

结构化文本 (ST)

编程参考手册

MITSUBISHI



三菱可编程逻辑控制器

MELSEC-Q

● 安全注意事项 ●

(使用之前务必阅读)

在使用本产品之前，应仔细阅读本手册及本手册中所介绍的关联手册，同时在充分注意安全的前提下正确地操作。

本手册中的注意事项仅记载了与本产品有关的内容。关于可编程控制器系统方面的安全注意事项，请参阅 CPU 模块的用户手册。

在“安全注意事项”中，安全注意事项被分为“危险”和“注意”这二个等级。



危险

表示错误操作可能造成灾难性后果，引起死亡或重伤事故。



注意

表示错误操作可能造成危险的后果，引起人员中等伤害或轻伤还可能使设备损坏。

注意根据情况不同，即使△注意这一级别的事项也有可能引发严重后果。

对两级注意事项都须遵照执行，因为它们对于操作人员安全是至关重要的。

妥善保管本手册，放置于操作人员易于取阅的地方，并应将本手册交给最终用户。

[设计方面的注意事项]

◇ 危险

- 通过个人计算机对运行中的可编程控制器进行数据变更、程序变更、状态控制时，应在可编程控制器系统的外部配置互锁电路，以保证整个系统总是能向安全的方向动作。
此外，在通过个人计算机对可编程控制器 CPU 进行在线操作时，应预先确定由于电缆的接触不良等导致数据通讯异常时的系统方面的处理方法等。

[启动、维护时的注意事项]

△ 注意

- 将个人计算机连接在运行中的可编程控制器 CPU 上进行在线操作(在可编程控制器 CPU 处于运行状态下的程序变更、强制输入输出操作、RUN-STOP 等运行状态的变更、远程操作)时，应在熟读手册，充分确认安全后进行操作。
此外，在可编程控制器 CPU 处于运行状态下进行程序变更(运行中写入)时，根据操作条件有时会发生程序被损坏等问题。应在充分了解 GX Developer 操作手册中记载的注意事项的基础上进行操作。

修订记录

* 手册编号在封底的左下角。

印刷日期	* 手册编号	修改内容
2007年03月	SH(NA)-080665CHN-A	第一版

日文手册原稿:SH-080365-E

本手册未被授予工业知识产权或其它任何种类的权利，亦未被授予任何专利许可证。三菱电机对使用本手册中的内容造成的工业知识产权问题不承担责任。

© 2007 三菱电机

前言

在此感谢贵方购买了三菱电机的产品。
在使用之前应熟读本书，在充分了解产品的功能/性能的基础上正确地使用本产品。
此外，请将本手册交给最终用户。

目录

安全注意事项	A - 1
修订记录	A - 2
目录	A - 3
关于手册	A - 5
手册的阅读方法	A - 6
本手册中使用的总称、简称	A - 7
1 概要	1 - 1 到 1 - 2
2 ST 程序创建步骤	2 - 1 到 2 - 2
3 ST 编程	3 - 1 到 3 - 16
新建用于 ST 的工程	3 - 1
定义标识	3 - 3
输入程序	3 - 8
对 ST 程序进行转换(编译).....	3 - 14
4 通过可编程控制器 CPU 进行读取、写入	4 - 1 到 4 - 2
5 调试程序	5 - 1 到 5 - 6
5.1 监视顺控程序	5 - 1
5.2 软元件测试	5 - 2
5.3 运行中写入	5 - 4
6 将程序保存到个人计算机中	6 - 1 到 6 - 2
7 ST 程序编辑时的便利的功能的介绍	7 - 1 到 7 - 2

8.1 创建 FB	8 - 1
新建工程	8 - 1
新建 FB	8 - 2
定义 FB 变量	8 - 5
通过 ST 语言创建 FB	8 - 9
8.2 将 FB 粘贴到主程序中	8 - 11
定义局部变量	8 - 11
创建主程序	8 - 13
8.3 在线	8 - 16
写入到可编程控制器中	8 - 16
监视顺控程序	8 - 17
确认程序的动作	8 - 18

关于手册

与本产品有关的手册如下表所示。
请根据需要参考本表。

关联手册

手册名称	手册编号
GX Developer 版本 8 操作手册(入门篇) 介绍 GX Developer 的系统配置、安装方法、启动方法等有关内容。 (另售)	SH-080355
GX Developer 版本 8 操作手册 介绍 GX Developer 的程序创建方法、打印输出方法、监视方法、调试方法等有关内容。 (另售)	SH-080311C
GX Developer 版本 8 操作手册(结构化文本篇) 介绍结构化文本程序创建的操作方法有关内容。 (另售)	SH-080666CHN
GX Developer 版本 8 操作手册(功能块篇) 介绍 GX Developer 的程序创建方法、打印输出方法等有关内容。 (另售)	SH-080639CHN
QCPU(Q 模式) 编程手册(结构化文本篇) 介绍结构化文本语言的编程方法有关内容。 (另售)	SH-080363
QCPU(Q 模式)/QnACPU 编程手册(公共指令篇) 介绍顺控指令、基本指令以及应用指令的使用方法有关内容。 (另售)	SH-080450CHN

备注

各操作手册及结构化文本(ST)编程参考手册与软件包一起被刻录在 CD-ROM 中。
希望单独购买编程手册时，由于配备有另售的印刷品，因此请通过上表的手册编号(型号代码)订购。

关于该手册 ...

本参考手册是面对初次使用 GX Developer 版本 8 软件包 (以下简称为 GX Developer) 创建结构化文本 (以下简称为 ST) 程序的读者为对象的说明书。

“第 1 章 概要”中记述了 ST 语言的概要及 MELSEC-Q 系列中 ST 语言的特点。在“第 2 章~6 章”中,通过样本程序对从使用 ST 语言创建程序的方法起至调试、保存为止的一系列基本操作方法进行了介绍。

“第 7 章”中介绍了 GX Developer 中配备的便利功能。

“第 8 章”作为应用篇,通过由梯形图创建的主程序,介绍了将由 ST 语言创建的功能块 (FB) 用于梯形图程序的程序创建方法。

在“第 4、5、8 章”中,介绍用于可编程控制器 CPU 的方法。

编程手册 ...

“QCPU(Q 模式)编程手册(结构化文本篇)”介绍使用 GX Developer 进行结构化文本 (ST) 编程。适用于具有可编程控制器、梯形图程序相关知识及编程经验的用户,以及具有 C 语言等高级语言相关知识及编程经验的用户。

操作手册 ...

“GX Developer 版本 8 操作手册(结构化文本篇)”是详细说明了使用 GX Developer 创建结构化文本程序的操作方法等有关内容的说明书。在需要了解操作的详细信息时请参阅该手册。

希望了解解除结构化文本编程以外的信息时 ...





请参阅“GX Developer 版本 8 操作手册”或“GX Developer 版本 8 操作手册(入门篇)”。

本手册中使用的总称、简称

在本手册中，将 GX Developer 软件包、可编程控制器 CPU 通过以下的总称、简称表示。需要标明相关型号时，将记述模块的型号。

总称/简称	内容 / 对象模块
ST	结构化文本的简称。
FB	功能块的简称。
GX Developer	产品型号为 SW8D5C-GPPW、SW8D5C-GPPW-A、SW8D5C-GPPW-V、SW8D5C-GPPW-VA 的产品统称名。
基本模式 QCPU	功能版本 B 以后的 Q00JCPU、Q00CPU、Q01CPU 的总称。
高性能模式 QCPU	Q02 (H) CPU、Q06CPU、Q12HCPU、Q25HCPU 的总称。
过程 CPU	Q12PHCPU、Q25PHCPU 的总称。
冗余 CPU	Q12PRHCPU、Q25PRHCPU 的总称。
QCPU(Q 模式)	Q00J、Q00、Q01、Q02 (H)、Q06H、Q12H、Q12PH、Q12PRH、Q25H、Q25PH、Q25PRHCPU 的总称。

本手册中使用的符号及内容如下所示：

符号	内容	示例
Point	记述了作为该项目相关知识应预先了解的内容，预先了解可带来方便的内容。	 要点
[]	菜单栏的菜单名	[Project]
()	工具栏中的图标	
<< >>	对话框的选项卡名	<<Select file>>
	对话框的指令按钮	 按钮

ST语言是...?

ST语言是指，关于开放/控制中的逻辑记述方式所制定的国际标准 IEC61131-3 中定义的语言。

在 ST 语言中支持运算符、控制语句、函数，可以进行如下记述：

- 通过条件语句进行的选择分支、通过重复语句进行的重复等的控制语句
- 使用运算符(*、/、+、-、<、>、=等)的公式
- 用户定义的功能块(FB)的调用
- 函数的调用(MELSEC函数/IEC函数)
- 包含汉字等的全角字符的注释记述

特点是...?

以下介绍 MELSEC-Q 系列中 ST 程序的主要特点。

☞ 通过部件化可以提高设计效率。

在 ST 语言中通过预先执行部件化操作将常用的处理定义为功能块 (FB)，可以调用各程序的必要部分。

由此，在提高了程序开发效率的同时，也减少了程序错误，提高了程序的质量。

☞ 可以在系统运行过程中更改程序 (运行中写入)。

可以在不停运可编程控制器 CPU 的状况下，对正在执行的程序进行部分变更。

☞ 可以与其它语言程序结合使用。

由于也支持除 ST 语言以外的其它语言，因此可以使用适用于处理对象的语言以提高编程效率。

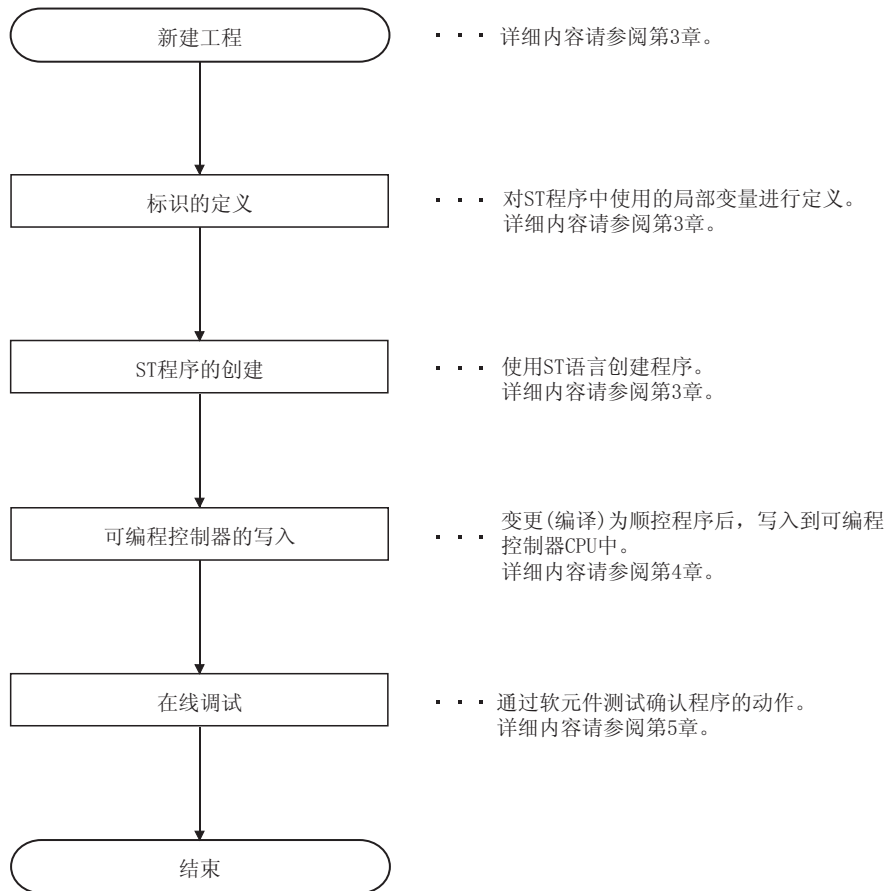
例如，通过创建梯形图程序进行顺序控制，通过编制 ST 语言进行运算处理。

由于可支持多种语言，可对较大范围内的用途执行最佳控制。

☞ 配备了丰富的函数群。

在 ST 程序中，配备了对应于 MELSEC-Q 系列用的各种通用指令的 MELSEC 函数、IEC61131-3 中定义的 IEC 函数。

从 ST 程序的创建起至在线调试为止的基本步骤的流程图如下所示：
以下示例是仅通过 ST 程序创建的程序。



备注

关于各操作的详细内容，请参阅关联手册中记载的“GX Developer 操作手册”。

在 3 章中介绍了从 ST 程序的输入起至转换(编译)为止的一系列基本操作。
该章中介绍的项目如下所示:

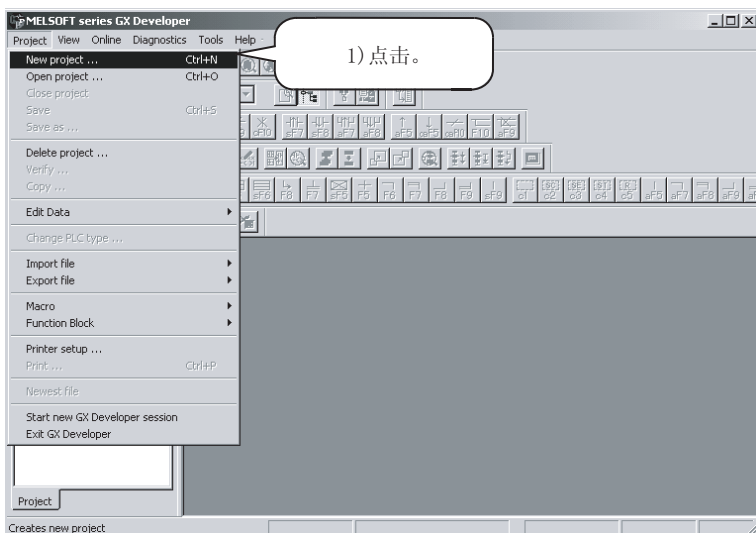
- ☞ 新建用于 ST 的工程。
- ☞ 定义使用 ST 程序的标识。
- ☞ 创建 ST 程序。
- ☞ 将所创建的 ST 程序转换(编译)为可执行的顺控程序。
- ☞ 发生转换(编译)错误时,对程序进行修正。

新建用于ST的工程

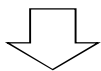
3

■ 新建用于 ST 的工程

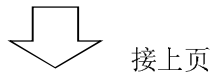
以下介绍新建工程的操作方法。



1) 点击菜单[Project(工程)] →
[New project(新建工程)]。

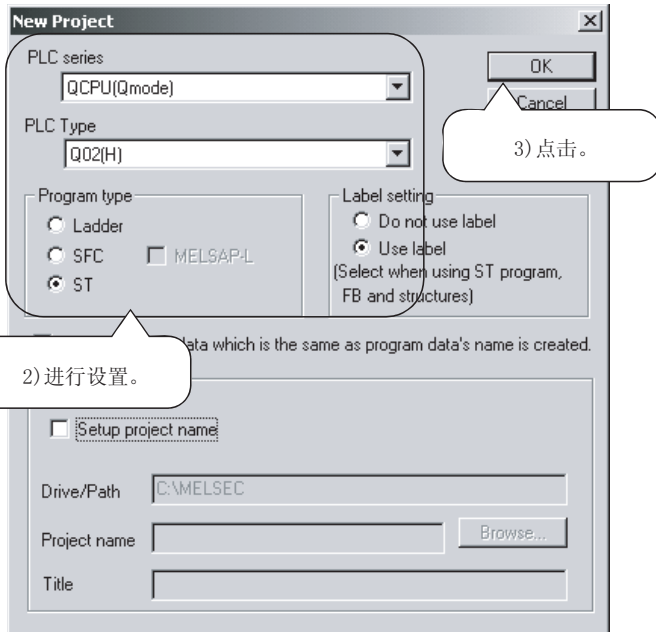


转下页



接上页

在新建工程对话框中进行设置。

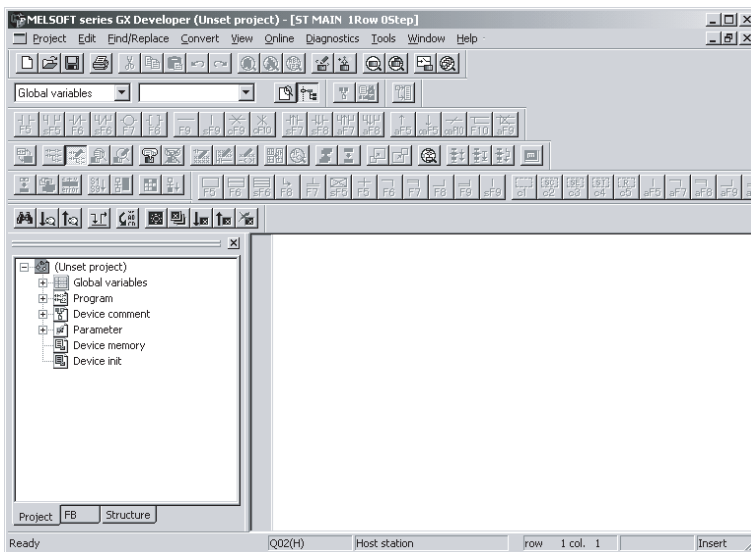


2) 输入以下内容：

- PLC 系列 : QCPU(Q 模式)
- PLC 类型 : Q02 (H)
- 标识设置 : 使用标识
- 程序类型 : ST

3) 点击 **OK** 按钮。

3



4) 新建用于 ST 的工程。

* 打开 ST 编辑画面，进入可输入 ST 程序的状态。

备注

在此，将 PLC 类型设置为“Q02 (H)”。

可兼容 ST 程序的可编程控制器 CPU 有以下几种类型：

基本模式 QCPU	高性能模式 QCPU	过程 CPU	冗余 CPU
Q00CPU	Q02CPU	Q12PHCPU	Q12PRHCPU
Q00JCPU	Q02HCPU	Q25PHCPU	Q25PRHCPU
Q01CPU	Q06HCPU		
	Q12HCPU		
	Q25HCPU		

定义标识

如何定义标识?

使用标识时，需要明确作为标识使用的变量。这称为“定义标识”。如果对使用了未定义的标识的程序进行转换(编译)将会发生错误，将无法创建顺控程序。

标识分为全局变量及局部变量这2种类型。

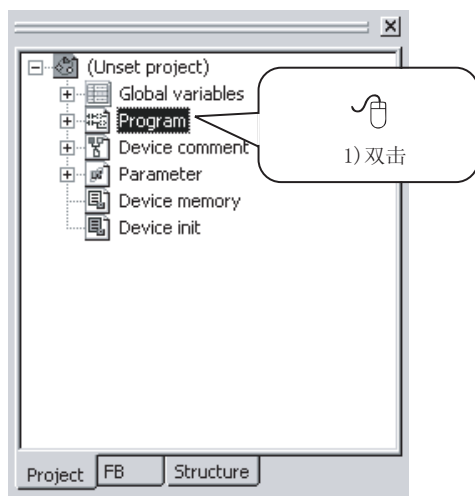
全局变量可被用于全部工程。局部变量只能被用于进行了标识定义的程序。

在此，试对随后输入的程序示例中使用的局部变量进行实际定义。

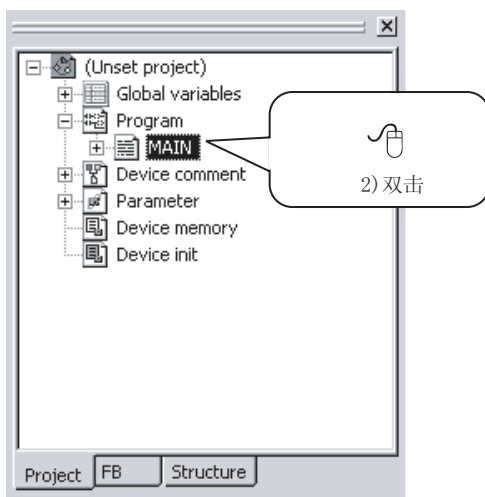
■ 显示局部变量(局部标识)设置画面

以下介绍对局部变量进行定义的操作方法。

关于全局变量，请参阅“GX Developer 操作手册”。

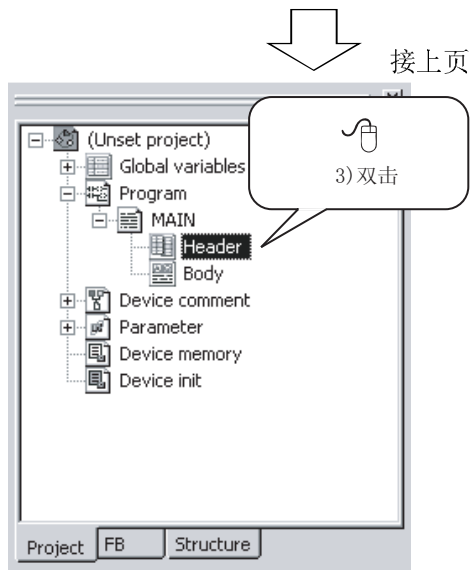


1) 双击<<Program(工程)>>选项卡的“Program(程序)”。

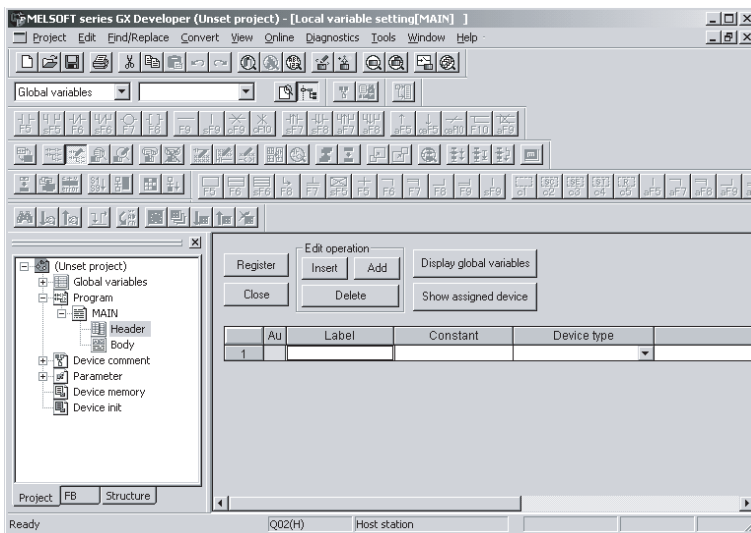


2) 双击“MAIN”。

转下页



3) 双击“Header(局部标识)”。



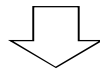
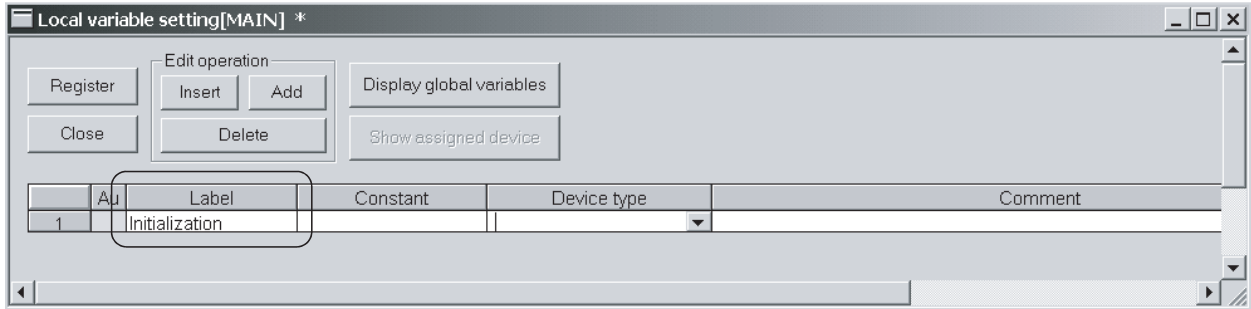
4) 显示局部变量(局部标识)设置画面。

■ 设置局部变量(局部标识)

1) 输入标识名。

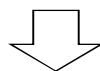
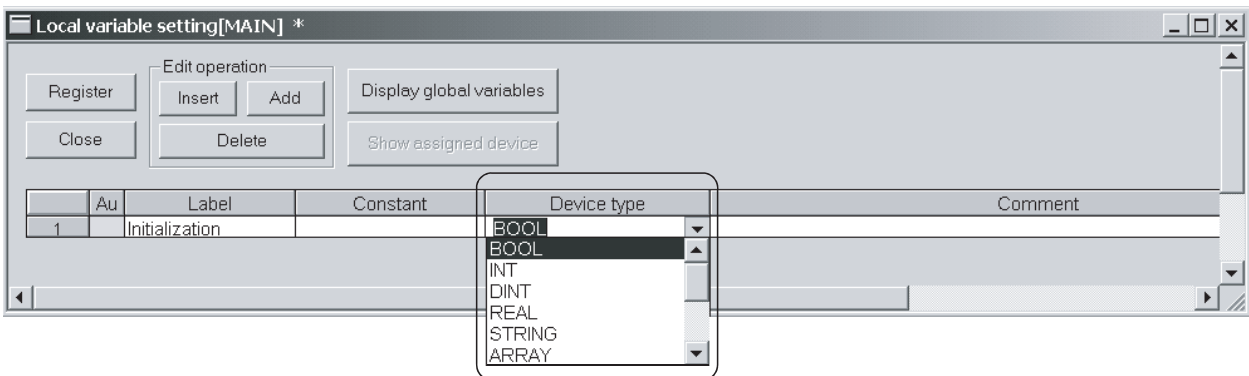
在半角 16 个字符(全角 8 个字符以内)的范围内输入标识名。标识名中不能使用保留字及实际软元件等字符串。应输入其它的标识。

* 关于保留字, 请参阅“GX Developer 操作手册”。



2) 输入软元件类型。

直接输入或者从列表框内选择。

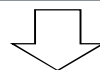
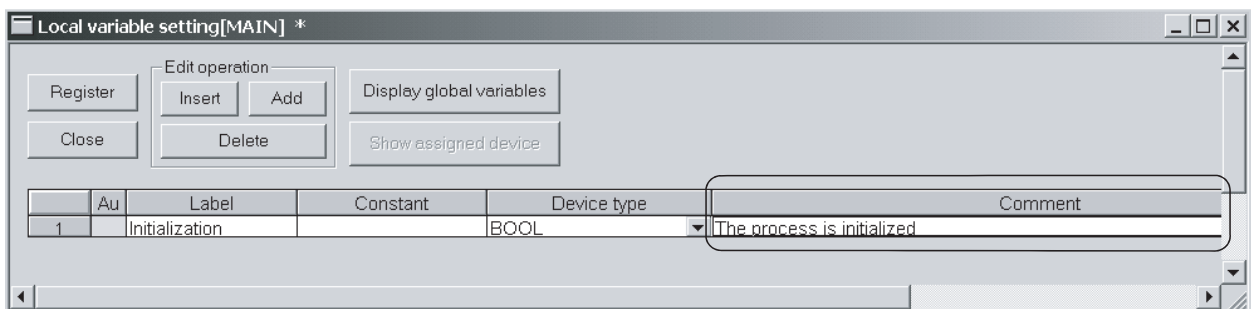


3) 在标识中输入注释。

应在半角 64 个字符(全角 32 个字符以内)的范围内输入。

注释可以以标识信息的工具条形式显示。

* 关于标识信息, 请参阅“7 章 ST 程序编辑时的便利功能的介绍”或者“GX Developer 操作手册(结构化文本篇)”。



转下页



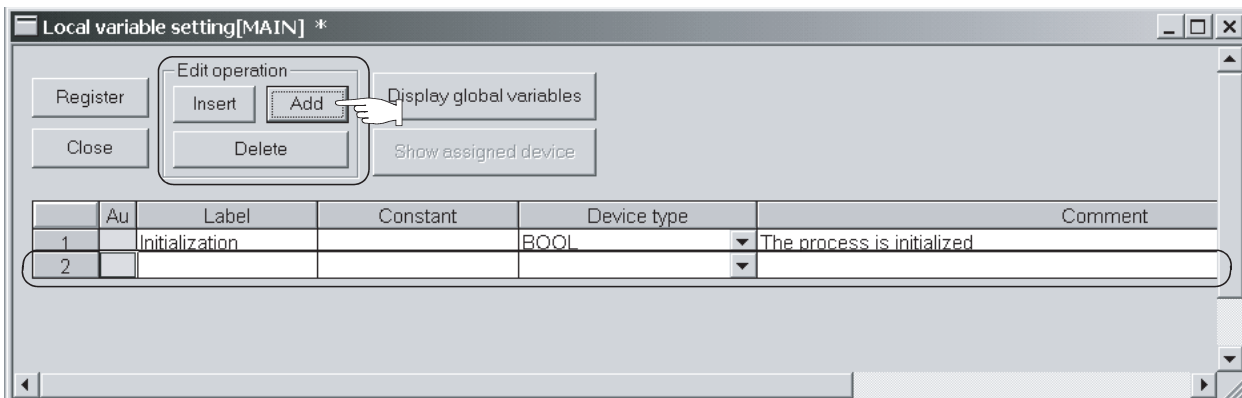
接上页

4) 连续输入标识时，点击编辑操作的 **Insert(插入行)** 或者 **Add(追加行)** 按钮进行追加。
编辑操作如下所示：

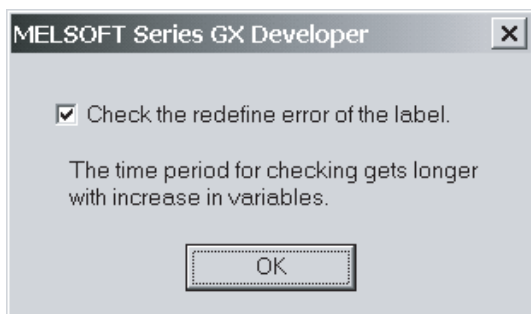
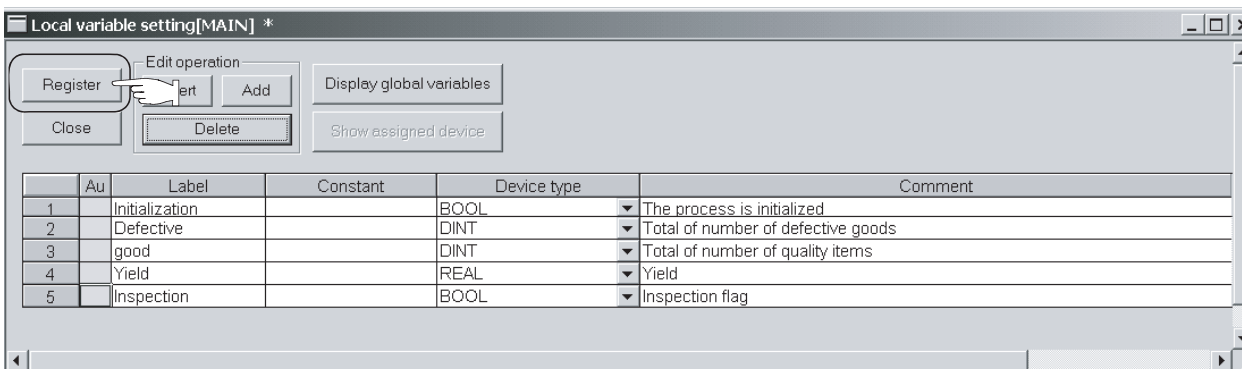
Insert(插入行) 按钮..... 在当前的单元格位置插入空白行。

Add(追加行) 按钮..... 在当前的单元格位置的 1 行下插入空白行。

Delete(删除行) 按钮..... 当前的单元格位置的 1 行将被删除。



5) 输入完毕后，点击 **Register(登录)** 按钮。



点击 **OK** 按钮。



转下页



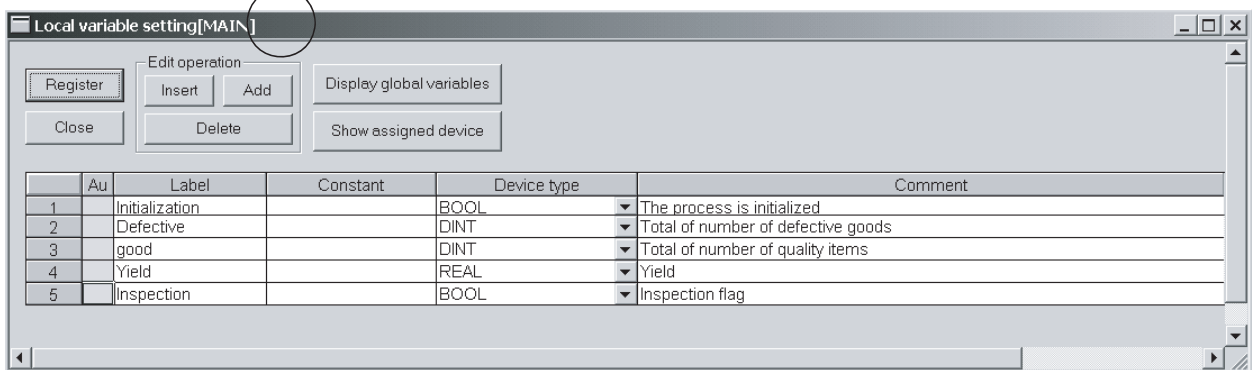
接上页



登录完毕。

点击 **OK** 按钮。

登录后，标题栏显示的“*”将消失。



备注

关于局部变量的详细内容，请参阅关联手册中记载的“GX Developer 操作手册”。

输入程序

输入时应注意?

可以使用 ST 编辑画面以文本格式自由地输入程序，在输入时应注意以下几点：

- 输入空白时应使用半角空格、`Tab` 键、`Enter` 键。
全角空格将不被作为空白处理。（转换（编译）时将被检测为错误。）但是，注释文内可以使用全角空格。
- 如果输入定义的标识、控制语句、注释，字符颜色将发生变化。
当字符颜色未发生变化时，可能是输入出错，或者标识未定义。

以下输入实际的列表-1 的程序。

列表-1.

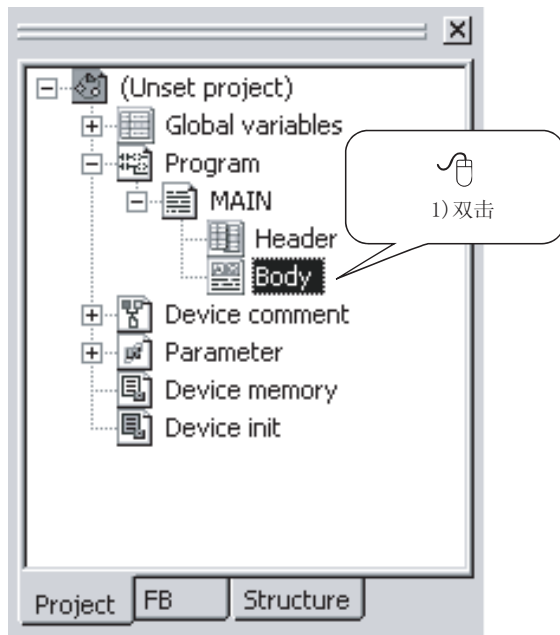
```
IF      Initialization THEN
    good := 0; Defective := 0; Yield := 0.0;

ELSE
    IF  Inspection THEN
        good := good + 1;

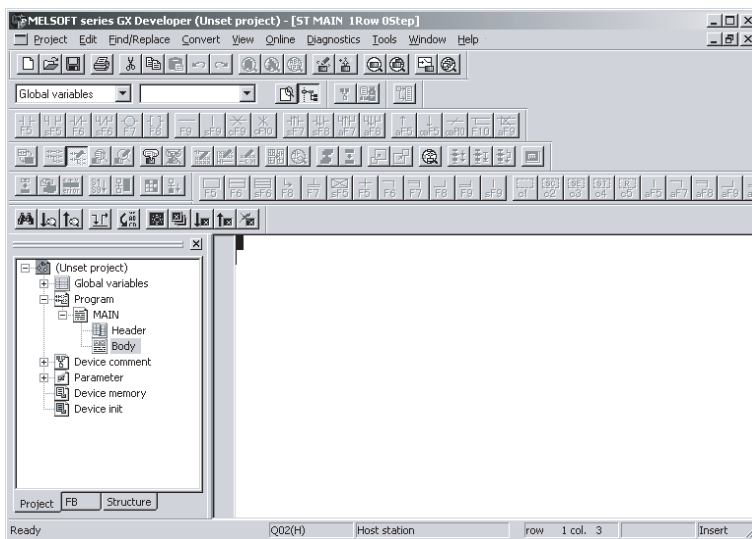
    ELSE
        Defective := Defective + 1;
    END_IF;

    Yield := DINT_TO_REAL(good)/DINT_TO_REAL(good + Defective);
END_IF;
```

■ 显示 ST 编辑画面



1) 用鼠标双击<<Project(工程)>>选项卡的“Body(程序本体)”。



2) 显示 ST 编辑画面。

■ 输入字符



输入“IF”。

* 即使以小写字母输入控制语句时也将被自动地转换为大写字母。

■ 输入标识

标识的输入有直接输入标识名的方法及使用标识选择功能的方法。此外，若要使用标识选择功能，需要预先对标识进行定义。

以下介绍使用标识选择功能进行输入的方法。

1) 点击

2) 选择“Initialization”。

3) 点击 按钮。

4) 标识名“Initialization”被插入。

1) 点击菜单的 [Edit (编辑)] → [Select label (标识选择)]。

2) 选择输入的标识。

3) 点击 按钮。

4) 插入标识。

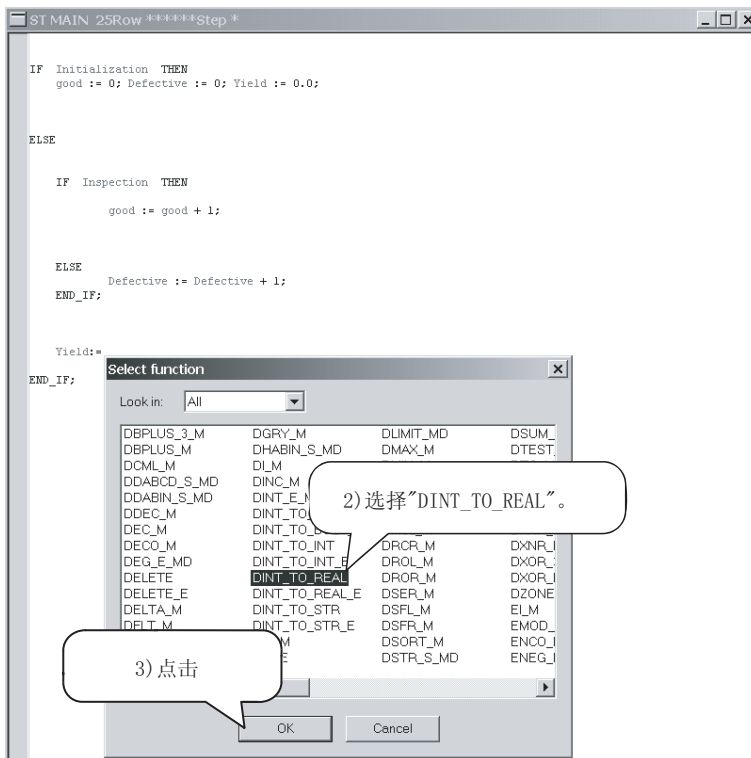
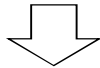
■ 输入函数

以大写字母输入函数。输入方法有直接输入方法及使用函数选择功能的输入方法。

以下介绍使用函数选择功能进行输入的方法。

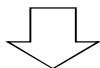


1) 点击菜单的[Edit(编辑)]→[Select function(函数选择)]。



2) 选择输入的函数名。

3) 点击 按钮。



转下页



```

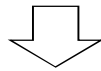
ST MAIN 25Row *****Step *
IF Initialization THEN
  good := 0; Defective := 0; Yield := 0.0;
ELSE
  IF Inspection THEN
    good := good + 1;
  ELSE
    Defective := Defective + 1;
  END_IF;
  Yield:=DINT_TO_REAL(
END_IF;
    
```

显示函数自变量的类型

4) 已被插入

4) 函数名被插入。

* 以工具条的形式显示函数自变量的类型。



```

ST MAIN 25Row *****Step *
IF Initialization THEN
  good := 0; Defective := 0; Yield := 0.0;
ELSE
  IF Inspection THEN
    good := good + 1;
  ELSE
    Defective := Defective + 1;
  END_IF;
  Yield:=DINT_TO_REAL(good)/DINT_TO_REAL(good + Defective);
END_IF;
    
```

5) 完毕

5) 参考以工具条形式显示的函数自变量的类型进行自变量的输入后，输入完毕。

■ 输入注释

注释不会对程序的动作产生任何影响。如果记述了程序的处理内容，进行了何种处理将会一目了然。

首先在程序的第 1 行中输入注释。

```

ST MAIN 26Row *****Step *
(* Initialization is done. *)
IF Initialization THEN
  good := 0; Defective := 0; Yield := 0.0;
ELSE
  IF Inspection THEN
    good := good + 1;
  ELSE
    Defective := Defective + 1;
  END_IF;
    
```

* 注释是以表示注释的开始的“(*)”及表示结束的“(*)”围住进行输入。





```

ST MAIN 25Row *****Step *
(* Initialization is done. *)
IF Initialization THEN
  good := 0; Defective := 0; Yield := 0.0;
(* The normal operation is processed. *)
ELSE
  (* Is the inspection passing? *)
  IF Inspection THEN
    good := good + 1; (* The number of non-defective articles is added. *)
    (* The inspection is failing. *)
  ELSE
    Defective := Defective + 1; (* The number of defective goods is added. *)
  END_IF;
  (* The yield is calculated. *)
  Yield:=DINT_TO_REAL(good)/DINT_TO_REAL(good + Defective);
END_IF;
    
```

参考左边的示例输入注释。
(列表-2)

至此，程序的输入完毕。

要 点

- 标识信息的显示
标识信息可以以工具条的形式显示。
操作 : 将鼠标指针对准标识位置
显示内容 : 标识名 → 标识类型 → 标识注释 → 软元件*
*: 表示软元件已被转换 (编译)。

```

ST MAIN 25Row 143Step
(* Initialization is done. *)
IF Initialization THEN
  good := 0; Defective := 0; Yield := 0.0;
(* The normal operation is processed. *)
ELSE
  (* Is the inspection passing? *)
  IF Inspection THEN
    good := good + 1; (* The number of non-defective articles is
    good->LOCAL->Total of number of quality items->D12284
    (* The inspection is failing. *)
  ELSE
    Defective := Defective + 1; (* The number of defective goods is add
    
```

显示标识信息

- 显示颜色的更改
可以更改控制语句、注释、标识的字符串、ST 编辑画面的背景色等。
操作: 菜单[Tools(工具)] → [Change display color(显示颜色更改)]
- 自动缩进的设置
可以设置按下 **Enter** 键时的字缩进及按下 **Tab** 键时的标签宽度。
操作: 菜单[Tools] → [ST editor settings (ST 编辑画面设置)]
有关详细内容请参阅“GX Developer 操作手册(结构化文本篇)”。

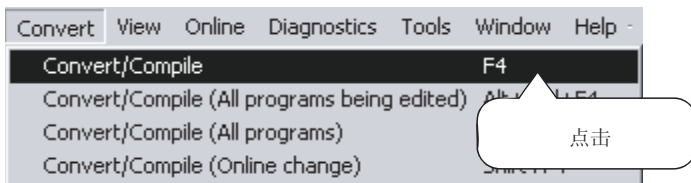
对ST程序进行转换(编译)

何为转换(编译)?

将通过 ST 编辑画面创建的程序变为可在可编程控制器 CPU 中执行的顺控程序，这个过程被称为转换(编译)。

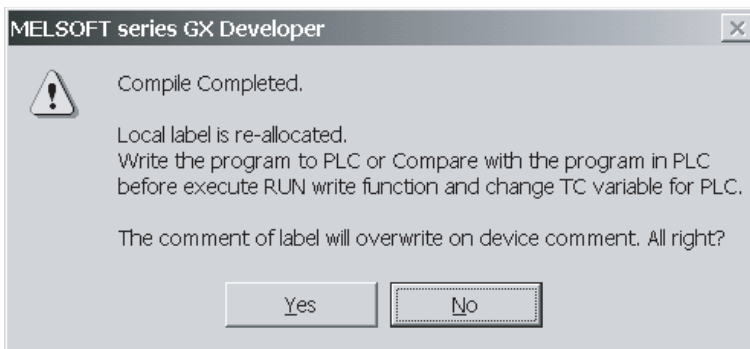
■ 进行转换(编译)

以下通过创建的程序说明转换(编译)的操作方法。



- 1) 点击菜单的 [Convert (转换)] → [Convert/Compile (转换/编译)]。

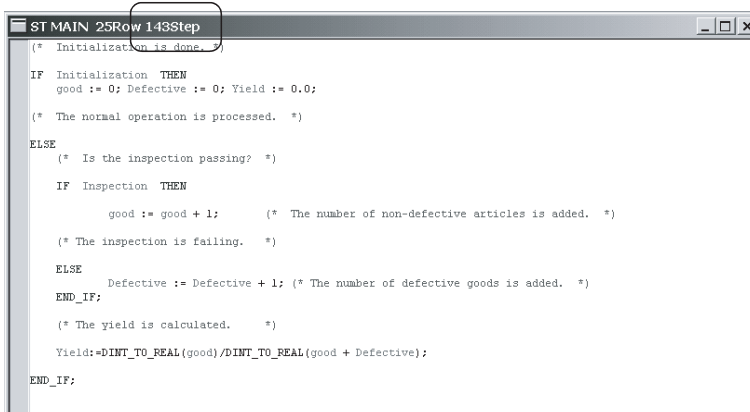
- (1) 正常结束时
将显示以下信息。



转换(编译)完毕。

显示左边的确认画面后，点击 按钮。

如果点击 按钮，有时会显示“Comment data to be referred to (comment by program) does not exist (参照目标的注释数据(程序分类注释)不存在。)”的信息。



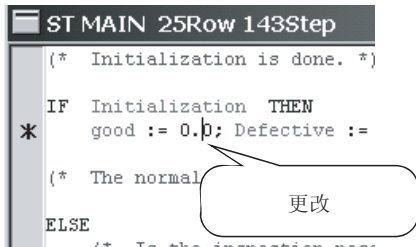
* 转换(编译)正常结束后标题栏中将显示步数。

(2) 发生错误时

将显示编译出错显示(详细)对话框。

以下对实际编译出错时的调试操作进行确认。

1) 更改程序使之发生编译错误。



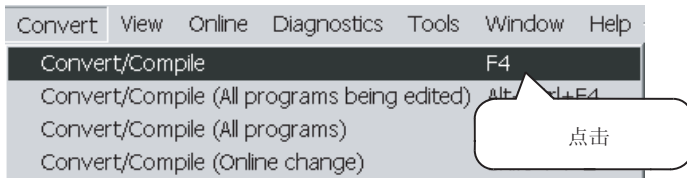
```
ST MAIN 25Row 143Step
(* Initialization is done. *)
IF Initialization THEN
* good := 0.0; Defective :=
(* The normal
ELSE
(* To the inspection next
```

更改

更改列表-2 的第 3 行。

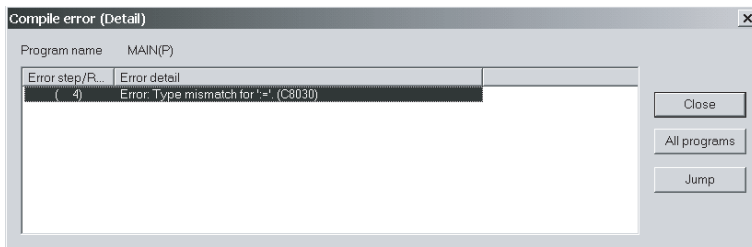
RYOUHIN := 0; → RYOUHIN := 0. 0;

2) 进行转换(编译)。



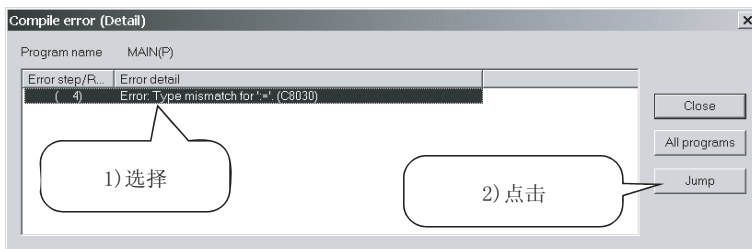
点击菜单的[Convert(转换)]→
[Convert/Compile(转换/编译)]。

3) 发生编译错误, 显示对话框。



确认出错的步/行及出错内容。

4) 确认发生错误的行。



- 1) 通过鼠标选择出错内容。
- 2) 点击 **Jump(跳转)** 按钮。

转下页



- 5) 调查出错原因，对不正常的之处进行修正。
ST 编辑画面中将显示表示出错位置的标记。
确认出错内容及程序内容，对程序进行修正。

```

ST MAIN 25Row *****Step *
(* Initialization is done. *)
IF Initialization THEN
  good := 0.0; Defective := 0; Yield := 0.0;
  The normal operation is processed. *)
  (* The inspection is failing. *)
  good := good + 1; (* The number of non-defective articles is added. *)
  (* The inspection is failing. *)
ELSE
  Defective := Defective + 1; (* The number of defective goods is added. *)
END_IF;
(* The yield is calculated. *)
Yield:=DINT_TO_REAL(good)/DINT_TO_REAL(good + Defective);
END_IF;
  
```

表示出错位置的标记

修正列表-2 的第 3 行。

good := 0. 0; → good := 0;

点击菜单的 [Convert (转换)] →
[Convert/Compile (转换/编译)]。

要点

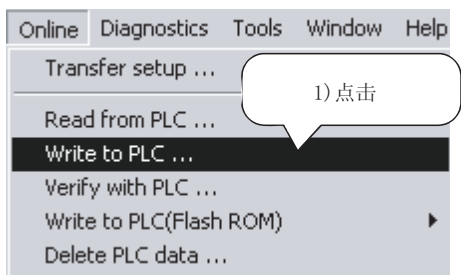
有时会发生出错位置与实际修正位置不相同的现象。
应通过“编译出错显示(详细)”对话框中显示的出错内容及出错位置表示标记所指示的行的程序内容对不正常的位置进行确定。

在 4 章中介绍通过可编程控制器 CPU 将转换(编译)的顺控程序写入到可编程控制器 CPU 的步骤以及读取顺控程序的步骤。

■ PLC 写入

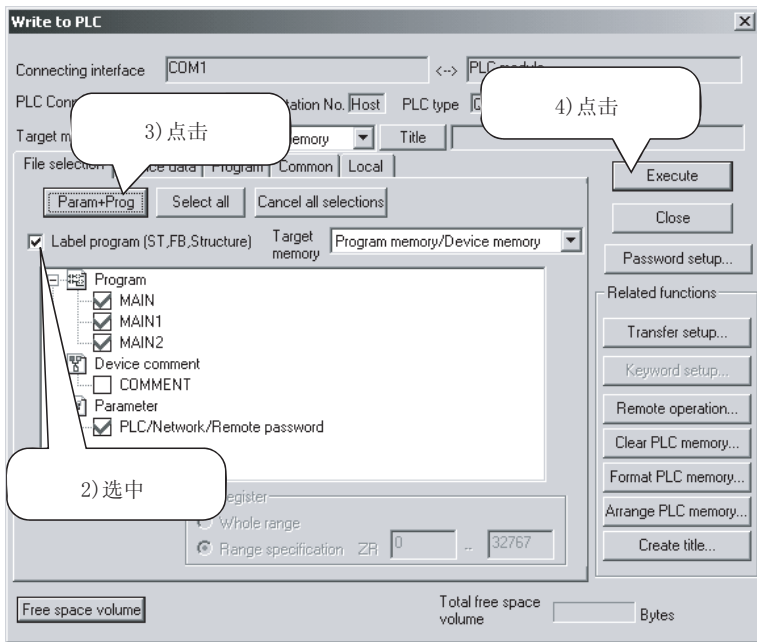
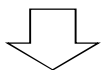
以下介绍 PLC 写入的操作方法。

显示 PLC 写入对话框，将程序及参数写入到可编程控制器 CPU 中。



* PLC 写入时，应将可编程控制器 CPU 置于 STOP 状态。

1) 点击菜单的[Online 在线]→[Write to PLC (PLC 写入)]。

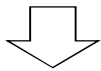


2) 在<<File selection(文件选择)>>选项卡中勾选“Label program (ST, FB, structure) 写入标识程序(ST、FB、结构体)”的选择框。

* 未在选择框中勾选时，只能写入实际程序。

3) 点击“Param + Prog(参数+程序)”。

4) 点击 **Execute(执行)** 按钮。



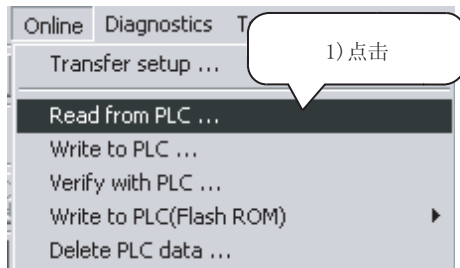
* 对可编程控制器 CPU 进行复位，进入运行状态。

发生了错误时，应在 GX Developer 的菜单[Diagnostics(诊断)]→[PLC diagnostics (PLC 诊断)]中确认出错内容。

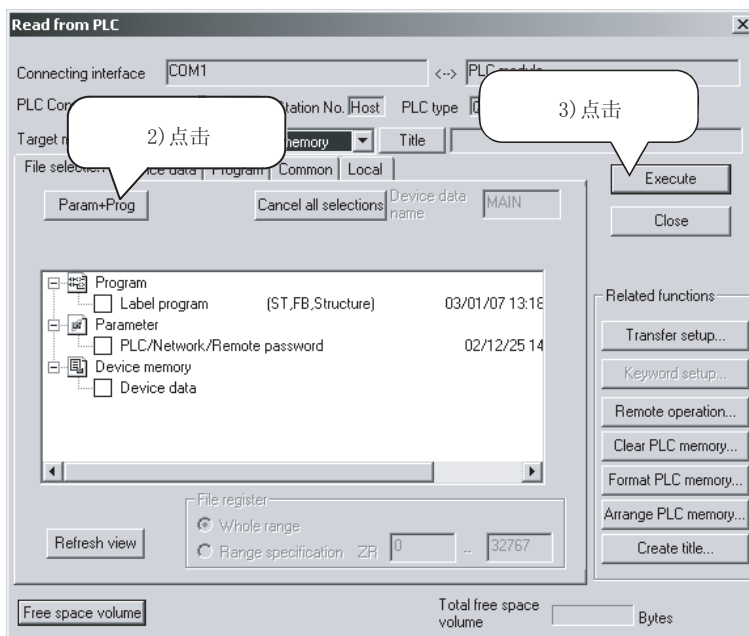
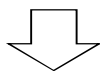
■ PLC 读取

以下介绍 PLC 读取的操作方法。

显示 PLC 读取对话框，从可编程控制器 CPU 中读取程序及参数。

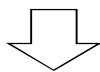


1) 点击菜单的 [Online (在线)] → [Read from PLC (PLC 读取)]。



2) 在 <<File selection (文件选择)>> 选项卡中点击 “Param + Prog (参数+程序)”。

3) 点击 **Execute (执行)** 按钮。



发生了错误时，应在 GX Developer 的菜单 [Diagnostics (诊断)] → [PLC diagnostics (PLC 诊断)] 中确认出错内容。

在 5 章中介绍使用监视功能/软元件测试功能对可编程控制器 CPU 中写入的顺控程序进行在线调试的操作等有关内容。

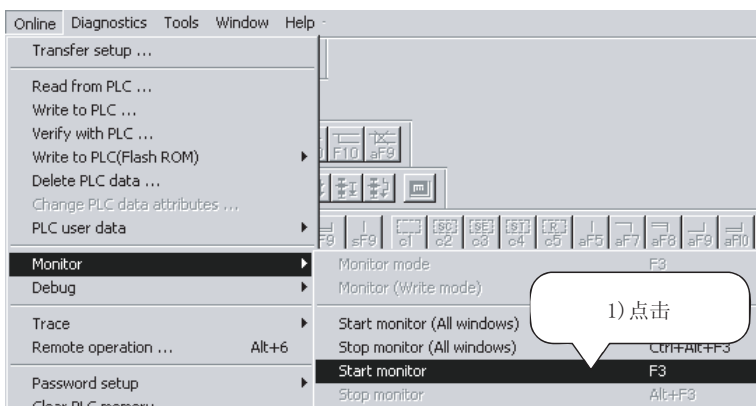
在该章中介绍的项目如下所示：

- ☞ 监视顺控程序。
- ☞ 更改位软元件的值进行软元件测试。
- ☞ 对顺控程序的进行部分更改，对处于运行状态的可编程控制器 CPU 进行写入。

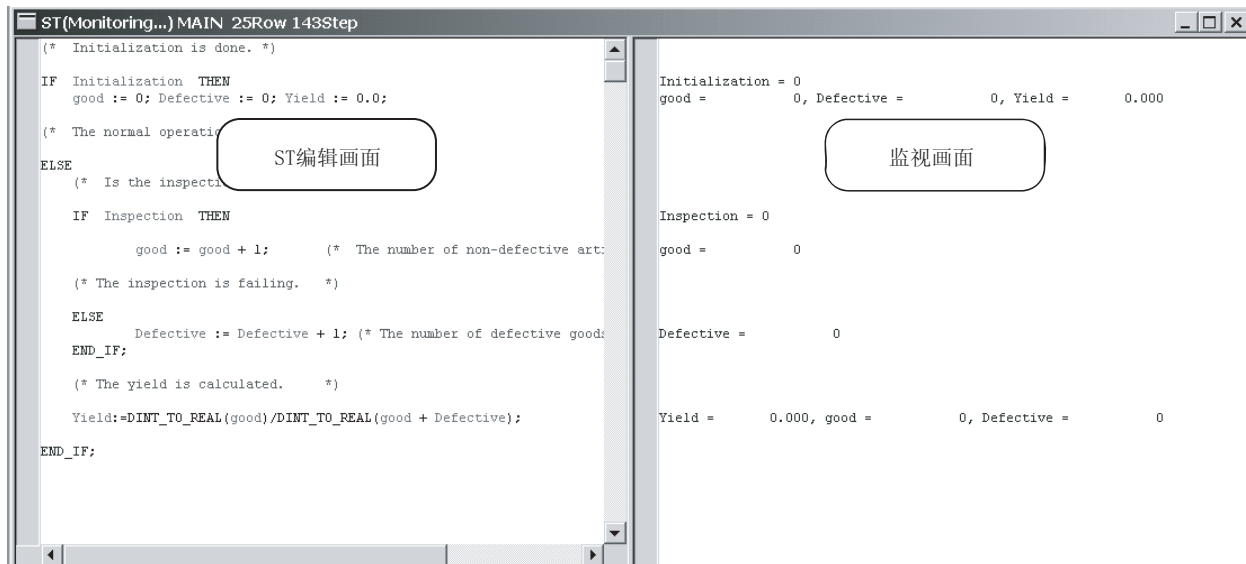
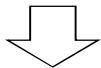
5.1 监视顺控程序

以下介绍监视顺控程序的操作方法。

■ 显示监视画面



- 1) 点击菜单的 [Online (在线)] → [Monitor (监视)] → [Start monitor (开始监视)]。



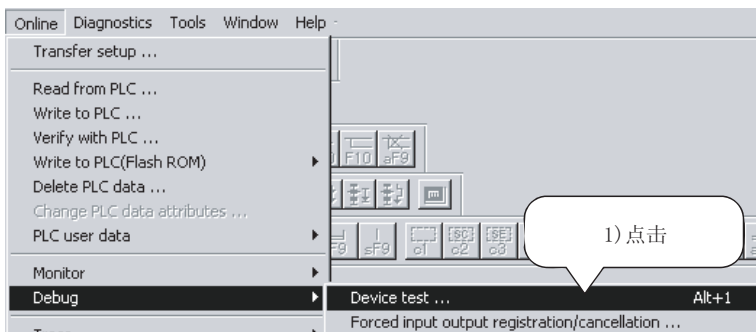
在 ST 编辑画面中显示的标识显示在监视画面的相同一行中。

5.2 软元件测试

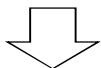
可以直接更改可编程控制器 CPU 内的标识(位软元件/字软元件)的值。
以下对位软元件的值进行更改后对程序的动作进行确认。

■ 确认程序的动作

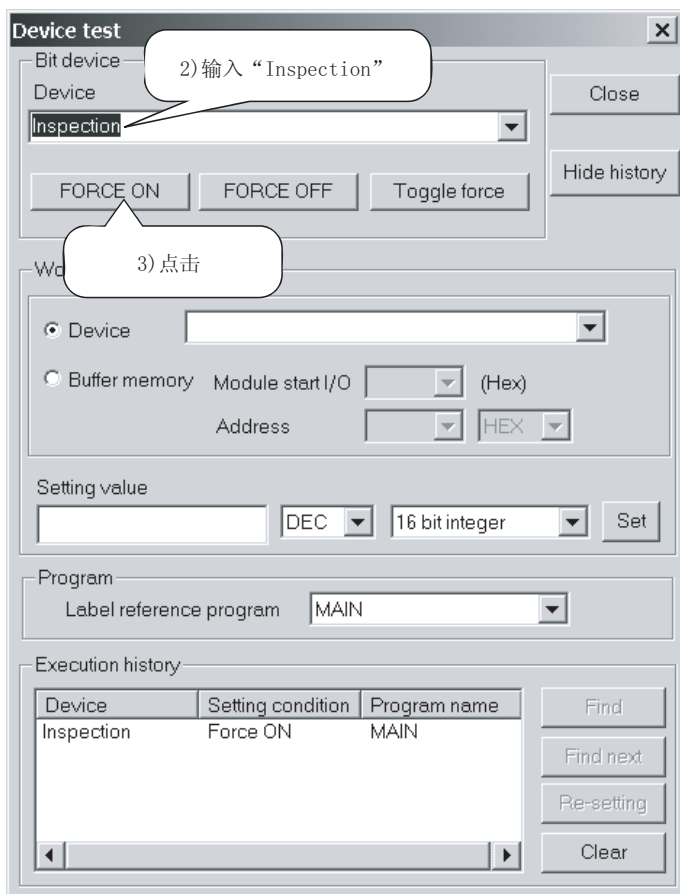
以下介绍位软元件的值的更改操作。



- 1) 点击菜单的 [Online (在线)] → [Debug (调试)] → [Device test (软元件测试)]。



对表示位软元件的标识“Inspection”强制执行 ON。



- 2) 在位软元件中输入“Inspection”。
- 3) 点击 **FORCE ON(强制 ON)** 按钮。





在监视画面中进行确认。

```

ST(Monitoring...) MAIN 25Row 143Step
(* Initialization is done. *)
IF Initialization THEN
  good := 0; Defective := 0; Yield := 0.0;
(* The normal operation is processed. *)
ELSE
  (* Is the inspection passing? *)
  IF Inspection THEN
    good := good + 1; (* The number of non-defective a
  (* The inspection is failing. *)
  ELSE
    Defective := Defective + 1; (* The number of defective go
  END_IF;
  (* The yield is calculated. *)
  Yield:=DINT_TO_REAL(good)/DINT_TO_REAL(good + Defective);
END_IF;
  
```

Initialization = 1
 good = 503493, Defective = 1, Yield =

Inspection = 1
 good = 503493

Defective = 1

Yield = 1.000, good = 503493, Defective =

ON
 增加了“good”的值。

请对其它标识的值也进行更改，对程序的动作进行确认。

要点

GX Developer 对于由 ST 语言创建的程序支持以下的调试功能：

- 执行程序至任意设置的中断点为止进行调试的中断执行
- 以 1 行为单位执行程序、进行调试的逐行执行

详细内容请参阅“GX Developer 操作手册(结构化文本篇)”中记述的调试功能。

5.3 运行中写入

可编程控制器 CPU 处于 RUN 状态时，可以更改部分顺控程序。
这称为运行中写入。

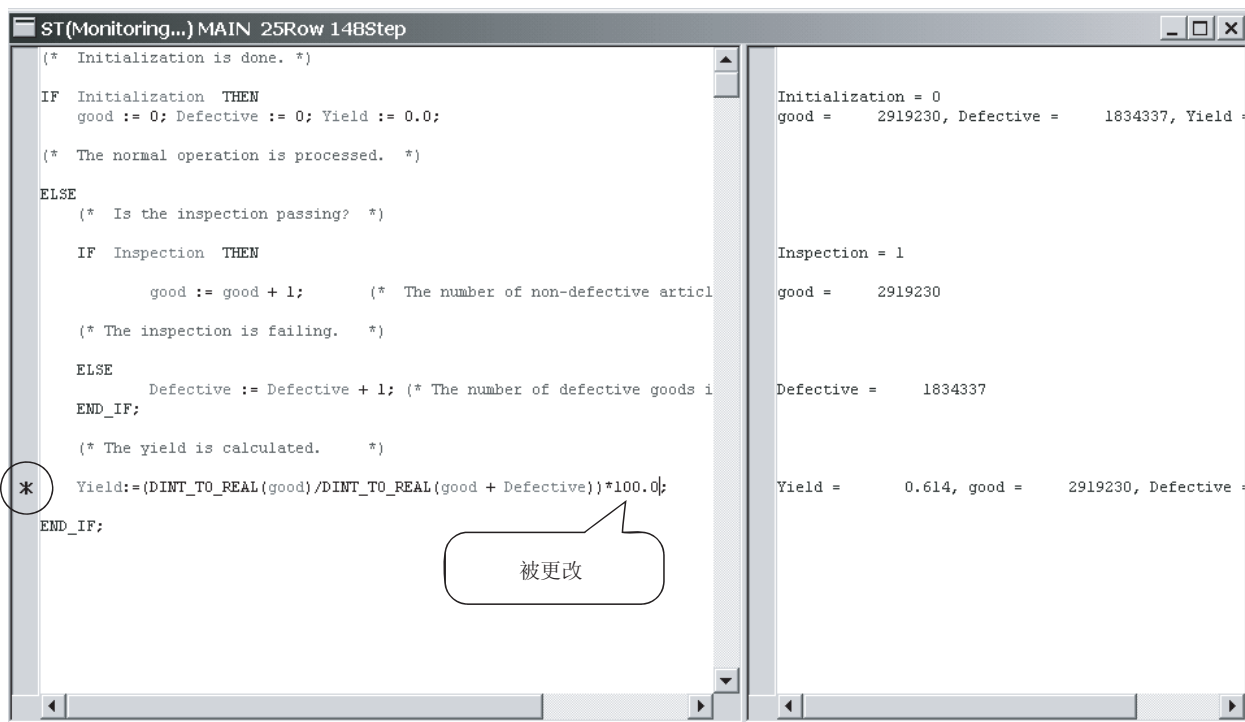
以下更改部分顺控程序进行实际的运行中写入。

■更改部分程序进行运行中写入

更改“Yield”的计算公式，进行运行中写入。

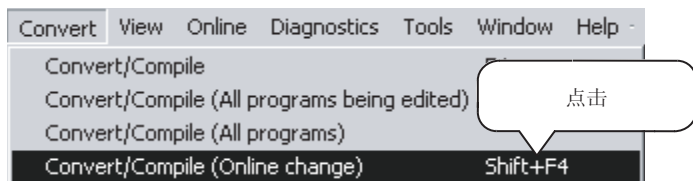
1) 更改部分程序。

```
Yield := DINT_TO_REAL (good)/DINT_TO_REAL (good + Defective);
      ↓
Yield := (DINT_TO_REAL (good)/DINT_TO_REAL (good + Defective))*100.0;
```



在更改行的指示栏中，显示有表示运行中写入对象行的“*”。

2) 执行运行中写入。

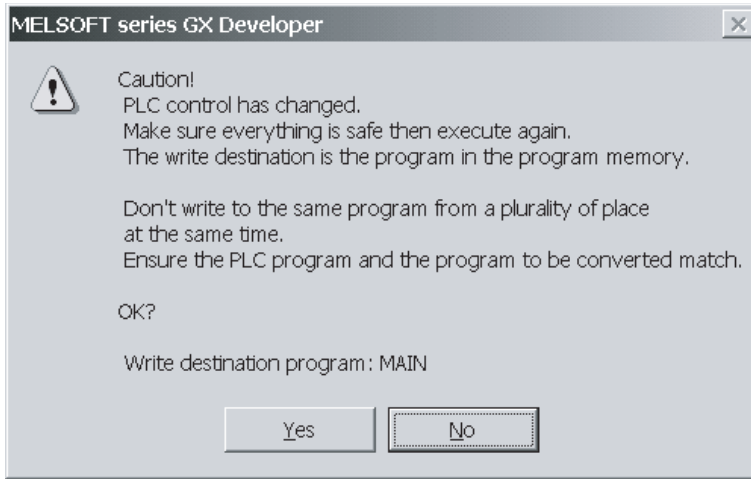


点击菜单的[Convert (转换)]→
[Convert/Compile (Online change)
(转换/编译(运行中写入))].



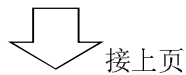


3) 显示确认信息。



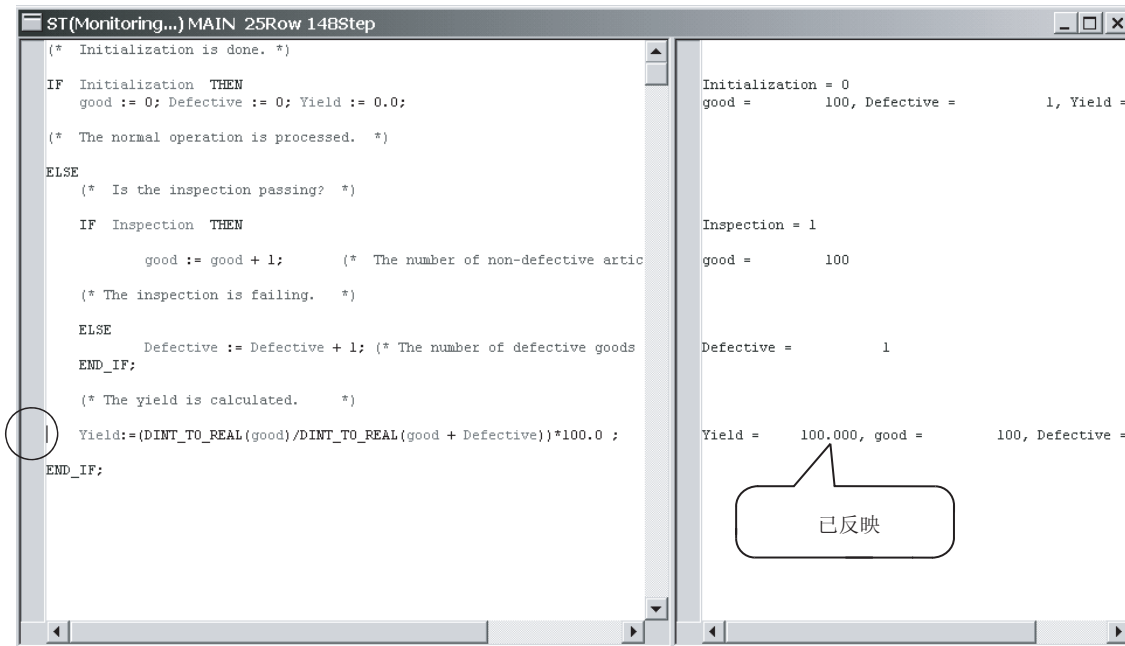
点击 按钮。





4) 运行中写入完毕。

点击 **OK** 按钮。

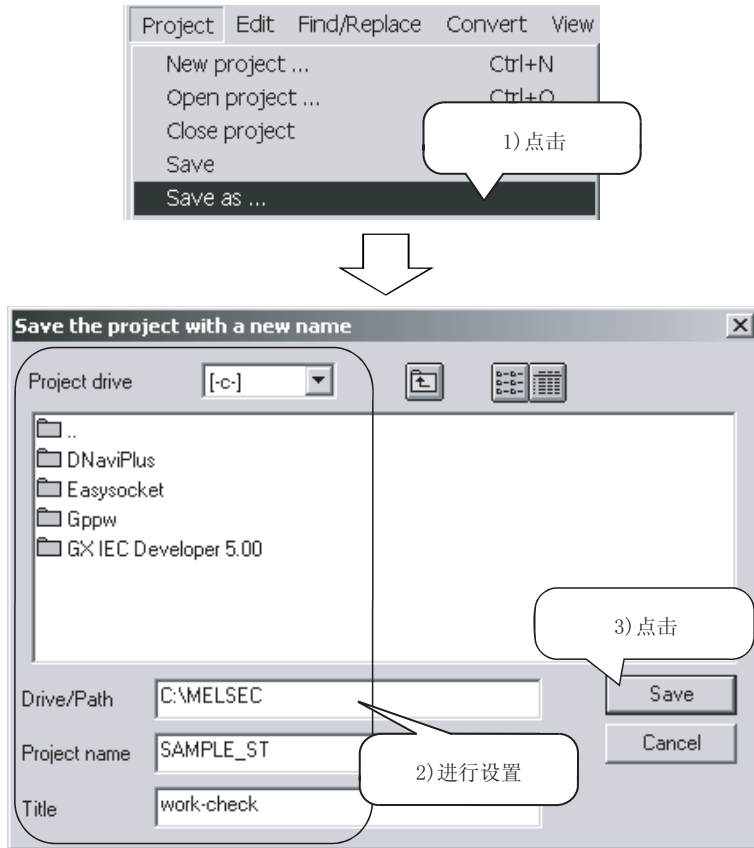


表示运行中写入对象行的“*”消失。
在监视画面中，确认“Yield”的当前值已变化。

在 6 章中，介绍对完成的工程进行附加名称后保存(另存为)等有关内容。

■ 保存工程

对所创建的工程进行附加名称后保存(另存为)。



1) 点击菜单[Project(工程)]→[Save as(另存为)]。

2) 输入以下内容：

- Drive/Path(驱动器/路径)：
C:\MELSEC
- Project name(工程名)：SAMPLE_ST
- Title(标题)：work-check

3) 点击 按钮。

所创建的程序被保存为以下样式。

Drive/Path(驱动器/路径)：C:\MELSEC
Project name(工程名)：SAMPLE_ST
Title(标题)：work-check

至此，ST用工程的从新建起至保存为止、从程序的输入起至在线调试为止的一系列操作完毕。
应对这些操作充分理解后再阅读第8章。

本章介绍编辑 ST 程序时预先了解可带来便利的功能。

若要了解详细内容，请参阅“GX Developer 操作手册(结构化文本篇)”。

(1) 窗口拆分

在编辑较大程序的过程中，希望确认已编辑的内容时，需要一页一页地滚动画面去阅览，十分麻烦 ...

在这种情况下可以使用“Window division(窗口拆分)”。

选择菜单的[Window(窗口)] → [Divide into two (拆分为 2 个)]后，将显示被拆分上、下的 2 个画面。可以在拆分的窗口中分别进行滚动、编辑。

(2) 书签

希望跳转至特定的行时，需要从程序的最初开始查找，十分麻烦 ...

在这种情况下可以使用“Bookmark(书签)”功能。

预先在菜单的[Find/Replace(查找/替换)] → [Bookmark setting/release(书签设置/解除)]或者

[Find/Replace(查找/替换)] → [Find(查找)] → “Bookmark setting(书签设置)”中设置书签。

可以选择菜单的[Find/Replace(查找/替换)] → [Bookmark list(书签一览)]，在书签一览对话框中选择任意的行后进行跳转。

(3) 标识信息的显示

希望了解标识中分配的软元件时 ...

在这种情况下可以使用“Label information(标识信息)”功能。

将鼠标指针对准标识，将以工具条的形式显示以下内容，可以一目了然地确认其内容。

Label name → Label type → Label comment → Device (标识名 → 标识类型 → 标识注释 → 软元件)
--

is displayed in the tool tip format for at-a-glance confirmation of the contents.

注：• 在已转换(编译)的程序中有效。

- 在局部变量(局部标识)设置画面中执行“Show assigned device(显示自动分配软元件)”也可进行确认。

(4) 函数选择

希望输入函数时却忘记了函数名 ...

在这种情况下可以使用“Select function(函数选择)”功能。

选择菜单的[Edit(编辑)] → [Select function(函数选择)]，可以在函数选择对话框中选择显示的函数名。此外，如果插入函数，将以工具条形式显示函数自变量的类型，可供输入自变量时参考。

(5) 显示颜色、字体的更改

编辑画面上的字符串的颜色相同难以辨认，希望更改字的大小时 ...

在这种情况下，可以使用“Change display color(显示颜色变更)”、“Font(字体变更)”功能。

选择菜单的[Tools(工具)] → [Change display color(显示颜色变更)]后，将显示颜色变更对话框，可以选择注释、控制语句、字符颜色、标识、背景颜色。通过更改显示颜色使阅读更加容易。

选择菜单的[Tools(工具)] → [字体变更]后，将显示字体变更对话框，可以选择字体的类型、式样、大小。通过设置使使用变得容易，提高了可操作性。

备忘录

7

在 1~7 章中介绍了创建 ST 程序的基本的操作方法及功能。

在 8 章中介绍通过 ST 语言创建功能块(FB)，将其粘贴到梯形图程序中使用的程序的创建有关内容。

该章中介绍的主要项目如下所示：

- ☞ 新建 FB。
- ☞ 定义 FB 变量。
- ☞ 创建 ST 程序。
- ☞ 通过 ST 语言创建 FB。
- ☞ 将通过 ST 语言创建的 FB 用于主程序。

8.1 创建 FB

何为功能块(FB)?

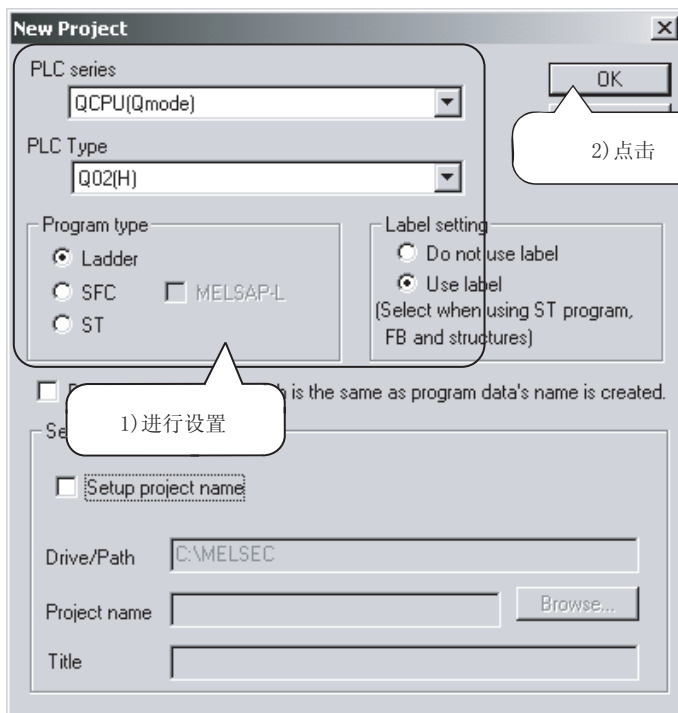
FB 是指，将常用的处理作为部件预先进行定义，可以用于各程序的必要部分。

新建工程

■ 创建新工程

以下介绍用于通过梯形图创建主程序的新建工程的方法。

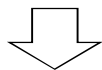
点击菜单的[Project(工程)] → [New project(新建工程)]，显示新建工程对话框。



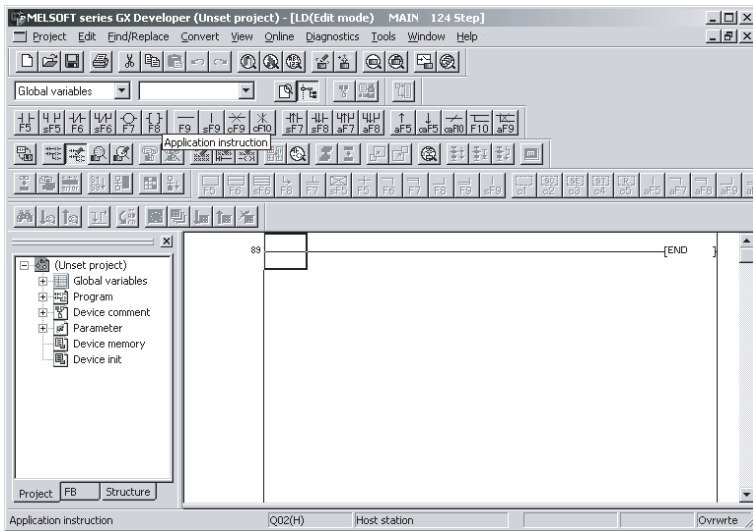
1) 输入以下内容：

- PLC series(PLC 系列)：
QCPU(Q 模式)
- PLC type(PLC 类型)：
Q02(H)
- Label setting(标识设置)：
Use label(使用标识)
- Program type(使用标识)：
Ladder(梯形图)

2) 点击 按钮。



转下页

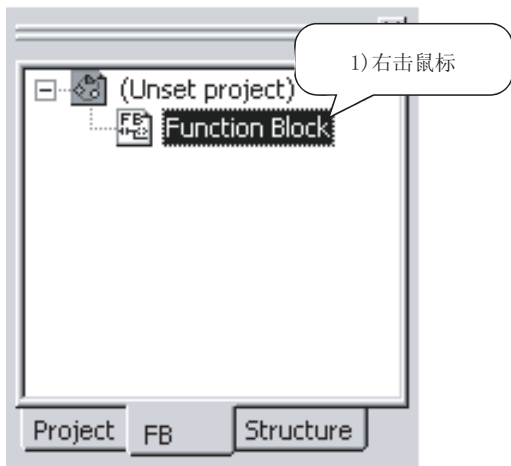


3) 新建工程完毕。

新建FB

■ 添加 FB

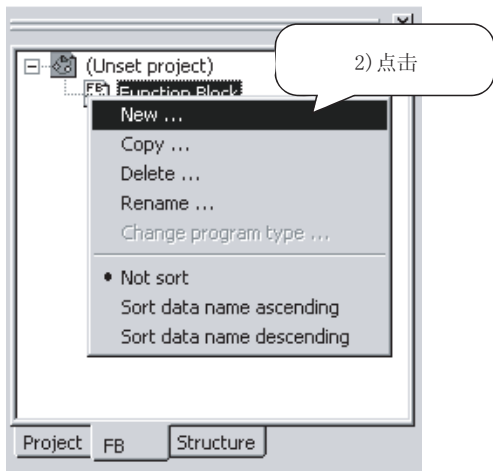
以下介绍新建通过 ST 语言创建的 FB 的操作方法。



1) 用鼠标右击<<FB>>选项卡的“Function Block(功能块)”，显示菜单。



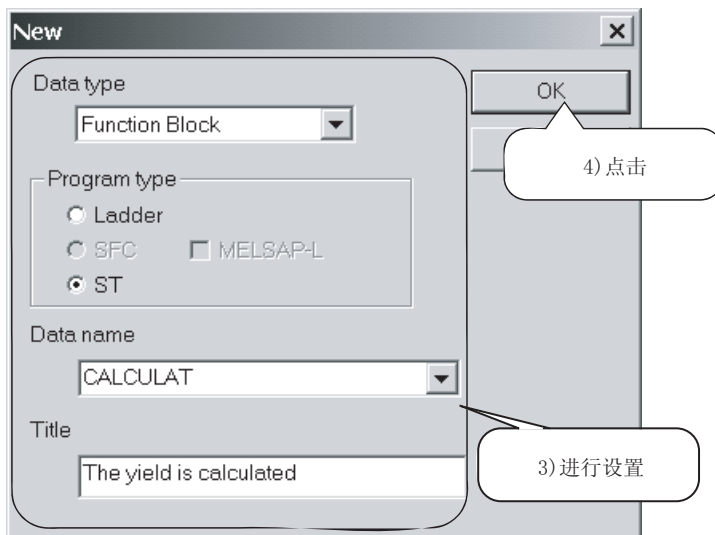
接上页



2) 点击“New(新建)”。

↓

在新建对话框中进行设置。



- 3) 输入以下内容：
- Data type(数据类型) :
Function Block(功能块)
 - Program type(程序类型) :
ST
 - Data name(新建数据名) :
CALCULAT
 - Title(标题) :
The yield is calculated

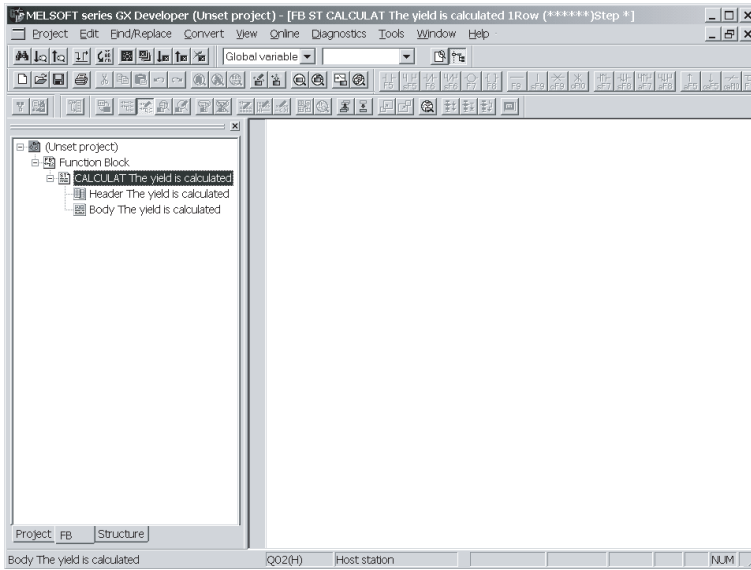
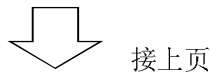
4) 点击 **OK** 按钮。

↓



5) 点击 **Yes** 按钮。

转下页



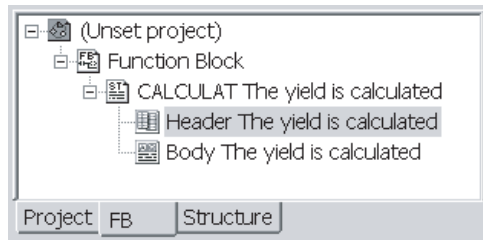
6) 数据名:CALCULAT 将被新建。

定义FB变量

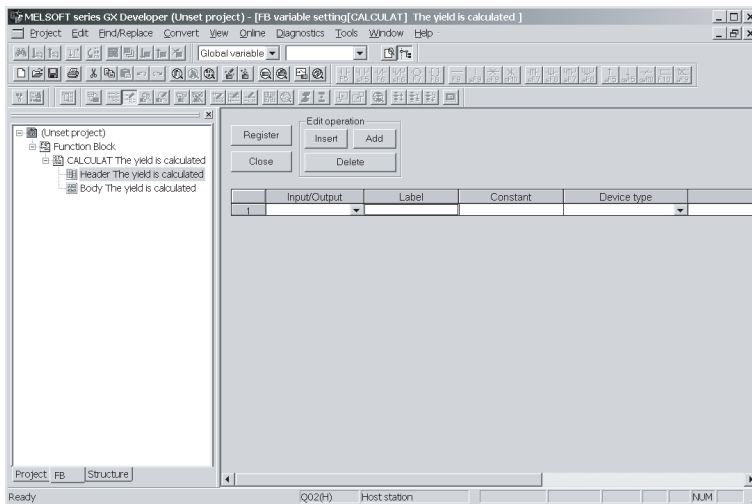
FB 内使用的标识被称为 FB 变量(FB 标识)。

■ 显示 FB 变量(FB 标识) 设置画面

以下介绍定义 FB 变量(FB 标识) 的操作方法。



1) 双击<<FB>>选项卡的“Header(FB 变量)”。



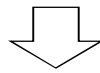
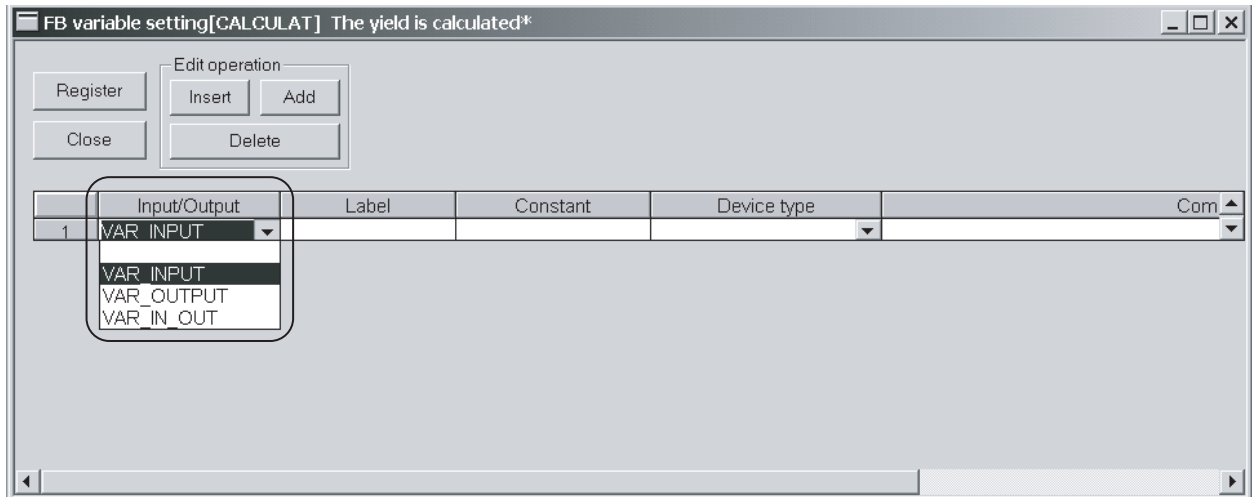
2) 显示 FB 变量(FB 标识) 设置画面。

■ 设置 FB 变量(FB 标识)

1) 选择输入输出类型

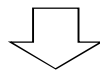
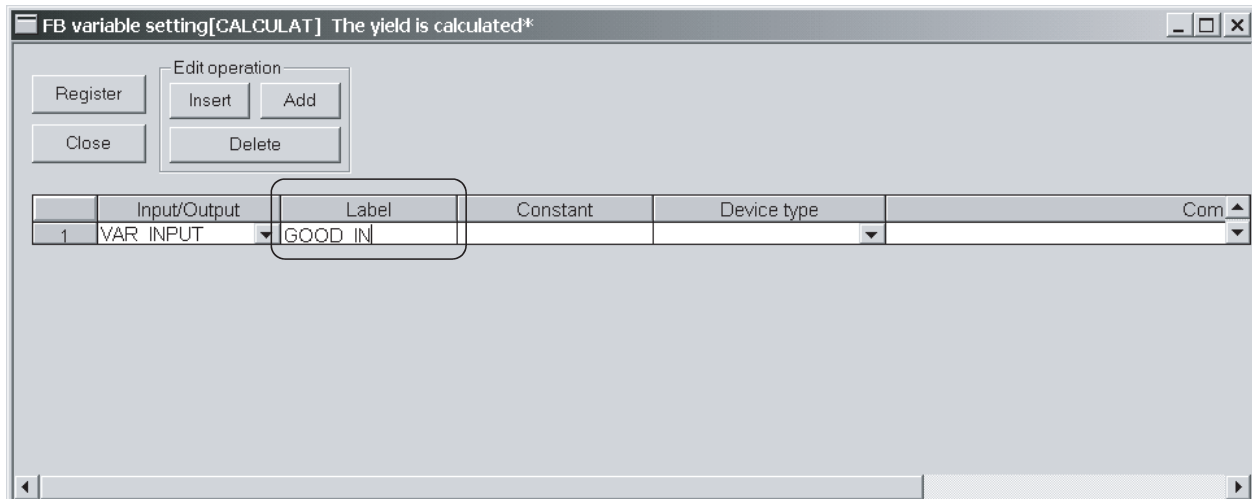
选择标识的类型。类型有以下 4 种：

- 输入变量..... 从 FB 外部输入的变量
- 输出变量..... 输出到 FB 外部的变量
- 输入输出变量... 具有输入及输出功能的变量
- “空栏”..... FB 内部使用的变量



2) 输入标识名。

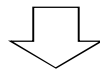
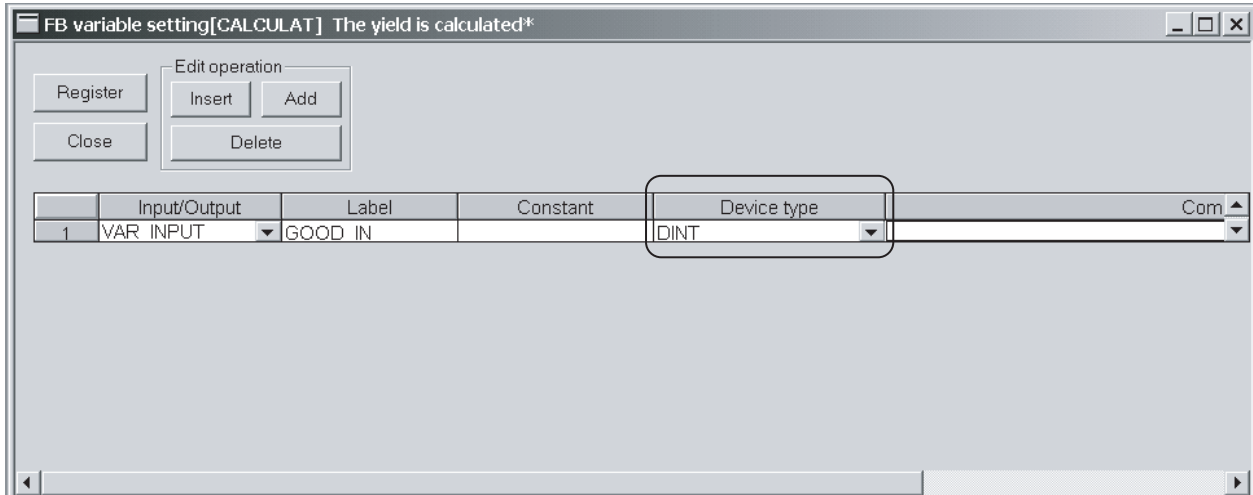
在半角 16 个字符(全角 8 个字符)范围内输入标识名。



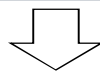
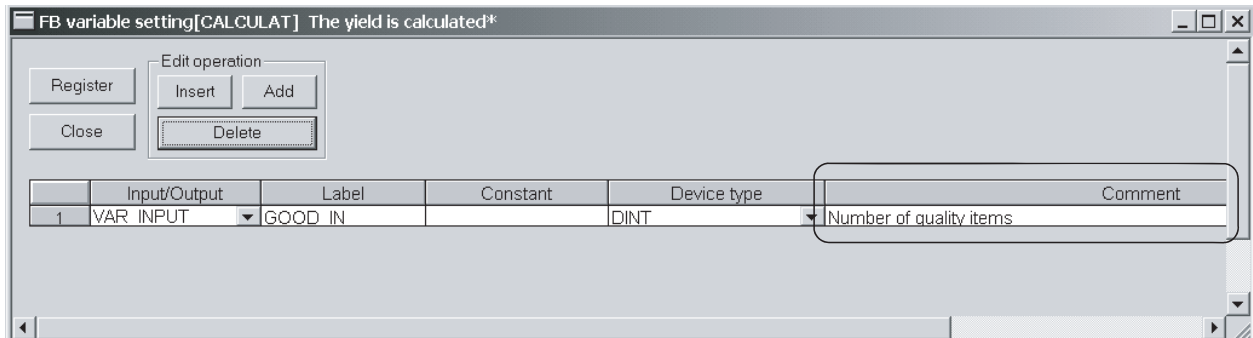
转下页



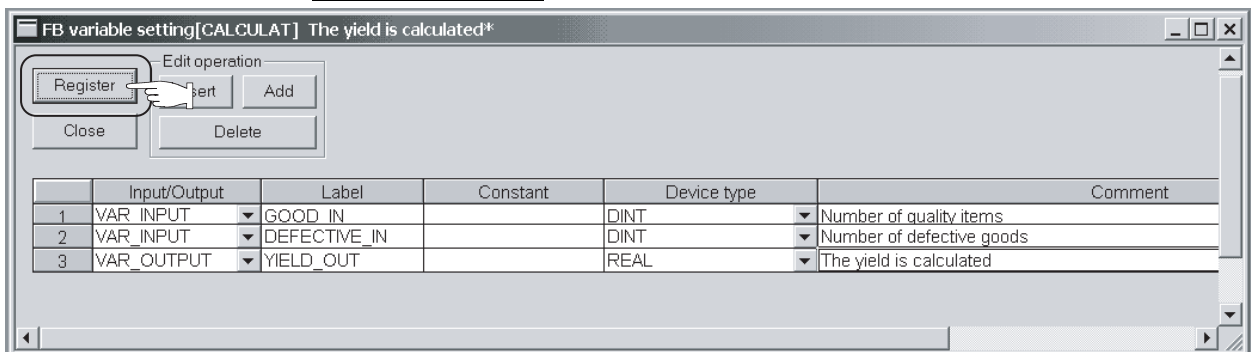
- 3) 输入软元件类型。
直接输入或者从列表框中选择。



- 4) 在标识中输入注释。
在半角 64 个字符 (全角 32 个字符) 的范围内设置。

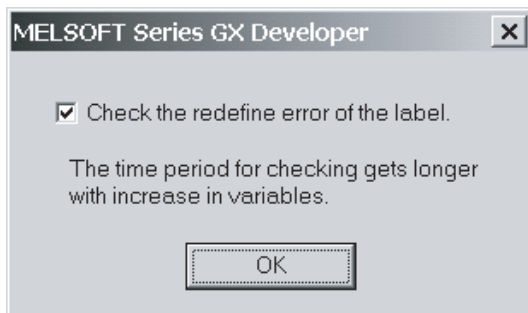


- 5) 输入结束后, 点击 **Register (登录)** 按钮。

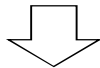




接上页



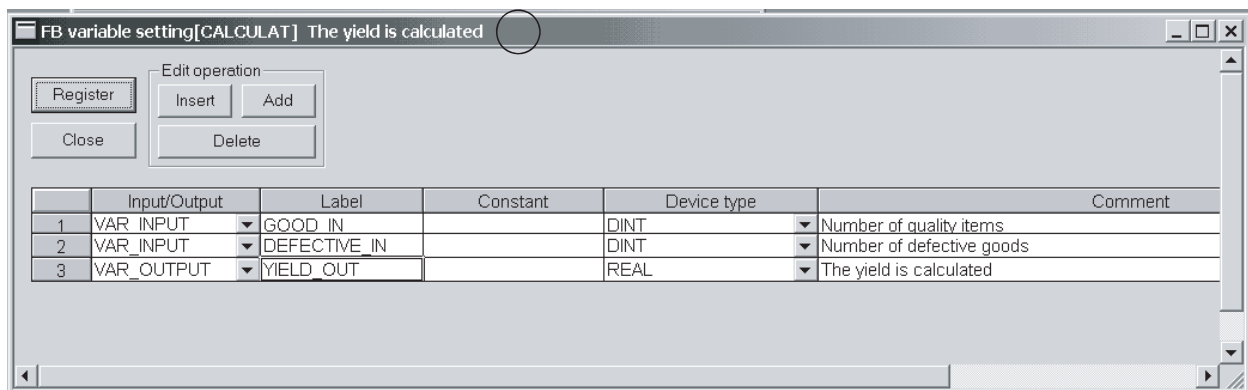
点击 **OK** 按钮。



登录结束。
点击 **OK** 按钮。



登录之后，标题栏中显示的“*”将消失。



备注

有关详细内容请参阅关联手册中记载的“GX Developer 操作手册(功能块篇)”。

通过ST语言创建FB

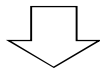
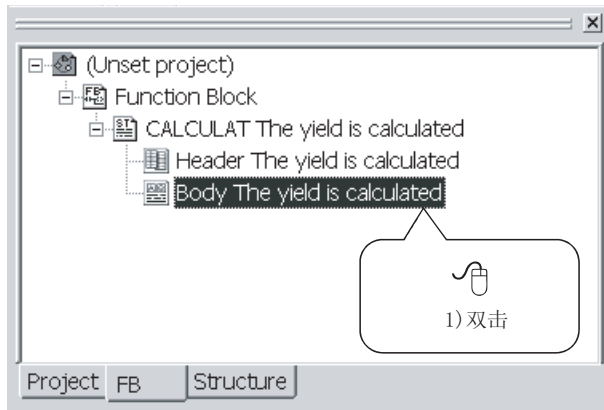
以下介绍有关列表-3 的程序的输入操作。

列表-3

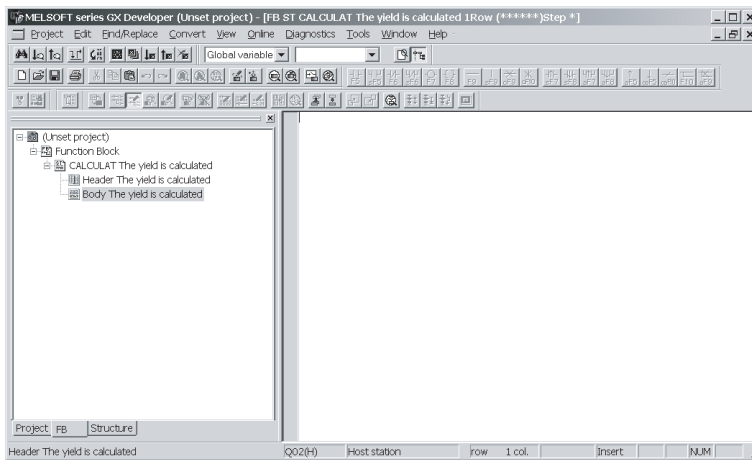
```
YIELD_OUT := DINT_TO_REAL(GOOD_IN)/DINT_TO_REAL(GOON_IN + DEFECTIVE_IN);
```

■显示 FB 定义画面

1) 双击“Body(程序本体)”。



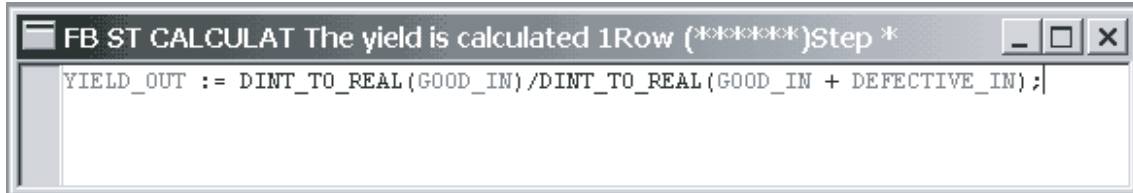
2) 显示 FB 定义画面。



■ 输入程序

将 FB 程序本体以与第 3 章中进行的主程序的输入相同的方法进行输入。

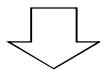
输入列表-3 的程序。

**■ 对 FB 进行转换 (编译)**

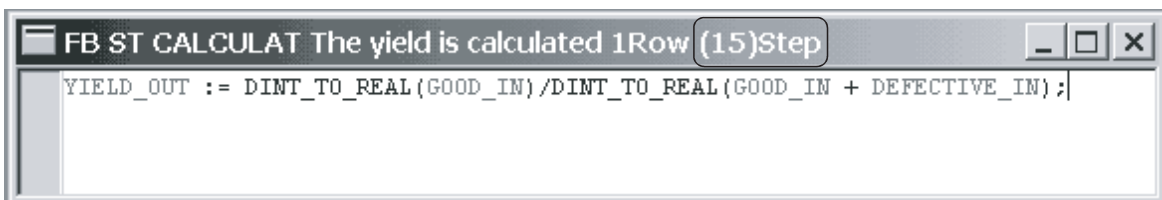
点击菜单的[Convert (转换)]→[转换/编译]进行转换 (编译)。



编译处理完毕。
点击 按钮。



转换 (编译) 正常结束后标题栏中将显示步数。



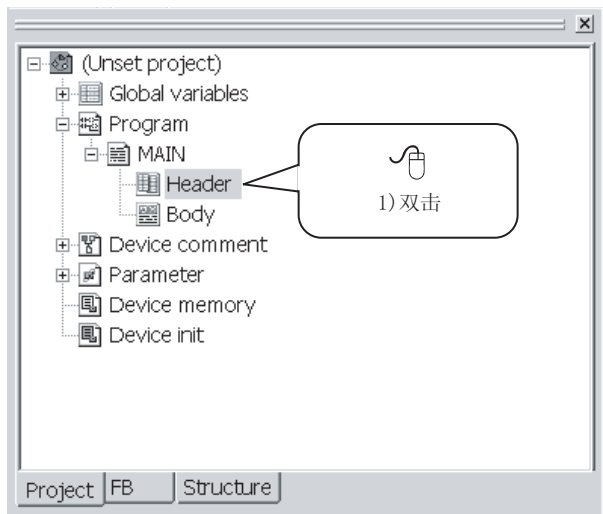
8.2 将 FB 粘贴到主程序中

使用 8.1 节中创建的 FB 创建主程序(梯形图)。

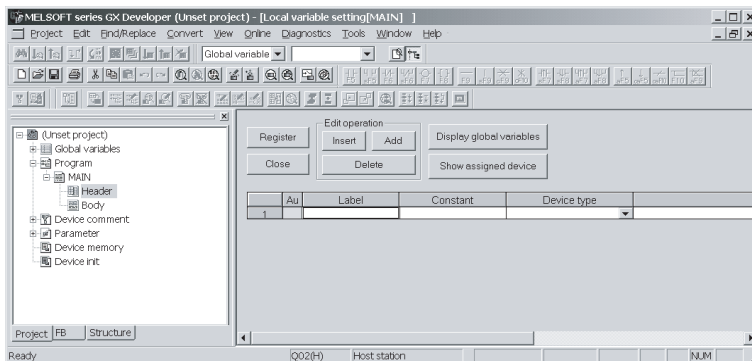
定义局部变量

定义主程序中使用的标识。

■ 显示局部变量(局部标识)设置画面



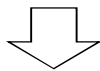
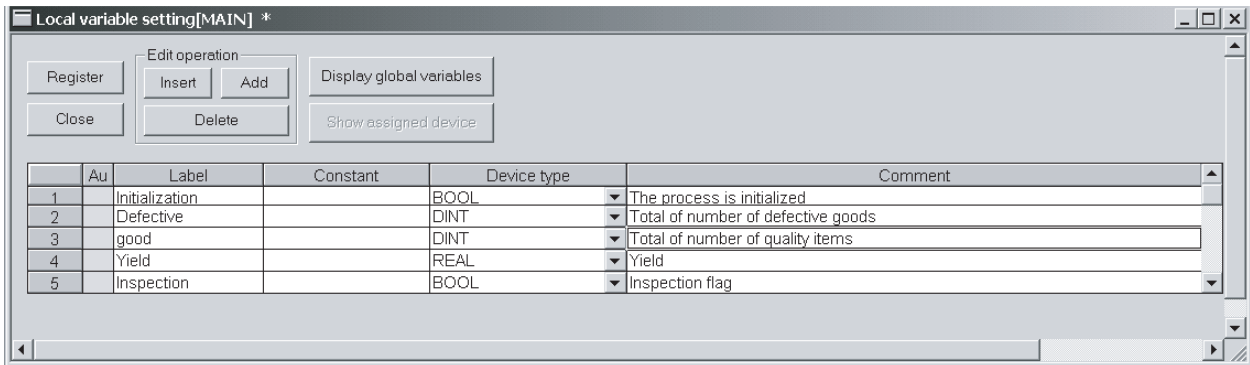
1) 双击“Header(局部标识)”。



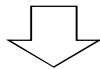
2) 显示局部变量(局部标识)设置画面。

■ 设置局部变量 (局部标识)

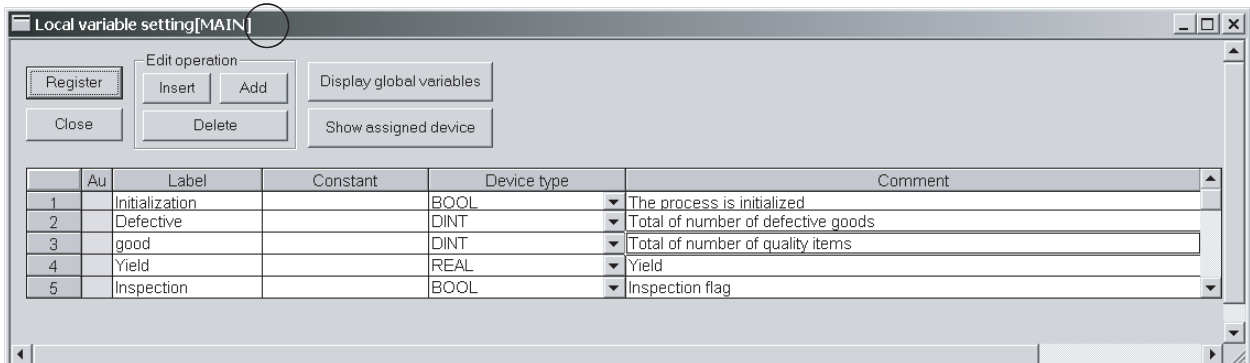
参阅第 3 章进行如下设置:



输入结束后, 点击
Register (登录) 按钮。
 局部变量的登录完毕。
 点击 **OK** 按钮。



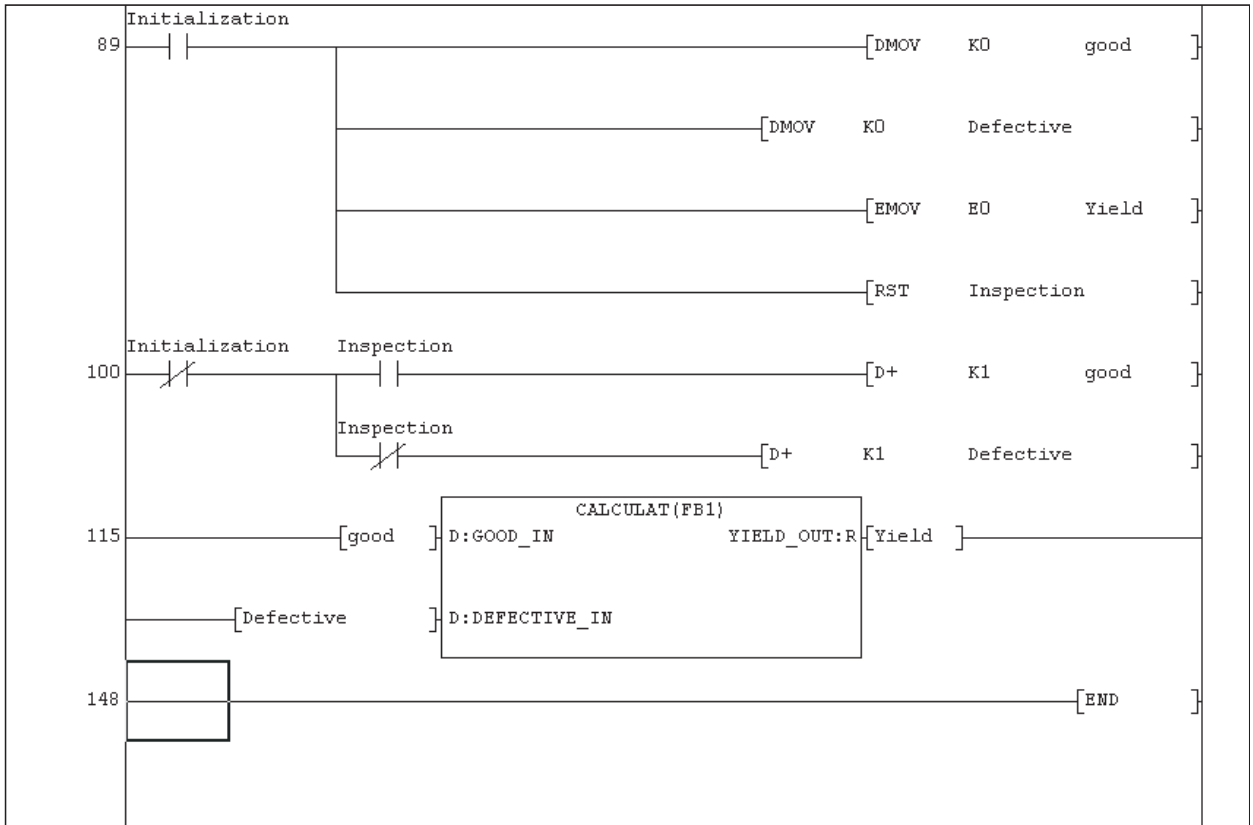
登录后, 标题栏中显示的“*”将消失。



创建主程序

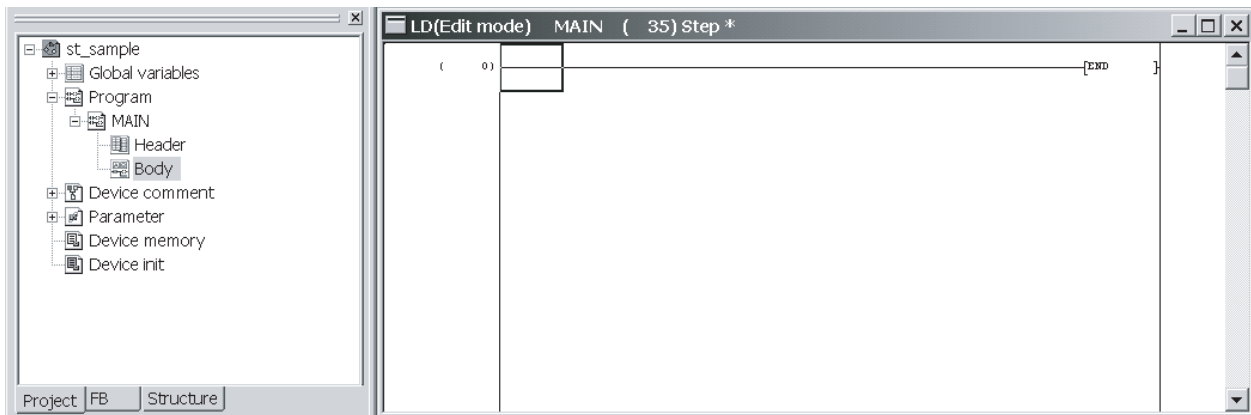
显示主程序的编辑画面，输入下图(列表-4)的程序。

列表-4

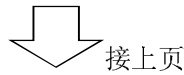


■ 显示编辑画面

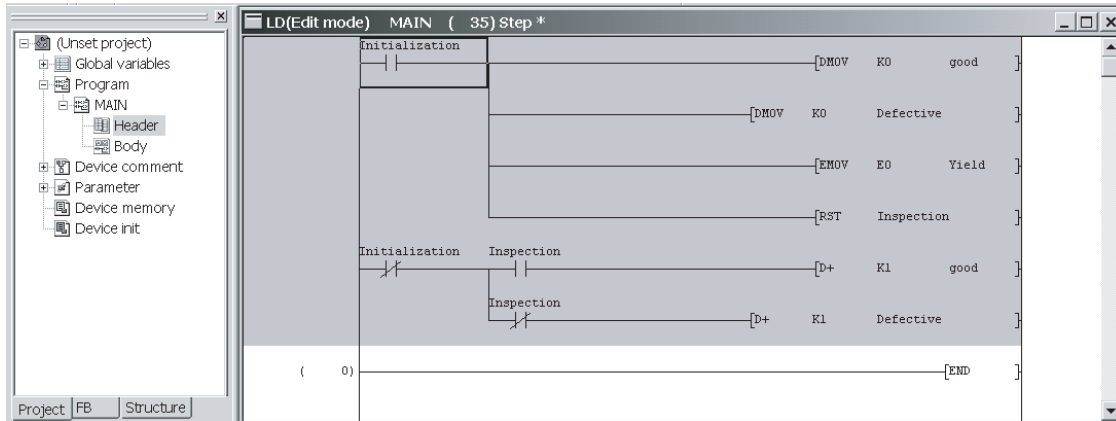
双击<<Project (工程)>>选项卡的“Body(程序本体)”，显示编辑画面。



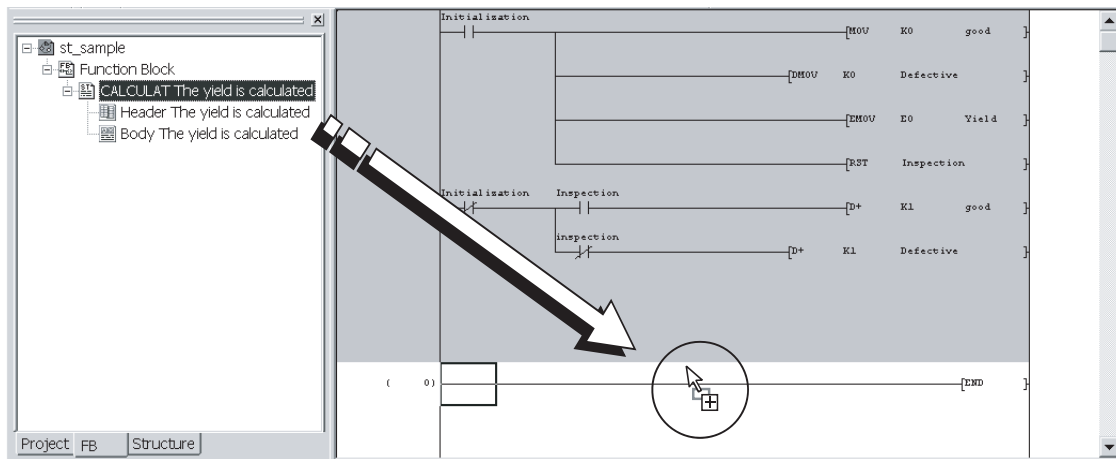
↓ 转下页



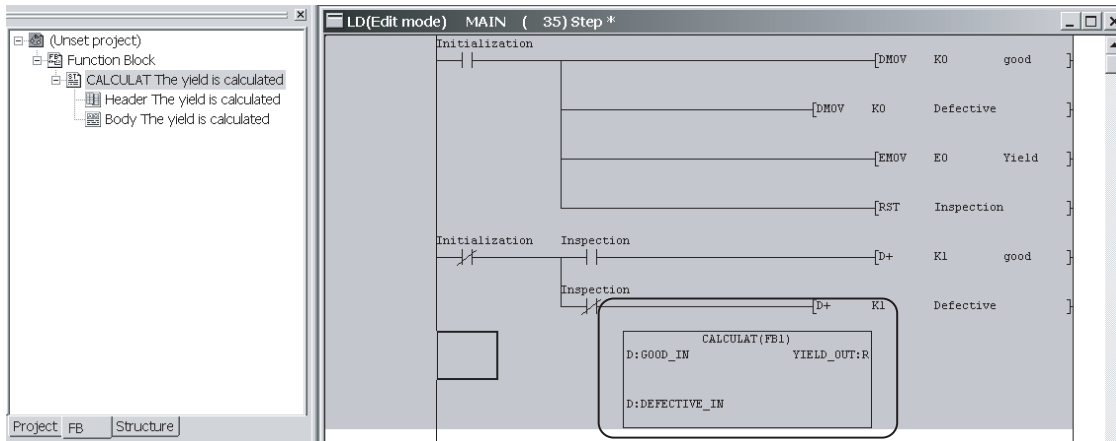
■ 输入梯形图程序
参阅下图进行输入。



■ 粘贴 FB
切换至<<FB>>选项卡，用鼠标将 FB 程序拖放至希望插入的位置。

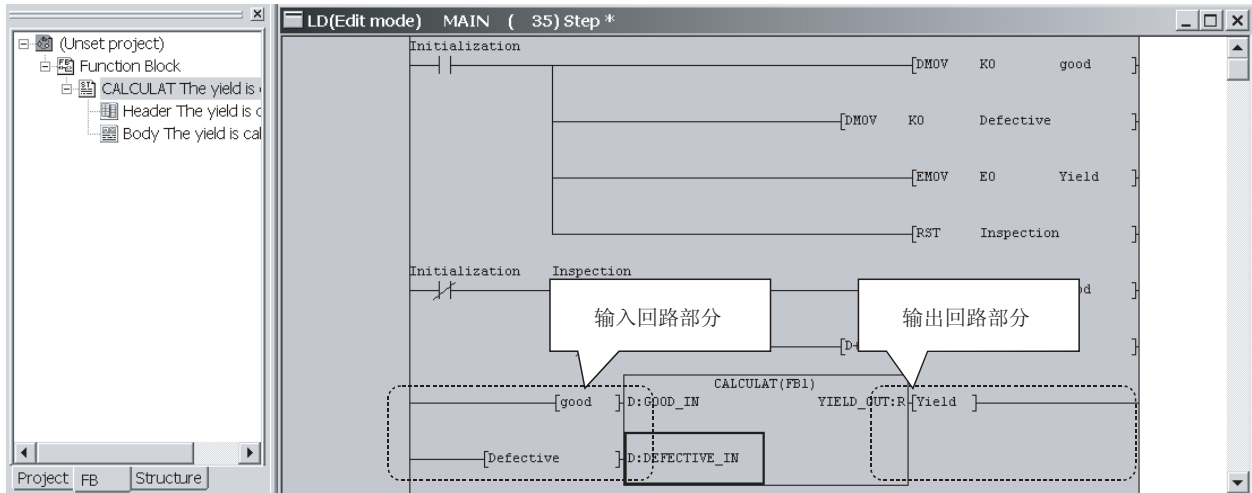


FB 将被插入到主程序中。

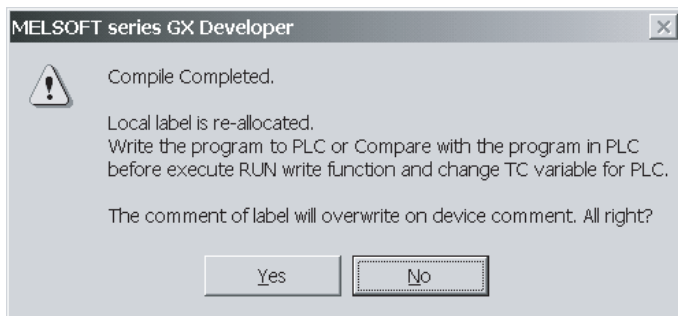




- 输入输入回路部分及输出回路部分
参阅下图，输入输入回路部分及输出回路部分。



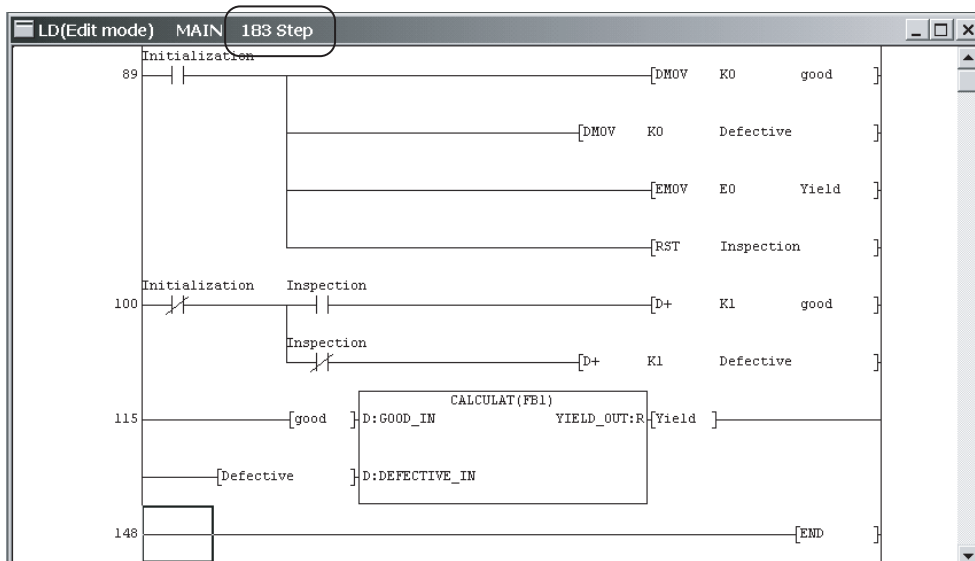
- 转换 (编译)
双击菜单的 [Convert (转换)] → [转换/编译] 进行转换 (编译)。



转换 (编译) 完毕。
点击 **No** 按钮。



转换 (编译) 结束后，标题栏中将显示步数。



8.3 在线

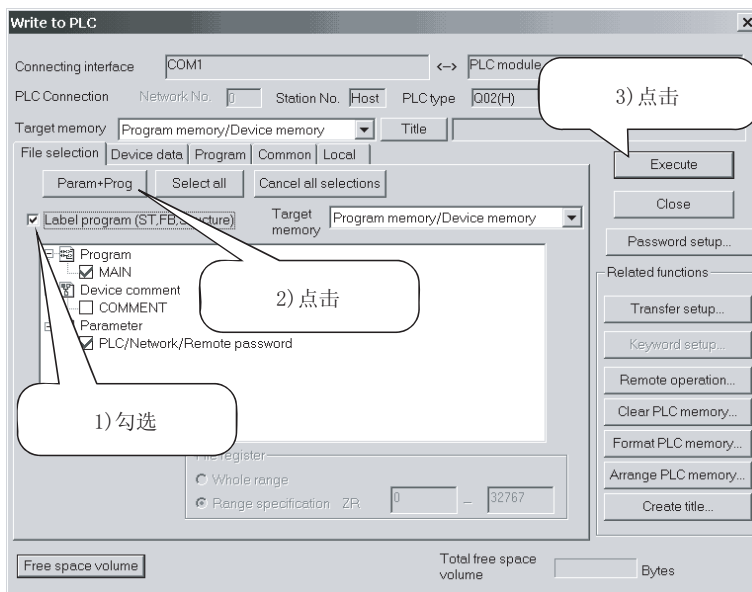
将顺控程序写入到可编程控制器 CPU 中，使用监视功能/软元件测试功能对程序的动作进行确认。

写入到可编程控制器中

■ PLC 写入

参阅第 4 章，进行 PLC 写入。

通过菜单[Online(在线)]→[Write to PLC(PLC 写入)]显示 PLC 写入对话框。



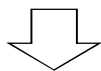
* PLC 写入时，应将可编程控制器 CPU 置于 STOP 状态。

1) 在<<File Selection(文件选择)>>选项卡中勾选“Label program (ST,FB,structure) (标签程序(ST、FB、结构体))”

* 如果未勾选，则只写入实际程序。

2) 点击“Param + Prog(参数+程序)”。

3) 点击 **Execute(执行)** 按钮。



4) PLC 写入完毕。

* 对可编程控制器 CPU 进行复位，进入 RUN 状态。

发生错误时，应通过 GX Developer 的菜单[Diagnostics(诊断)]→[PLC diagnostics(PLC 诊断)]确认出错内容。

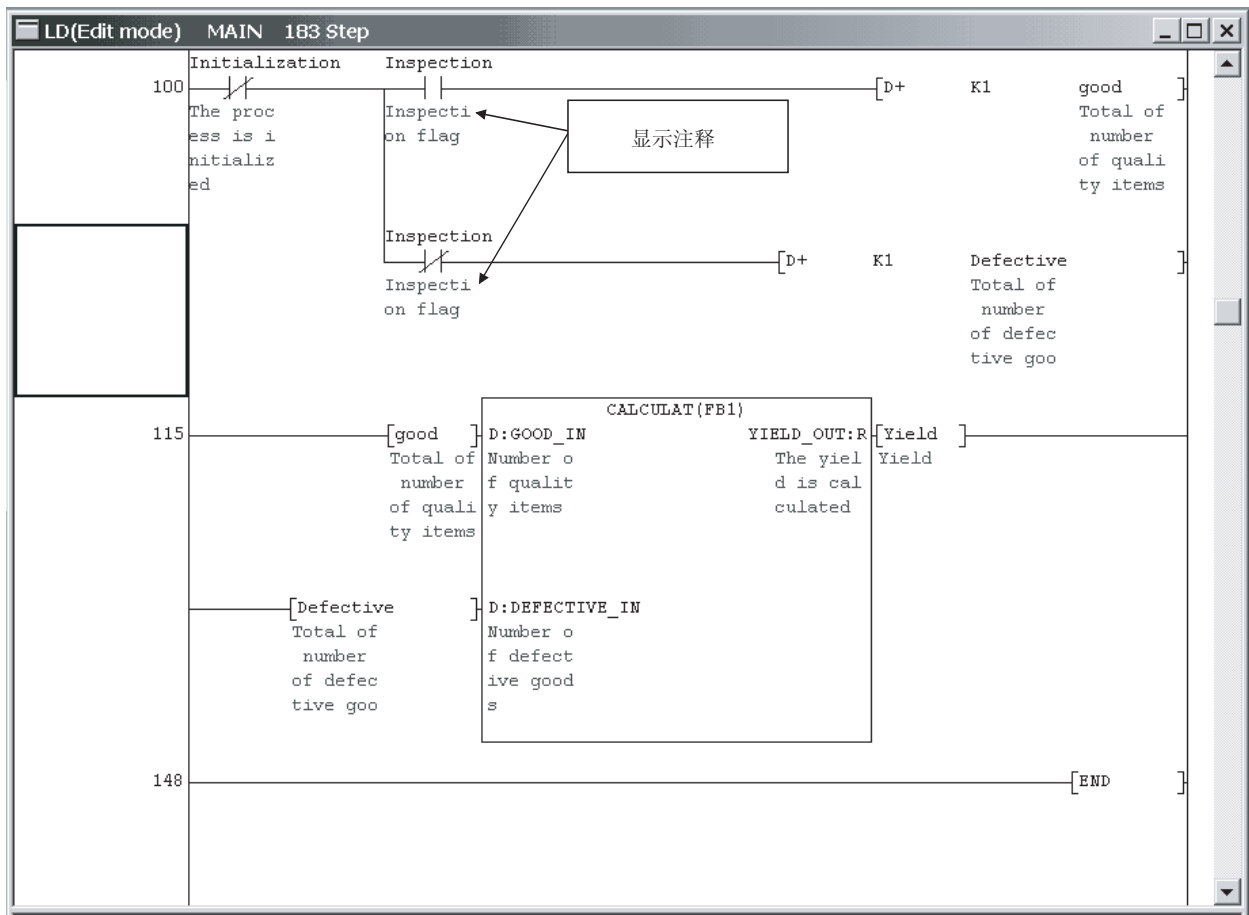
监视顺控程序

监视顺控程序，进行确认。

监视的开始/停止操作如下所示：

- 进行监视时
[Online (在线)] → [Monitor (监视)] → [Monitor mode (监视模式)]
- 停止监视时
[Online (在线)] → [Monitor (监视)] → [Stop monitor (监视停止)]
- 重启监视时
[Online (在线)] → [Monitor (监视)] → [Start monitor (开始监视)]

监视画面



在局部变量(局部标识)设置画面中设置的注释可以通过选择菜单的[View(显示)] → [Comment(注释显示)]显示。

确认程序的动作

更改可编程控制器 CPU 内的位软元件的值，确认程序的动作。

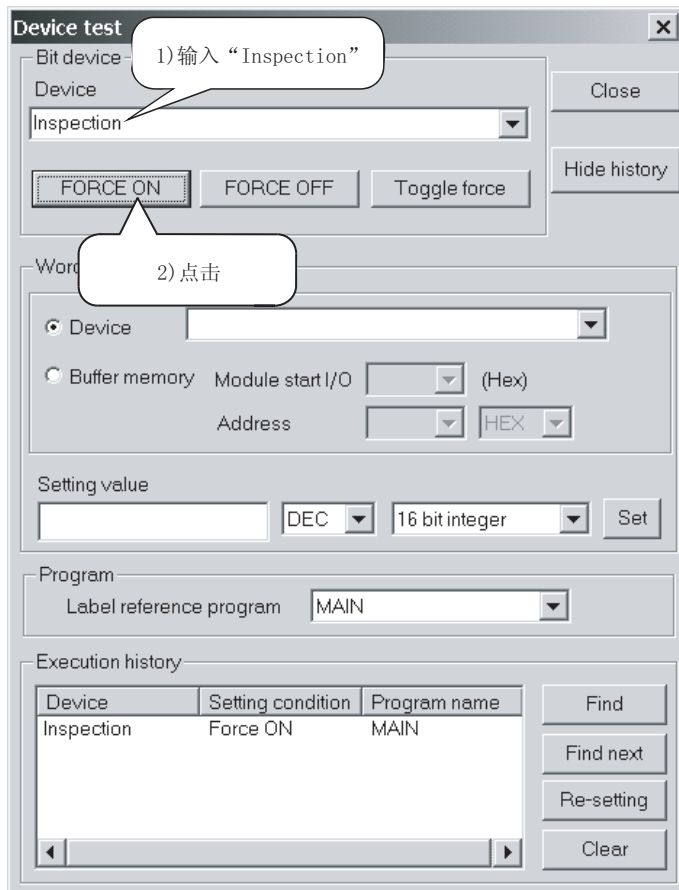
■ 进行软元件测试

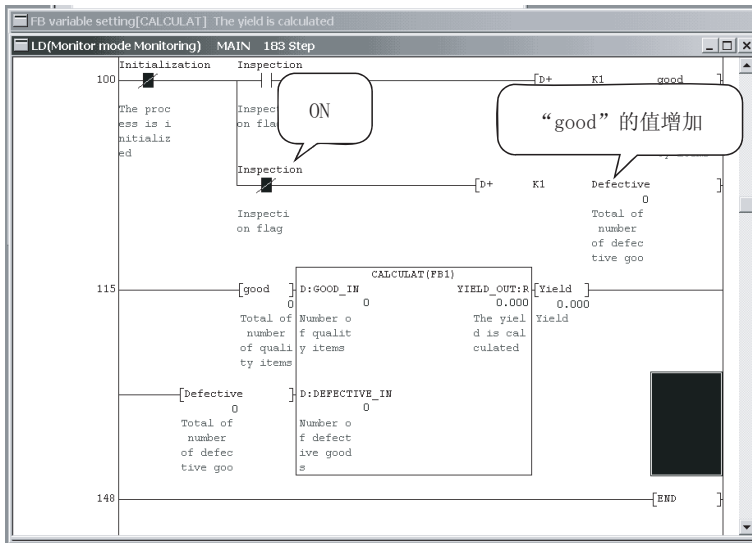
参阅 5.2 节，更改位软元件的值。

通过选择菜单的[Online(在线)]→[Debug(调试)]→[Device test(软元件测试)]显示软元件测试对话框。

1) 在位软元件中输入“Inspection”。

2) 点击 **FORCE ON(强制 ON)** 按钮。





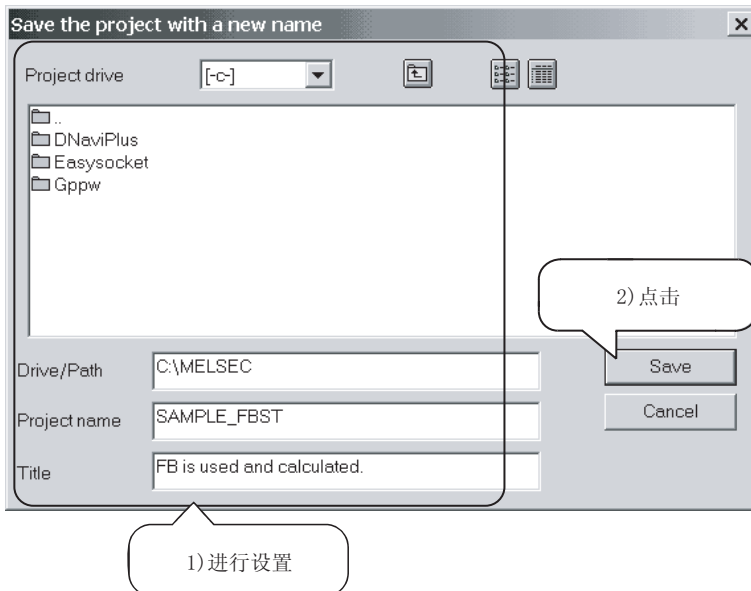
3) 确认程序动作正常。

* 应对其它的标识的值也进行变更，确认程序的动作。

■ 保存工程

参阅第 6 章，将创建的工程附加名称后保存(另存为)。

点击菜单的[Project(工程)]→[Save as(另存为)]，显示工程的另存为对话框。



1) 输入以下内容：

- Drive/Path(驱动器/路径)：
C:\MELSEC
- Project name(工程名)：
SAMPLE_FBST
- Title(标题)：
FB is used and Calculated
(使用 FB 进行计算)

2) 点击 **Save (保存)** 按钮。

所创建的程序被保存为以下样式。

Drive/Path(驱动器/路径) : C:\MELSEC

Project name(工程名) : SAMPLE_FBST

Title (标题) : FB is used and calculated (使用 FB 进行计算)

关于创建程序的一系列操作方法就介绍到这里。
如果想要进一步了解，建议参阅“关联手册”中记载的
各个手册。

索引

[B]

编辑

- 标识信息的显示 7-1
- 窗口拆分 7-1
- 函数选择 7-1
- 书签 7-1
- 显示颜色的更改 7-1
- 字体的更改 7-1

标识

- 定义标识 3-3

[C]

程序

- 函数选择功能 3-11、7-1
- 函数自变量 3-12
- 输入标识 3-10
- 输入函数 3-11
- 输入时的注意事项 3-8
- 输入输入回路部分及输出回路部分 8-15
- 输入梯形图程序 8-14
- 输入注释 3-12
- 输入字符 3-9

[G]

工程

- ST用工程的新建 3-1
- 保存工程 6-1、8-19
- 新建工程 8-1

功能块 (FB)

- 何为功能块 (FB) 8-1

[J]

监视

- 监视顺控程序 5-1、8-17

局部变量

- 设置局部变量 3-5、8-12
- 输入标识名 3-5
- 输入软元件类型 3-5
- 显示设置画面 3-3、8-11
- 在标识中输入注释 3-5

[P]

- PLC 读取 4-2
- PLC 写入 4-1、8-16

[R]

软元件测试

- 程序的動作确认 5-2、8-18

[S]

- ST 语言 1-1
- 设置 FB 变量 8-6
 - 输入标识名 8-6
 - 输入软元件类型 8-6
 - 选择输入输出类型 8-6
 - 在标识中输入注释 8-7

[X]

- 显示 FB 变量设置画面 8-5
- 显示 FB 定义画面 8-9
 - 输入程序 8-10
- 显示 ST 编辑画面 3-9
- 新建 FB 8-2

[Y]

- 运行中写入 5-4

[Z]

- 粘贴 FB 8-14
- 转换 (编译)
 - 编译出错显示 (详细) 3-15
 - 出错位置表示标记 3-16
 - 对 FB 进行转换 (编译) 8-10
 - 何为转换 (编译) 3-14
 - 执行转换 (编译) 3-14

结构化文本 (ST)

编程参考手册

技术服务热线:

800-828-9910

服务时间: **9:00~12:00**
13:00~17:00 (节假日除外)

三菱电机自动化(上海)有限公司

地址: 上海市黄浦区新昌路80号智富广场4楼

邮编: 200003

电话: 021-61200808 传真: 021-61212444

网址: www.mitsubishielectric-automation.cn

书号	SH(NA)-080665CHN-A(0703)STC
印号	STC-ST-P-RM(0703)

内容如有更改
恕不另行通知