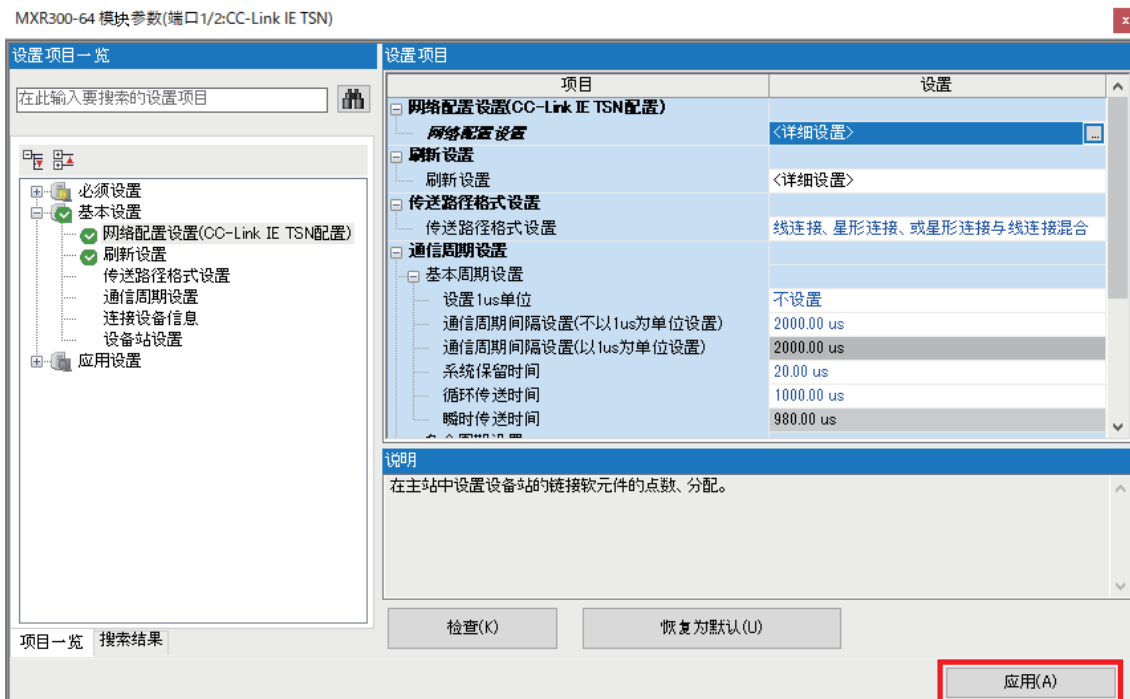
更改网络同步通信设置后，重新分配软件元件编号可能会缩短链接刷新时间、通信周期间隔和循环传送时间。关于详细内容，请参阅下述章节。

96页 缩短刷新时间

4. 点击[反映设置并关闭]按钮。



5. 点击[应用]按钮应用设置。



注意事项

由于网络同步通信设置只允许CC-Link IE TSN Class为“Class B”的设备站进行“同步”通信，因此该设置可能会对所使用设备站和可实现的系统造成限制。请根据要实现的系统内容判断能否使用该技术。

7.8 设置循环传送分散

概要

CC-Link IE TSN Class为“Class B”且通信周期设置为“中速”或“低速”的设备站，可以通过在通信周期内分散各站的循环传送时机来缩短基本周期的通信周期间隔和循环传送时间。

执行步骤

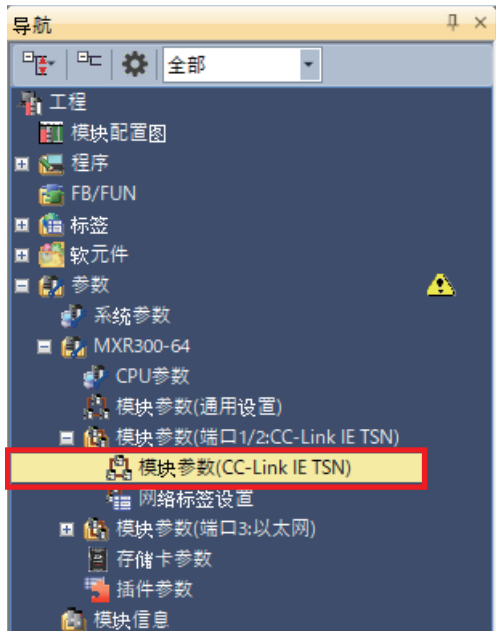
以下述网络配置为例，说明分散CC-Link IE TSN Class为“Class B”且通信周期设置为“中速”或“低速”的设备站的循环传送时机的步骤。

台数	型号	站号	站类型	网络	RS-设置	RS-设置	RS-设置	RS-设置	网络同步通信设置	通信周期	设备	CC-Link IE TSN Class	循环传送分散
0	本站	0	主站										
1	K22GR12A4-16D	1	远程站		16 0000 000F	16 0000 000F	4 0000 0003	4 0000 0003	同步	基本周期		CC-Link IE TSN Class B	不分散
2	K22GR12A4-16D	2	远程站		16 0010 001F	16 0010 001F	4 0004 0007	4 0004 0007	异步	低速		CC-Link IE TSN Class B	不分散
3	K22GR12A4-16D	3	远程站		16 0020 002F	16 0020 002F	4 0008 000B	4 0008 000B	同步	基本周期		CC-Link IE TSN Class B	不分散
4	K22GR12A4-16D	4	远程站		16 0030 003F	16 0030 003F	4 000C 000F	4 000C 000F	设置	中速		CC-Link IE TSN Class B	不分散
5	K22GR12A4-16D	5	远程站		16 0040 004F	16 0040 004F	4 0010 0013	4 0010 0013	设置	中速		CC-Link IE TSN Class B	不分散
6	K22GR12A4-16D	6	远程站		16 0050 005F	16 0050 005F	4 0014 0017	4 0014 0017	异步	低速		CC-Link IE TSN Class B	不分散
7	K22GR12A4-16D	7	远程站		16 0060 006F	16 0060 006F	4 0018 001B	4 0018 001B	设置	异步		CC-Link IE TSN Class B	不分散
8	K22GR12A4-16D	8	远程站		16 0070 007F	16 0070 007F	4 001C 001F	4 001C 001F	设置	基本周期		CC-Link IE TSN Class B	不分散
9	K22GR12A4-16D	9	远程站		16 0080 008F	16 0080 008F	4 0020 0023	4 0020 0023	异步	中速		CC-Link IE TSN Class B	不分散
10	K22GR12A4-16D	10	远程站		16 0090 009F	16 0090 009F	4 0024 0027	4 0024 0027	异步	基本周期		CC-Link IE TSN Class B	不分散
11	K22GR12A4-16D	11	远程站		16 00A0 00AF	16 00A0 00AF	4 0028 002B	4 0028 002B	异步	中速		CC-Link IE TSN Class B	不分散
12	K22GR12A4-16D	12	远程站		16 00B0 00BF	16 00B0 00BF	4 002C 002F	4 002C 002F	异步	低速		CC-Link IE TSN Class B	不分散

操作步骤

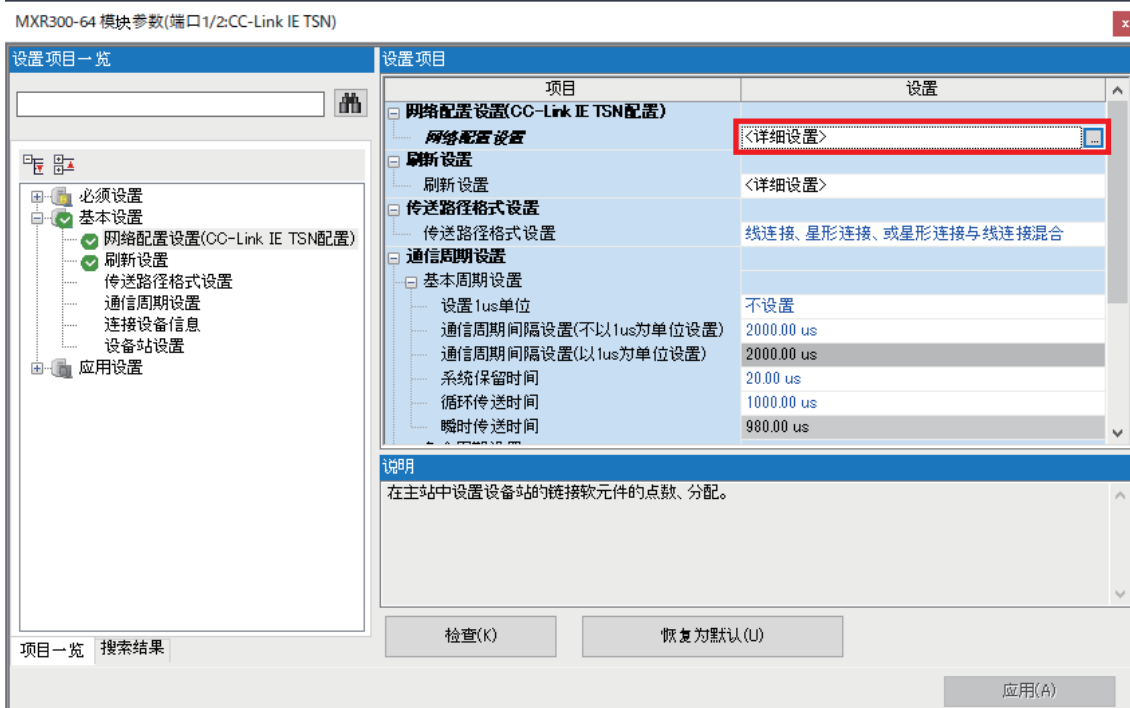
1. 打开模块参数(CC-Link IE TSN)。

双击导航窗口⇒[参数]⇒机型名⇒[模块参数(端口1/2:CC-Link IE TSN)]⇒[模块参数(CC-Link IE TSN)]



2. 在基本设置中打开网络配置设置。

双击[基本设置]⇒[网络配置设置(CC-Link IE TSN配置)]⇒[网络配置设置]的<详细设置>



3. 将CC-Link IE TSN Class为“Class B”且通信周期设置为“中速”或“低速”的设备站的“循环传送分散设置”设置为“分散”。



要点

如果未显示“循环传送分散设置”项目，请点击[详细显示]按钮显示。

4. 点击[反映设置并关闭]按钮。

CC-Link IE TSN配置(起始I/O : 0000)

CC-Link IE TSN配置(I) 编辑(E) 视图(V) 取消设置并关闭(N) **反映设置并关闭(R)**

检测连接/切断的设备 简易显示

模式设置: 在线(单播模式) 分配方法(A): 点数/起始

台数	型号	IP地址	子网掩码	默认网关	保留/出错无效站	网络同步通信设置	通信周期设置	站信息	CC-Link IE TSN Class	循环传送分散设置
0	本站	192.168.4.253								
1	NZ2GH12A4-16D	192.168.4.1			无设置	同步	基本周期		CC-Link IE TSN Class B	不分散
2	NZ2GH12A4-16D	192.168.4.2			无设置	不同步	低速		CC-Link IE TSN Class B	分散
3	NZ2GH12A4-16D	192.168.4.3			无设置	同步	基本周期		CC-Link IE TSN Class B	不分散
4	NZ2GH12A4-16D	192.168.4.4			无设置	同步	中速		CC-Link IE TSN Class B	分散
5	NZ2GH12A4-16D	192.168.4.5			无设置	不同步	中速		CC-Link IE TSN Class B	分散
6	NZ2GH12A4-16D	192.168.4.6			无设置	同步	低速		CC-Link IE TSN Class B	分散
7	NZ2GH12A4-16D	192.168.4.7			无设置	不同步	低速		CC-Link IE TSN Class B	分散
8	NZ2GH12A4-16D	192.168.4.8			无设置	不同步	基本周期		CC-Link IE TSN Class B	不分散
9	NZ2GH12A4-16D	192.168.4.9			无设置	同步	中速		CC-Link IE TSN Class B	分散
10	NZ2GH12A4-16D	192.168.4.10			无设置	不同步	基本周期		CC-Link IE TSN Class B	不分散
11	NZ2GH12A4-16D	192.168.4.11			无设置	同步	中速		CC-Link IE TSN Class B	分散
12	NZ2GH12A4-16D	192.168.4.12			无设置	同步	低速		CC-Link IE TSN Class B	分散

站号1 站号2 站号3 站号4 站号5 站号6 站号7 站号8 站号9 站号10 站号11 站号12

站号0: 主站
总站数: 12
线状/星状

5. 点击[应用]按钮应用设置。

MXR300-64 模块参数(端口 1/2:CC-Link IE TSN)

设置项目一览

在此输入要搜索的设置项目

必须设置
基本设置
网络配置设置(CC-Link IE TSN配置)
刷新设置
传送路径格式设置
通信周期设置
连接设备信息
设备站设置
应用设置

设置项目

项目	设置
网络配置设置(CC-Link IE TSN配置)	
网络配置设置	<详细设置>
刷新设置	<详细设置>
传送路径格式设置	线连接、星形连接、或星形连接与线连接混合
通信周期设置	
基本周期设置	
设置1us单位	不设置
通信周期间隔设置(不以1us为单位设置)	2000.00 us
通信周期间隔设置(以1us为单位设置)	2000.00 us
系统保留时间	20.00 us
循环传送时间	1000.00 us
瞬时传送时间	980.00 us

说明
在主站中设置设备站的链接元件的点数、分配。

检查(K) 恢复为默认(L) **应用(A)**

注意事项

由于只能为CC-Link IE TSN Class为“Class B”且通信周期设置为“中速”或“低速”的设备站设置循环传送分散，因此该设置可能会对使用的设备站和可实现的系统造成限制。请根据要实现的系统内容判断能否使用该技术。

修订记录

*本手册号在封底的左下角。

修订日期	*手册编号	修改内容
2025年6月	BCN-89999-9837-A	第一版

日文手册编号：BCN-89999-9684-B

本手册不授予工业产权或任何其他类型的权利，也不授予任何专利许可。三菱电机对由于使用了本手册中的内容而引起的涉及工业产权的任何问题不承担责任。

© 2025 MITSUBISHI ELECTRIC CORPORATION

资讯与服务

关于资讯与服务，请向当地三菱电机代理店咨询。
请访问三菱电机官网查找当地三菱电机代理店。

MITSUBISHI ELECTRIC Factory Automation Global Website
Locations Worldwide
www.MitsubishiElectric.com/fa/about-us/overseas/

商标

The company names, system names and product names mentioned in this manual are either registered trademarks or trademarks of their respective companies.

In some cases, trademark symbols such as ‘™’ or ‘®’ are not specified in this manual.

BCN-89999-9837-A(2506)

三菱电机自动化(中国)有限公司

地址：上海市虹桥路1386号三菱电机自动化中心

邮编：200336

电话：86-21-2322-3030 传真：86-21-2322-3000

官网：<https://www.MitsubishiElectric-FA.cn>

技术支持热线 **400-821-3030**



内容如有更改 恕不另行通知