MITSUBISHI

三菱可编程控制器
培训教材

MELSEC

结构化文本(ST)编程

(GX Developer用)

● 安全注意事项 ●

（使用前请务必仔细阅读）

设计系统时，请务必阅读相关手册，同时请充分考虑安全问题。

此外，实习时请充分注意以下各点，正确进行操作。

【实习注意事项】

⚠ 危险

- 通电中请不要触摸端子，以免发生触电事故。
- 打开安全盖时，请确认已切断电源，确保充分安全后再进行作业。
- 请不要用手接触可动部分。

⚠ 注意

- 请听从讲师的指示进行实习。
- 请不要擅自取下实习机的模块，或者改变连线，
  以免发生故障、误动作、伤害以及火灾。
- 装卸模块时，请先关断电源。
  通电中进行操作，可能引起模块故障或者触电。
- 实习机(X/Y载台等)出现异味或者异常声音时，请按下[电源开关]或者[紧急开关]，停止机器。
- 发生异常的情况下，请立即联系讲师。

再 版 记 录

※ 教科书号码记载于本教科书背面的左下方。

编制日期	教科书号码	修 订 内 容
2004年10月	SH（名称）=080536-A	初版

目　录

前　言

本教科书对MELSEC-Q系列QCPU的ST编程进行介绍。

<相关资料如下>

手册名称	手册号码 （型号代码）
QCPU 用户手册（硬件设计、维护检查篇） 对 CPU 模块、电源模块、基板模块、扩展电缆以及存储卡等规格进行说明。 （另售）	SH-080472 (13JP56)
QCPU 用户手册（功能说明、程序基础篇） 对 QCPU（Q 模式）下编写程序的必要功能、编程方法以及软元件等进行说明。 （另售）	SH-080473 (13JP57)
QCPU（Q 模式）/QnACPU 编程手册（通用命令篇） 对顺控命令、基本命令以及应用命令等的使用方法进行说明。 （另售）	SH-080021 (13JC00)
GX Developer 版本 8 操作手册 对 GX Developer 下的程序编写、打印、监控、调试方法等进行说明。 （另售）	SH-080356 (13KV69)
GX Developer 版本 8 操作手册（功能模块篇） 对 GX Developer 下的功能模块编辑、监视操作进行说明。 （另售）	SH-080359 (13JV72)
GX Developer 版本 8 操作手册（结构化文本篇） 对 GX Developer 下结构化文本程序的编写方法、打印输出方法进行说明。 （另售）	SH-080364 (13JV73)
结构化文本（ST）编程指南 面向初次使用结构化文本（ST）进行编程的人员。通过示例程序，对基本的操作方法和功能进行说明。 （另售）	SH-080365 (13JC12)
QCPU（Q 模式） 编程手册（结构化文本篇） 对结构化文本语言的编程方法进行说明。 （另售）	SH-080363 (13JC11)

备　忘　录

1 概要

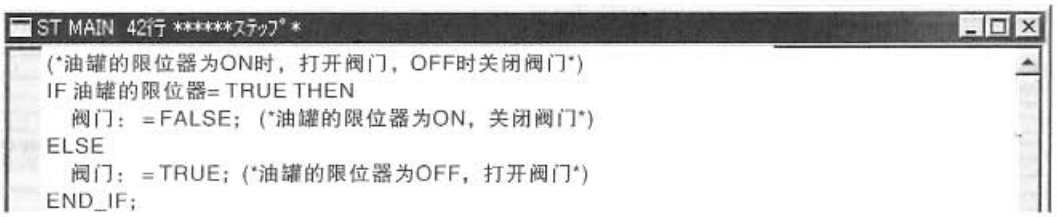
1.1 何谓结构化文本（ST）

ST语言是由国际标准IEC61131-3定义的关于开放式控制器下逻辑表达方式的规定语言。St支持运算符、控制结构和函数，如下几点所述。

- 根据条件的选择分支结构、循环控制结构
- 使用运算符（*,/,+,−,<,>,+等）的表达式
- 用户定义功能块（FB）的调用
- 函数的调用（MELSEC函数、IEC函数）
- 包含汉字等全角文字的注释

（1）文本形式的自由表达

ST语言可以使用半角英文字符及数字的文本形式进行表达。标识或者注释中也可以使用汉字等全角字符。



（2）能够以与C语言等高级语言同样的方式进行编程

ST语言同样可以使用C语言等高级语言中的条件选择结构和循环控制结构。因此可以简洁地编写具有良好可读性的程序。

```
(*生产线A~C控制*)
CASE 生产线 OF
  1: 开始开关: =TRUE; (*传送带运行*)
  2: 开始开关: =FALSE; (*传送带停止*)
  3: 开始开关: =TRUE; (*传送带停止，告警*)
     告警灯: =TRUE;
END_CASE;
IF 开始开关=TRUE THEN (*传送带运行，处理100次*)
FOR 处理次数: =0
  TO 100
  BY 1 D0
  处理数: =处理数+1;
 END FOR;
```

(3) 运算处理易于表达

对于列表、梯形图下难以表达的运算处理，ST语言可以简洁地进行清晰的表达，从而使程序具有良好的可读性，适合于需要进行复杂算术运算以及比较运算的应用领域。

```
CASE 生产线 OF
  1: Speed A:  = Distance_B/Hour_C*3600(*测量*)
    (*调用FB*)
    FB1(I_Test:= D0, O_Test:= D1, IO_Test:= D100)
  2: MO: = GT_E(X0, D0, D1, D2, D3, Result);
    (*如果执行条件为X0为ON，则D0~D3依次查询按照增序排列的运行设备*)
    (*并将结果存入Result中*)
    阀门:  =FALSE;
    RETURN;
END_CASE
```

1.2 特点

ST程序是指使用ST语言进行表达的程序。
通过使用GX Developer编写ST程序，可以在良好的操作环境下进行高效率的编程。

MELSEC-Q系列ST程序的主要特点如下。

(1) 模块化设计提高开发效率

通过ST语言，将经常使用的处理作为功能模块（FB）进行模块化的预定义，可以在各个程序的必要部分进行调用。这样，在提高程序开发效率的同时，还减少程序错误，提高程序品质。

(2) 可以在系统运行中更改程序（运行中写入）

不需要停止PLC CPU，在运行中即可对程序进行部分修改。

(3) 可同时使用其他语言的程序

MELSEC-Q系列还支持ST语言以外的语言，因此可以选择使用适于处理的语言，编写高效率的程序。
高性能QCPU与过程QCPU能够以文件为单位设置执行条件，1个PLC CPU中可以写入多个程序文件。
通过支持多语言，在宽广的应用范围内提供最佳的控制。

(4) 丰富的库函数

针对MELSEC-Q系列的ST程序，准备了适用MELSEC-Q系列各种共通命令的MELSEC函数，以及IEC61131-3中定义的IEC函数。

要　点

ST程序与顺控程序相比，可以使用的软元件类别和表示方法有部分不同。详细请参照附录1。

1.3 关于本教科书的内容

本教科书记述了ST语言的概要，使用GX Developer编写ST程序，以及将程序写入PLC・CPU的顺序。

（1）关于GX Developer的操作

GX Developer下编写ST程序的操作，请参照下列手册。

[GX Developer 版本8 操作手册（ST篇）]

需要ST编程之外的操作信息，请参照下列手册。

・[GX Developer 版本8操作手册]

・[GX Developer 版本8操作手册（基础篇）]

（2）关于ST编程

关于ST程序的详细编程方法，请参照下列手册。

该手册详细记述了ST程序中使用的函数、控制结构的书写、表达示例、数据类型的表达方式等。

[QCPU（Q模式）编程手册（结构化文本篇）]

（3）关于功能块（FB）

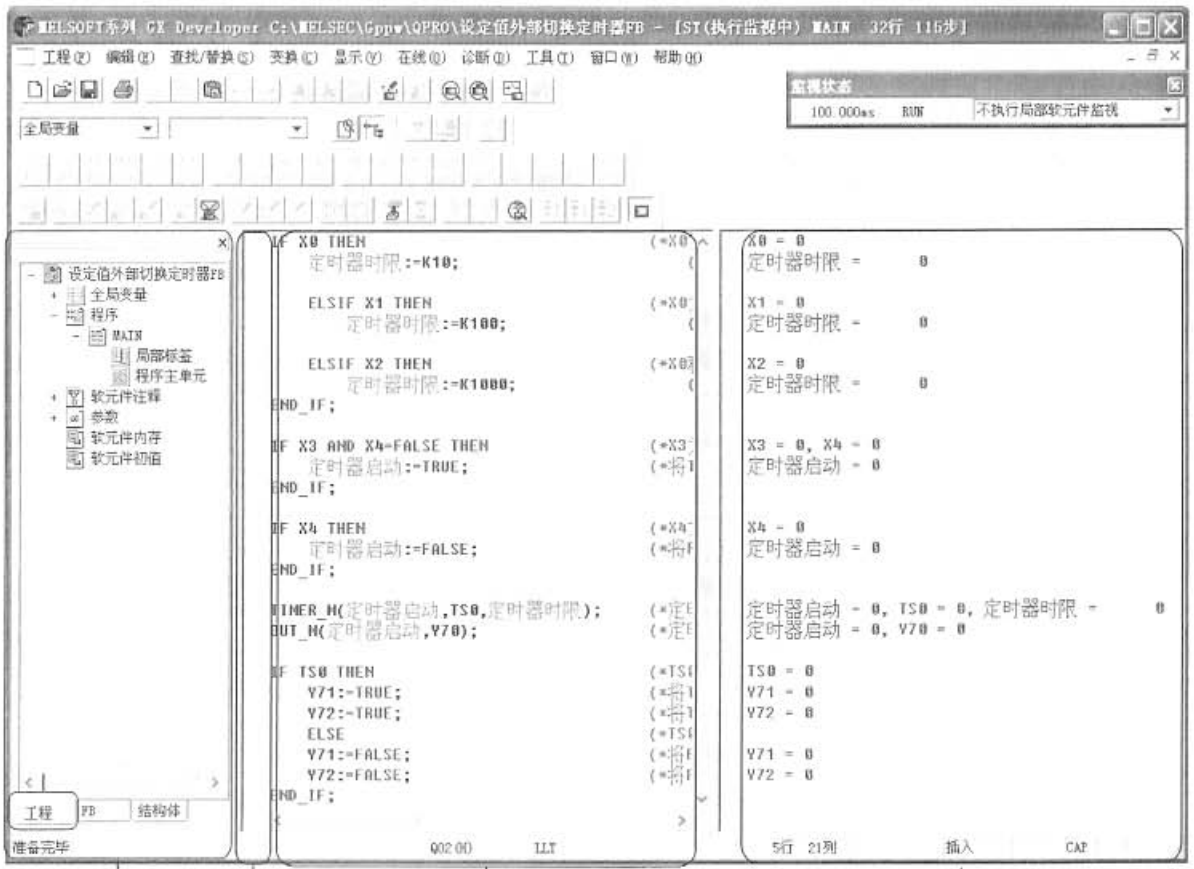
ST程序中使用功能块（FB）的情况下，请参照下列手册。

该手册对于将顺控程序中反复使用的回路部分进行模块化（FB）及其在顺控程序中进行重用的顺序进行了说明。

[QCPU（FB）程序教程]

本节对编写ST程序的画面构成及名称进行说明。

（1）编写顺控程序的画面（顺控程序画面）

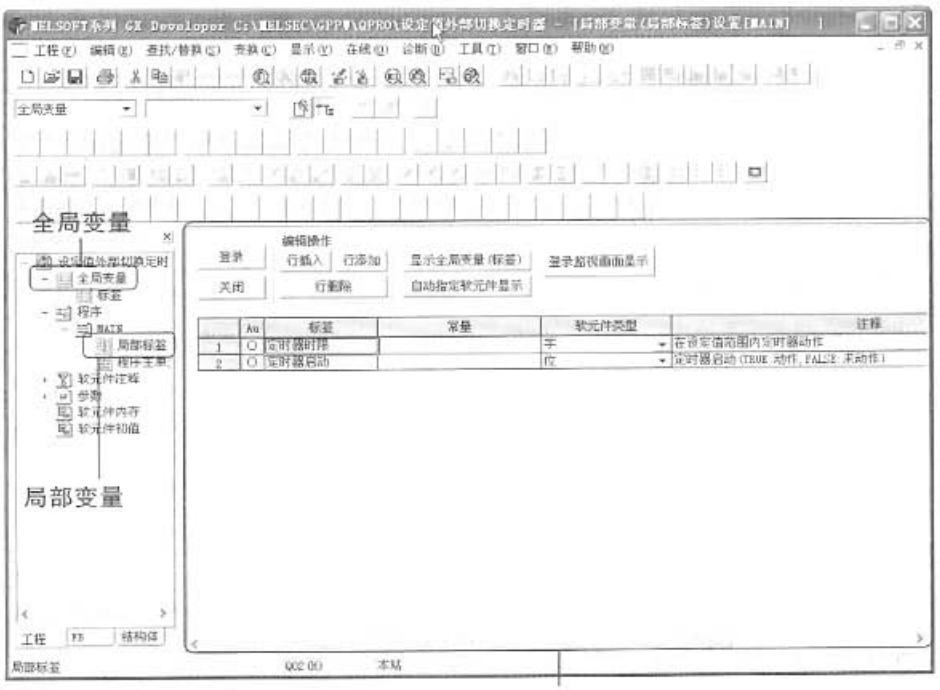


此为编辑顺控程序的画面。
顺控程序通过ST进行编写。
不可以使用列表形式。

名称	内　　容
工程分页	工程数据浏览中显示工程信息的分页。
工程数据浏览	将工程中的数据进行分类显示。可以直接调出程序编写画面、对话框等。
ST 编辑画面	编辑 ST 程序的画面。
监视画面	监视 ST 程序、确认 PLC CPU 动作状态的画面。
指示栏	显示编辑中的状态。

（2）建立全局变量和局部变量的画面

（全局变量设置画面/局部变量设置画面）



局部变量设置画面

例：局部变量设置画面的情况

对顺控程序中使用的全局变量和局部变量的变量类型、软元件类型等进行定义。

名称	内容
全局变量	工程中全部顺控程序均可使用的变量。
全局变量设置画面	设置全局变量的画面
局部变量	各顺控程序内部使用的变量。
局部变量设置画面	设置局部变量的画面

1.5 ST程序的编写顺序

对ST程序的编写顺序进行说明。

（1）功能模块（FB）的编写（仅在使用FB时）

ST程序中使用功能块（FB）的情况下，对FB定义程序和标签进行定义。

将编写的FB定义进行编译，则FB建立完成。

FB定义画面

备　注
关于功能块（FB）的设置方法，请参照[Q编程教程（FB篇）]。

（2）全局变量和局部变量的建立

顺控程序中使用的全局变量和局部变量需先进行定义。

全局变量可以在整个工程中进行使用。

局部变量仅可以在定义该标签的程序中使用。

此外，使用FB的情况下，在全局变量设置画面以及局部变量设置画面下对使用的FB标签进行定义。

[局部变量设置画面]

	Au	标签	常量	软元件类型	注释
1	○	设定值		字	
2	○	定时器启动		位	定时器启动(TRUE:动作, FALSE:未动作)
3	○	FB_TIMER	详细设置	FB(TIMER)	

（3）顺控程序的编写

在ST编辑画面下，使用ST语言输入顺控程序。

ST程序组要由以下形式构成。

（1）赋值语句

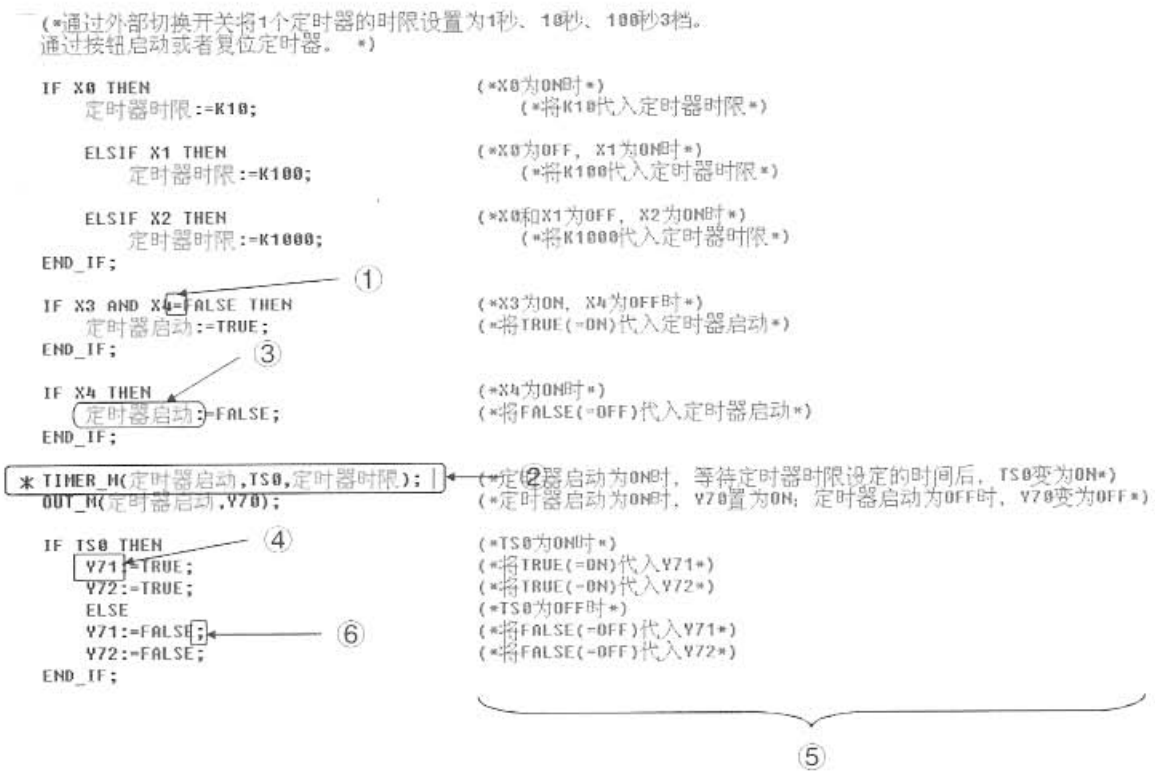
将右侧表达式的处理结果赋给左侧的标签或软元件。

（2）控制结构

・条件语句：条件成立时，执行相应的选择分支。

・循环语句：根据特定的变量或条件的状态，重复执行多条语句。

例）IF X0 THEN ~END _IF;



ST程序中使用的字符如下。

① 运算符：表达式、赋值语句中表示特定含义的字符代号。

② 函数：已定义的MELSEC函数、IEC函数名

③ 标签：用户定义的任意类型和名称的数据

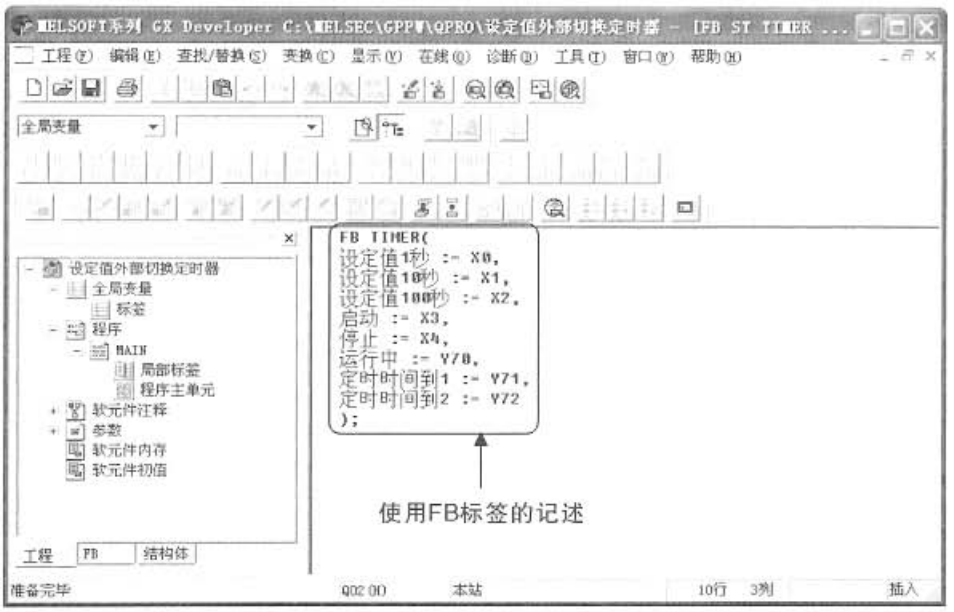
④ 软元件：MELSEC PLC中使用的软元件。通过软元件名、软元件号码进行识别。

⑤ 注释：程序中不作为控制处理对象的注释性文字。

⑥ 标点：为了明确程序结构而定义的具有特定含义的字符代号。

要　点

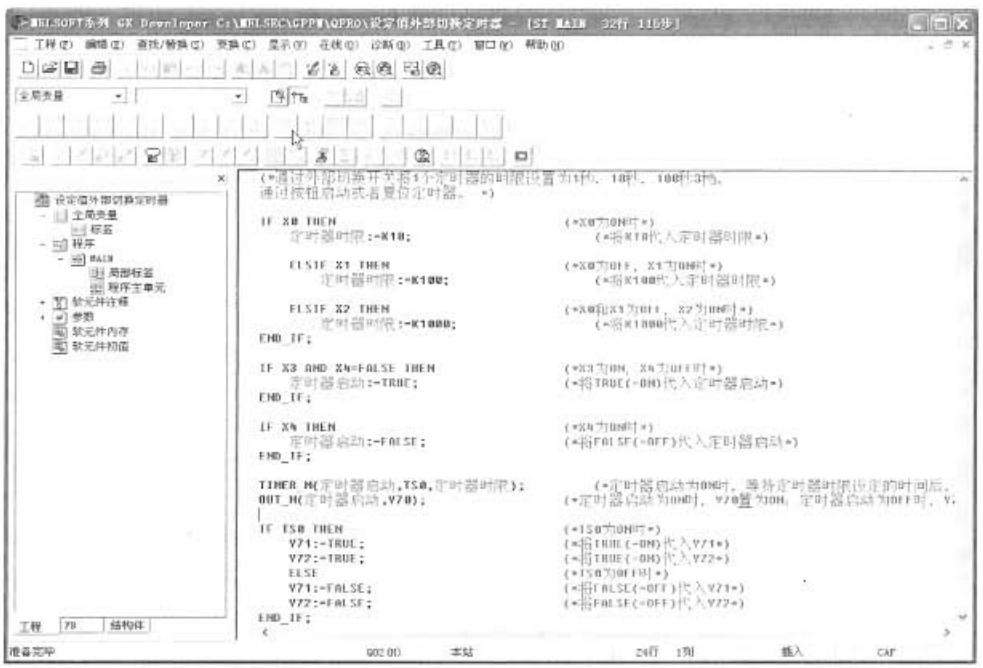
使用FB的情况下，在顺控程序按照如下的方式进行输入。



（4）顺控程序编译

将编写的顺控程序进行编译后，才可以使用。

编译完成后，顺控程序即建立完毕。



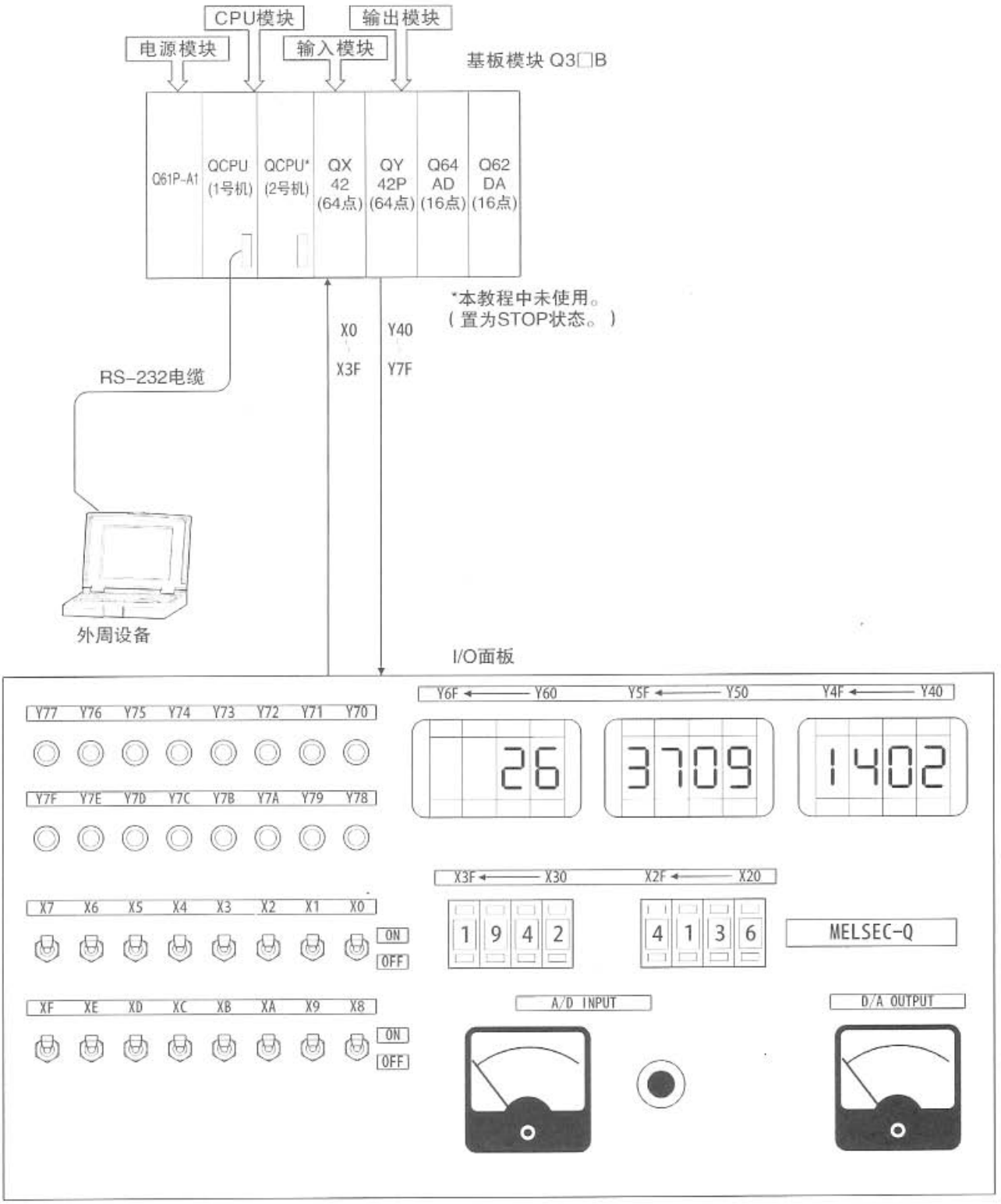
按照上述顺序，使用ST的顺控程序建立完毕。

备　忘　录

2 实习机系统、顺控程序的编写顺序

2.1 实习机的系统构成

实习使用的系统构成如下所示。



2.2 顺控程序的编写顺序

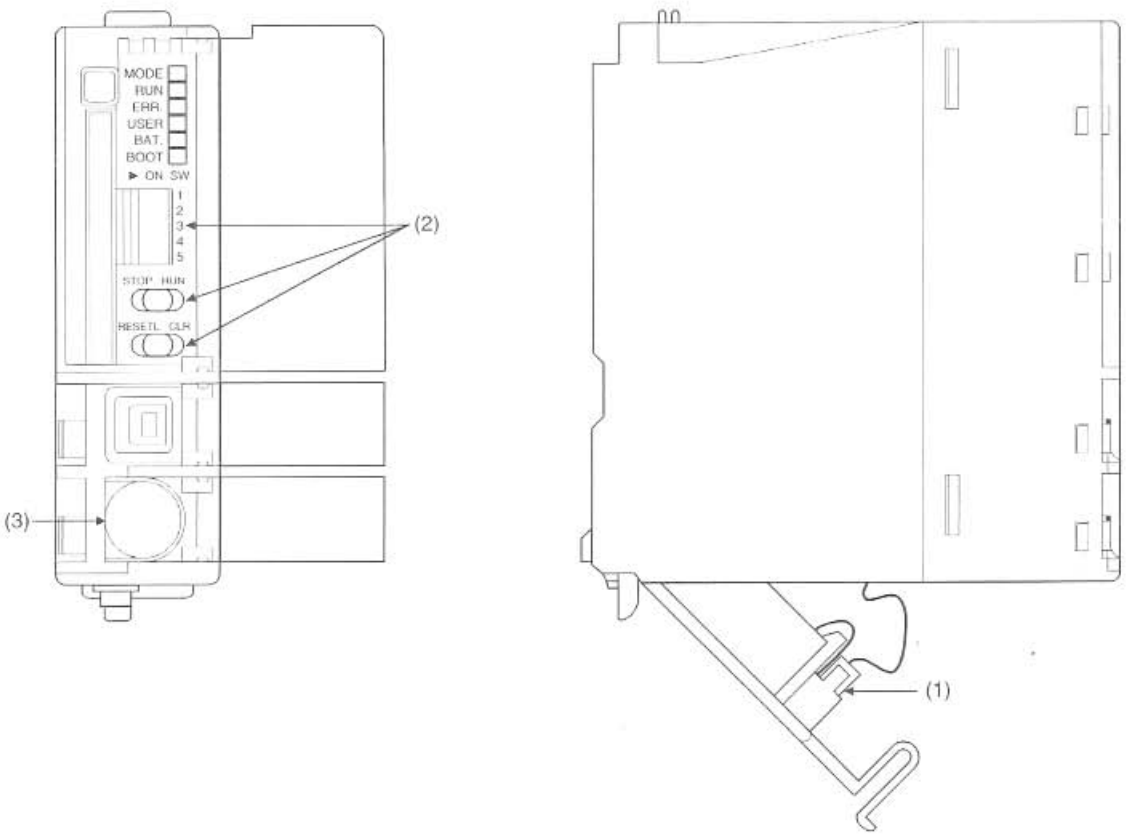
本教程中按照以下流程编写顺控程序。

2.3 CPU运行之前的准备

将顺控程序写入CPU之前，为了使得CPU在运行状态下正常动作，需要进行预先的准备工作。

2.3.1 CPU侧的连接

请按照下图所示，进行（1）～（3）连接器、开关的连接或者设置。
（下例为Q02HCPU，其他CPU同样。）



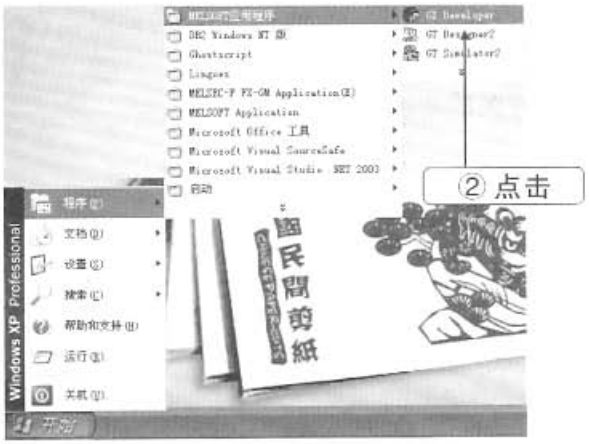
（1）连接电池
出厂时未连接电池，故需要连接电池连接器。

（2）设置开关
设置开关包括系统设置DIP开关与RUN/STOP开关.
① 系统设置DIP开关的设置
所有的开关均置为OFF。
（开关1为CPU写入、控制指示禁止开关置为OFF，解除禁止。）
② RUN/STOP开关的设置。
设置为STOP的位置。
③ RESET/L.CLR开关的设置
设置为中央的位置。

（3）连接RS-232电缆

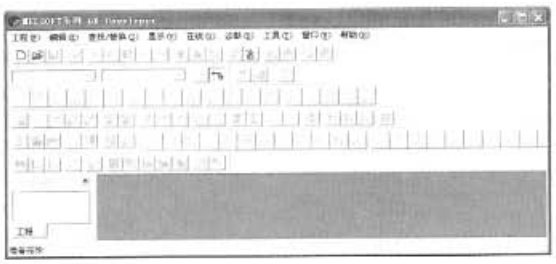
2.3.2 GX Developer 的操作

（1）新建ST程序的工程



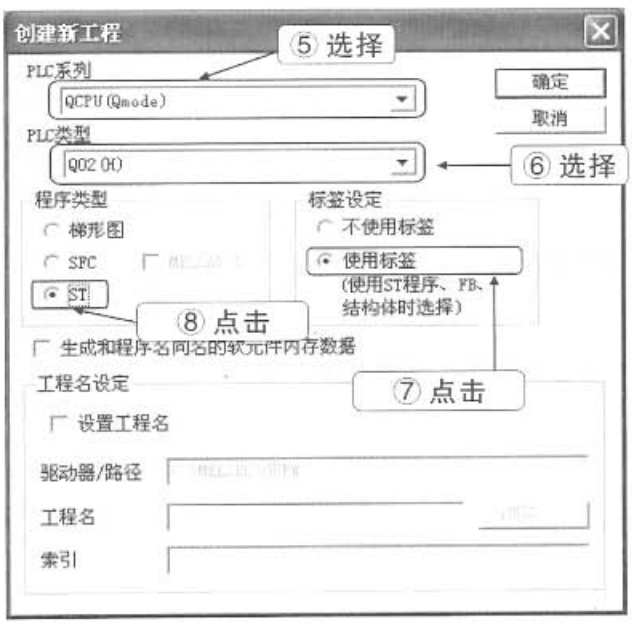
① 打开电脑电源。

② 点击[开始]－[程序]－[MELSOFT 应用程序]－[GX Developer]菜单。



③ 启动GX Developer。

④ 点击工具栏的 🗋 按钮。



⑤ 显示左侧的对话框，将PC系列设置为“QCPU(Q模式)”。

⑥ 将PC类型设置为“Q02(H)”。

⑦ 将标签设置设定为“使用标签”。

⑧ 将程序类别设置为“ST”。

转下页

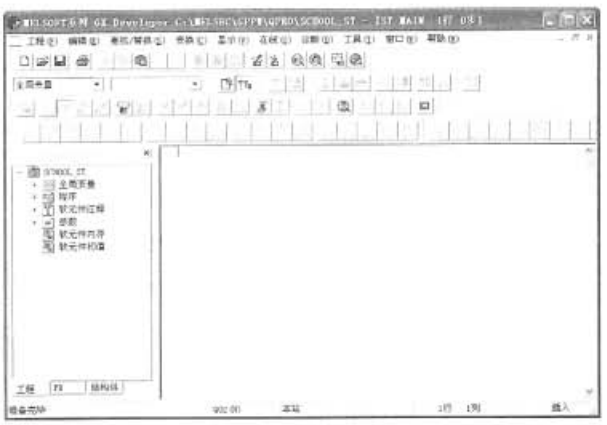
接上页

⑨ 选中“设置工程名”

⑩ 在驱动盘/路径栏中输入“C:\MELSEC\GPPW\QPRO”。

⑪ 在工程名栏中输入“SCHOOL_ST”。

⑫ 点击 确定 按钮



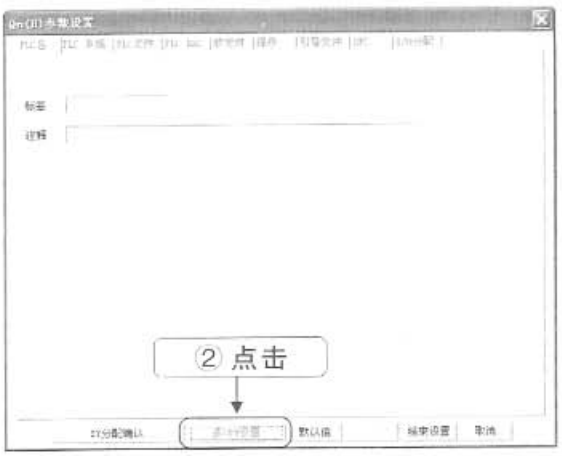
⑬ ST使用的工程新建完毕。

（2）多CPU参数设置（由1台CPU构成则不需设置）

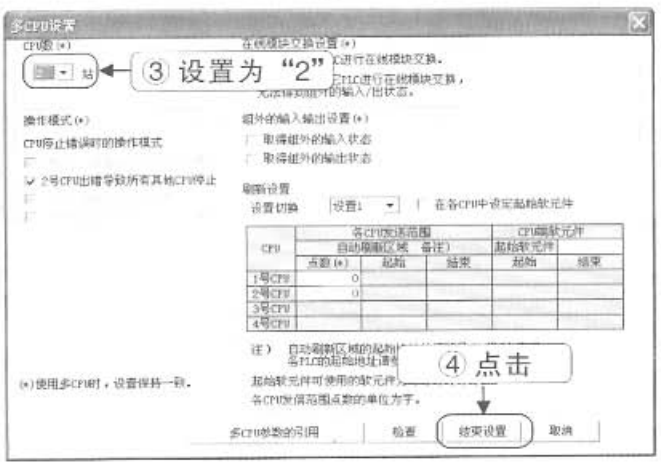
Q系列实习机为多CPU实习机结构，安装了2台CPU。

运行多CPU之前，需要进行参数设置，但本教程中未使用。

① 双击GX Developer的工程数据浏览中的“PLC参数”。



② 显示Qn（H）参数设置对话框，点击 多CPU设置 按钮。

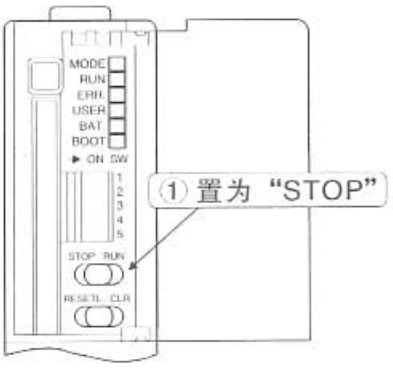


③ 将多CPU设置对话框的[CPU台数]设置为“2”台。

④ 点击 设置完成 按钮。

（3）格式化CPU的程序内存

对1号机的QCPU程序内存进行格式化。



① 将CPU(1号机、2号机)的[RUN/STOP]开关设置为[STOP]。

② 点击[在线]－[格式化PLC的内存]菜单。

③ 显示左侧的对话框，将目标存储器设置为“程序内存/软元件内存”。

④ 点击 执行 按钮。

⑤ 点击 是 按钮，进行格式化。

⑥ 格式化完成后，显示左侧对话框，点击 确定 按钮。

转下页

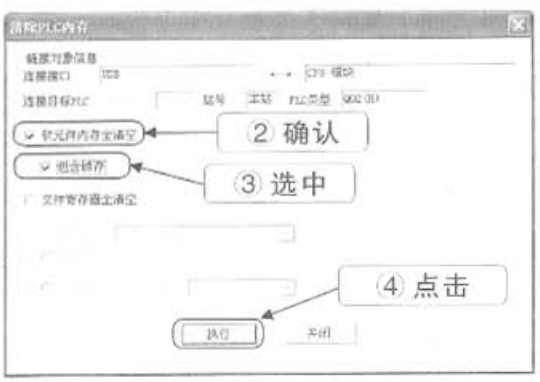
接上页

⑦点击 关闭 按钮，关闭对话框。

（4）全部清除CPU的软元件内存

对1号机的CPU软元件内存进行清除。

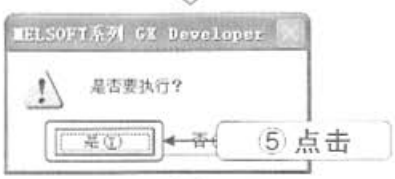
① 点击[在线]－[清除PLC内存]菜单。



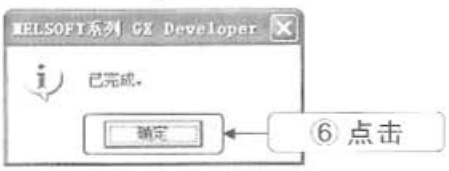
② 显示左侧的对话框，确认选中“软元件内存全部清除”。

③ 选中“包含锁存”

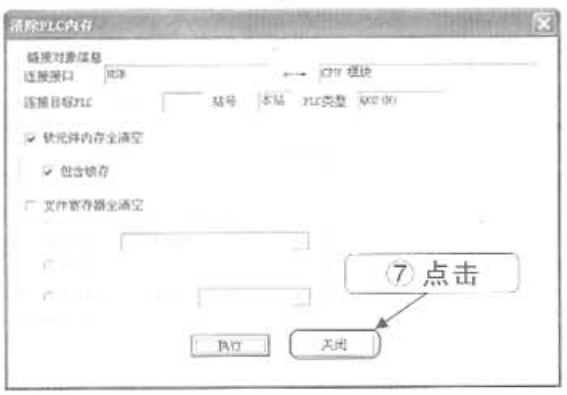
④ 点击 执行 按钮。



⑤ 点击 是 按钮，执行锁存软元件的清除操作。



⑥ 锁存软元件清除完成后，显示左侧的对话框，点击 确定 按钮。



⑦ 点击 关闭 按钮，关闭对话框。

(5) 向CPU写入

将设定的参数写入1号机的QCPU。

① 点击工具栏的 或者[在线]-[PC写入]菜单。

② 在[文件选择]分页下，点击 参数+程序 。

③ 点击 执行 按钮。

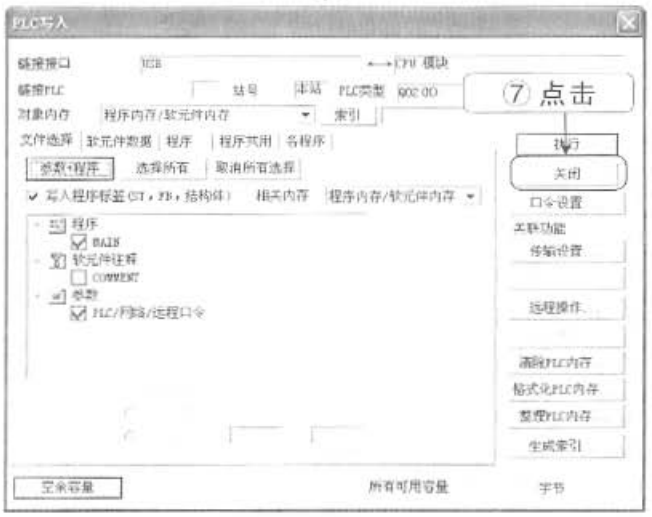
④ 参数或者程序已写入的情况下，确认覆盖，点击 是 按钮。

⑤ 显示正在写入对话框。

转下页

接上页

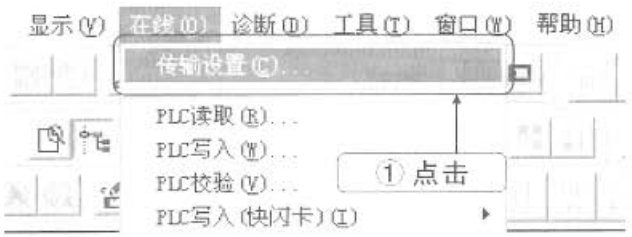
⑥写入完成后，显示[完成]消息，点击 确定 按钮。



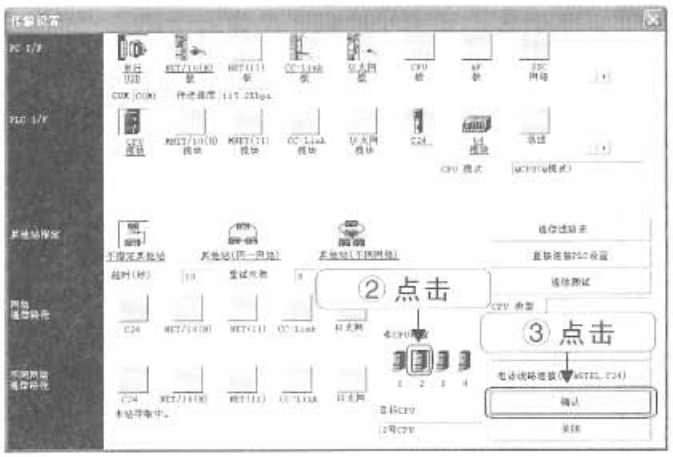
⑦点击 关闭 按钮，关闭对话框。

（6）向2号机QCPU的写入

对于2号机，写入与1号机相同的参数。



① 点击[在线]→[传输设置]。

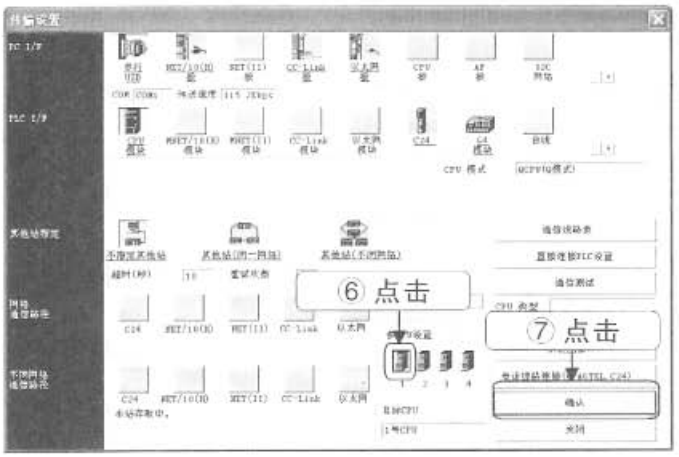


② 选择点击[多CPU设置]的“2”。

③ 点击 确认 按钮。

④ 对于2号机的QCPU，与1号机的QCPU相同，按照本项（3）～（5）的顺序进行操作。

- 格式化CPU的程序内存
- 全部清除CPU的软元件内存
- 向CPU写入参数



⑤ 点击[在线]→[传输设置]，显示传输设置的画面。

⑥ 选择点击[多CPU设置]的“1”。

⑦ 点击 确认 按钮。

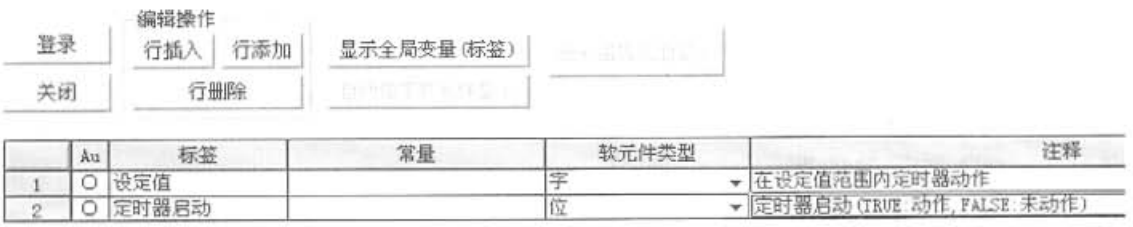
3 ST编程

本章在2.3.2项建立的工程下，使用ST语言进行编程。

（1）顺控程序

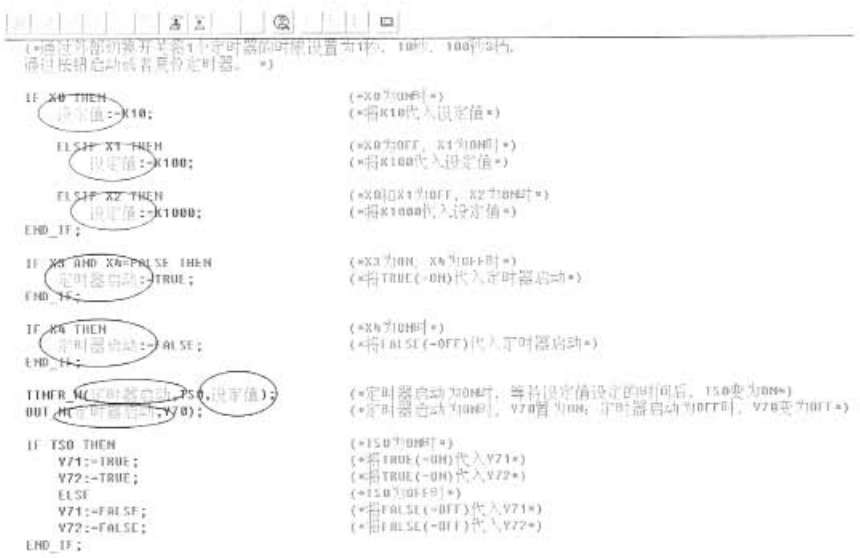
```
(*通过外部切换开关将1个定时器的时限设置为1秒、10秒、100秒3档。
通过按钮启动或者复位定时器。 *)

IF X0 THEN                              (*X0为ON时*)
    设定值:=K10;                        (*将K10代入设定值*)

    ELSIF X1 THEN                       (*X0为OFF，X1为ON时*)
        设定值:=K100;                   (*将K100代入设定值*)

    ELSIF X2 THEN                       (*X0和X1为OFF，X2为ON时*)
        设定值:=K1000;                  (*将K1000代入设定值*)
END_IF;

IF X3 AND X4=FALSE THEN                 (*X3为ON，X4为OFF时*)
    定时器启动:=TRUE;                   (*将TRUE(=ON)代入定时器启动*)
END_IF;

IF X4 THEN                              (*X4为ON时*)
    定时器启动:=FALSE;                  (*将FALSE(=OFF)代入定时器启动*)
END_IF;

TIMER_M(定时器启动,TS0,设定值);         (*定时器启动为ON时，等待设定值设定的时间后，TS0变为ON*)
OUT_M(定时器启动,Y70);                  (*定时器启动为ON时，Y70置为ON；定时器启动为OFF时，Y70变为OFF*)

IF TS0 THEN                             (*TS0为ON时*)
    Y71:=TRUE;                          (*将TRUE(=ON)代入Y71*)
    Y72:=TRUE;                          (*将TRUE(=ON)代入Y72*)
    ELSE                                (*TS0为OFF时*)
    Y71:=FALSE;                         (*将FALSE(=OFF)代入Y71*)
    Y72:=FALSE;                         (*将FALSE(=OFF)代入Y72*)
END_IF;
```

（2）全局变量和局部变量的定义



	Au	标签	常量	软元件类型	注释
1	○	设定值		字	在设定值范围内定时器动作
2	○	定时器启动		位	定时器启动(TRUE:动作,FALSE:未动作)

对顺控程序中使用的标签（变量）的类别、常数值、软元件类型、注释进行设置。
局部标签的设置必须与顺控程序主单元对应。
本例中程序的编写未使用全局变量。



：标签

① 双击“工程”分页的“程序”。

② 点击“MAIN”。

转下页

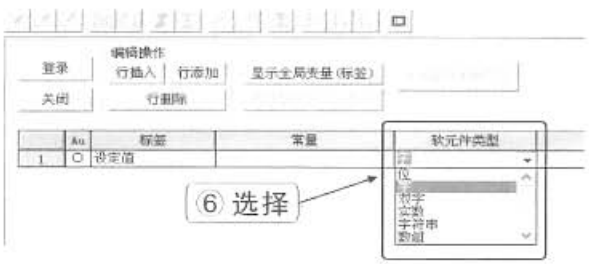
接上页

③ 双击“局部变量”。

④ 显示局部变量（局部标签）设置画面。

⑤ 设置标签名。在“标签”项目中输入“设定值”。

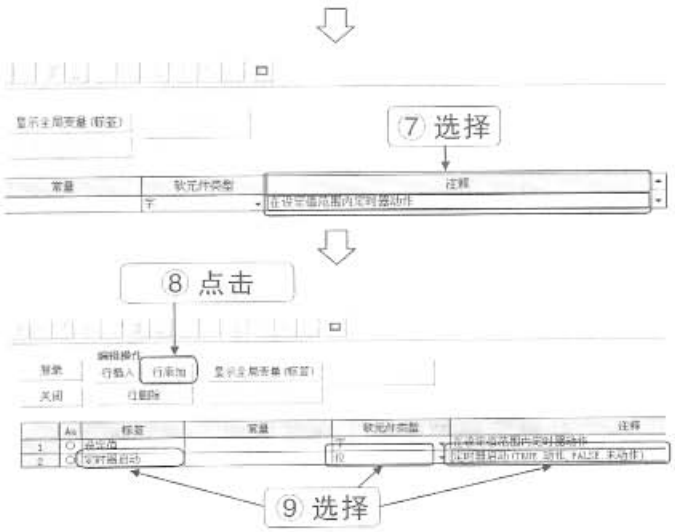
- 标签名限制在半角16字符以内（全角8字符以内）。
- ST程序的函数名、控制结构中使用的保留字符以及实软元件中使用的字符串不可以设置为标签名。



⑥ 设置“软元件类型”为“字”。

转下页

接上页

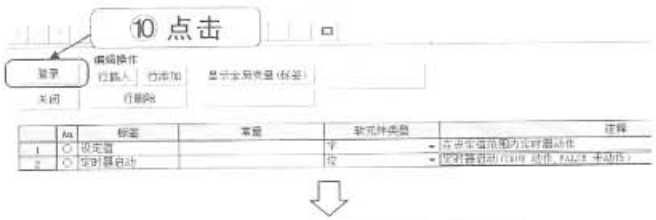


⑦ 在“注释”项目中输入“在设定值范围内定时器动作”。

- 注释限制在半角64字符以内（全角32字符以内）。

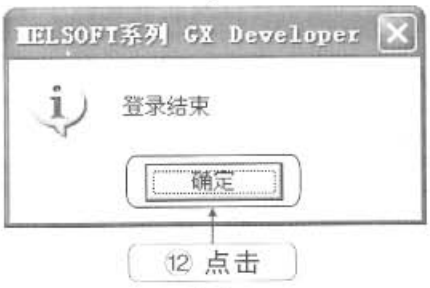
⑧ 点击 行添加 按钮。

⑨ 第2个标识设置如下.
标签：“定时器启动”
软元件类型：“位”
注释：“定时器启动（TRUE：动作，FALSE：未动作）”



⑩ 输入结束后，点击 登录 按钮。

⑪ 显示左侧对话框。点击 确定 按钮。



⑫ 点击 确定 按钮。

转下页

接上页

⑬ 标签的登录完成。

备　注

关于登录至标签的注释

从GX Developer的菜单选中[显示]–[标签信息]的情况下，在ST编辑画面中输入的标签上移动鼠标，则显示标签信息，可以对登录的注释进行确认。

标签信息的显示形式如下。

・未编译的情况下

标签名–>标签类别–>标签注释

・已编译的情况下

标签名–>标签类型–>标签注释–>软元件

对于全局变量，标签类别显示为“GLOBAL”，对于局部变量，标签类别显示为“LOCAL”。

要　点

（1）关于在局部标签设置画面下登录的标签。
局部标签设置画面下登记的标签，在程序编译时自动分配软元件。
自动分配软元件按照自动分配软元件的设置进行分配。
关于自动分配软元件的详细信息请参照3.3节。

（2）关于全局变量
标签也可以使用在工程全部顺控程序内均可使用的全局变量。
全局变量设置的详细信息请参照[GX Developer 操作手册]。

使用3.1节定义的局部变量编写顺控程序。

3.2.1 输入程序之前

在ST编辑画面下，能够以文本的形式自由地编写程序。但输入代码时需要注意以下几点。

（1）关于空格

输入空格，请使用半角Space・ Tab 键・ Enter 键。
全角空格不作为空白处理，编译时出错。
但是注释内可以使用全角空格。

（2）输入错误的确认方法

・输入已定义的标识、控制语句、注释时，字符的颜色改变。
如果字符的颜色没有改变，则可能是输入错误，或者标识未定义。
・控制语句以小写字母输入，也会自动转换为大写字母。

（3）关于本程序中使用的条件语句和函数

① 条件语句

・IF・・・ELSIF语句

布尔表达式（条件表达式）1的值为真（TRUE）时，执行语句1。布尔表达式1的值为假（FALSE）且布尔表达式2为真（TRUE）时，执行语句2。布尔表达式2的值为假（FALSE）且布尔表达式3的值为真（TRUE）时，执行语句3。

```
IF<布尔表达式1>THEN
    <语句1・・・>
ELSIF<布尔表达式2>THEN
    <语句2・・・>
ELSIF<布尔表达式3>THEN
    <语句3・・・>
END_IF;
```

布尔表达式：条件表达式　可使用的数据类型 BOOL（等同于梯形图的位）

② 函数

· OUT_M

将执行条件输出至指定的软元件。

[使用示例]

将执行条件X0输出至Y70。

```
OUT_M ( X0, Y70 ) ;
        ↑   ↑
        ①   ②
```

①：执行条件 可使用的数据类型 BOOL（等同与梯形图中的位）

②：ON/OFF控制对象 可使用的数据类型 BOOL（等同与梯形图中的位）

[参考]

与本函数等价的梯形图如下所示。

· TIMER_R

定时器（低速定时器、低速积分定时器）的线圈置为ON，经过设定的时间后（计测值>=设定值），则接点导通（A接点的情况）/非导通（B接点的情况）。

ST程序中，定时器、计数器的接点、线圈、当前值、使用各个独立的软元件进行表记。

（详细请参照附录1）

[使用示例]

执行条件X0为ON，则TS0为ON，计测至TVALUE后，时间到（计测值>=设定值），接点变为导通（A接点的情况）/非导通（B接点的情况）。

```
TIMER_M ( X0, TS0, TValue ) ;
          ↑   ↑     ↑
          ①   ②     ③
```

①：执行条件（仅在TRUE时执行函数）可使用的数据类型 BOOL（等同与梯形图中的位）

②：计测值TS、C软元件或STS、STC软元件 可使用的数据类型BOOL（等同与梯形图中的位）

③：定时器的设定值，定时器的设定值指定为常数的情况下，仅可以指定为10进制数。可使用的数据类型 ANY16（等同与梯形图中的字）

[参考]

与本函数等价的梯形图如下所示。

要 点

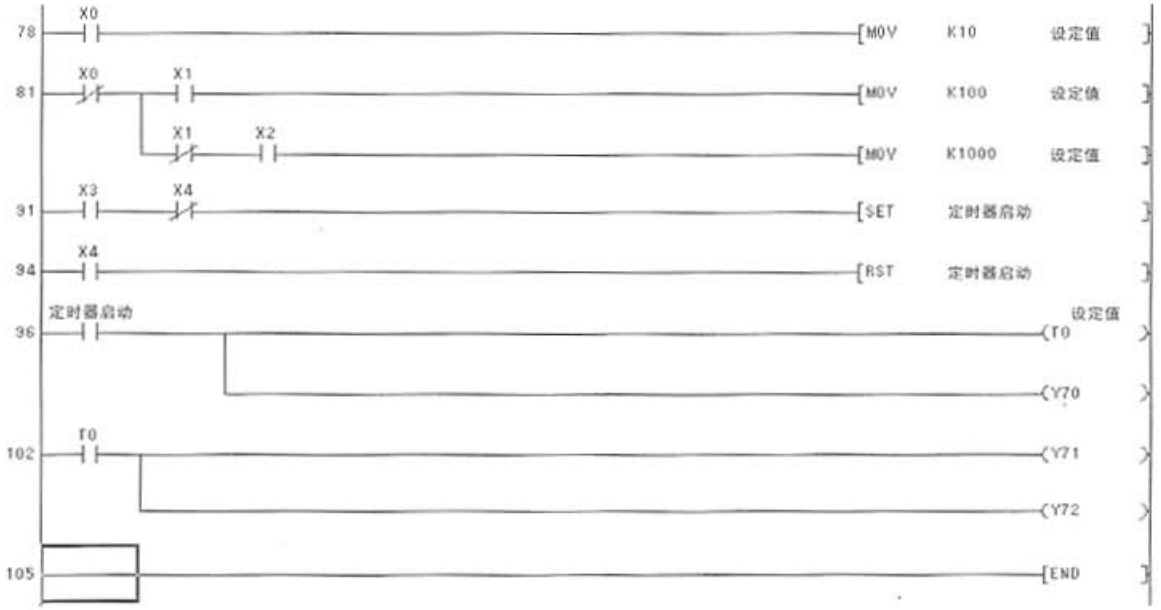
关于ST程序内使用的控制结构以及函数的详细信息请参照[QCPU（Q模式）编程手册（结构化文本篇）]

3.2.2 输入程序

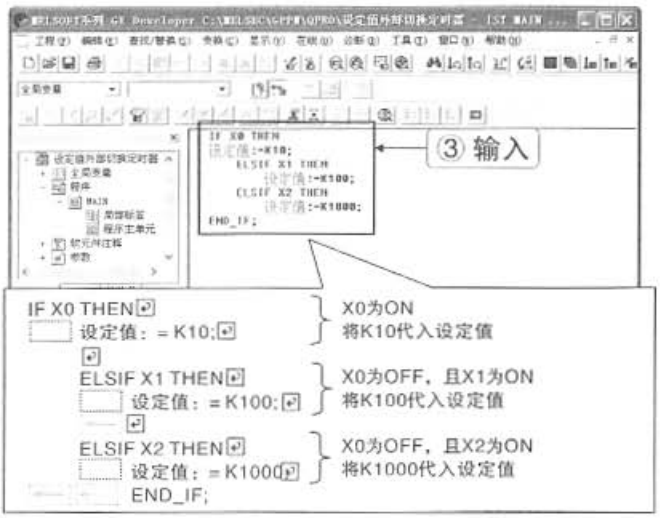
输入以下的ST程序。
与ST程序的动作等同的梯形图程序作为参考记载如下，以用于确认程序的动作内容等。

```
(*通过外部切换开关将1个定时器的时限设置为1秒、10秒、100秒3档。
通过按钮启动或者复位定时器。 *)

IF X0 THEN                          (*X0为ON时*)
    设定值:=K10;                    (*将K10代入设定值*)

    ELSIF X1 THEN                   (*X0为OFF，X1为ON时*)
        设定值:=K100;               (*将K100代入设定值*)

    ELSIF X2 THEN                   (*X0和X1为OFF，X2为ON时*)
        设定值:=K1000;              (*将K1000代入设定值*)
END_IF;

IF X3 AND X4=FALSE THEN             (*X3为ON，X4为OFF时*)
    定时器启动:=TRUE;               (*将TRUE(=ON)代入定时器启动*)
END_IF;

IF X4 THEN                          (*X4为ON时*)
    定时器启动:=FALSE;              (*将FALSE(=OFF)代入定时器启动*)
END_IF;

TIMER_M(定时器启动,T50,设定值);     (*定时器启动为ON时，等待设定值设定的时间后，T50变为ON*)
OUT_M(定时器启动,Y70);              (*定时器启动为ON时，Y70置为ON；定时器启动为OFF时，Y70变为OFF*)

IF T50 THEN                         (*T50为ON时*)
    Y71:=TRUE;                      (*将TRUE(=ON)代入Y71*)
    Y72:=TRUE;                      (*将TRUE(=ON)代入Y72*)
    ELSE                            (*T50为OFF时*)
    Y71:=FALSE;                     (*将FALSE(=OFF)代入Y71*)
    Y72:=FALSE;                     (*将FALSE(=OFF)代入Y72*)
END_IF;
```

参考）与本ST程序动作等同的梯形图程序



① 双击“工程”分页的“程序主体”。



Enter：键输入位置　　Tab键输入位置
Back Space键输入位置

② 显示ST编辑画面。

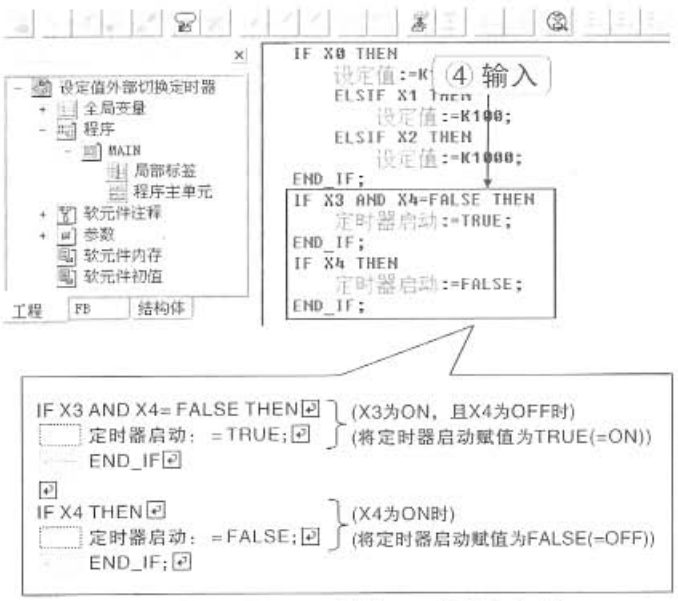
③ 点击第一行的开头附近，显示光标后，输入如下代码。

```
IF X0 THEN
    设定值: =K10;
    ELSIF X1 THEN
        设定值: =K100;
    ELSIF X2 THEN
        设定值: =K1000;
END_IF;
```

- 常数为10进制的情况下可以直接输入。
  例）设定值: =10;
- 为了提高程序的可读性，使用 TAB 键/ Enter 键/ Back Space 键/ Space 键输入或者删除半角空白、制表符以及换行，以保证格式的规整。

转下页

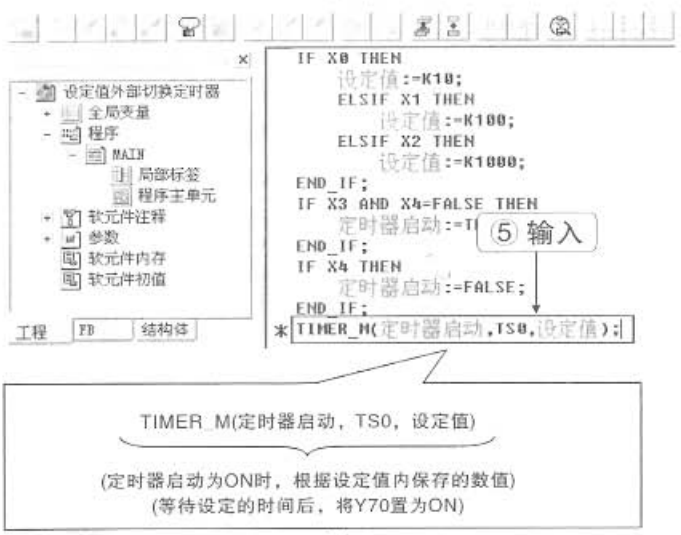
接上页



④ 继续输入以下代码。

```
IF X3 AND X4=FALSE THEN
   定时器启动: =TURE;
END_IF;

IF X4 THEN
   定时器启动: =FALSE;
END_IF;
```

“X3 AND X4=FALSE”也可表述为“X3&NOT X4”。



⑤ 在④之后继续输入如下代码。

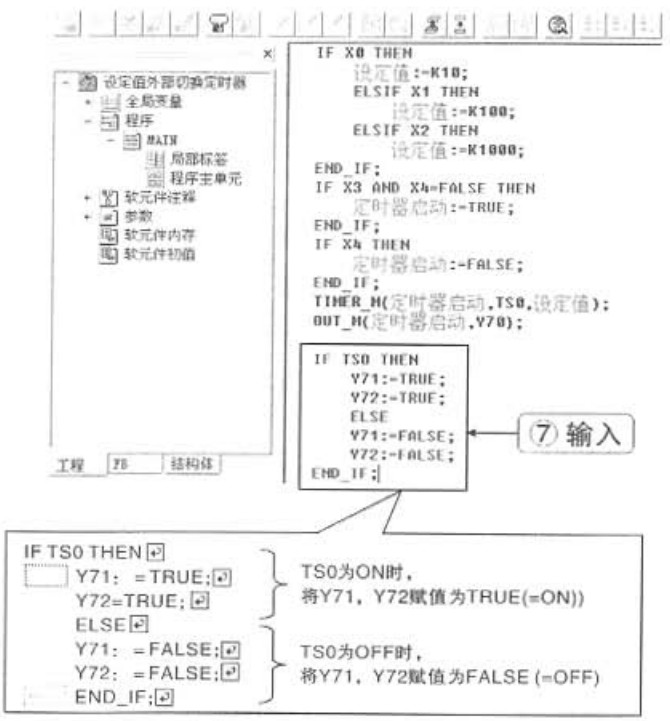
```
TIMER_M(定时器启动，TS0，设定值)
```

转下页

接上页

⑥ 在⑤之后继续输入如下代码。

```
OUT_M ( 定时器启动，Y70 ) ;
```



⑦ 在⑥之后继续输入如下代码。

```
IF TS0 THEN
    Y71: = TRUE;
    Y72: = TRUE;
  ELSE
    Y71: = FALSE;
    Y72: = FALSE;
END_IF;
```

备　注

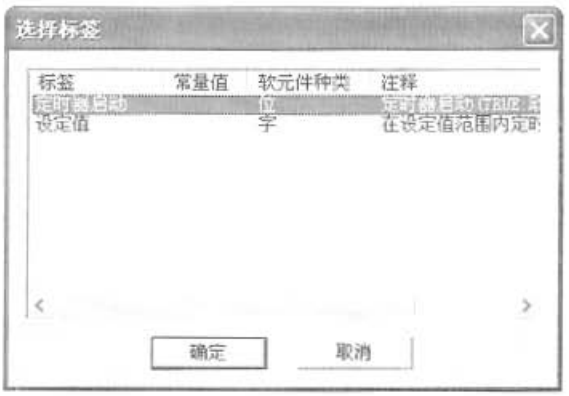
标签和函数除了直接输入的方式以外，还有使用“标签选择功能”、“函数选择功能”进行输入的方式。

（1）使用标签选择功能的情况

① 按照下列任意一项的顺序使用标签选择功能。

- 从菜单选择[编辑]→[标签选择]
- 从右键快捷菜单选择[标签选择]
- 按下 F11 键

② 按照①的顺序进行操作，则显示如下标签选择对话框。
点击选择希望输入的标签，并点击 OK 按钮，
则选择的标签被输入在光标的位置



标签选择对话框

（2）使用函数选择功能的情况

① 按照下列任意一项的顺序使用函数选择功能。

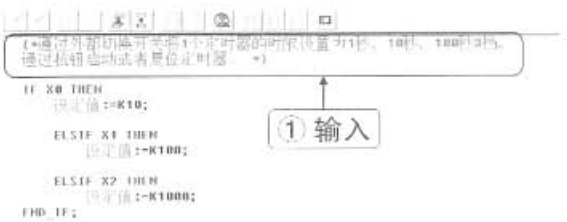
- 从菜单选择[编辑]→[函数选择]
- 从右键快捷菜单选择[函数选择]
- 按下 Shift + F11 键

② 按照①的顺序进行操作，则显示如下函数选择对话框。
点击选择希望输入的函数，并点击 OK 按钮，
则选择的函数被输入在光标的位置

函数选择对话框

3.2.3 添加注释

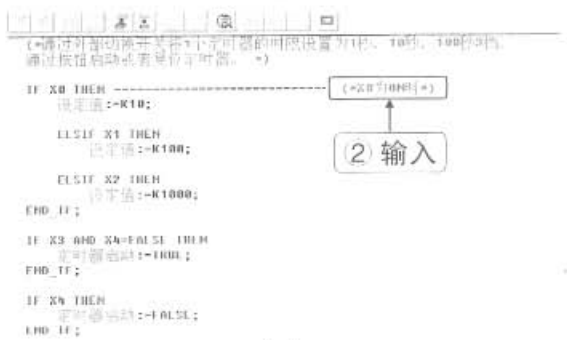
在3.2.2项编写的程序中添加注释。
使用“(*”和“*)”将注释文字包围起来。
使用注释可以提高程序的可读性。



① 使光标显示在第1行开头位置，从第1行的开头输入如下注释。

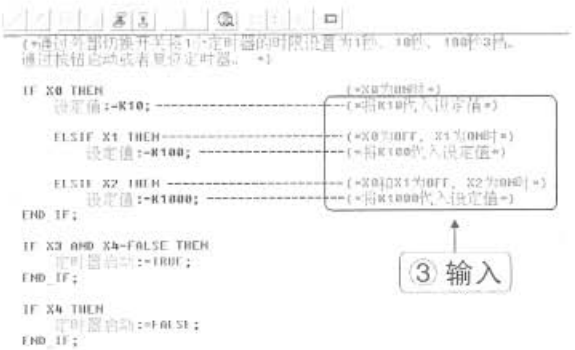
```
(*通过外部切换开关将1个定时器的时限设置为1秒、10秒、100秒3档。↵
通过按钮启动或者复位定时器。
*)↵
↵
```

↵ : Enter键的输入位置。



② 将光标移动至第4行[IF X0 THEN]的右侧，按下 Tab 键，输入如下注释。

```
(*X0为ON时*)
```



③ 在5～11行输入以下注释。

```
(*将K10代入设定值*)
(*X0为OFF，X1为ON时*)
(*将K100代入设定值*)
(*X0和X1为OFF，X2为ON时*)
(*将K1000代入设定值*)
```

转下页

接上页

④ 在14～30行输入以下注释

```
(*X3为ON，X4为OFF时*)
(*将TURE（=ON）代入定时器启动*)
(*X4为ON时*)
(*将FALSE(=OFF)代入定时器启动*)
(*定时器启动为ON时，等待设定值设
定的时间后，TS0变为ON*)
(*定时器启动为ON时，Y70置为ON，
定时器启动为OFF时，Y70变为OFF*)
(*TS0为ON时*)
(*将TURE（=ON）代入Y71*)
(*将TURE（=ON）代入Y72*)
(*TS0为OFF时*)
(*将FALSE（=OFF）代入Y71*)
(*将FALSE（=OFF）代入Y72*)
```

至此，顺控程序的注释添加完毕。

要　点

（1）请注意输入为如下形式的情况下，不被识别为注释。

```
开关将1个定时器的时限设置为1秒、10秒、100秒3档，
若复位定时器。 *)

                    (*X0为ON时*)
0;                  (*将K10代入设定值*)

HEN                 (*X0为OFF，X1为ON时*)
:=K100;             (*将K100代入设定值*)

HEN                 (*X0和X1为OFF，X2为ON时*)
:=K1000;            (*将K1000代入设定值* )

ELSE THEN           (*X3为ON，X4为OFF时*)X3为ON时*)
:=TRUE;             (*将TRUE(=ON)代入定时器启动*)
```

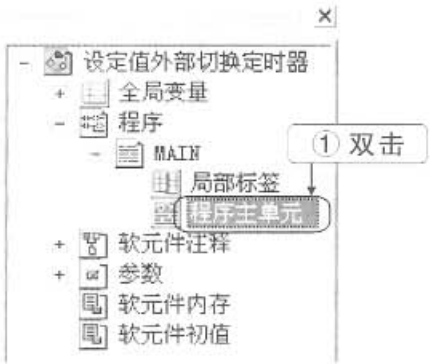
“(”和“*”之间有空格，故不被识别。

注释中包含注释，故不被识别。

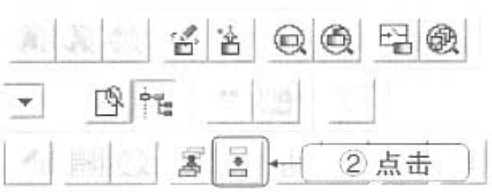
（2）与梯形图中使用的申明（statement）、说明（note）、软元件注释（comment）不同。

3.3 将顺控程序编译为可以在PLC CPU上运行的程序（编译（Compile））

对顺控程序进行编译（Compile），使其可以在PLC CPU上运行。



① 从工程数据浏览双击“程序”→“MAIN”→“程序主单元”。



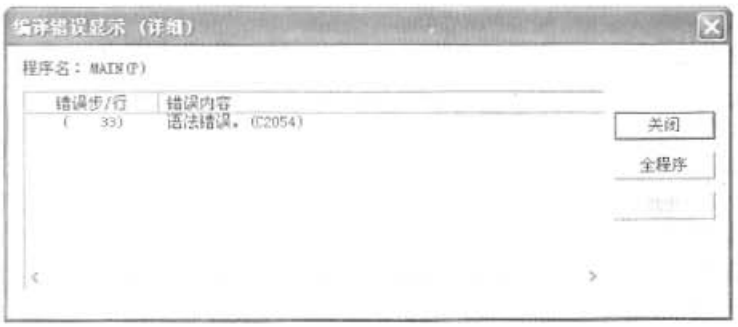
② 点击 按钮。

③ 显示左侧对话框，点击 否 按钮。

④ 顺控程序的编译完成。

要　点

顺控程序中存在错误的情况下，编译时显示如下对话框。
请点击选择错误内容，点击 跳过 按钮。
光标移动到发生编译错误的位置，请对程序进行修改。

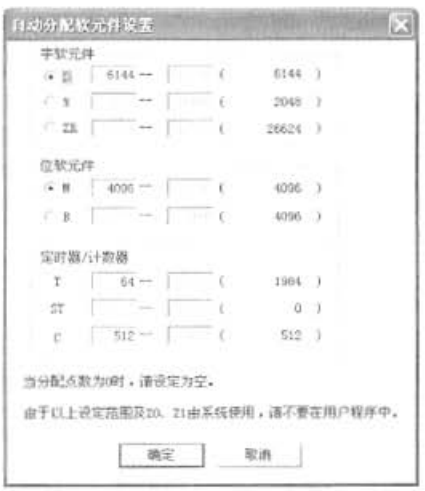


备　注

关于已编译的顺控程序
已编译（Compile）的顺控程序，按照自动分配软元件的设置，将软元件分配至局部标签。
编写顺控程序，请避开自动分配的软元件。

（1）自动分配使用的软元件的范围
自动分配软元件的设置，默认情况下范围如下。
① 字软元件：D6144～D12287
② 位软元件：M4096～M8191
③ 定时器：T64～T2047
④ 计数器：C512～C1023
分配至各标识的软元件，从最终号码（D12287，M8191，T2047，C1023）开始顺次使用。

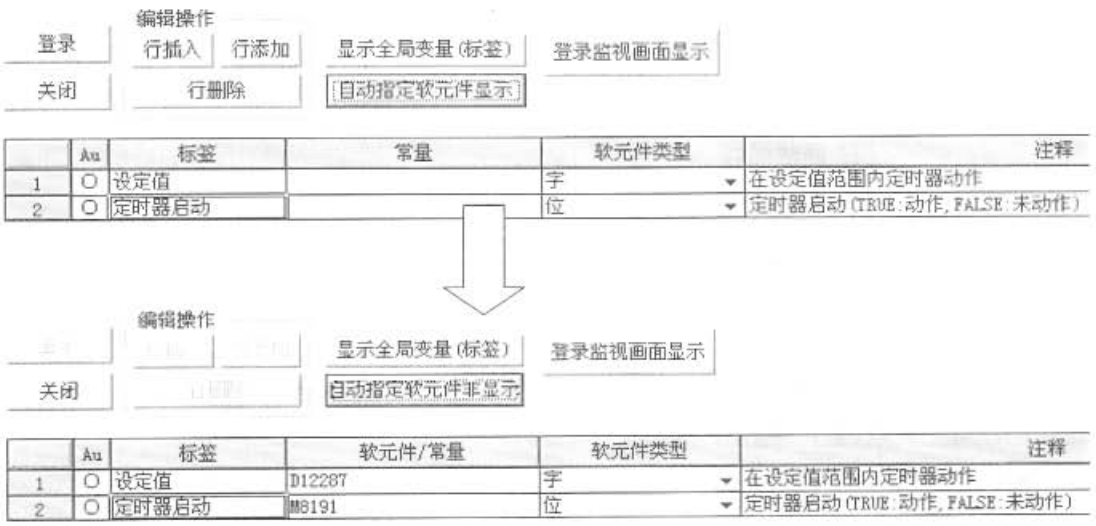
更改软元件范围的情况下，显示ST编辑画面时，点击[编辑]→[自动分配软元件设置]菜单，在自动分配软元件设置画面下进行更改。



自动分配软元件设置画面

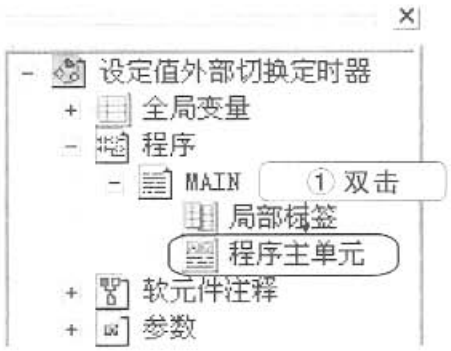
（2）自动分配软元件的确认方法

编译（Compile）后，希望对实际自动分配的软元件进行确认的情况下，在局部变量(局部标签)设置画面下，点击“自动分配软元件显示按钮”。自动分配至各标签的软元件显示在“软元件/常量”项目中。



3.4 将顺控程序写入PLC · CPU（PC写入）

将已编译的顺控程序写入PLC · CPU。



① 从工程数据浏览双击“程序”→“MAIN”→“程序主单元”。

② 点击工具栏的 按钮。

③ 显示PLC写入对话框，点击 参数+程序 按钮。

④ 设置目标存储器为“程序存储器/软元件存储器”。

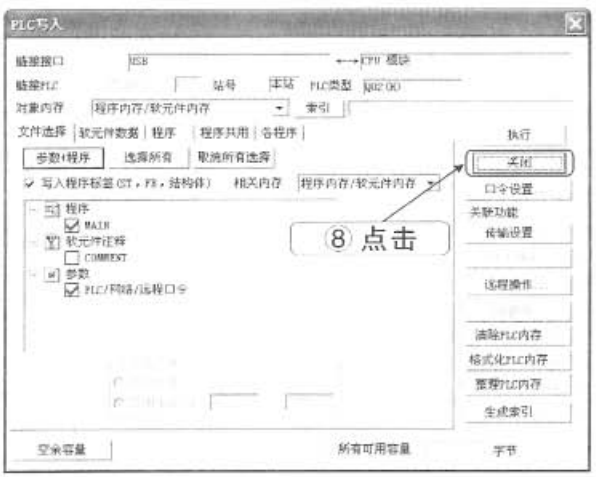
⑤ 点击 执行 按钮。

⑥ 参数或者程序已经写入的情况下，需要确认覆盖，点击 是 按钮。

转下页

接上页

⑦ 写入完成后，显示[完成]消息，点击 确定 按钮。



⑧ 点击 关闭 按钮，关闭对话框。

3.5 从PLC CPU读出顺控程序（PLC读取）

将写入PLC CPU的顺控程序读出。

① 点击工具栏的 按钮，保存工程数据。

② 点击工具栏的 按钮，新建工程。

③ 显示左侧对话框，将PLC系列设置为“QCPU（Q模式）”。

④ 将PLC类型设置为“Q02（H）”。

⑤ 将标签设置设定为“使用标签”。

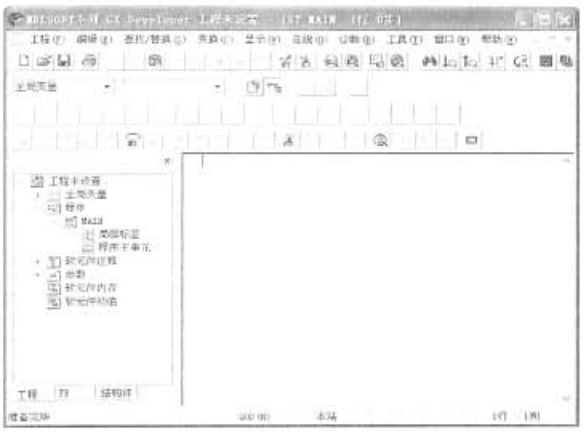
⑥ 将程序类型设置为“ST”。

⑦ 点击 确定 按钮。

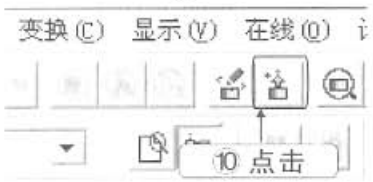
⑧ 确认结束工程，点击 是 按钮。

转下页

接上页



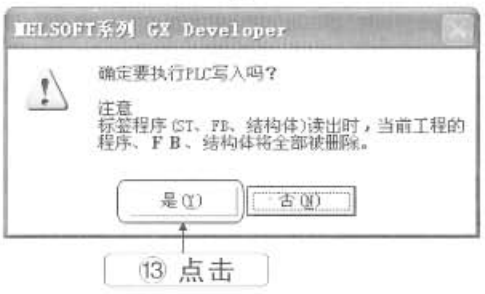
⑨ 工程被新建。



⑩ 点击工程的 按钮（或者F3键）。

⑪ 显示PLC读取对话框，点击 参数+程序 按钮。

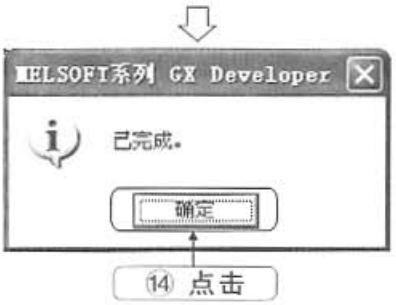
⑫ 点击 执行 按钮。



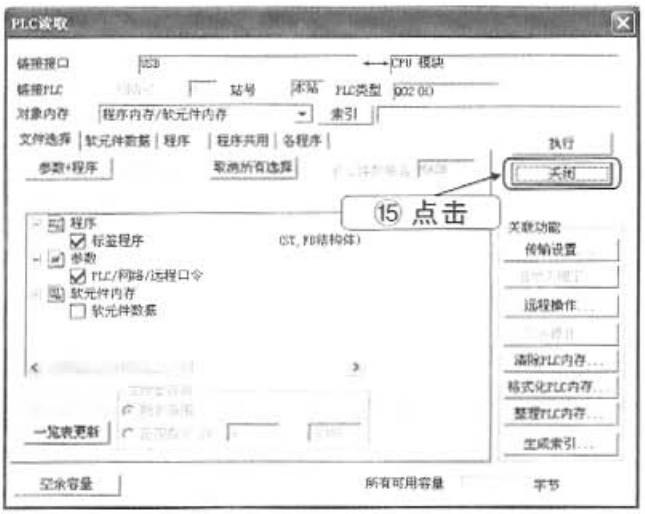
⑬ 点击 是 按钮，确认PLC读取。

转下页

接上页



⑭ PLC读取完成后，显示[完成]消息，点击 确定 按钮。

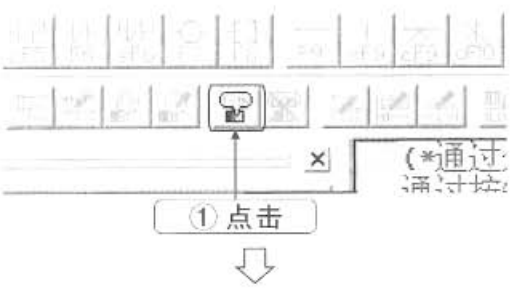


⑮ 点击 关闭 按钮，关闭对话框。

3.6 监视、测试顺控程序

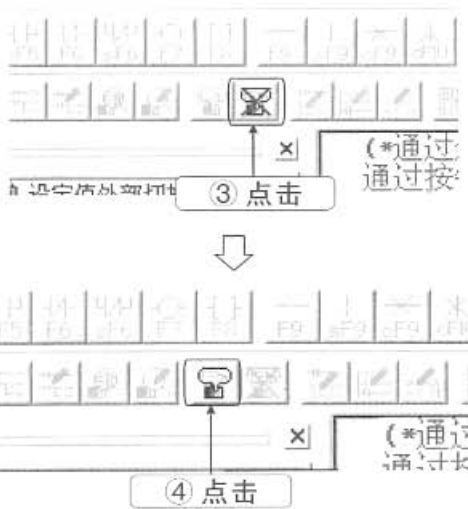
3.6.1 监视顺控程序

对写入PLC・CPU的顺控程序进行监视，以确认PLC・CPU的动作状态。



① 点击工具栏的 按钮(或按下F3键)，开始顺控程序的监视。

② 在ST画面下显示的标签显示在监视画面下的相同行。



③ 停止监视时，点击工具栏的 按钮（或者按下Alt+F3键）。

④ 继续进行监视时，点击工具栏的 按钮（或者按下F3键）。

备　注

在ST画面下各行使用的标签在监视画面下的相同行以“标签 = 监视值”的形式显示。
1行中有多个相同变量的情况下，只显示第1个变量，第2个及以后的变量均不显示。

变量类型	ST 编辑画面	监视画面	备　注
位	Input：= TRUE; Input：= FALSE;	Input = 1 Input = 0	TRUE　1 FALSE　0
字	Word1 := –32767;	Word1 = –32767	10 进制数 6 字符
		Word1 = H8001	16 进制数 5 字符
实数	Result := 340282.338;	Result = 3.403e+005	
字符串	Str1：= “ABCDEFGH”；	Str1 = 16961	10 进制数　字符串的第一个字以 10 进制数显示　6 字符
		Str1 = H4241	16 进制数　字符串的第一个字以 16 进制数显示　6 字符

要　点

- 10进制数和16进制数可以通过[在线]→[监视]→[当前值监视切换（10进制）]或者[当前值监视切换（16进制]进行切换。
- 监视画面的背景颜色与ST编辑画面相同，显示文字可以通过[工具]→[显示色改变]，使用所选择的显示颜色。

3.6.2 测试顺控程序（软元件测试）

对PLC・CPU的位软元件进行强制ON/OFF操作，更改字软元件的当前值，以确认所编写的顺控程序的动作处理。

① 按照3.6.1的顺序进行操作，进行监视模式。

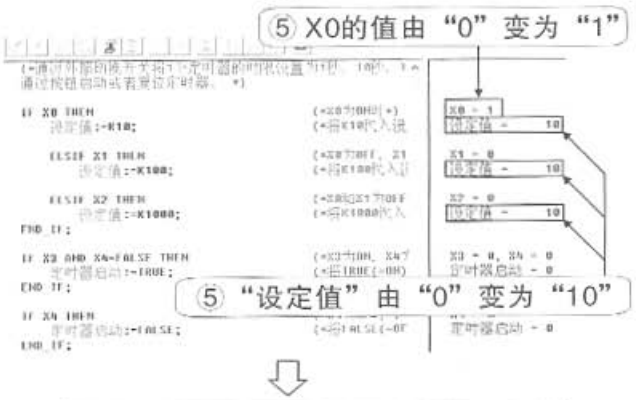
② 点击菜单的[在线]→[调试]→[软元件测试]。

③ 显示软元件测试对话框，输入“X0”至位软元件。

④ 点击 强制ON 按钮。

转下页

接上页



⑤ 将监视画面的X0值更改为“1”，并将设定值更改为“10”。

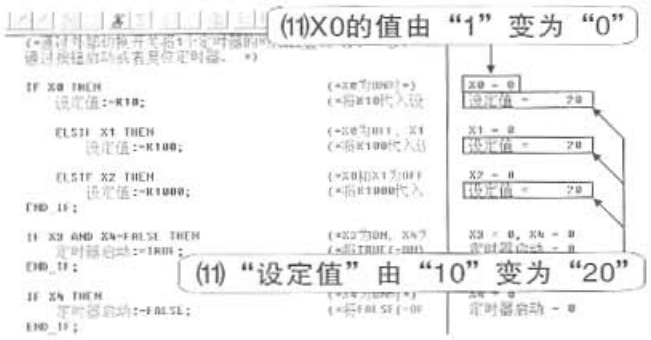
⑥ 输入“X0”至软元件测试对话框中位软元件的软元件栏。

⑦ 点击 强制OFF 按钮。

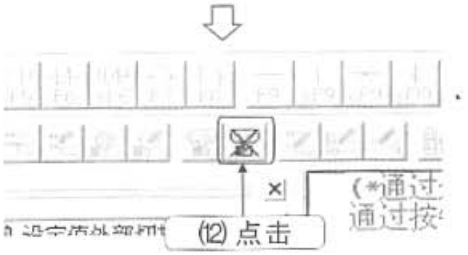
⑧ 选中字软元件的软元件复选框，输入“设定值”。

⑨ 输入设定值为“20”。

⑩ 点击 设置 按钮。



(11) 监视画面的X0值被更改为“0”，并将设定值更改为“20”。



⑫ 点击工具栏的 按钮，结束监视。

已完成监视、测试的顺控程序，如果未按照预期进行动作，或者希望提高程序效率的情况下，需要修改程序。
在此，请按如下方法改变局部标签。

（1）顺控程序的修改点

```
(*通过外部切换开关将1个定时器的时限设置为1秒、10秒、100秒3档。
通过按钮启动或者复位定时器。 *)
IF X0 THEN                          (*X0为ON时*)
    设定值:=K10;                    (*将K10代入设定值*)
ELSIF X1 THEN                       (*X0为OFF, X1为ON时*)
    设定值:=K100;                   (*将K100代入设定值*)
ELSIF X2 THEN                       (*X1为OFF, X2为ON时*)
    设定值:=K1000;                  (*将K1000代入设定值*)
END_IF;

IF X3 AND X4=FALSE THEN             (*X3为ON, X4为OFF时*)
    定时器启动:=TRUE;               (*将TRUE(=ON)代入定时器启动*)
END_IF;

IF X4 THEN                          (*X4为ON时*)
    定时器启动:=FALSE;              (*将FALSE(=OFF)代入定时器启动*)
END_IF;

TIMER_M(定时器启动,T0,设定值);      (*定时器启动为ON时，等待设定值设定的时间后，T0变为ON*)
OUT_M(定时器启动,Y70);              (*定时器启动为ON时，Y70置为ON；定时器启动为OFF时，Y70变为OFF*)

IF T0 THEN                          (*T0为ON时*)
    Y71:=TRUE;                      (*将TRUE(=ON)代入Y71*)
    Y72:=TRUE;                      (*将TRUE(=ON)代入Y72*)
    ELSE                            (*T0为OFF时*)
    Y71:=FALSE;                     (*将FALSE(=OFF)代入Y71*)
    Y72:=FALSE;                     (*将FALSE(=OFF)代入Y72*)
END_IF;
```

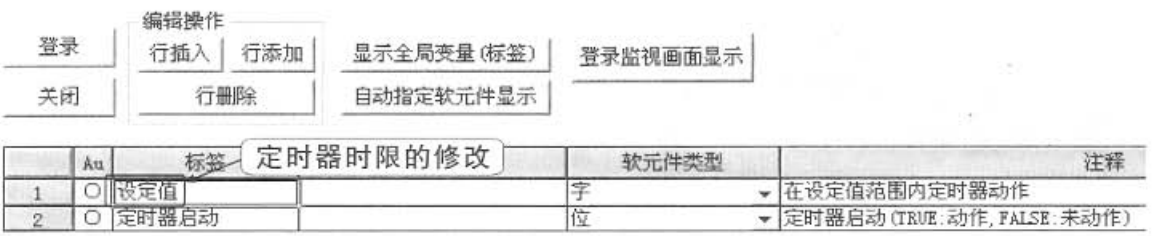
定时器时限的修改

定时器时限的修改

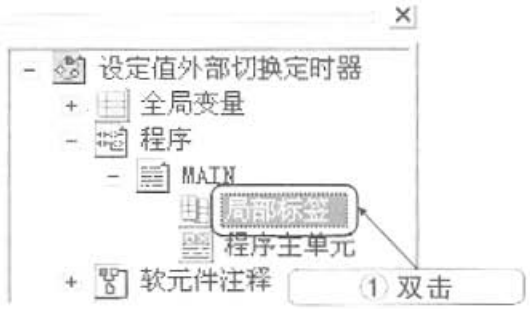
定时器时限的修改

定时器时限的修改

（2）局部标签的修改点



	Au	标签		软元件类型	注释
1	○	设定值		字	在设定值范围内定时器动作
2	○	定时器启动		位	定时器启动(TRUE:动作, FALSE:未动作)



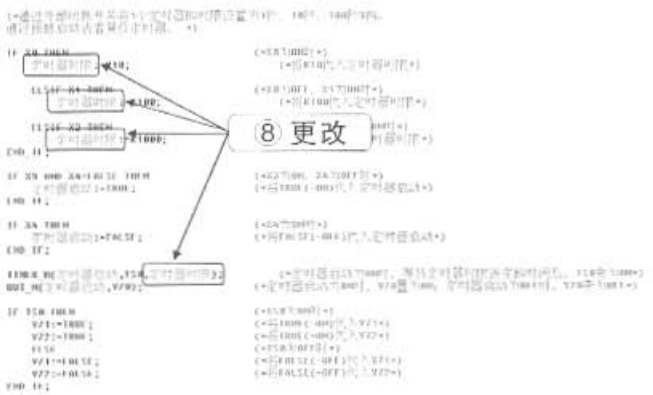
① 双击“局部标签”。

② 显示局部变量（局部标签）设置对话框。

③ 将标签名“设定值”改为“定时器时限”。

④ 点击 登记 按钮。

⑤ 双击“程序主单元”



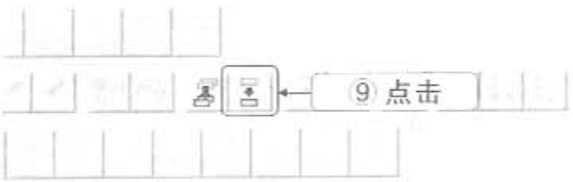
⑥ 显示ST编辑画面。

⑦ 局部标签“设定值”的文字颜色变为黑色，确认其已不是标签。

⑧ “设定值”已更改为“定时器时限”。

转下页

接上页



⑨ 点击工具栏的 按钮，编译(Compile)顺控程序。

检查出错误的情况下，请参照3.3节进行修改。

⑩ 编译(Compile)完成后，显示左侧对话框，点击 否 按钮。

⑪ 按照与3.4节相同的顺序，将顺控程序写入PLC CPU。

要　点

（1）关于修改的程序数据

从PLC CPU读取新程序的情况下，请预先建立使用标签的工程。

（2）关于调试功能

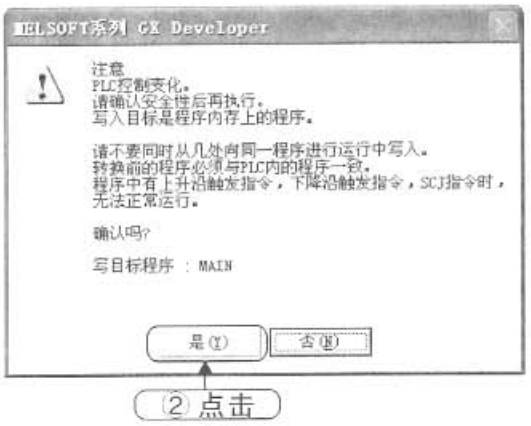
对于使用ST语言编写的程序，GX Developer支持以下调试功能。
使用调试功能时，需要使用GX Simulator Ver.6.16以上版本。

- [执行至断点]：程序执行至任意设定的断点处，进行调试。
- [单行执行]：单行执行程序，进行调试。

关于调试功能的详细信息，请参照[GX Developer 操作手册[结构化文本篇]]。

将修改后的顺控程序写入运行状态中的PLC CPU。
运行中写入可以将修改后程序的编译以及向PLC CPU的写入操作一次执行完毕。

① 点击[变换]→[变换/编译（运行中写入）]菜单。



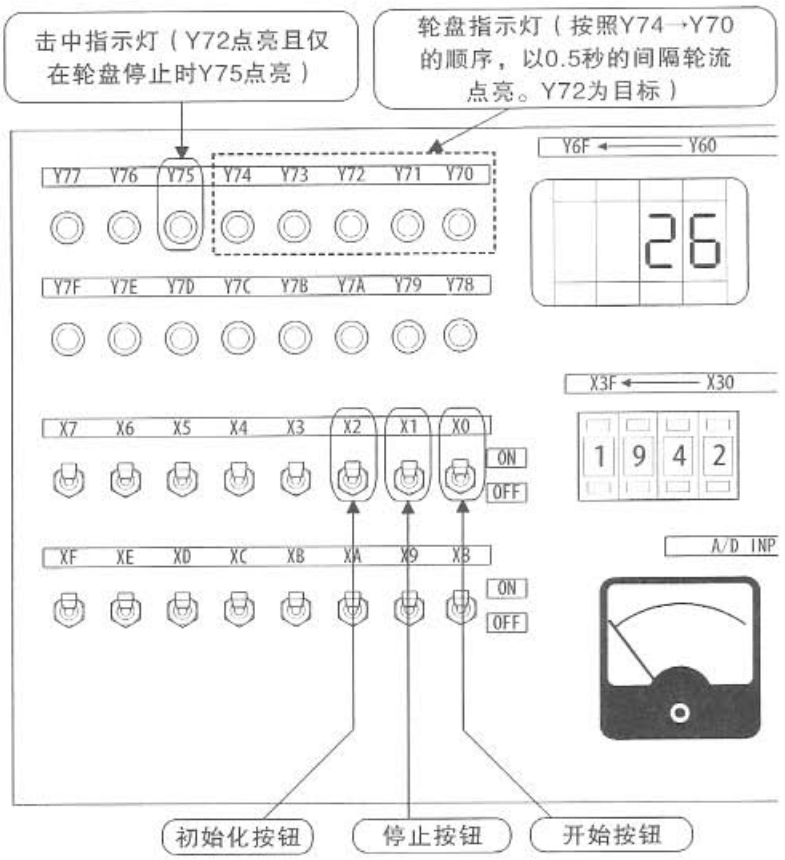
② 显示运行中写入确认对话框，点击 是 按钮。

③ 显示运行中写入完成消息，点击 是 按钮。

4 练习问题

4.1 练习问题1 轮盘程序

按下开始按钮（X0），指示灯开始闪烁，按下停止按钮时（X1），如果轮盘的指示灯停止在“击中”的位置，则显示击中（Y75）。
按下初始化按钮（X2），则显示击中的指示灯熄灭，可以再次使用轮盘。



4.1.1 程序内容

路径名	A:￥SCHOOL
工程名	练习1
程序名	MAIN

[顺控程序]

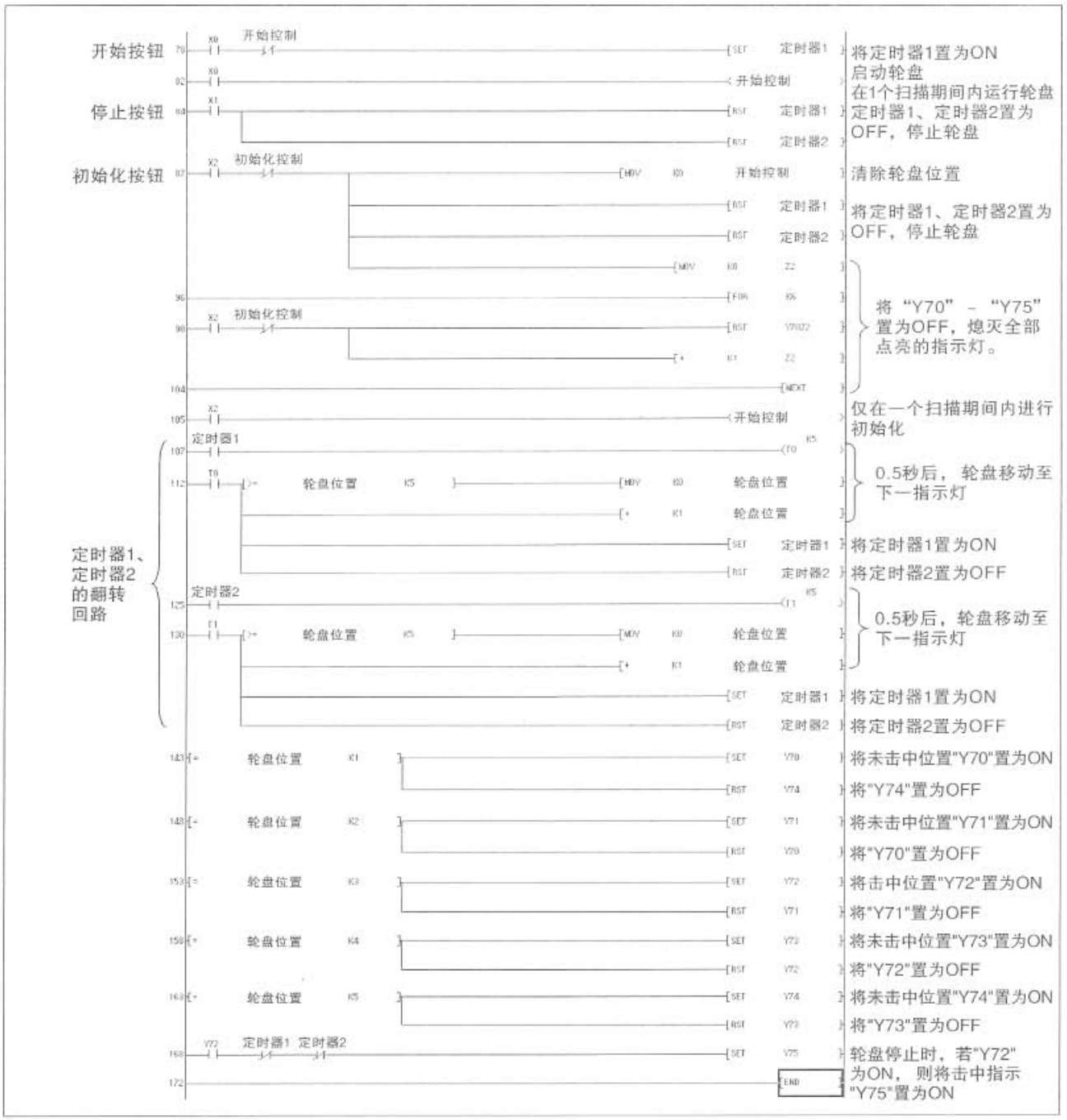
・ST程序

```
(开始)
IF X0 AND 开始控制=FALSE THEN                    (开始按钮 "X0" 为ON, "开始控制" 为OFF时)
   定时器1: =TRUE;                               ( "定时器1" 变为ON, 轮盘启动)
END_IF;

OUT_M(X0, 开始控制)                              (仅在 "X0" 为ON期间, 将 "开始控制" 置为ON)
(停止)
IF X1 THEN (轮盘停止按钮 "X1" 为ON时)            (轮盘停止按钮 "X1" 为ON时)
   定时器1: =FALSE;                              ( "定时器1" 变为OFF, 轮盘停止)
   定时器2: =FALSE;                              ( "定时器2" 变为OFF, 轮盘停止)
END_IF;

(初始化)
IF X2 AND 初始化控制 =FALSE THEN                 (初始化按钮 "X2" 为ON, "初始化控制" 为OFF时)
   轮盘位置: =K0; 定时器1: =FALSE; 定时器2: =FALSE;  (轮盘停止)
   FOR Z2: =0 TO 5 BY 1 DO                       (击中指示, 轮盘的指示灯全部置为OFF)
      Y70Z2: =FALSE;
   END_FOR;
END_IF;

OUT_M(X2, 初始化控制)                            (仅在 "X2" 为ON期间, 将 "初始化控制" 置为ON)

(轮盘实动部分, 翻转)
TIMER_M(定时器1, TS0, K5);                       (定时器设定=轮盘速度)

IF TS0 THEN
   IF 轮盘位置>=5 THEN
      轮盘位置: =K0;
   END_IF;
      轮盘位置: =轮盘位置+1; 定时器2: =TRUE; 定时器1: =FALSE;        (指示灯的位置移动一位)


TIMER_M(定时器2, TS1, K5);                       (定时器设定=轮盘速度)

IF TS1 THEN
   IF 轮盘位置>=5 THEN
      轮盘位置: =K0;
   END_IF;
      轮盘位置: =轮盘位置+1; 定时器1: =TRUE; 定时器2: =FALSE;        (指示灯的位置移动一位)
END_IF;

(每经过一定时间, 点亮对应的指示灯)
CASE 轮盘位置 OF

   1: Y74: =FALSE;                               ( "轮盘位置" 为1时, 将 "Y74" 置为OFF)
      Y70: =TURE;                                (将 "Y70" 置为ON。(未击中位置))

   2: Y70: =FALSE;                               ( "轮盘位置" 为2时, 将 "Y70" 置为OFF)
      Y71: = TURE;                               (将 "Y71" 置为ON。(未击中位置))

   3: Y71: =FALSE;                               ( "轮盘位置" 为3时, 将 "Y71" 置为OFF)
      Y72: = TURE;                               (将 "Y72" 置为ON。(击中位置))

   4: Y72: =FALSE;                               ( "轮盘位置" 为4时, 将 "Y72" 置为OFF)
      Y73: = TURE;                               (将 "Y73" 置为ON。(未击中位置))

   5: Y73: =FALSE;                               ( "轮盘位置" 为5时, 将 "Y73" 置为OFF)
      Y74: = TURE;                               (将 "Y74" 置为ON。(未击中位置))

END_CASE;

IF Y72 AND 定时器1 = FALSE AND 定时器2 = FALSE THEN   (仅在轮盘停止时, 进行击中判断)
   Y75: =TRUE;                                   (将 "Y75" 置为ON, 显示击中)
END_IF;
```

参考）梯形图程序

与上页的ST程序动作相同的梯形图程序。

确认ST程序的内容、将梯形程序替换成ST程序的情况下可供参考。



[局部变量定义]

	Au	标签	常量	软元件类型
1	○	开始控制		位
2	○	初始化控制		位
3	○	定时器1		位
4	○	定时器2		位
5	○	轮盘位置		字

4.1.2 关于程序中出现的控制结构（CASE条件语句 FOR…DO结构）

・CASE 条件语句

[说明]

与整数表达式的值相一致，则首先执行该值所对应的语句。没有值一致的情况下，则执行ELSE之后的语句。

CASE条件语句的结果返回整数值。

该条件语句在需要根据单一整数值或者复杂表达式结果的整数值而执行选择语句的情况下使用。

```
CASE <整数表达式> OF
        <整数选择值1> : <语句1・・・>
        <整数选择值2> : <语句2・・・>
                    ⋮
        <整数选择值n> : <语句n・・・>
ELSE
        <语句n+1>
END_CASE;
```

・CASE条件语句中的<整数表达式>可以指定的数据类型

整型（INT）、双精度整型（DINT）。可以指定字软元件以及字/双字型标签。

・FOR・・・DO结构

[说明]

FOR・・・DO结构根据重复变量的值，反复执行某些语句。

```
FOR<重复变量初始化>
    TO<最终值表达式>
    BY<增量表达式>DO
      <语句・・・>
END_FOR;
```

重复变量表达式初始化：对作为重复变量使用的数据进行初始化。

最终值表达式、增量表达式：已初始化的重复变量按照增量表达式递增或者递减。到达最终值之前，重复进行处理。

・FOR结构语句的<最终值表达式、增量表达式>可以使用的数据类型

可以指定整数值或者运算式结果的整数值。

4.1.3 动作内容及确认

在实习机上进行本程序的动作确认。

实习机上的动作确认

① 将“X0”置为ON，轮盘启动，“Y70～Y74”顺次点亮。
② 将“X1”置为ON，轮盘停止，“Y70～Y74”的其中之一保持ON状态。
③ 轮盘停止时，仅在“Y72”为ON的情况下，表示击中的“Y75”为ON。
④ 将“X2”置为ON，则“Y70～Y75”全部变为OFF，轮盘可以再次进行使用。

4.2 练习问题2 包含FB的顺控程序

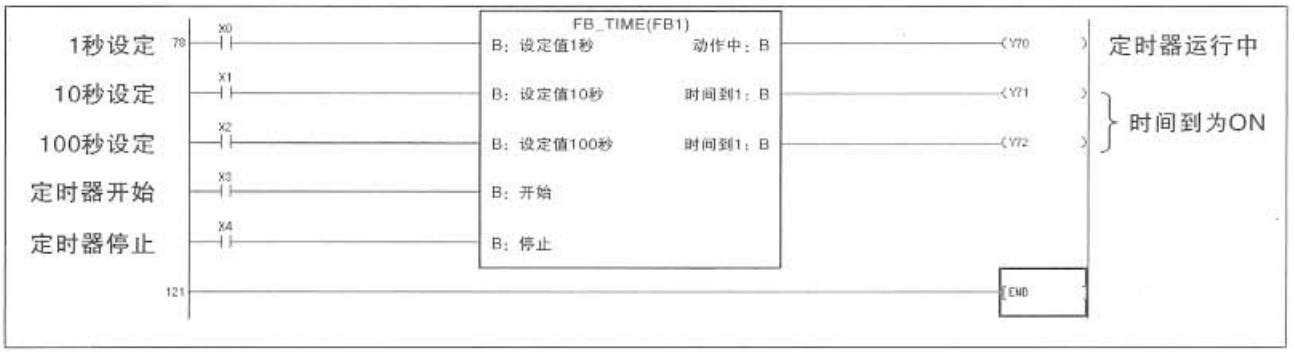
4.2.1 程序内容

路径名	A:￥SCHOOL
工程名	练习2
程序名	MAIN

将第3章编写的ST程序 [设定值可外部切换的定时器]进行功能模块化(FB)，在顺控程序中使用。
1个定时器的时限通过外部切换开关可以设置为1秒、10秒、100秒3档。
X0、X1、X2、X3作为按钮。
[顺控程序]
・ST程序

```
FB_TIER(
设定值1秒：=X0,
设定值10秒：=X1,
设定值100秒：=X02,
启动：=X3,
停止：=X4,
动作中：=Y70,
时间到1：=Y71,
时间到2：=Y72
);
```

参考）梯形图程序
与上述的ST程序动作相同的梯形图程序。
确认ST程序的内容，将梯形程序替换成ST程序的情况下可供参考。



[局部变量定义]
顺控程序中调用的FB，在局部变量设置画面下进行定义。

	Au	标签	常量	软元件类型
1	○	FB_TIMER	详细设置	FB(TIMER)

[FB程序：TIMER]

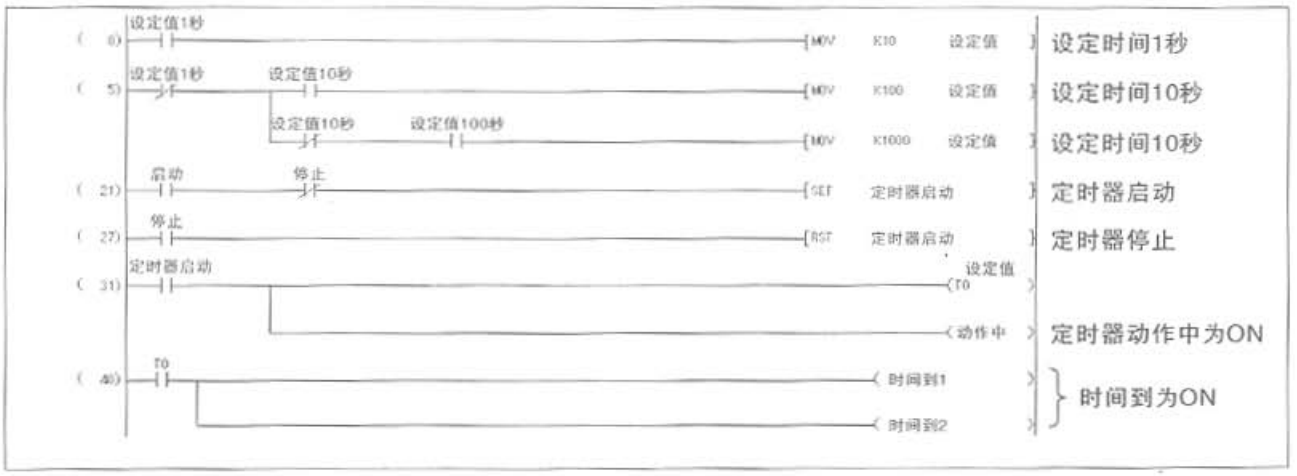
・ST程序

```
IF 设定值1秒 THEN                    (输入变量“设定值1秒”为ON时)
  设定值：=K10;                      (将K10代入内部变量“设定值”)

  ELSEIF 设定值10秒 THEN             (“设定值1秒”为OFF，输入变量“设定值10秒”为ON时)
  设定值：=K100;                     (将K100代入“设定值”)

  ELSEIF 设定值100秒 THEN            (“设定值1秒”，“设定值10秒”为OFF)
    设定值：=K1000;                  (输入变量“设定值100秒”为ON时)
END_IF;                              (将K1000代入“设定值”)

IF 启动 AND 停止=FALSE THEN          (输入变量“启动”为ON，“停止”为OFF时)
  定时器启动：=TRUE;                 (内部变量“定时器启动”变为ON，定时器启动)
END_IF;

IF 停止 THEN                         (输入变量“停止”为ON时)
  定时器启动：=FALSE;                (将“定时器启动”置为OFF，停止定时器)
END_IF;

TIMER_M(定时器启动，TS0，设定值)     (“定时器启动”为ON时，等待设定值的时间后，将TS0置为ON)
OUT_M(定时器启动，动作中)            (将“定时器启动中”存放的ON/OFF值，反应到“动作中”的值里)


IF TS0 THEN          (TS0为ON时)     (TS0为ON时)
  定时器时间到1：=TRUE;              (输出变量“时间到1”置为ON)
  定时器时间到2：=TRUE;              (输出变量“时间到2”置为ON)
  ELSE
  定时器时间到1：=FALSE;             (输出变量“时间到1”置为OFF)
  定时器时间到2=FALSE;               (输出变量“时间到2”置为OFF)
END_IF;
```

参考）梯形图程序

与上述的ST程序动作相同的梯形图程序。

确认ST程序的内容、将梯形程序替换成ST程序的情况下可供参考。



[FB变量定义]

在FB变量设置画面下对FB定义程序内使用的标签进行定义。

	输入输出	标签	常量	软元件类型
1	输入变量	设定值1秒		位
2	输入变量	设定值10秒		位
3	输入变量	设定值100秒		位
4	输入变量	启动		位
5	输入变量	停止		位
6	输出变量	动作中		位
7	输出变量	时间到1		位
8	输出变量	时间到2		位
9		设定值		字
10		定时器启动		位

4.2.2 FB的使用顺序

ST程序中使用FB的情况下，操作顺序如下所示。

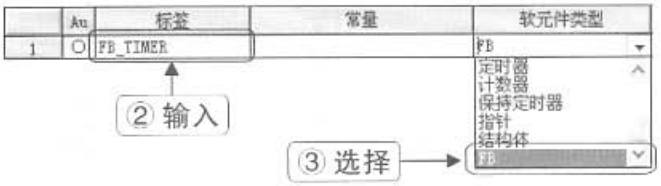
（1）FB定义的生成

对顺控程序中使用的FB变量进行定义。

关于FB变量的定义，请参考[Q编程教科书（FB篇）]。

（2）FB名的定义

将编写的FB用于ST程序的情况下，在局部变量设置画面下对FB名进行定义。



① 打开局部变量设置画面。

② 输入所使用的FB的标签名(FB_TIMER)。

③ 点击 ▼ 按钮，选择“FB”。

④ 显示“功能模块定义选择”对话框。

⑤ 点击选择使用的FB定义（TIMER）。

⑥ 点击 确定 按钮。

⑦ FB标签的定义完成。

（3）在顺控程序中调用FB

在顺控程序中通过FB名调用FB。

关于FB的调用，说明如下。

- FB中所使用的输入变量、输入输出变量需要全部列出。
- 对于输入变量、输入输出变量请务必指定数值。
  （对于输出变量，如果不需要其结果，可以省略）

（例）FB定义："TIMER"的情况

[FB定义内容]

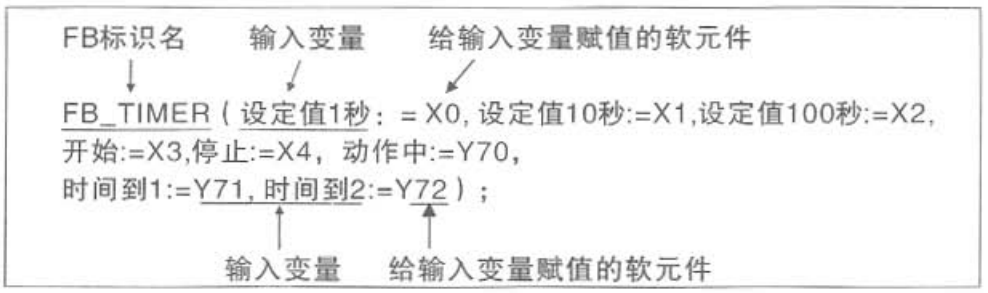
FB标签名：FB_TIMER

输入变量：设定值1秒、设定值10秒、设定值100秒、开始、停止。

输出变量：动作中、时间到1、时间到2。

[使用示例]

ST程序中，调用形式如下。



在FB名之后添加"."，并指定输出变量名，可以获得FB的输出。

[使用示例]

```
Y70：=FB_TIMER.动作中；
```

通过这种方式获取输出值的情况下，请在FB调用执行之后进行。在调用之前执行则出错。

要　点

重复调用同一FB的情况。
在局部变量设置画面下设置的FB，ST程序中只能使用1次，多次使用则会出错。
对同一FB多次使用的情况下，需要根据使用次数，重新定义FB标识。

（1）在局部变量设置画面，FB出错。

	Au	标签	常量	软元件类型
1	○	FB_TIMER	详细设置	FB(TIMER)
2	○	FB_TIMER2	详细设置	FB(TIMER)
3	○	FB_TIMER3	详细设置	FB(TIMER)

（2）ST程序中，调用形式如下
[使用示例]

```
FB_TIMER（设定值1秒:=X0, 设定值10秒:=X1,设定值100秒:=X2,
          开始:=X3,停止:=X4，动作中:=Y70,
          时间到1:=Y71, 时间到2:=Y72）;

FB_TIMER2（设定值1秒:=X5, 设定值10秒:=X6,设定值100秒:=X7,
          开始:=X8,停止:=X9，动作中:=Y73,
          时间到1:=Y74, 时间到2:=Y75）;

FB_TIMER3(设定值1秒:=X10, 设定值10秒:=X11,设定值100秒:=X12,
          开始:=X13,停止:=X14，动作中:=Y76,
          时间到1:=Y77, 时间到2:=Y78）;
```

4.2.3 动作内容及确认

在实习机上进行本程序需动作的确认。

实习机上的动作确认

① 将“X0”、“X1”、“X2”的其中之一置为ON，设置定时器时间。
② 将“X3”置为ON，则定时器启动，“Y70”变为ON。
③ 经过①中设置的时间后，定时器时间到，“Y71”和“Y72”变为ON。
④ 将“X4”置为ON，定时器复位，可以再次使用。

附录

附录1 ST程序规范

ST程序的规范以及可使用的软元件说明如下。

（1）程序大小

每个程序文件最大839680（半角）字节。

> **要　点**
>
> 统计文件内字符数的情况下，需要注意如下。
> - CR，LF作为2个字符处理。
> - 全角文字作为2个字符处理。
> - 半角空格作为1个字符处理。
> - TAB制表符作为1个字符处理。

（2）可使用的软元件

ST程序中可以使用的软元件名如下所示。

软元件的点数可以在参数设置中更改。

分类	类别	软元件	表示	备注
内部用户软元件	位	输入	X	
		输出	Y	
		内部继电器	M	
		锁存继电器	L	
		报警器	F	
		链接继电器	B	
		数据寄存器	SB	
	字	链接寄存器	D	
		链接特殊寄存器	W	
		链接特殊寄存	SW	
内部系统软元件	位	特殊继电器	SM	
	字	特殊寄存器	SD	
文件寄存器	字	文件寄存器	R	
			ZR	
MELSECNET/10(H) 链接直接软元件	位	输入	Jn\X	
		输出	Jn\Y	
		链接继电器	Jn\B	
		链接特殊继电器	Jn\SB	
		链接寄存器	Jn\W	
	字	链接特殊寄存器	Jn\SW	
智能功能模块	字	缓冲寄存器	Un\G	
变址寄存器	字	变址寄存器	Z	Z0，Z1 不可使用
常数	位/字/双字	10 进制常数	K	
		16 进制常数	H	
	实数	实常数	E	
	字符串	字符串常数	“ABC”等	
其他	位	SFC 模块	BL	
	位	SFC 移位软元件	BL\TR	
	位	SFC 步进继电器	BL\S	
	位	直接输入	DX	
	位	直接输出	DY	

(3) 仅可在ST程序中使用的软元件

ST程序中，定时器、计数器的接点、线圈、当前值均作为独立的软元件进行表示和使用。

定时器、计数器的接点、线圈、当前值的软元件表示和类别如下所示。

分类	类别	软元件	表示	备注
内部用户软元件	字	定时器接点	TS	
		定时器线圈	TC	
		累积定时器接点	STS	
		累积定时器线圈	STC	
		计数器接点	CS	
		计数器线圈	CC	
	位	定时器当前值	TN/T	
		积分定时器当前值	STN/ST	
		计数器当前值	CN/C	

使用示例

(1) [ST程序]

```
M0:=TS0;
```

[等价列表程序]

```
LD  T0
OUT M0
```

(2) [ST程序]

```
COUNTER_M (X0, CC20, 10);
```

[等价列表程序]

```
LD X0
OUT C20 K10
```

附录2 标签、FB名称中不可使用的字符串

进行ST编程时，不可以作为标识、FB名使用的字符串列示如下。
软元件名、顺控命令、SFC命令、应用命令中使用的字符串不可在标识、FB名中使用。
使用下表列示的字符串的情况下，进行登录/编译时，提示出错。

	标签、FB 名中不可使用的字符串
A	A,ACJ,ADD,ANB,AND,ANDF,ANDN,ANDP,ANI,ANY,ANY_BIT,ANY_DATE,ANY_DERIVED, ANY_ELEMENTARY,ANY_INT,ANY_MAGNITUDE,ANY_NUM,ANY_REAL,ANY_SIMPLE,ANY_STRING,ARRAY
B	B,BCD(P),BEND,BIN(P),BKBCD(P),BKBIN(P),BL,BLOCK,BMOV(P), BOOL,BOOL_TO_BYTE(DINT,DWORD,INT,REAL,SINT,UDINT,UINT,USINT,WORD),BYTE(DINT,DWORD,INT, REAL,SINT,TIME,UDINT,UINT,USINT,WORD)_TO_STRING, BYTE_TO_BOOL(DINT,DWORD,INT,REAL,SINT,UDINT,UINT,USINT,WORD), B_BCD_TO_DINT(INT,SINT),BXCH(P),BY,BYTE
C	C,CASE,CAL,CALC,CALCN,CJ,CML(P)
D	D,DBCD(P),DBIN(P),DBL(P),DCML(P),DDEC(P),DEC(P),DELTA(P),DFLT(P),DGBIN(P),DGRY(P),DI,D INC(P),DINT,DINT(P),DINT_TO_BCD(BOOL,BYTE,DWORD,INT,REAL,SINT,TIME,UDINT,UINT,USINT,WO RD),DIV,DMOD,DMOV(P),DNEG(P),DWORD,DWORD_TO_BOOL(BYTE,DINT,INT,REAL,SINT, UDINT,UINT,USINT,WORD),DX,DXCH(P),DY,D_BCD_TO_DINT(INT,SINT)
E	E,ELSE,ELSIF,END_CASE,END_FOR,END_IF,END_REPEAT,END_WHILE,EGF,EGP,EI,EMOV(P),END,ENEG(P),EQ,EQ(GE,GT,LE,LIMIT,LT,MAX,MIN,NE,SEL)_STRING,EXIT
F	F,FD,FEND,FF,FLT(P),FMOV(P),FOR,FX,FY
G	G,GBIN(P),GE,GOEND,GRY(P),GT
H	H
I	I,IMASK,INC(P),INT,INT(P), INT_TO_BOOL(BYTE,DINT,DWORD,REAL,SINT,UDINT,UINT,USINT,WORD),INV,IRET
J	J,JMP,JMPC,JMPCN
K	K
L	L,LD,LDF,LDI,LDN,LDP,LE,LED,LEDA,LEDB,LEDC,LEDR,LINT,LREAL,LT,LWORD
M	M,MC,MCR,MEF,MEP,MOD,MOV(P),MPP,MPS,MRD,MTR,MUL
N	N,NE,NEG(P),NOP,NOPLF,NOT
O	OF,OR,ORB,ORF,ORI,ORN,ORP,OUT(H)
P	P,PAGE,PCHK,PLF,PLS,PLSY,PWM
Q	Q
R	R,RAMP,RCJ,READ,REAL,REAL_TO_BOOL(BYTE,DINT,DWORD,INT,SINT,UDINT,UINT,USINT,WORD),RECV ,REPEAT,RETURN,REQ,RET,RETC,RETCN,RFRP,RFS,ROTC,RST,RTOP

	标签、FB 名中不可使用的字符串
S	S,SB,SCJ,SD,SEND,SEND,SET,SFCP,SFCPEND,SFT(P),SG,SINT, SINT_TO_BOOL(BYTE,DINT,DWORD,INT,REAL,UDINT,UINT,USINT,WORD),SM,SPD,SREAD,ST, STEPC,STEPD,STEPG,STEPI,STEPID,STEPIR,STEPISC,STEPISE,STEPIST,STEPN,STEPR,STEPSC, STEPSE, STEPST,STMR,STN,STOP,STRING,STRING_TO_BYTE(DINT,DWORD,INT,REAL,SINT,TIME, UDINT,UINT,USINT,WORD),SUB,SW,SWAP(P),SWRITE,SZ
T	T,THEN,TIME,TO,TR,TRANA,TRANC,TRANCA,TRANCO,TRANCOC,TRANJ,TRANL,TRANO,TRANOA, TRANOC,TRANOCA,TRANOCJ,TRANOJ,TRUNC_DINT(INT,SINT),TTMR
U	U,UDCNT1(P),UDCNT2(P),UDINT,UDINT_TO_BOOL(BYTE,DINT,DWORD,INT,REAL,SINT,UINT,UNTIL, USINT,WORD),UINT,UINT_TO_BOOL(BYTE,DINT,DWORD,INT,REAL,SINT,UDINT,USINT,WORD), ULINT,USINT,USINT_TO_BOOL(BYTE,DINT,DWORD,INT,REAL,SINT,UDINT,UINT,WORD)
V	V,VAR,VAR_CONSTANT,VAR_EXT,VAR_EXTERNAL,VAR_EXTERNAL_CONSTANT, VAR_EXTERNAL_FB,VAR_EXTERNAL_PG,VAR_GLOBAL,VAR_GLOBAL_CONSTANT, VAR_GLOBAL_FB,VAR_GLOBAL_PG,VAR_IN_OUT,VAR_INPUT,VAR_OUTPUT, VAR_TEMP,VD,VOID
W	W,WHILE,WORD,WORD(P),WORD_TO_BOOL(BYTE,DINT,DWORD,INT,REAL,SINT,UDINT, UINT,USINT), WRITE,WSTRING,W_BCD_TO_DINT(INT,SINT)
X	XOR,XORN
Y	Y
Z	Z,ZCOM,ZNRD,ZNRF,ZNTO,ZNWR,ZR
函数群	MELSEC 函数群、IEC 函数群

使用和上表完全一致的字符串则出错。（不区分大小写）
软元件名在末尾添加0～F字符的情况下，也提示出错。
例 XFFF，M100

其他在FB定义时的注意事项
1. 不可以使用空格。
2. 第1个字符不可以使用数字。
3. 下列字符不可以使用。
   (，)，*，/，+，−，<，>，=，&，
   !，“，#，$，%，’，~，^，|，@，‘，[，]，{，}，;，:，,，.，?，¥，_
下划线位于字符串结尾，或者两个以上连用的情况下出错。

备　忘　录

三菱电机自动化(上海)有限公司
地址:上海市南京西路288号创兴金融中心17F
邮编:200003
电话:(021)2322 3030 传真:(021)2322 3000
网址:http://www.meas.cn