

[illegible]

Introduction

- Please be sure to read and understand Precautions and Introductions in *CX-Programmer Operation Manual SFC* (W469-E1), *CX-Programmer Operation Manual* (W446-E1) and *CX-Programmer Operation Manual Function Block/Structured Text* (W447-E1) before using the product.
- This guide describes the basic operation procedure of CX-Programmer. Refer to the Help or the Operation Manual of the PDF file for detailed descriptions.
- To read the PDF files, you need Adobe Reader, a free application distributed by Adobe Systems.
- You can display the PDF files from the [Start] menu on your desktop after installing the CX-Programmer.
- The screen views used in this guide may be different from the actual view, and be subject to change without notice.
- The product names, service names, function names, and logos are the trademarks or registered trademarks of the respective company.
- "TM" and the (R) mark do not appear with product or company names in this guide.
- The product names of the other companies may be abbreviated in this guide.
- Microsoft product screen shot(s) reprinted with permission from Microsoft Corporation.

Table of Contents

Section 1 SFC Overview	1-1
1-1 Introduction of SFC	1-1
1-1-1 Describing Programs with SFC	1-1
1-1-2 Correspondence of Steps and Programs	1-1
1-2 What Can CX-Programmer Do?	1-2
1-2-1 Choosing a Programming Language for Your Application	1-2
1-3 SFC Operation	1-4
1-3-1 Elements of SFC	1-4
1-3-2 SFC Program Operation	1-5
 Section 2 Creating Programs with SFC	 2-1
2-1 Workflow	2-1
2-2 Designing Programs	2-2
2-2-1 Parking Gate System Operation	2-2
2-2-2 Action Flow Diagram	2-5
2-2-3 Programming Example	2-6
2-3 Creating Projects	2-9
2-4 Creating SFC Programs	2-11
2-5 Creating Steps and Transitions	2-13
2-5-1 Renaming Steps/Transitions	2-13
2-5-2 Adding Steps/Transitions	2-15
2-5-3 Creating Transitions	2-18
2-5-4 Creating Loops to Initial Steps	2-22
2-6 Compiling	2-24
2-6-1 Checking for Program Errors	2-24
2-6-2 Saving Projects	2-24
2-7 Simulated Transition Test	2-25
2-7-1 Online Connection to Simulator	2-25
2-7-2 Setting Transition Conditions	2-26
2-8 Creating Action Blocks	2-28
2-8-1 Creating Actions	2-28
2-8-2 Assigning Action Qualifiers	2-30
2-9 Simulated Operation Test	2-32

Appendix A-1

 A-1 Reference A-1

 A-1-1 Main WindowA-1

 A-1-2 Action Qualifier (AQ) ListA-3

 A-2 Creating SFC Elements A-4

 A-2-1 Simultaneous Sequence Divergences/ConvergencesA-4

 A-2-2 Divergences/ConvergencesA-6

 A-3 Online Editing A-9

 A-3-1 Transferring Programs to PLCA-9

 A-3-2 Online EditingA-11

Section 1

SFC Overview

This section explains the features and the operations of SFC.

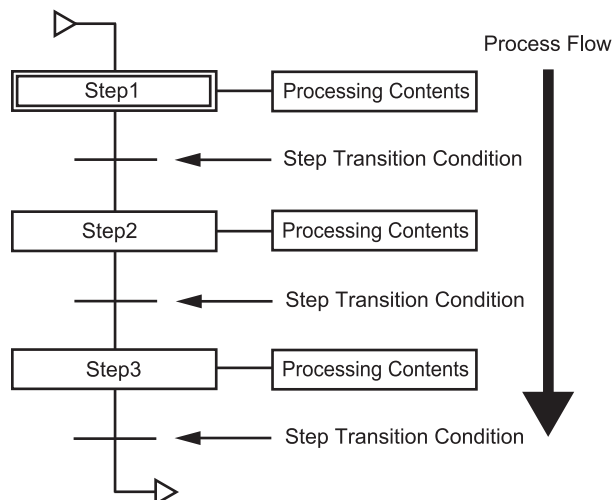
1-1 Introduction of SFC

This section provides an overview of SFC.

1-1-1 Describing Programs with SFC

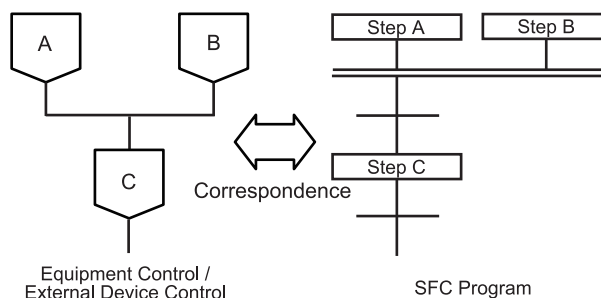
Sequential Function Chart (hereafter called SFC) is a graphical programming language that displays the process flow as a diagram, thereby allowing the user to control the sequential processes by describing the transition conditions and actions for each step.

SFC is suitable for understanding the processing order and status transition of a program.



1-1-2 Correspondence of Steps and Programs

Program controls correspond to the actual step flow, making it easy to understand the program as a whole. This reduces the workload in debugging and maintaining the programs.



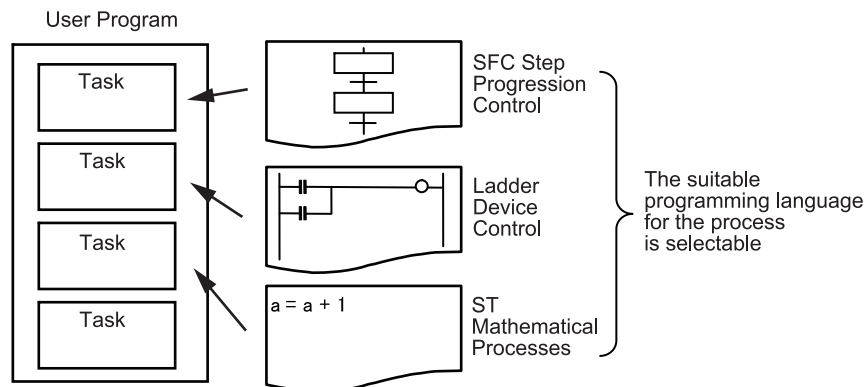
1-2 What Can CX-Programmer Do?

This section explains the features and the usage of SFC.

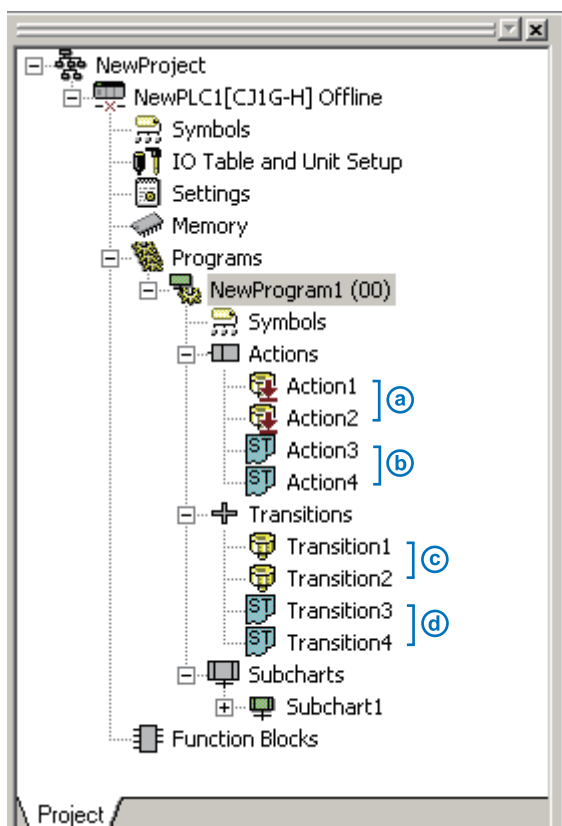
1-2-1 Choosing a Programming Language for Your Application

Each program can be coded in any combination of SFC, ladder, and ST (Structured Text). From the three, select the suitable programming language for the process.

For example, you can code step progression with SFC, device control with ladder, and arithmetic processes with ST.

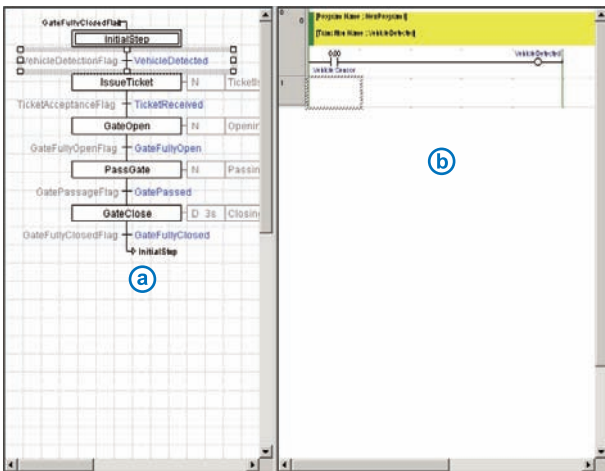


Flow and order of controls can be coded with SFC, while actions within steps and transitions between steps can be coded with ladder, ST or Boolean variables (contacts).



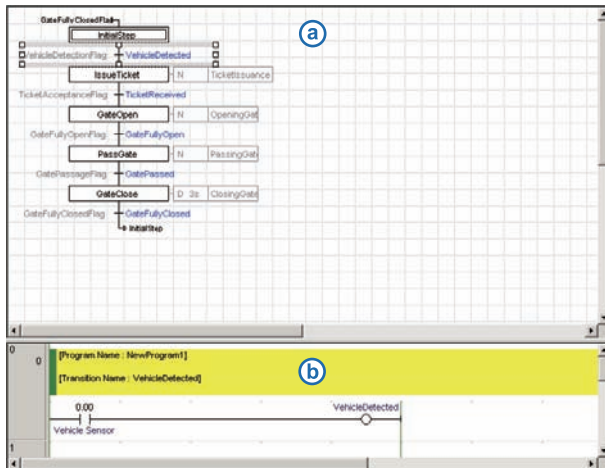
- (a) : Action Program (Ladder)
- (b) : Action Program (ST)
- (c) : Transition Program (Ladder)
- (d) : Transition Program (ST)

The SFC editor allows simultaneous viewing and editing of both the SFC chart and action/transition programs. This allows for process and transition conditions to be viewed and edited while maintaining a perspective on the program as a whole, allowing the user to fully utilize the features of SFC.



(a) : SFC View

(b) : Program View



Info

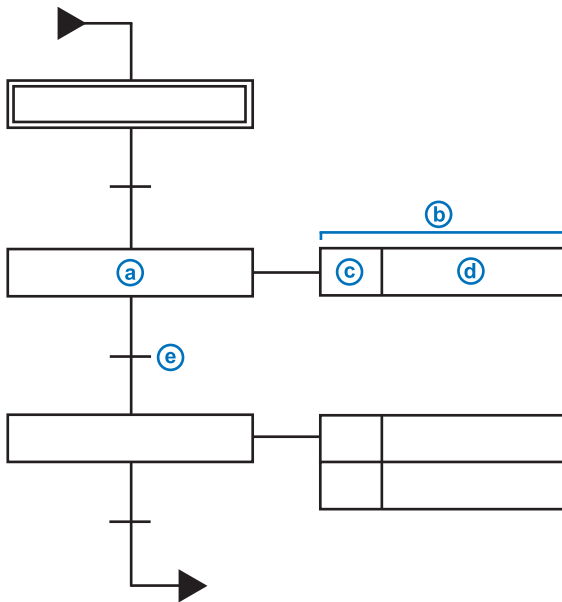
The SFC editor can display the SFC view and the Program view in horizontal or vertical alignment.

To display the views in vertical alignment, check **SFC Horizontal Split** on the **SFC** tab of the **Options** dialog box.

1-3 SFC Operation

This section explains the elements of SFC and the operation of SFC programs.

1-3-1 Elements of SFC



(a) : Step

A step is an element of SFC programs, and represents a single process within the overall process flow. When a step becomes active, action blocks assigned to that step are executed.

The first step of a program is referred to as the "initial step".

(b) : Action Block

An action block contains the step processes (actions) for a single step.

(c) : Action Qualifier

An action qualifier defines the execution timing and status retention preference for each action.

(d) : Action Name

For each action, specify a Boolean variable (contact) or an action program name.

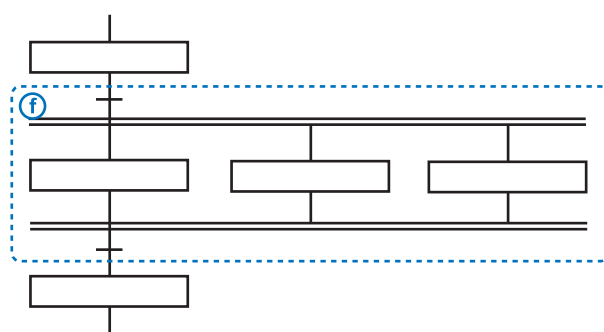
(e) : Transition

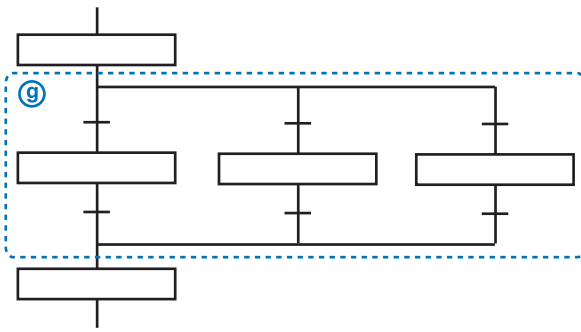
A transition represents the condition that transfers the active status from the step before the transition to the step after the transition.

(f) : Simultaneous Sequence Divergence/Convergence

A simultaneous sequence divergence is a structure in which a single transition is followed by multiple steps. When the transition condition is satisfied, all of the connected steps become active simultaneously.

A simultaneous sequence convergence is a structure in which multiple steps are followed by a single transition. Active status is transferred when the transition condition is satisfied and after all the steps have been activated.





g : Divergence/Convergence

A divergence is a structure in which a single step is followed by multiple transitions. The active status is transferred to the transition for which the condition is satisfied.

A convergence is a structure in which multiple transitions are each followed by a step. This structure merges a flow that has been branched.

Info

For details on SFC elements, refer to *CX-Programmer Operation Manual SFC*.

1

SFC Overview

1-3-2 SFC Program Operation

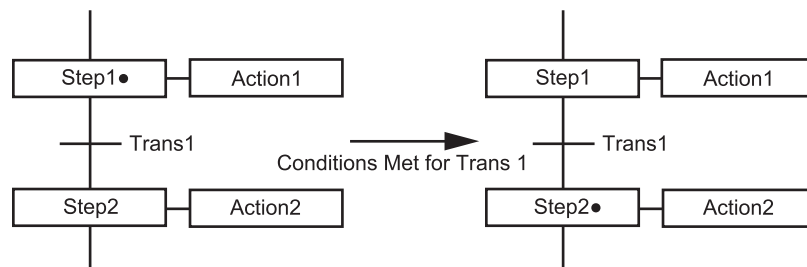
A step can be either active or inactive.

When a step becomes active, actions defined within that step are executed.

When the transition condition is satisfied, the step immediately before the transition becomes inactive, and the step immediately after the transition becomes active.

The following example illustrates a case in which transition condition "Trans1" is satisfied, and the active status is transferred from step 1 to step 2.

The bullet symbol indicates the active step.



Memo

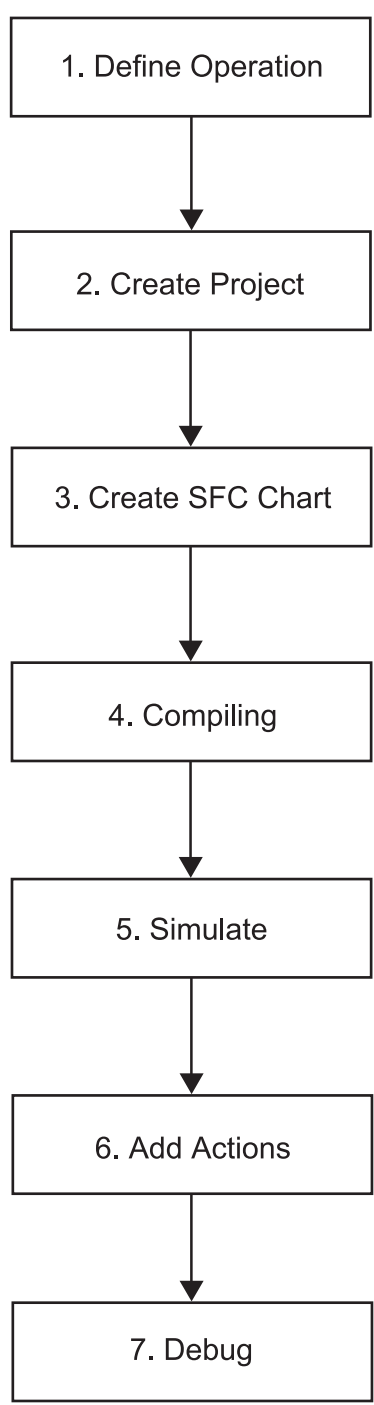
Section 2

Creating Programs with SFC

This section explains the process of coding an SFC program with CX-Programmer, using a parking gate system as an example.

2-1 Workflow

This section explains the SFC programming workflow, from creation to simulated debugging. When creating a program for actual use, test the program with the actual devices after completing the steps below.



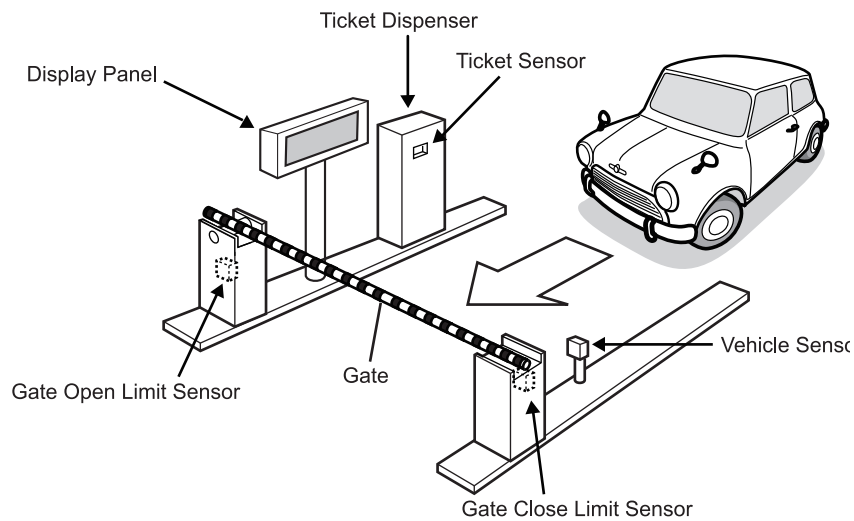
- 1. Define Operation
Define the operation of the system to perform.
Refer to *2-2 Designing Programs*.
- 2. Create Project
Start CX-Programmer, and begin programming.
Refer to *2-3 Creating Projects*.
- 3. Create SFC Chart
Organize the overall process flow into actions and transitions, and create an SFC chart. Assign the created program to a task. Create an SFC chart, and program the transition conditions.
Refer to *2-4 Creating SFC Programs*.
- 4. Compiling
Check for errors in the program.
Refer to *2-6 Compiling*.
- 5. Simulate
Using the simulator, check that the transitions between steps actually take place when the transition conditions are satisfied.
Refer to *2-7 Simulated Transition Test*.
- 6. Add Actions
Program the actions for each step.
Refer to *2-8 Creating Action Blocks*.
- 7. Debug
Test the operation of the program as a whole.
Refer to *2-9 Simulated Operation Test*.
For details on debugging, refer to *CX-Programmer Operation Manual*.

2-2 Designing Programs

Design the program based on system operation.

2-2-1 Parking Gate System Operation

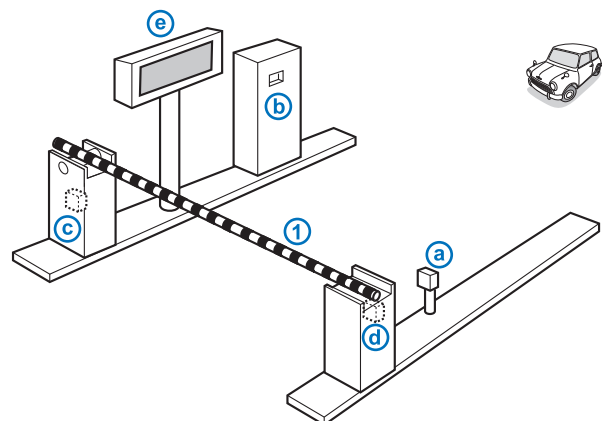
When a car approaches the gate, a ticket is issued. The gate is opened when the ticket is taken. When the car passes the gate, the gate is closed, and the system returns to the initial state. A display panel is used to display status messages. Transitions between actions are initiated by inputs from the sensors.



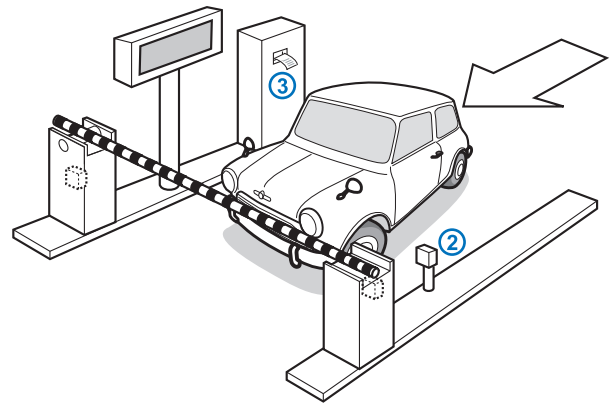
The program design process is outlined below.

- Define the operation of the parking gate system.
- Clarify the conditions by which transitions of action take place.
- Create an SFC chart based on action flow.
- Create the actual I/O device control programs for each action.

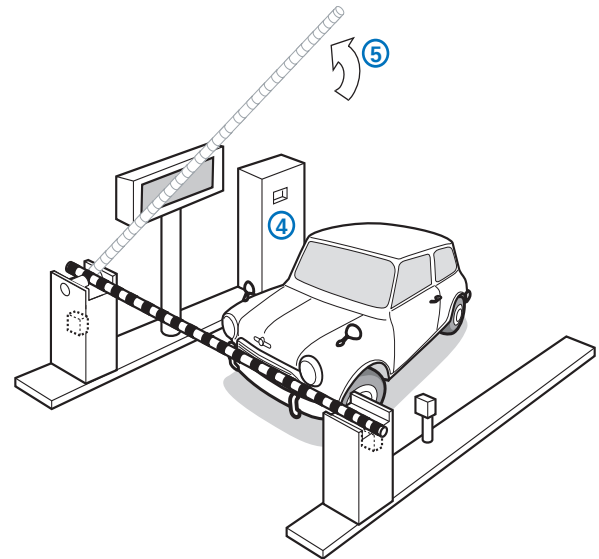
- ① When the system is waiting for a car (wait status), the gate is closed.
 - (a) : Vehicle sensor is OFF.
 - (b) : Ticket sensor is OFF.
 - (c) : Gate open limit sensor is OFF.
 - (d) : Gate close limit sensor is ON.
 - (e) : Nothing is displayed on the display panel.



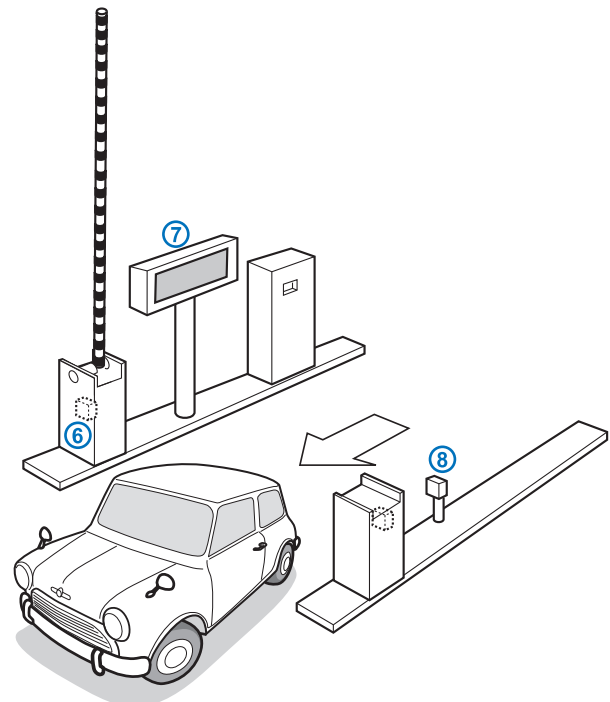
- ② When a car enters, the vehicle sensor turns ON.
- ③ When the vehicle sensor turns ON, a ticket is issued.
"Take ticket" is displayed on the display panel.



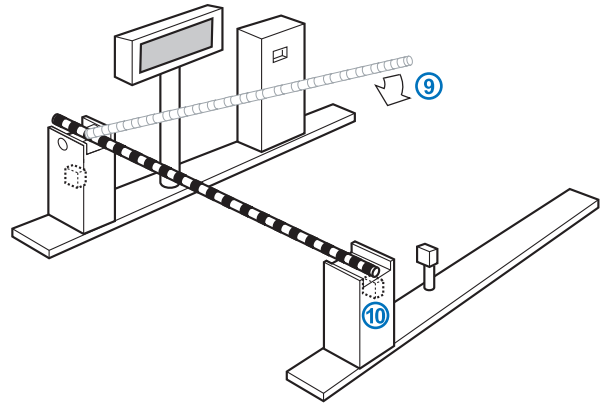
- ④ When the ticket is taken, the ticket sensor turns ON.
- ⑤ When the ticket sensor turns ON, the gate opens.
"Opening gate" is displayed on the display panel.
Gate close limit sensor turns OFF.



- ⑥ When the gate is fully open, the gate open limit sensor turns ON.
When the gate open limit sensor turns ON, the gate stops.
- ⑦ "Go" is displayed on the display panel.
- ⑧ When the car passes the gate, the vehicle sensor turns OFF.



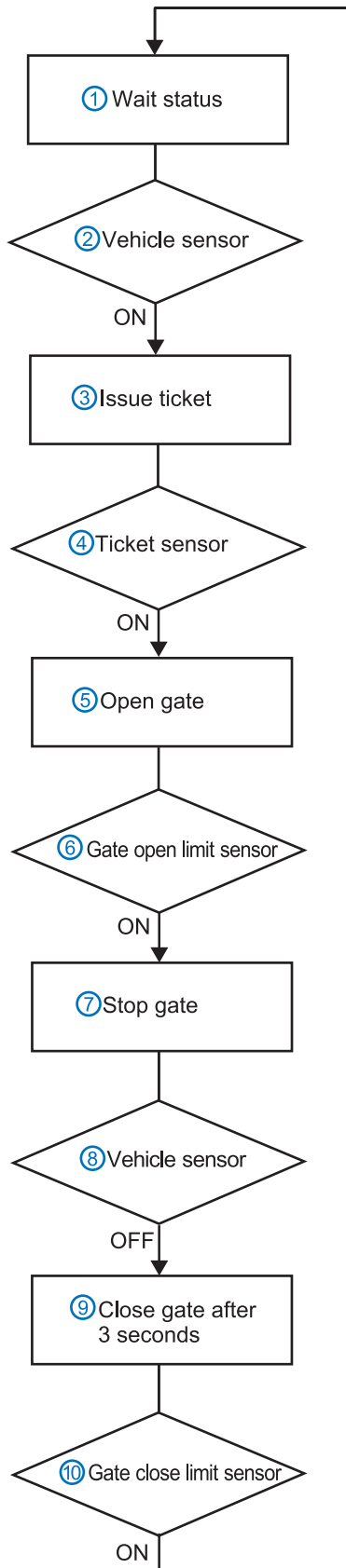
- ⑨ When the vehicle sensor turns OFF, the timer counts 3 seconds for safety, and then the gate closes.
"Closing gate" is displayed on the display panel.
The gate open limit sensor turns OFF.
- ⑩ When the gate is fully closed, the gate close limit sensor turns ON.
When the gate close limit sensor turns ON, the gate stops.
The system returns to the wait status.



2-2-2 Action Flow Diagram

The action flow diagram for the parking gate system is shown below.

The numbering in the diagram corresponds to the numbering in 2-2-1 *Parking Gate System Operation*.



- ① This is the initial state of the parking gate system (wait status).
Maintain wait status until the vehicle sensor turns ON.
- ② When a car enters, the vehicle sensor turns ON.
When the vehicle sensor turns ON, make a transition from ① to ③.
- ③ Activate the ticket dispenser and issue a ticket.
Display "Take ticket" on the display panel.
Maintain this state until the ticket sensor turns ON.
- ④ When the driver takes the ticket, the ticket sensor turns ON.
When the ticket sensor turns ON, make a transition from ③ to ⑤.
- ⑤ Rotate the gate operating motor in the normal direction to open the gate.
Display "Opening gate" on the display panel.
Maintain this state until the gate open limit sensor turns ON.
- ⑥ When the gate is fully open, the gate open limit sensor turns ON.
When the gate open limit sensor turns ON, make a transition from ⑤ to ⑦.
- ⑦ The gate is stopped at the open limit.
Display "Go" on the display panel. Maintain this state until the vehicle sensor turns OFF.
- ⑧ When the car passes the gate, the vehicle sensor turns OFF.
When the vehicle sensor turns OFF, make a transition from ⑦ to ⑨.

- ⑨ Wait 3 seconds after the transition, then rotate the gate operating motor in the reverse direction to close the gate.
Display "Closing gate" on the display panel. Maintain this state until the gate close limit sensor turns ON.
- ⑩ When the gate is fully closed, the gate close limit sensor turns ON.
When the gate close limit sensor turns ON, make a transition from ⑨ to ①.

2-2-3 Programming Example

An example program for the described parking gate system is shown below.
The numbering in the diagram corresponds to the numbering in 2-2-2 *Action Flow Diagram*.

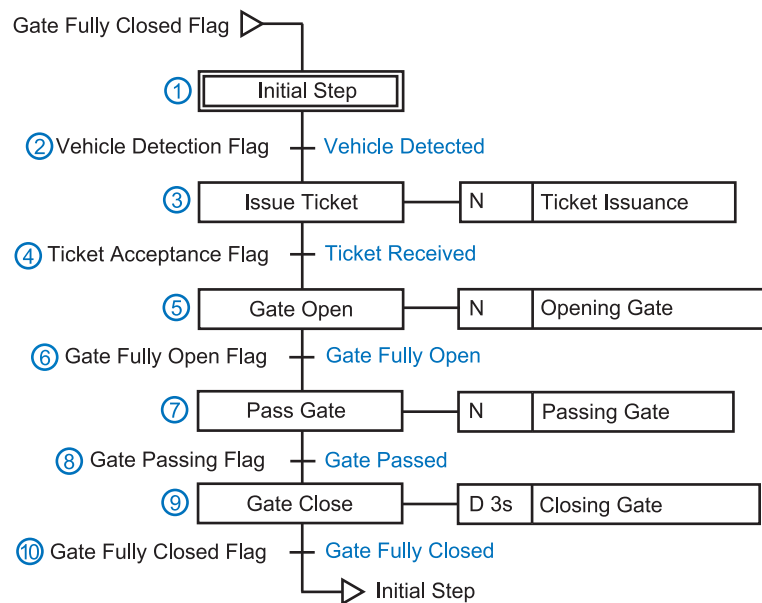
Note

The programs used in this manual are provided for educational purposes only, to aid in understanding how CX-Programmer works. When designing a program for actual use, be sure to make considerations for safety in terms of hardware devices and control methods.

■ SFC Chart

Create an SFC chart based on the system operation.

Use a Step to code steps, an Action Program for processes within a step, an Action Qualifier for execution timing, and a Transition for transition conditions.



■Memory Allocation

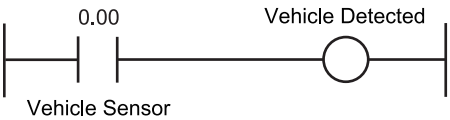
Set memory allocation for the external devices as follows.

External Device	I/O	Address
Vehicle Sensor	Input	0.00
Ticket Sensor	Input	0.01
Gate Open Limit Sensor	Input	0.02
Gate Close Limit Sensor	Input	0.03
Issue Ticket	Output	10.00
Normal Gate Operating Motor Rotation	Output	11.00
Reversed Gate Operating Motor Rotation	Output	11.01
Ticket Issuance Message	Output	20.00
Gate Opening Message	Output	20.01
Pass Gate Message	Output	20.02
Gate Closing Message	Output	20.03

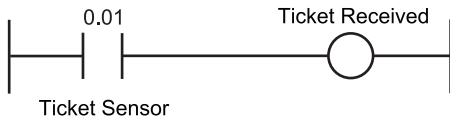
■Transitions

Set the transition conditions between steps. Example ladder programs are shown below.
For a ladder output coil, use the Boolean variable output, having the same name as the transition program.

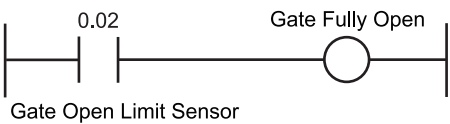
"Vehicle Detected" Program



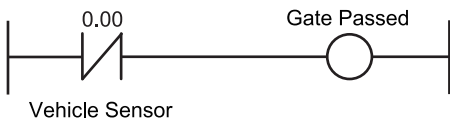
"Ticket Received" Program



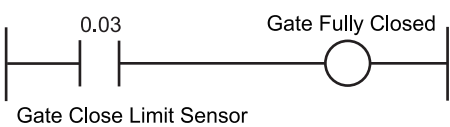
"Gate Fully Open" Program



"Gate Passed" Program



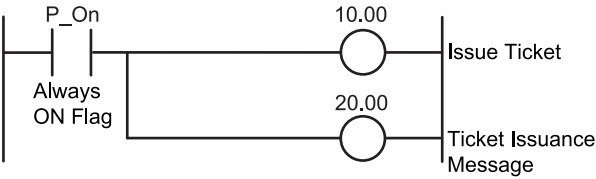
"Gate Fully Closed" Program



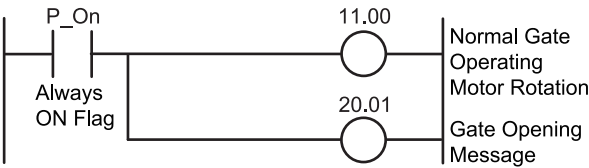
■Actions

Describe the processes for each step. Example ladder programs are shown below.

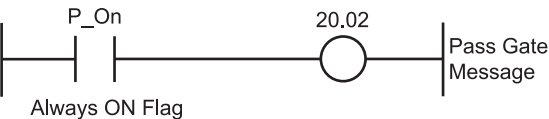
"Ticket Issuance" Program



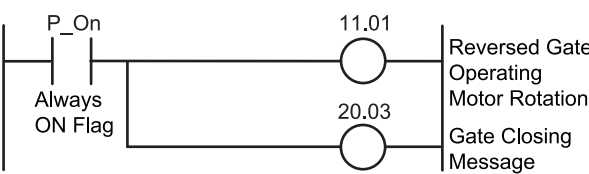
"Opening Gate" Program



"Passing Gate" Program



"Closing Gate" Program



2-3 Creating Projects

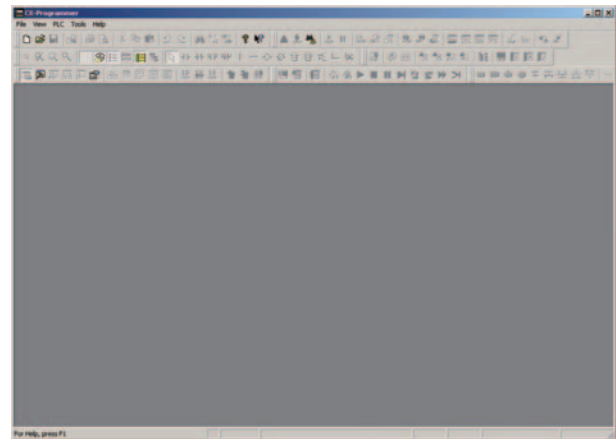
Start CX-Programmer, and select the device type.

1 Start CX-Programmer.

- ① On the desktop, select **Start - All Programs - OMRON - CX-One - CX-Programmer - CX-Programmer**.

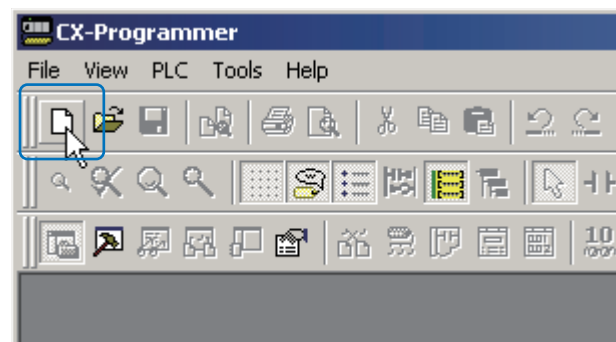
CX-Programmer will start, and the main window will be displayed.

For details of the main window, refer to A-1-1 *Main Window*.



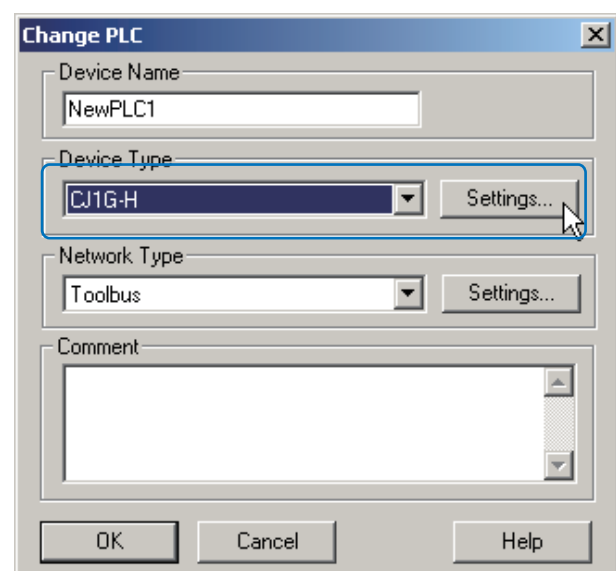
2 Create a new project.

- ① Click **New**.
The **Change PLC** dialog box will be displayed.

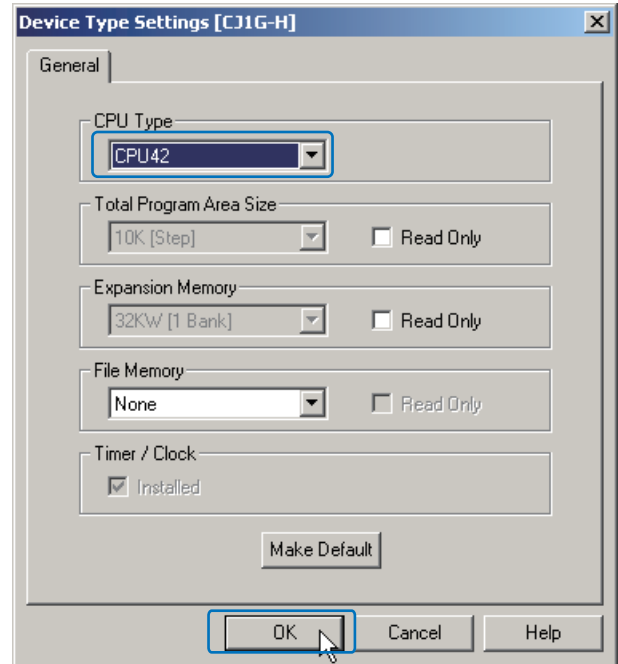


3 Select the device type and the network type.

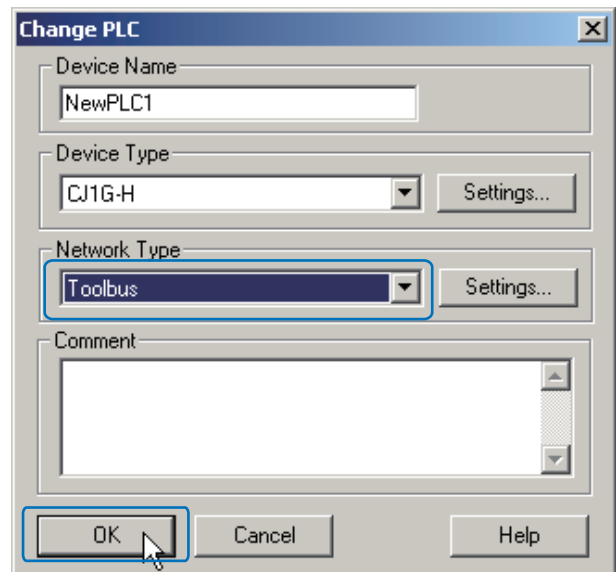
- ① Select the device type from the **Device Type** drop-down list.
- ② Click **Settings**.
The **Device Type Settings** dialog box will be displayed.
Settings for CJ1G-CPU42-H will be used here as an example.



- ③ Select the CPU from the **CPU Type** drop-down list.
- ④ Click **OK**.
The **Device Type Settings** dialog box will be closed.



- ⑤ Select the network type from the **Network Type** drop-down list.
For example, select **Toolbus**.
- ⑥ Click **OK**.
The **Change PLC** dialog box will be closed.



2-4 Creating SFC Programs

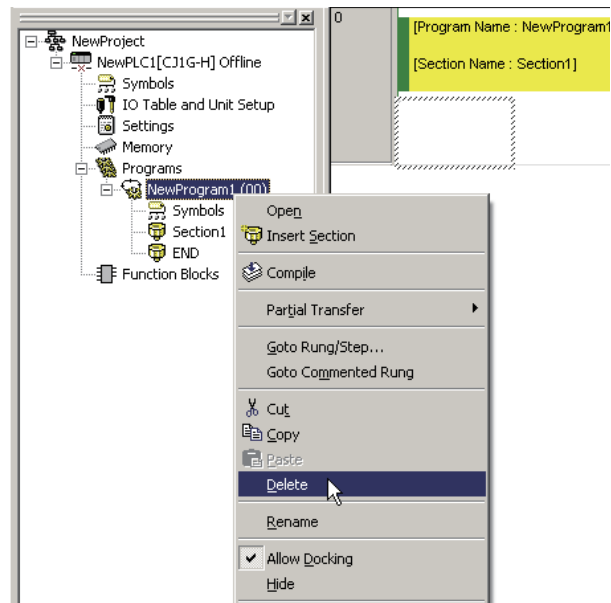
Add an SFC program.

When a new project is created, it will automatically be configured for ladder programming.

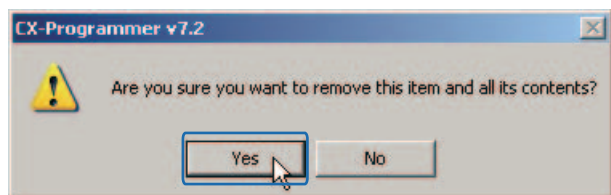
1 Delete the ladder program.

Ladder programs are not used in this example. Delete the existing ladder program and add an SFC program.

- ① On the project workspace, right-click **New Program 1 (00)**, and select **Delete** from the pop-up menu.
A dialog box will be displayed to confirm deletion.

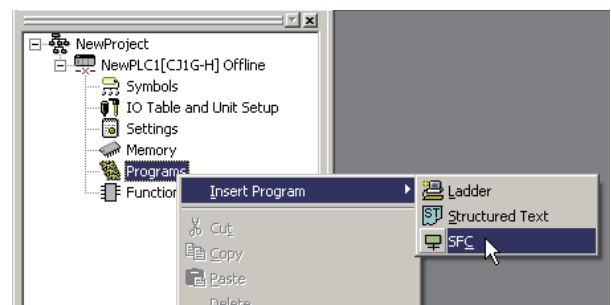


- ② Click **Yes**.
The default ladder program will be deleted, and the ladder editor will be closed.



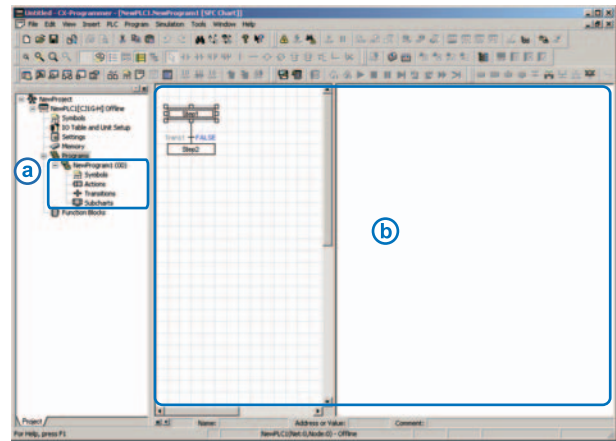
2 Add an SFC program.

- ① On the project workspace, right-click **Programs**, and select **Insert Program - SFC** from the pop-up menu.



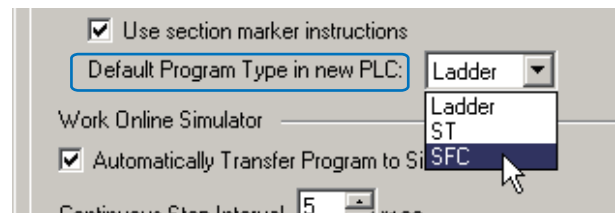
2-4 Creating SFC Programs

A new program **(a)** will be registered. The SFC editor **(b)** will be displayed.



Info

The default programming language for new projects can be changed to SFC. Select **Tools - Options**. The **Options** dialog box will be displayed. From the **PLC** tab, select **SFC** for **Default Program Type in new PLC**.



2

2-5

Creating Steps and Transitions

Create an SFC chart and set the transition conditions.

2-5-1

Renaming Steps/Transitions

When an SFC program is first created, two steps and a transition are displayed by default. Rename these SFC elements, so that the program content can be understood more easily.

1

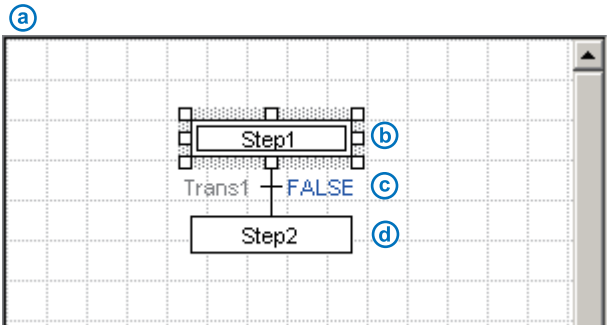
Rename the initial step.

By default, an initial step **(b)** , a transition **(c)** , and a step **(d)** will be displayed in the SFC view **(a)** .

Steps define a process, and transitions define the condition(s) for moving to the next process.

The step with the double-line frame is the initial step. Execution of an SFC chart begins at the initial step.

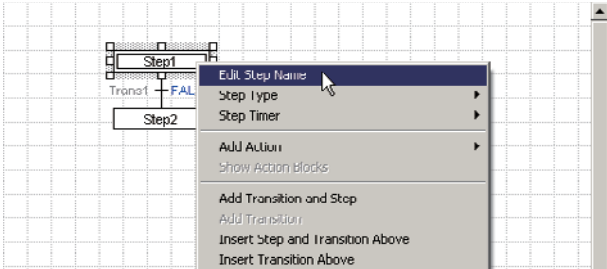
In this example, the step represents the parking gate system in the wait status.



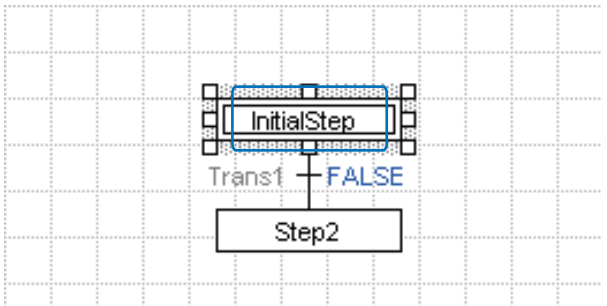
- ① Right-click **Step1**, and select **Edit Step Name** from the pop-up menu.

Info

Steps can be renamed without affecting operation.



- ② Type "Initial Step", and press the **Enter** key. The step will be renamed.



2 Rename the transition.

Here, the transition represents the condition for making a transition from the **Initial Step** process to the following **Issue Ticket** process.

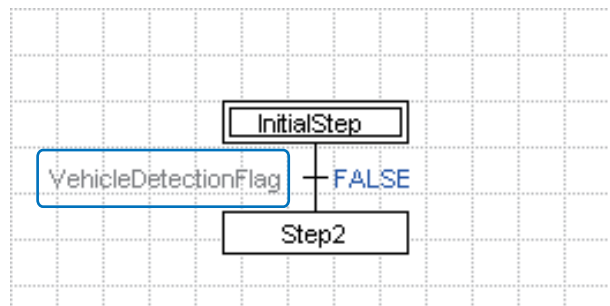
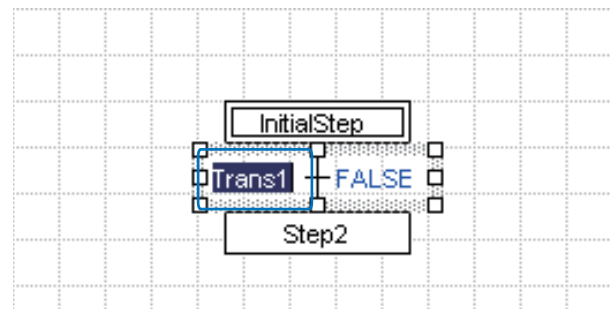
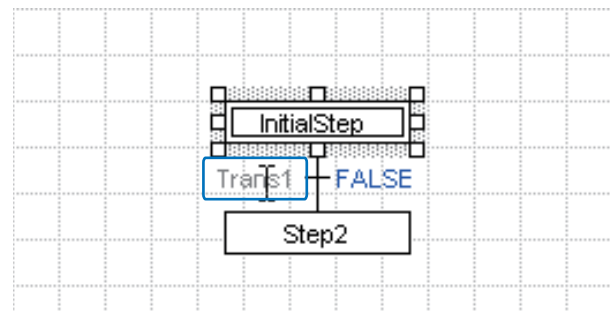
- 1 Double-click **Trans1**.

Trans1 will be selected.

Info

- If the transition is not selected, click on a blank area of the SFC view, and retry.
- Transitions can be renamed without affecting operation.
- Transition names can be hidden. Select **Tools - Options**. The **Options** dialog box will be displayed. Uncheck **Show Transition Name** on the **SFC** tab.

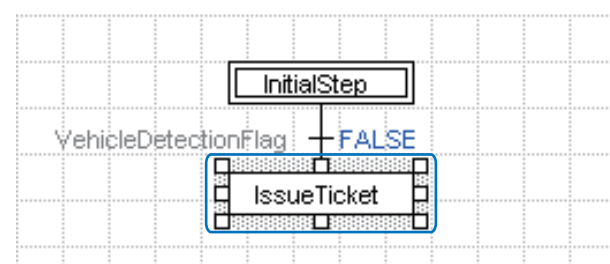
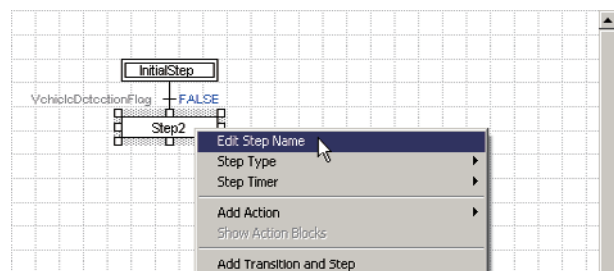
- 2 Type "Vehicle Detection Flag", and press the **Enter** key.
The transition will be renamed.



3 Rename the step.

In this example, the step represents the ticket issuing process.

- 1 Right-click **Step2**, and select **Edit Step Name** from the pop-up menu.
- 2 Type "Issue Ticket", and press the **Enter** key.
The step will be renamed.



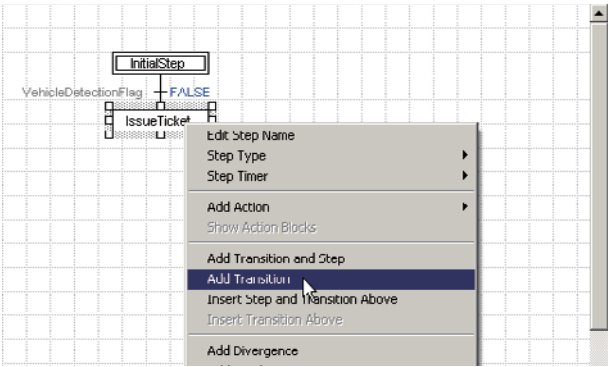
2-5-2 Adding Steps/Transitions

Create the subsequent processes.
Add steps and transitions to the SFC chart.

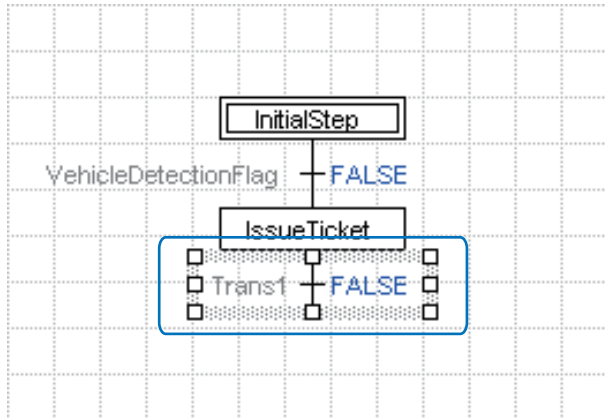
1 Add a transition.

- ① Right-click the **Issue Ticket** step, and select **Add Transition** from the pop-up menu.

Info You can also add a transition by pressing the **T** key while the step is selected.



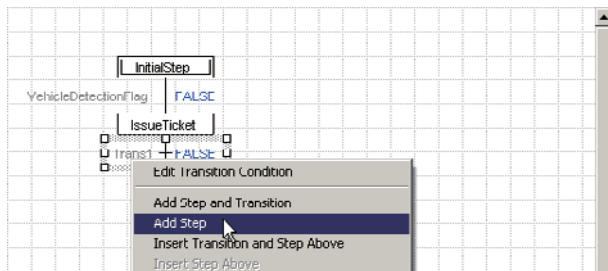
A transition will be added.



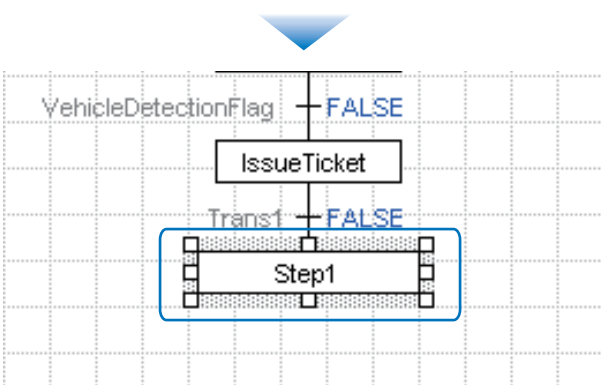
2 Add a step.

- ① Right-click the **Trans1** transition, and select **Add Step** from the pop-up menu.

Info You can also add a step by pressing the **S** key while the transition is selected.



A step will be added.

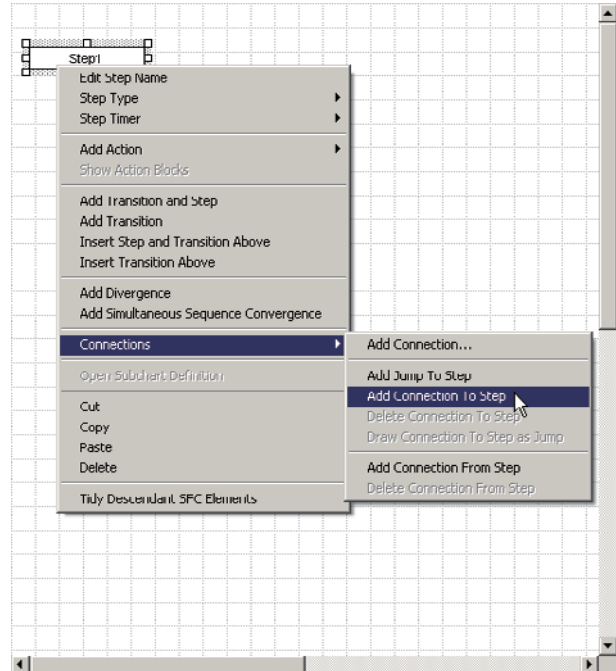


■ Creating Steps and Transitions

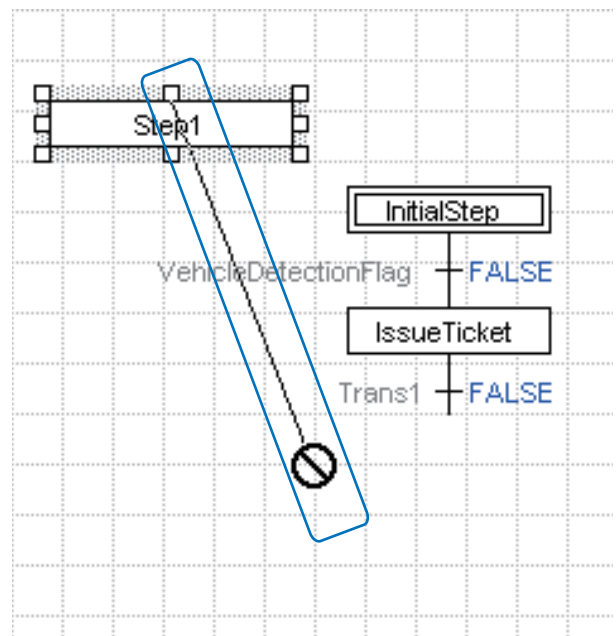
Steps and transitions can be created independently, and then later connected, using connectors. The procedure for connecting a step to a transition is detailed below.

Connect a step and a transition.

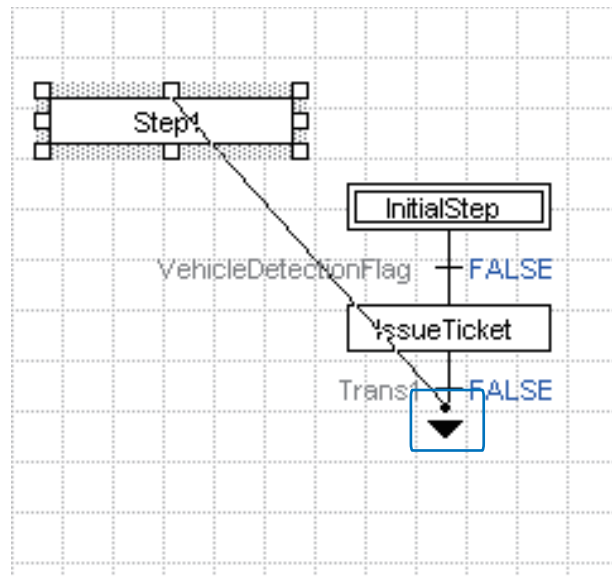
- ① Right-click the step, and select **Connections** - **Add Connection To Step** from the pop-up menu.



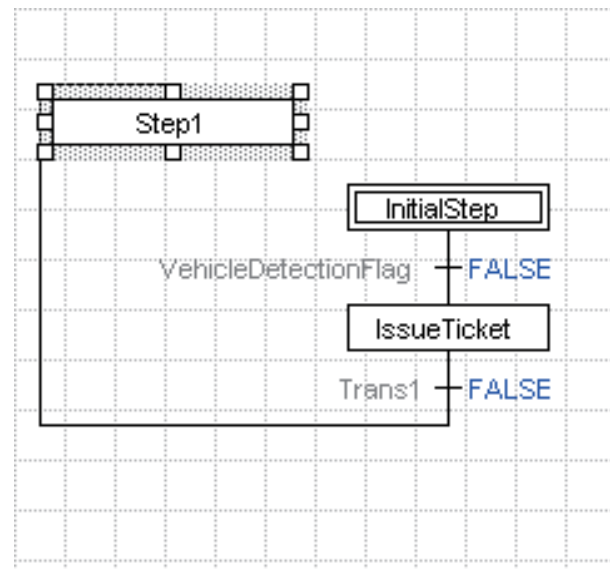
A connector extending from the step to the cursor will be displayed.



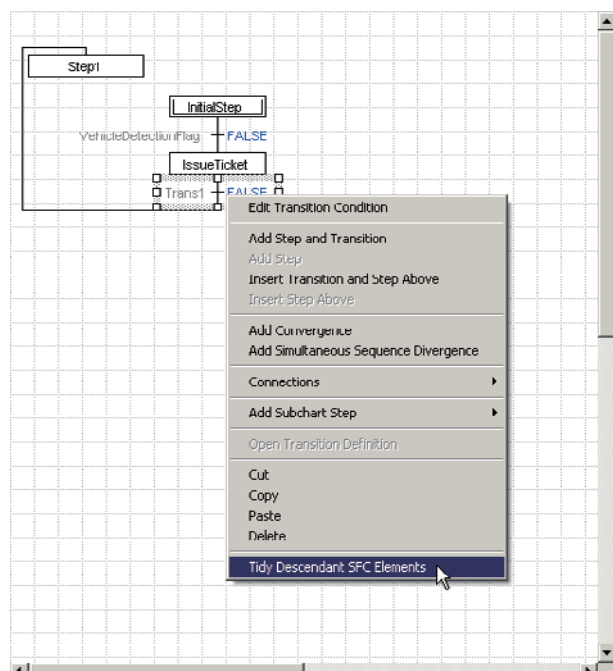
- ② Move the cursor to the target transition.
The cursor will become an arrow.



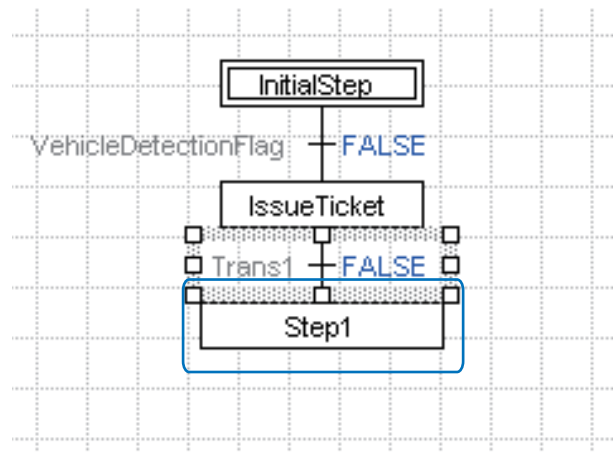
- ③ Left-click on the mouse.
The step and the transition will be connected.



- ④ Right-click the transition, and select **Tidy Descendant SFC Elements** from the pop-up menu.



The connected step will be aligned below the transition.



2-5-3 Creating Transitions

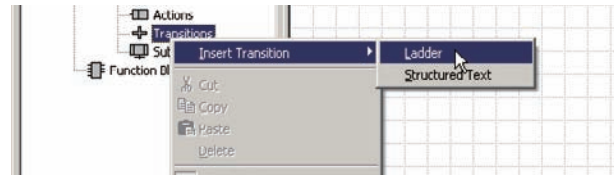
Create transition conditions as ladder programs.

1 Register a transition program.

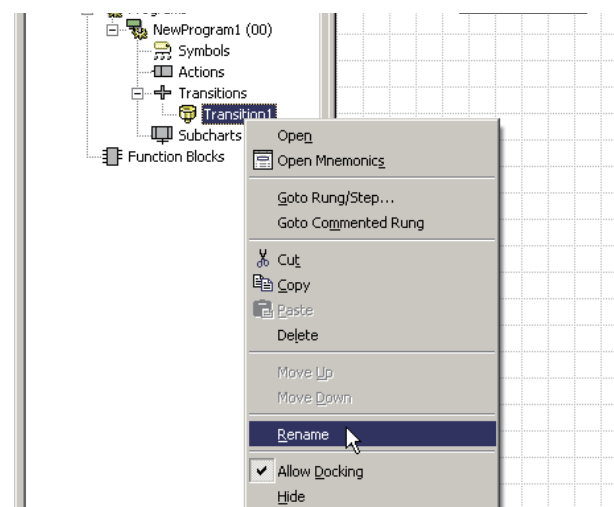
For the **Vehicle Detection Flag** transition in 2-5-1 *Renaming Steps/Transitions*, define the transition condition as a ladder program.

- ① Right-click the **Transitions** folder, and select **Insert Transition - Ladder** from the pop-up menu.

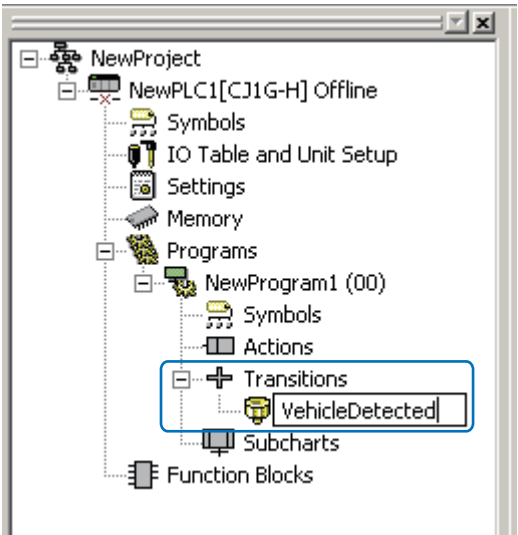
Transition1 transition program will be added to the **Transitions** folder.



- ② Right-click **Transition1**, and select **Rename** from the pop-up menu.



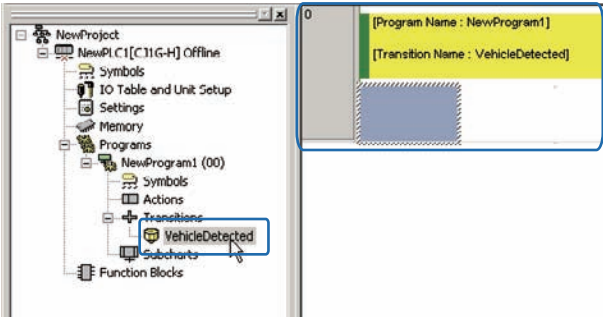
- ③ Type "Vehicle Detected".
The transition program will be renamed to **Vehicle Detected**.



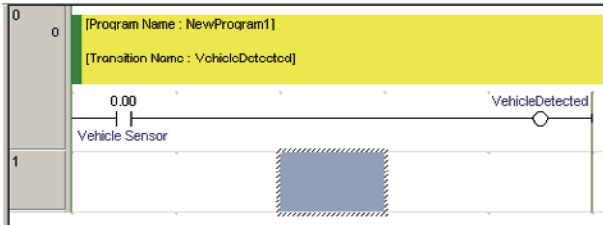
2 Create the program.

When the transition condition is satisfied, the value of the transition will change to TRUE. To determine whether a vehicle has entered, the input contact of the **Vehicle Sensor** will be used. Create a program so that the value of the transition becomes TRUE when the **Vehicle Sensor** turns ON.

- ① Double-click the **Vehicle Detected** program.
The ladder editor will be displayed.



- ② Create the ladder program.
Enter input contact **0.00** and the comment **Vehicle Sensor** for the **Vehicle Sensor**, and output coil **Vehicle Detected**. For the output coil, select the transition program name **Vehicle Detected** as a Boolean variable.



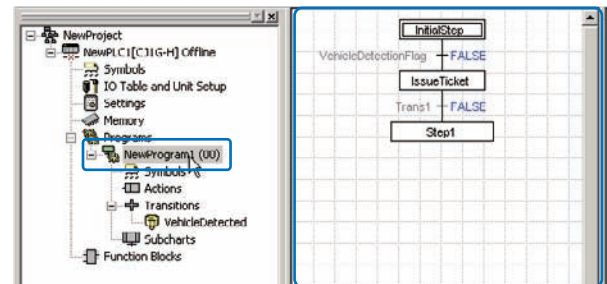
Info

- For details on creating ladder programs, refer to *CX-Programmer Operation Manual*.
- Note that if a timer counter bit output is used for a program, the output will be maintained even after the transition to the next step takes place.

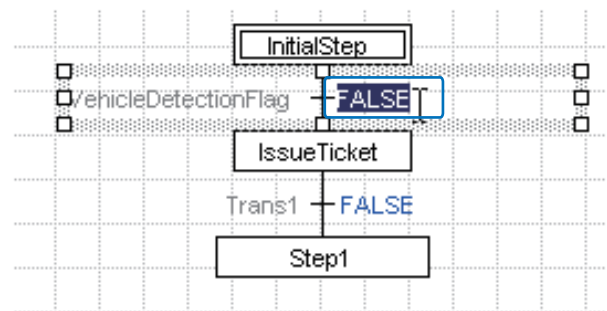
3 Assign the transition program.

Assign the **Vehicle Detected** program to the **Vehicle Detection Flag** transition.

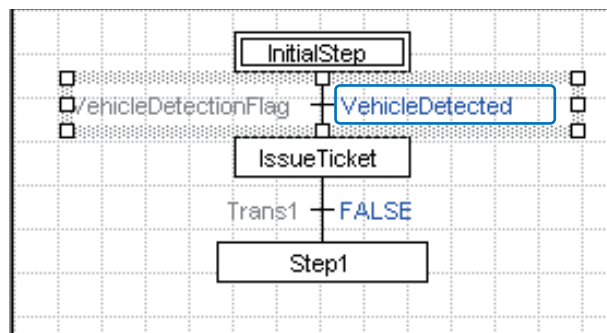
- ① Double-click the **NewProgram1 (00)** folder.
The SFC editor will be displayed.



- ② Double-click **FALSE**.
FALSE will be selected.

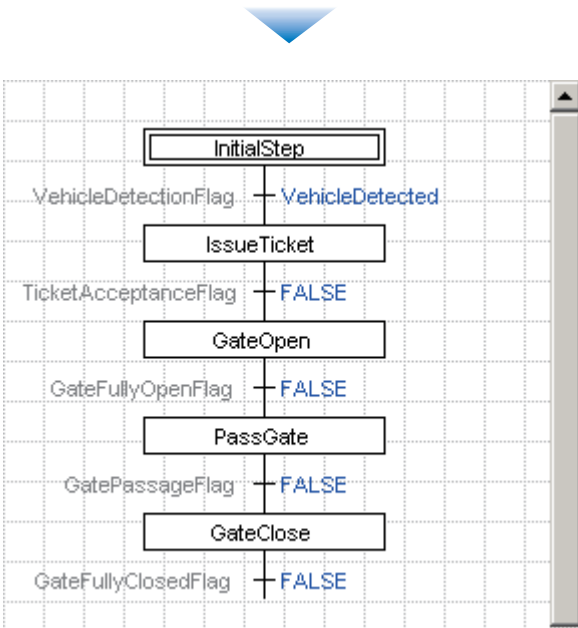


- ③ Type "Vehicle Detected", and press the **Enter** key.
The transition condition will be assigned.
The **Vehicle Detected** transition program will be displayed in the program view.



4 Create the remaining steps and transitions.

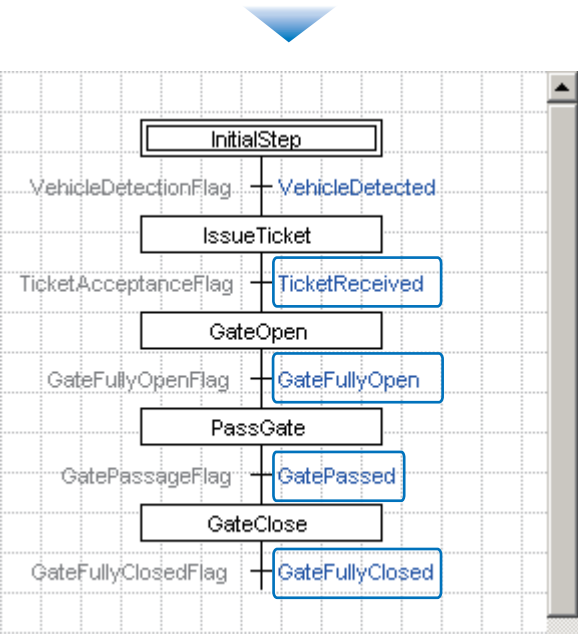
- ① Create all the other remaining steps and transitions.
For details on steps and transitions to be created, refer to 2-2-3 *Programming Example*.



5 Assign all transition programs.

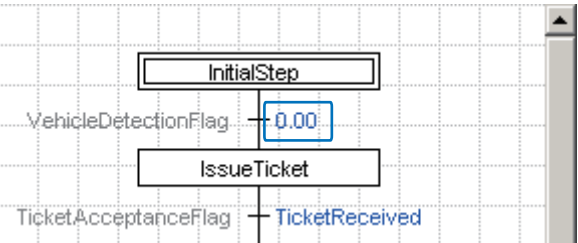
In this example, inputs from sensors are used as transition conditions.

- ① Assign transition programs to the transitions, in the same manner that the **Vehicle Detected** program was assigned to the **Vehicle Detection Flag** transition. For details on transition programs to be created, refer to 2-2-3 *Programming Example*.



Info

Sensor inputs can also be assigned to transitions as Boolean variable addresses. For example, the input contact (0.00) for the **Vehicle Sensor** can be assigned directly to the **Vehicle Detection Flag** transition.

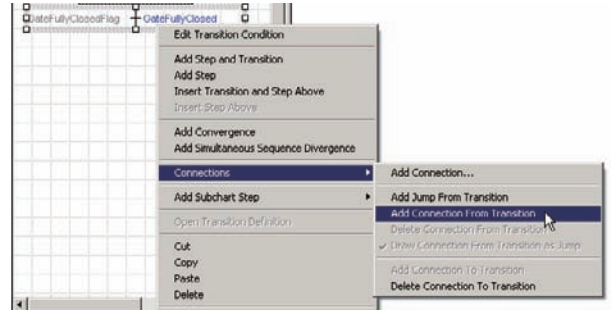


2-5-4 Creating Loops to Initial Steps

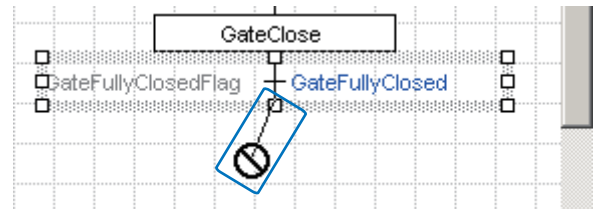
Add a connector extending from the **Gate Fully Closed Flag** transition to the **Initial Step**.

1 Draw a connector from the transition.

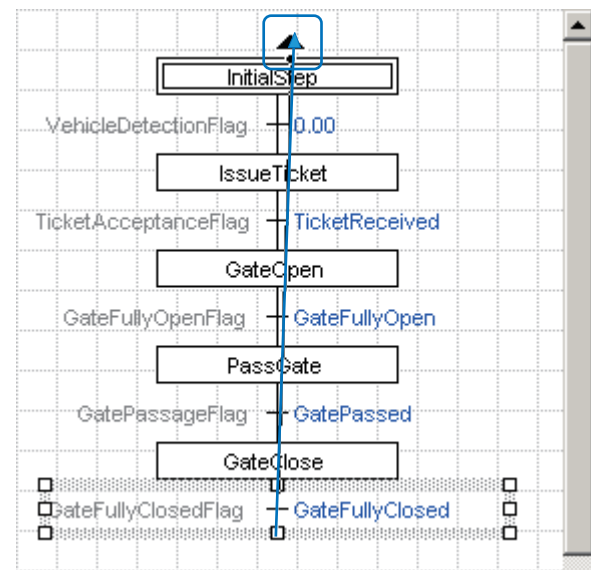
- Right-click the transition, and select **Connections - Add Connection From Transition** from the pop-up menu.



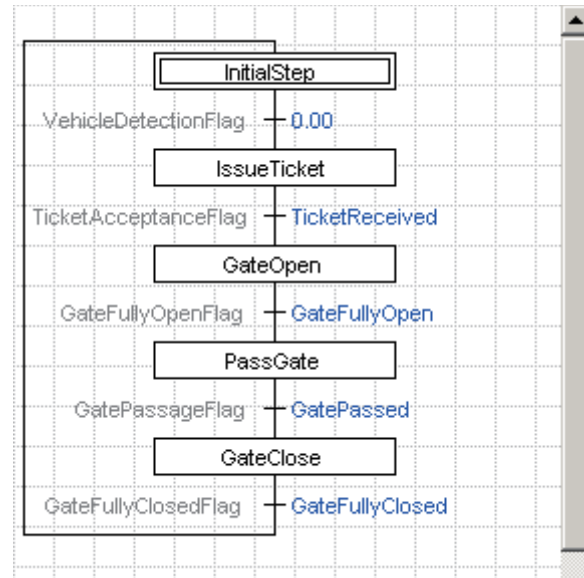
A connector extending from the transition to the cursor will be displayed.



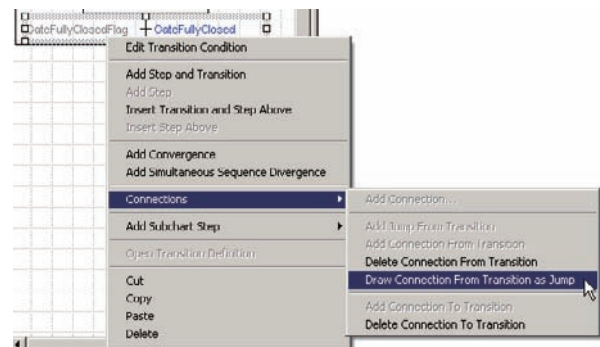
- Move the cursor to the **Initial Step**. The cursor will become an arrow.



- ③ Left-click on the mouse.
The step and the transition will be connected.
This connection is referred to as a "loop".

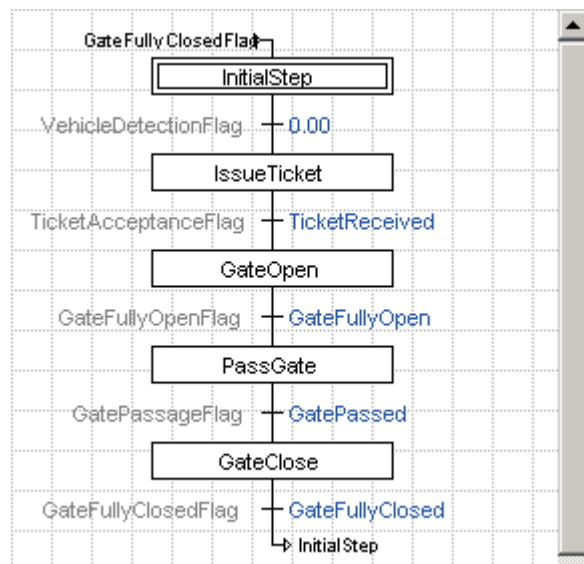


The loop can be displayed differently for easier viewing of the SFC chart.
Right-click the **Gate Fully Closed Flag** transition, and select **Connections - Draw Connection From Transition as Jump** from the pop-up menu.



A jump is functionally identical to a loop. They differ in appearance only.

The SFC chart is now completed, except for the actions.



2-6 Compiling

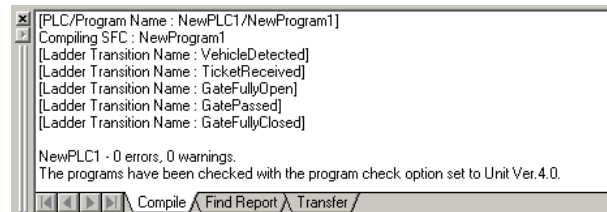
Test the program before creating actions for the steps.

2-6-1 Checking for Program Errors

Compile the program to check for errors.

1 Check all programs.

- ① Press the **F7** key.
Program compilation will begin.
Compilation results will be displayed in the output window.
If no error is detected, the program will operate properly. If an error is detected, correct the problem in SFC editor.

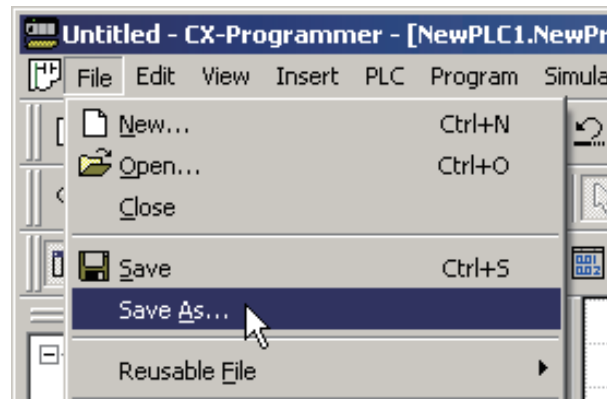


2-6-2 Saving Projects

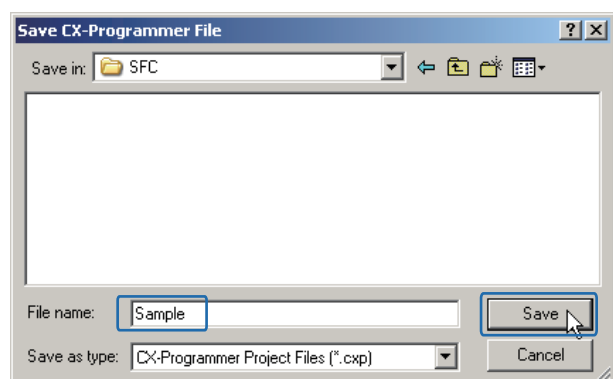
Save the created program.

1 Name and save the project.

- ① Select **File - Save As**.
The **Save CX-Programmer File** dialog box will be displayed.



- ② Enter a file name, and click **Save**.



2-7 Simulated Transition Test

Before creating the actions, check how the program operates when the transition conditions are satisfied. The simulator will allow you to test programs using only the computer.

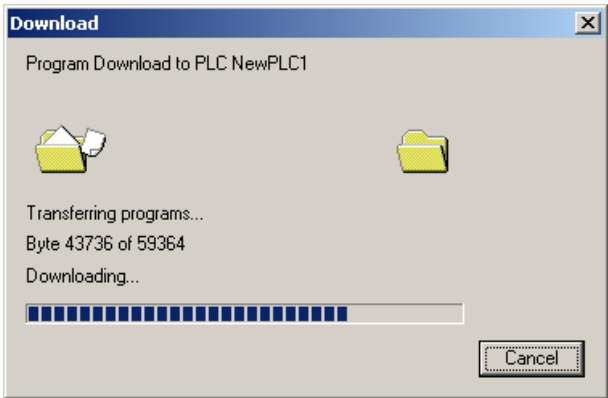
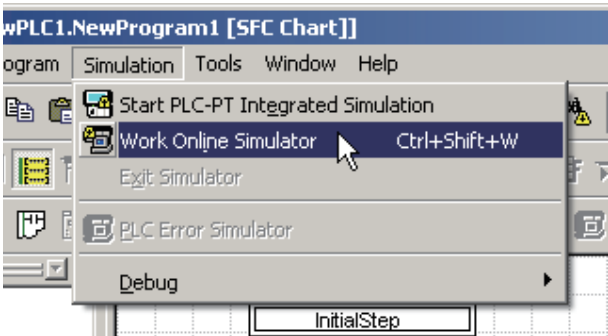
2-7-1 Online Connection to Simulator

Transfer the program to the simulator.

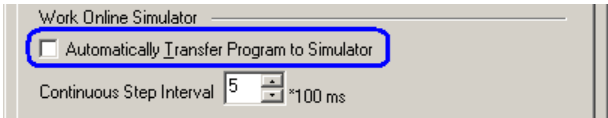
1 Establish an online connection.

- ① Select **Simulation - Work Online Simulator**.

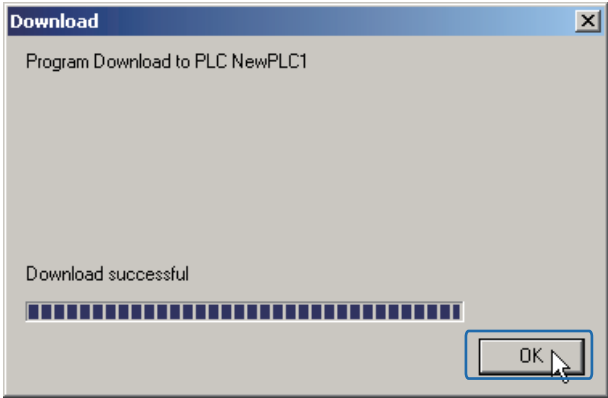
An online connection will be established between the computer and the simulator. The program will be automatically transferred to the simulator.



Info You can disable automatic program transfer to the simulator after establishing an online connection. Select **Tools - Options**. The **Options** dialog box will be displayed. Uncheck **Automatically Transfer Program to Simulator** in the **Work Online Simulator** section on the **PLC** tab.



- ② Click **OK**.
Online connection is now completed.

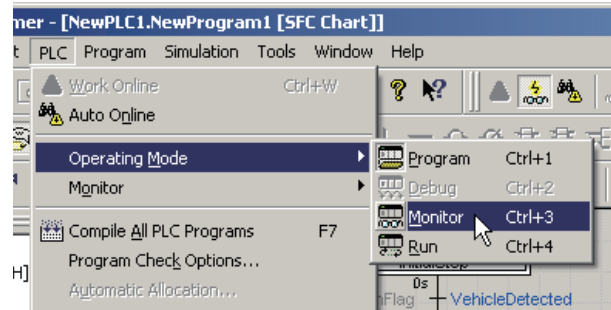


2-7-2 Setting Transition Conditions

Confirm that the transitions actually take place when the transition conditions are satisfied.

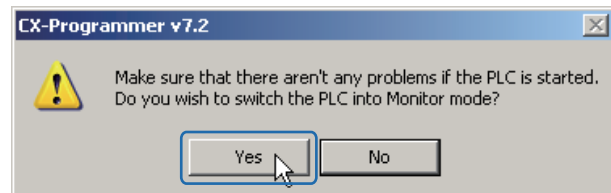
1 Switch to Monitor mode.

- ① Select **PLC - Operating Mode - Monitor**.
A confirmation dialog box will be displayed.

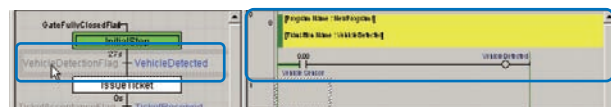


- ② Click **Yes**.
The operating mode for the PLC will be switched to Monitor mode.

The SFC editor will become greyed out.
The green frame represents the active step.



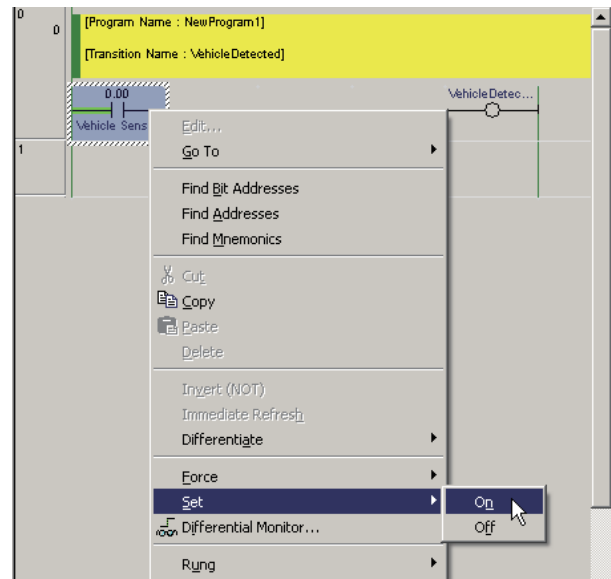
- ③ Click the **Vehicle Detection Flag** transition.
The **Vehicle Detected** program will be displayed in the program view.



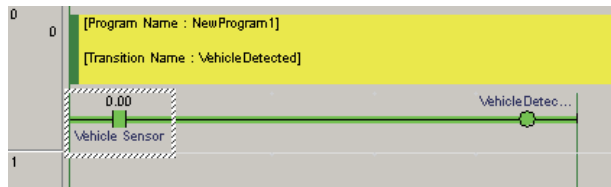
2 Turn the input contact ON.

- Right-click **Vehicle Sensor**, and select **Set - On** from the pop-up menu.

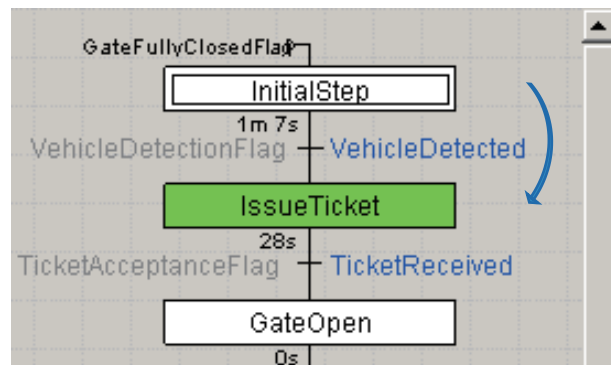
Info The contact can also be set to ON/OFF by double-clicking the contact, just as with any standard ladder program.



When the **Vehicle Sensor** turns ON, the **Vehicle Detected** program outputs ON, and the value for the **Vehicle Detection Flag** transition becomes TRUE.



Active status will be transferred from the **Initial Step** to the **Issue Ticket** step.



- Set or reset the other transition programs and confirm that the transitions take place.

3 End the simulation.

- Select **Simulation - Work Online Simulator**.
The simulator will go offline.
- Select **Simulation - Exit Simulator**.
The simulator will close.

2-8 Creating Action Blocks

Create the processes for the steps as action blocks.

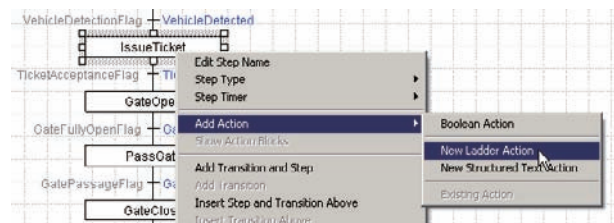
2-8-1 Creating Actions

Create the actions using ladder.

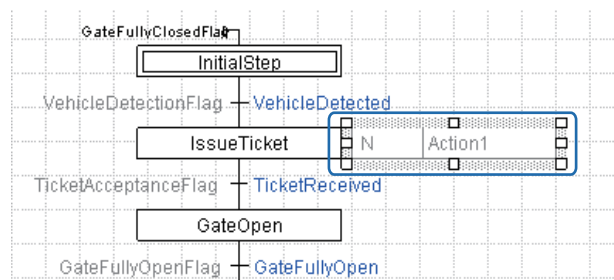
1 Register an action program.

The **Initial Step** does not require an action block, since there is nothing there to process. Create a ladder action program for the **Issue Ticket** step.

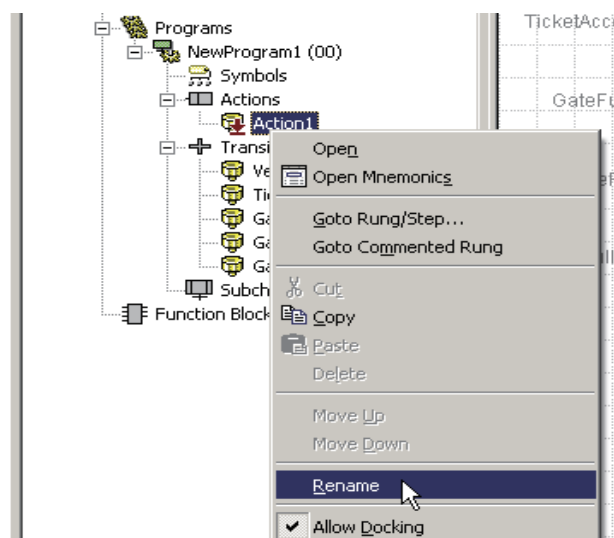
- Right-click the **Issue Ticket** step, and select **Add Action - New Ladder Action** from the pop-up menu.



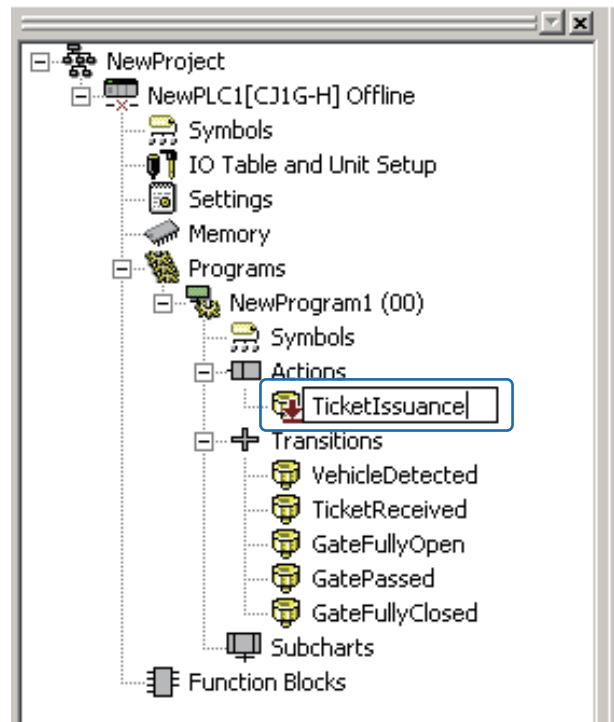
An action block will be added.



- Right-click **Action1**, which has been added to the project workspace, and select **Rename** from the pop-up menu.



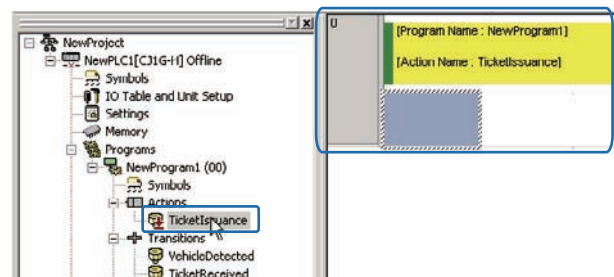
- ③ Type "Ticket Issuance".
The action program will be renamed to **Ticket Issuance**. The action will also be renamed automatically on the SFC chart.



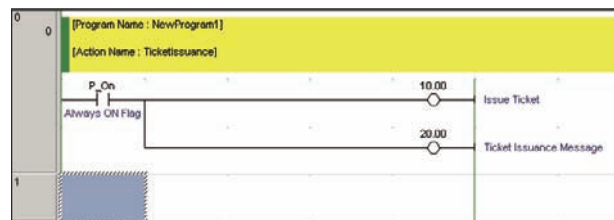
2 Create the program.

In the **Issue Ticket** step, issue a ticket and display the "Take ticket" message.
Create a program that starts these two processes when the step becomes active.

- ① Double-click the **Ticket Issuance** program.
The ladder editor will be displayed.



- ② Create the ladder program.
To turn an input ON when the step becomes active, use P_On (always On flag). Enter output contact **10.00** for the ticket dispenser, and output contact **20.00** for the display panel. For details on the programs to be created and the comments to be entered, refer to 2-2-3 *Programming Example*.



Info

For details on creating ladder programs, refer to *CX-Programmer Operation Manual*.

3 Create the remaining actions.

In the **Gate Open** step, open the gate and display the "Opening gate" message.

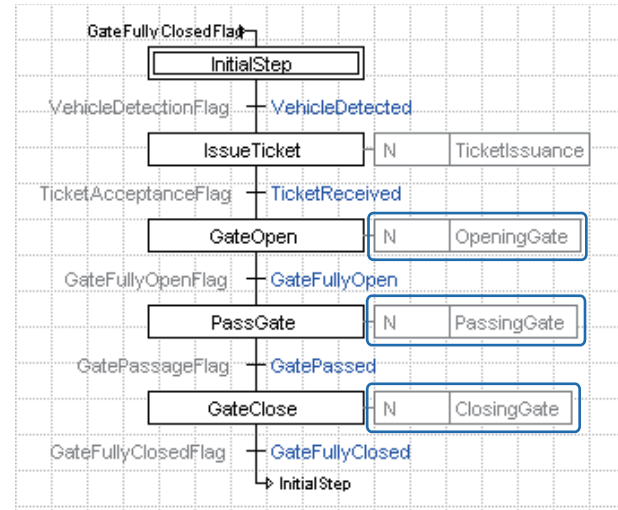
In the **Pass Gate** step, display the "Go" message.

In the **Gate Close** step, close the gate and display the "Closing gate" message.

In all steps, the outputs are reset at the end of the step.

Create action blocks for the remaining steps, in the same manner as for the **Issue Ticket** step.

For details on action programs to be created, refer to *2-2-3 Programming Example*.



2-8-2 Assigning Action Qualifiers

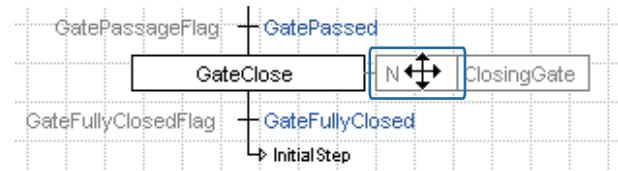
Set the execution timing for the actions.

1 Select an action qualifier.

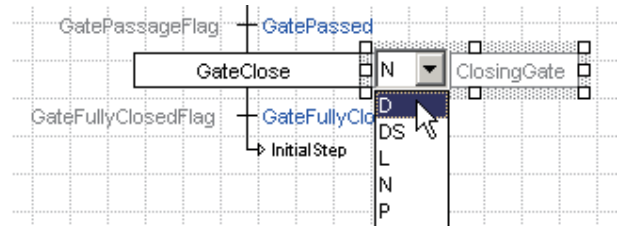
In the **Gate Close** step, the gate will be closed.

Wait until 3 seconds after the vehicle has passed the gate before beginning to close the gate. The execution timing for an action is set by using an action qualifier (AQ).

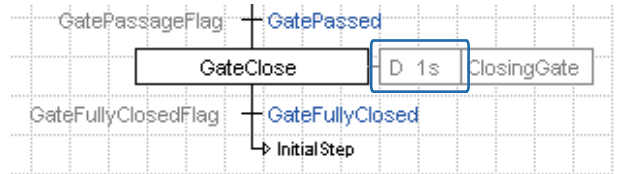
- ① Double-click the **N** AQ for the **Closing Gate** action.
A drop-down list will be displayed.
For details on AQ, refer to *A-1-2 Action Qualifier (AQ) List*.



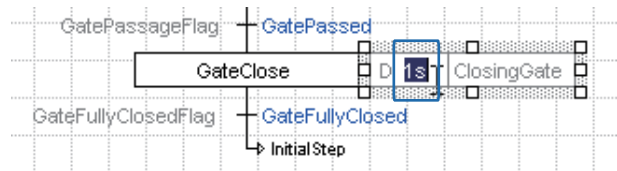
- ② Select **D** from the drop-down list.



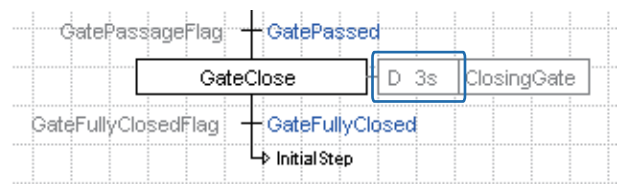
D 1s will be displayed as the AQ. This means that the action will be executed 1 second after the step becomes active.



- ③ Double-click **1s**.



- ④ Type "3", and press the **Enter** key.
D 3s will be displayed as the AQ.



The SFC chart is now fully completed.
Save the project.

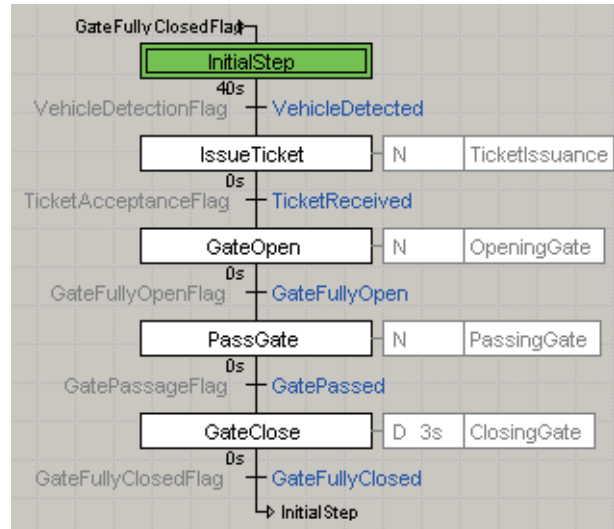
2-9 Simulated Operation Test

Once again, transfer the program to the simulator, and check the operation of the program as a whole, including the actions.

In 2-7 *Simulated Transition Test*, the transition conditions were tested. In this example, the operation of the actions will be tested.

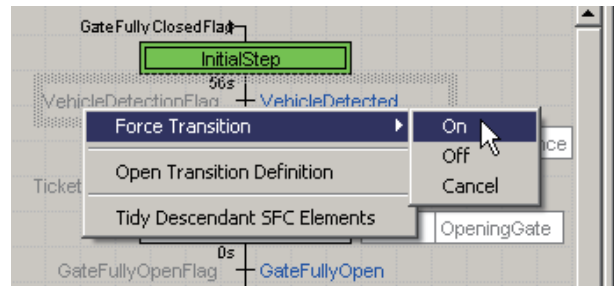
- 1 Establish an online connection.
Switch to Monitor mode.

The SFC editor will become greyed out.
Refer to 2-7 *Simulated Transition Test*.

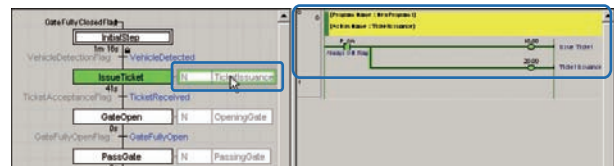


- 2 Turn the transitions ON.

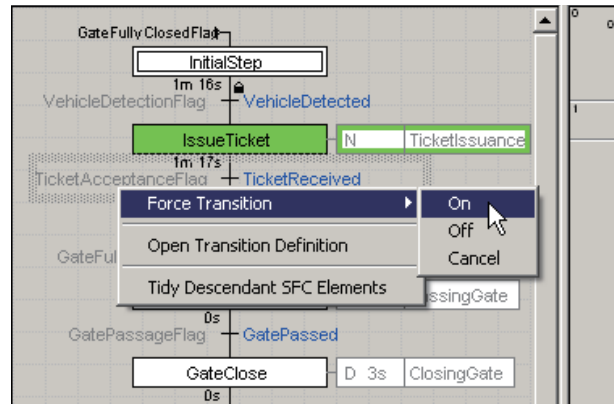
- ① Right-click the **Vehicle Detection Flag** transition, and select **Force Transition - On** from the pop-up menu.
The active status will be transferred from the **Initial Step** to the **Issue Ticket** step.
When a transition is force-set, the icon () will be displayed on the transition.



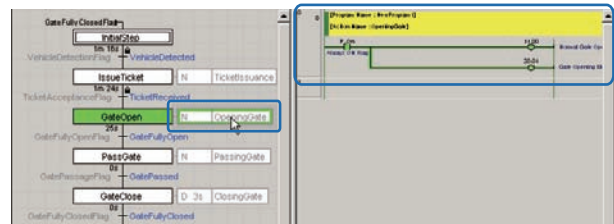
- ② Double-click the **Ticket Issuance** action.
Check the execution status of the action program in the program view.



- ③ Right-click the **Ticket Acceptance Flag** transition, and select **Force Transition - On** from the pop-up menu. The active status will be transferred from the **Issue Ticket** step to the **Gate Open** step.



- ④ Double-click the **Opening Gate** action. Check the execution status of the action program in the program view.

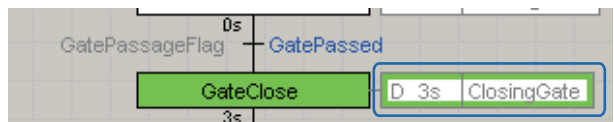
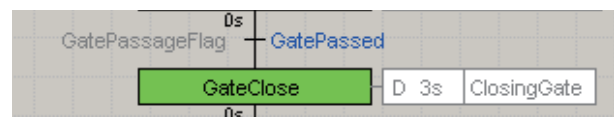


- ⑤ Click the **Ticket Issuance** action again. In the program view, confirm that outputs **10.00** (Issue Ticket) and **20.00** (Ticket Issuance Message) are both turned OFF.



3 Test the operation of the actions.

- ① Force-set the **Gate Fully Open Flag** transitions. Check the operation of the actions as the steps become active/inactive. Since the **Vehicle Sensor** in the **Gate Passed** transition program is OFF, the **Gate Passage** transition condition is satisfied. Active status will be transferred immediately from the **Pass Gate** step to the **Gate Close** step. The action block in the **Gate Close** step will be triggered 3 seconds after the transition.
- ② When the **Gate Fully Closed Flag** transition condition is satisfied, the **Initial Step** will become active once again. Test the program again in the same manner to double-check. The program operation test is now completed.

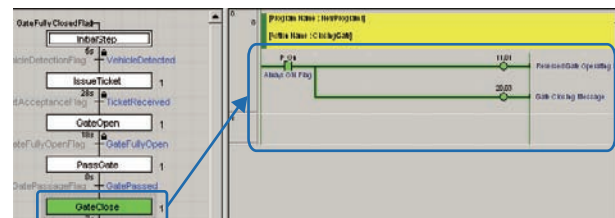
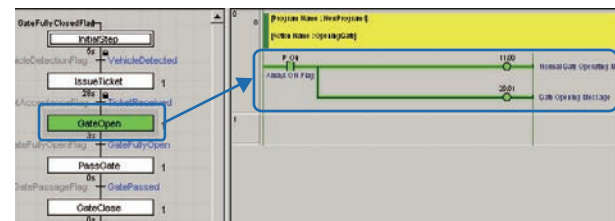
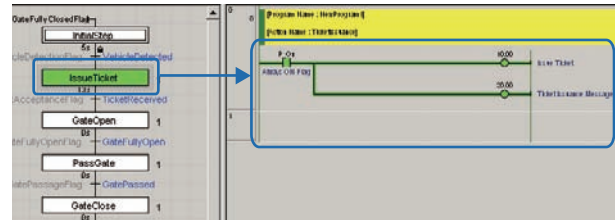


4 End the simulation.

- ① Select **Simulation - Work Online Simulator**.
The simulator will go offline.
- ② Select **Simulation - Exit Simulator**.
The simulator will close.

Info

Action blocks can be hidden for easier viewing of the steps and the transitions. When the action blocks are hidden, the action programs for the active step will be displayed as the active status is transferred.

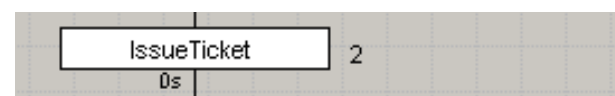
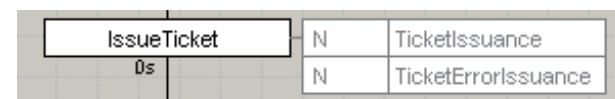


Info

Action blocks can be hidden in either of the following ways.

- Right-click a step and select **Show Action Blocks** from the pop-up menu.
- Right-click on a blank area of the SFC view, and select **Hide All Action Blocks** from the pop-up menu.

If there are multiple action blocks, select the action to be displayed by selecting its number from the list.

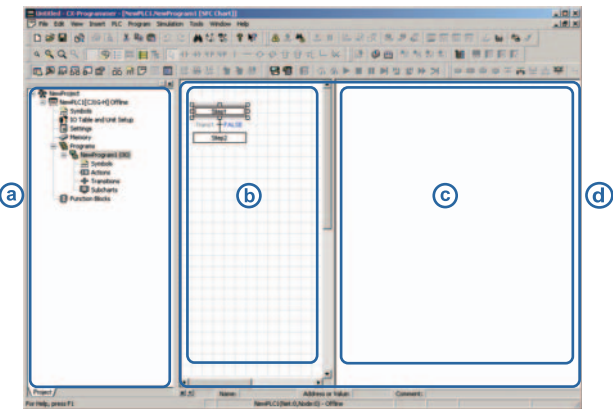


Appendix

A-1 Reference

A-1-1 Main Window

This section explains the CX-Programmer project workspace and the SFC editor.
For details on uncovered functions of the main window, and on ladder programming and ST programming, refer to *CX-Programmer Operation Manual* and *CX-Programmer Operation Manual Function Block/Structured Text*.

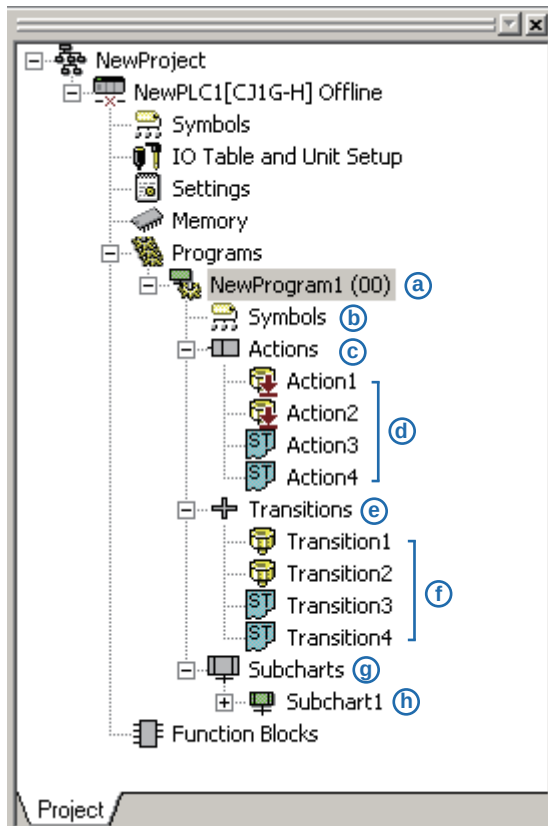


- (a) : Project Workspace
- (b) : SFC View
- (c) : Program View
- (d) : SFC Editor

■ Project Workspace

The elements of SFC programs are hierarchically displayed.

The project workspace displays elements for the entire project.



(a) : Program

SFC programs are registered under the **Programs** item.

Task allocation will be necessary to execute the newly added SFC program.

(b) : Symbols

Register local variables for use in SFC programs.

(c) : Actions folder

(d) : Action Programs

Register the step processes as action programs. Code the action programs in ladder or ST.

To make an association with an action program, specify its name within an action block on the SFC chart.

(e) : Transitions folder

(f) : Transition Programs

Register the step transition conditions as transition programs. Code the transition programs in ladder or ST.

To make an association with a transition program, specify its name within a transition on the SFC chart.

(g) : Subcharts folder

(h) : Subchart Programs

Register the subchart programs created for structuring purposes (i.e. reuse). Code the subchart programs in SFC.

To make an association with a subchart program, specify its name within a subchart step on the SFC chart.

A-1-2 Action Qualifier (AQ) List

AQ Symbol	Name	Function	Timer Value
D	Delay	The action is executed when the user-specified time has elapsed since the activation of the step. If the step becomes inactive before the user-specified time expires, the action will not be executed.	Required
DS	Delay Set	The operation is the same as for "SD". Unlike "SD", however, the action will not be performed if the step becomes inactive before the user-specified delay time expires. Execution can be aborted with the "R" AQ.	Required
L	Limit	The action is executed when the step is activated, and is terminated when the user-specified time expires. If the step becomes inactive before the user-specified time expires, execution will be aborted.	Required
N	Normal (Default)	The action is executed for as long as the step is active.	-
P	Pulse	If a final scan is enabled, the action is executed for 2 cycles when the step is activated. If a final scan is disabled, the action is executed for 1 cycle when the step is activated.	-
P1	Rising Pulse	The action is executed once when the step becomes active.	-
P0	Falling Pulse	The action is executed once when the step becomes inactive.	-
R	Reset	Stops and resets the action when the steps is activated, if the action is being executed by "S", "SL", "SD", or "DS". Actions being executed by other AQs will be reset, but not stopped. When an action is reset, the following operations are performed: - OUT/OUT NOT instruction: OFF - TIM/TIMH instruction: Reset - Other timer, counter, or shift register: Maintained	-
S	Set	The action is executed when the step is activated, and continues to be executed even after the step becomes inactive. Execution can be aborted with the "R" AQ.	-
SD	Set Delay	The action is executed when the user-specified time has elapsed since the activation of the step, and continues to be executed even after the step becomes inactive. Execution can be aborted with the "R" AQ.	Required
SL	Set Limit	The action is executed when the step is activated, and is terminated when the user-specified time expires. Unlike "L", the action continues to be executed even after the step becomes inactive. Execution can be aborted with the "R" AQ.	Required

For details on action qualifiers, refer to *CX-Programmer Operation Manual SFC*.

A-2 Creating SFC Elements

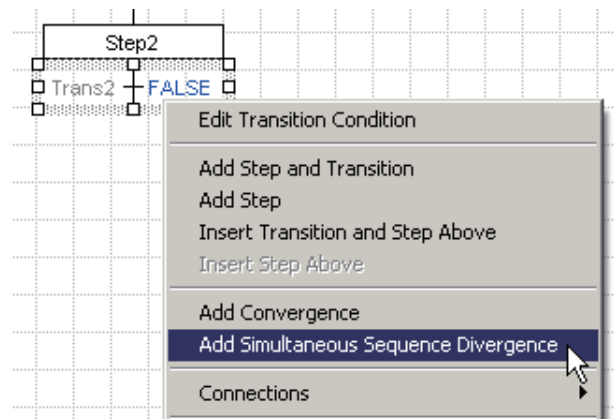
This section explains how to create more complex SFC elements, such as simultaneous sequence divergences, simultaneous sequence convergences, divergences, and convergences.

A-2-1 Simultaneous Sequence Divergences/Convergences

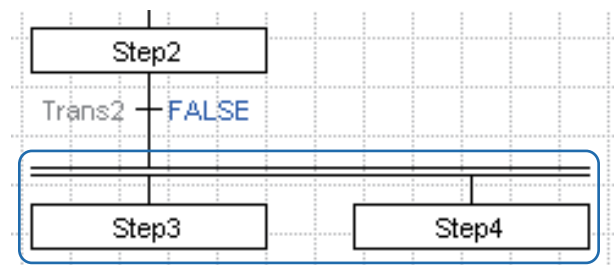
This section explains how to add a simultaneous sequence divergence/convergence.

1 Create a simultaneous sequence divergence.

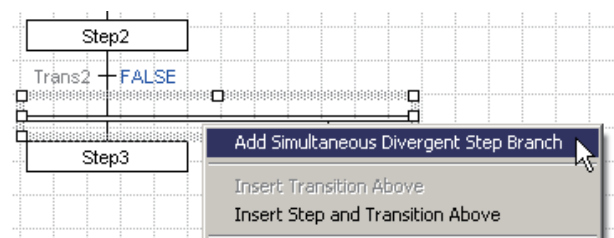
- ① Right-click a transition, and select **Add Simultaneous Sequence Divergence** from the pop-up menu.



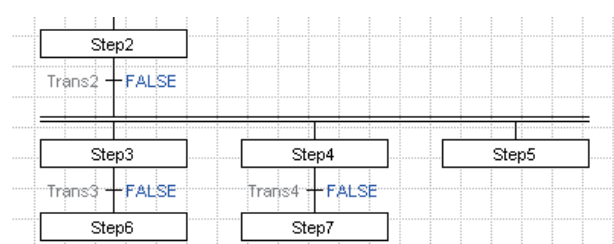
A double-line representing a simultaneous sequence divergence will be added. Two diverted steps will also be added to the divergence.



- ② Right-click the double-line, and select **Add Simultaneous Divergent Step Branch** from the pop-up menu. Another step will also be added to the divergence.

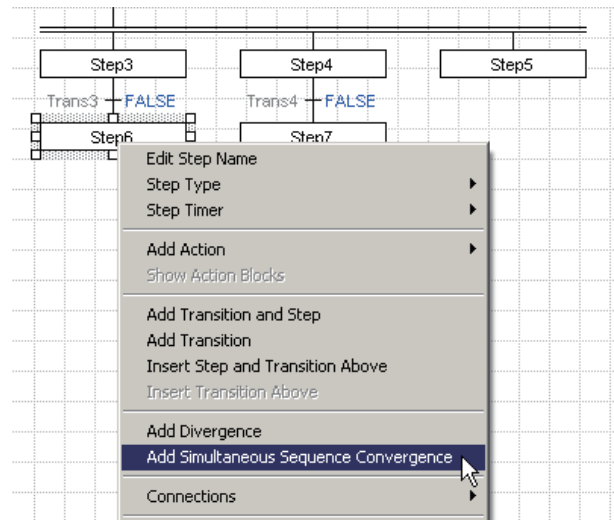


- ③ Add transitions and steps to each of the diverted steps. Complete the diverted SFC chart.

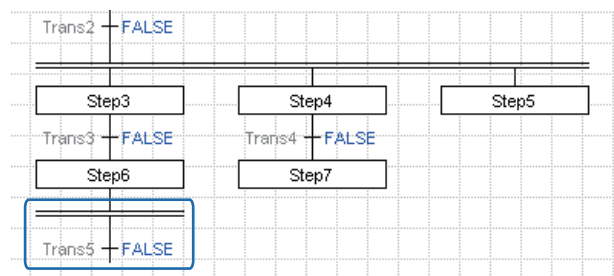


2 Create a simultaneous sequence convergence.

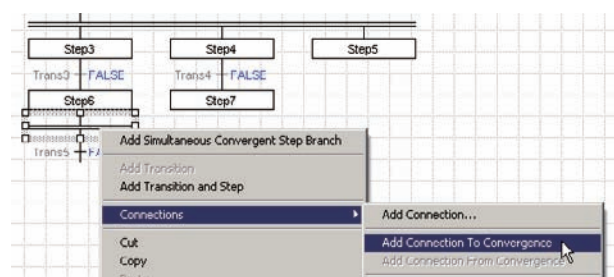
- ① Right-click a diverted step, and select **Add Simultaneous Sequence Convergence** from the pop-up menu.



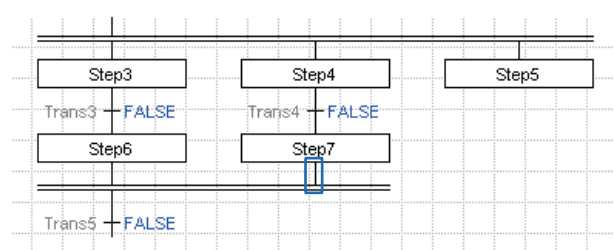
A double-line representing a simultaneous sequence convergence, and a transition will be added. A connector extending from the step to the double-line will also be added.



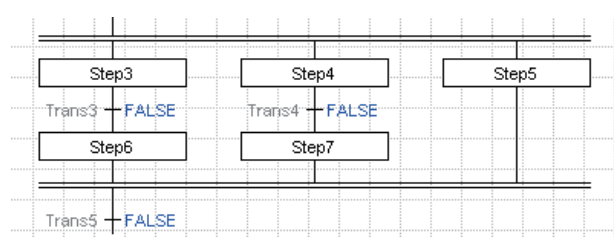
- ② Right-click the double-line, and select **Connections - Add Connection To Convergence** from the pop-up menu.



- ③ Move the cursor to the step to be merged, and click.
The step will be connected to the double-line.



- ④ Complete the simultaneous sequence convergence by repeating the above procedure.

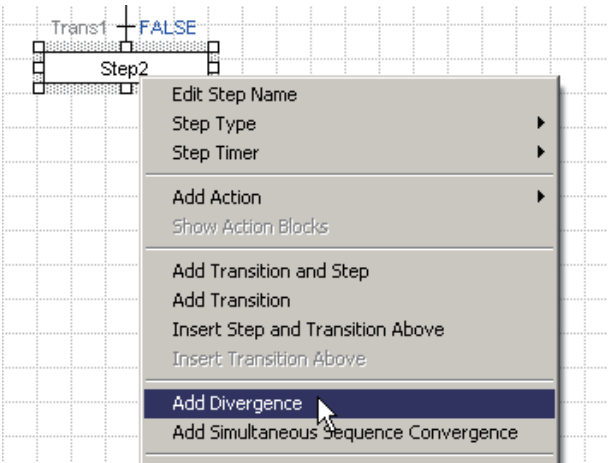


A-2-2 Divergences/Convergences

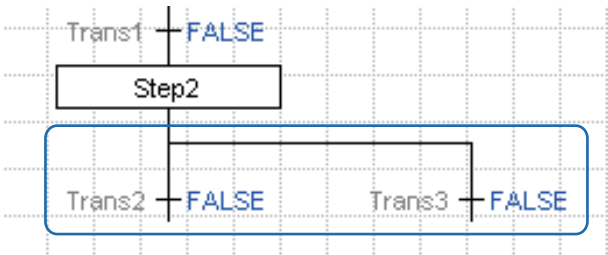
This section explains how to add a divergence/convergence.

1 Create a divergence.

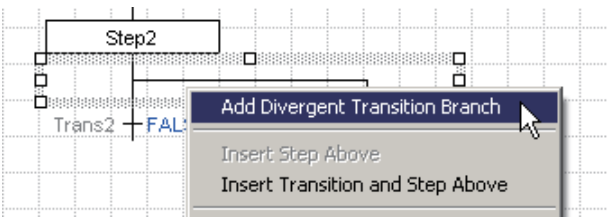
- 1 Right-click a step, and select **Add Divergence** from the pop-up menu.



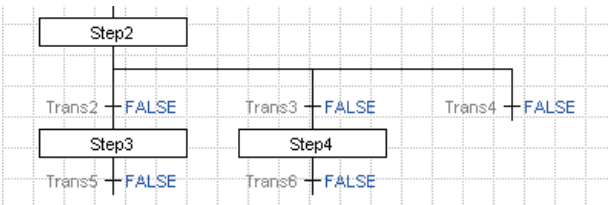
A diverted connector will be added. Two transitions are also added.



- 2 Right-click the connector, and select **Add Divergent Transition Branch** from the pop-up menu. Another diverted transition will be added to the connector.

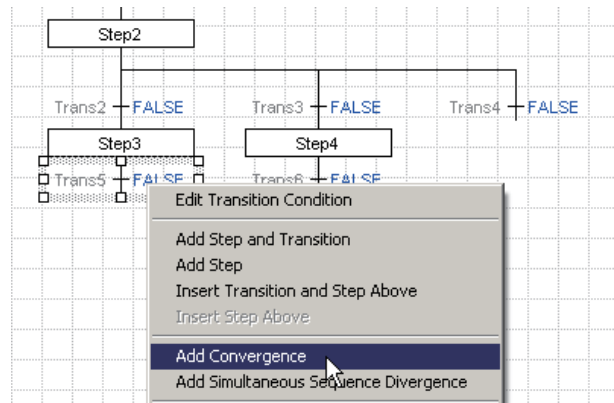


- 3 Add steps and transitions to each of the diverted transitions. Complete the diverted SFC chart.

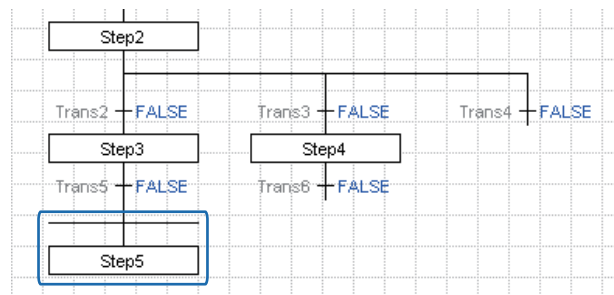


2 Create a convergence.

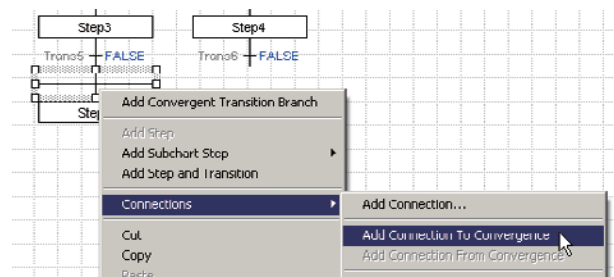
- ① Right-click a transition, and select **Add Convergence** from the pop-up menu.



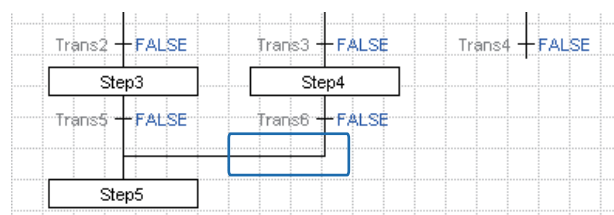
A convergence line and a step will be added. A connector extending from the transition to the convergence line will also be added.



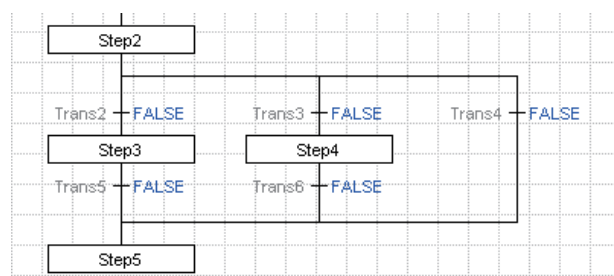
- ② Right-click the convergence line, and select **Connections - Add Connection To Convergence** from the pop-up menu.



- ③ Move the cursor to the transition to be merged, and click. The transition will be connected to the convergence line.



- ④ Complete the convergence by repeating the above procedure.

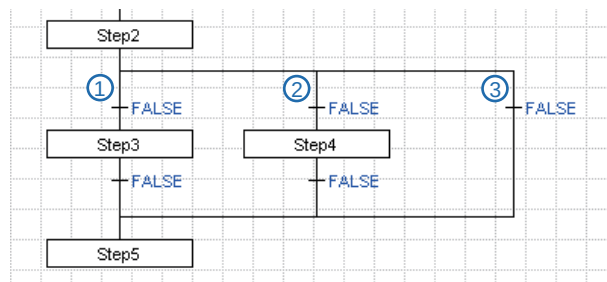


■ Transition Precedence of Diverged Branches

By default, transition conditions for diverged branches are tested from left to right.

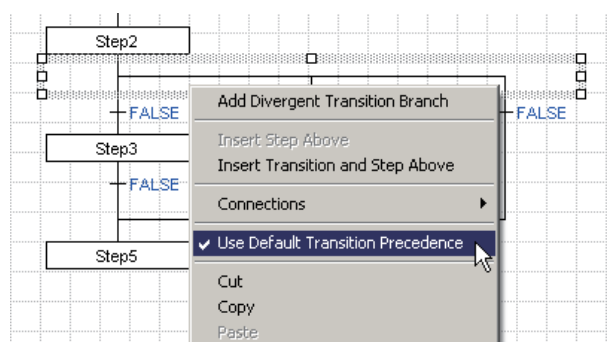
The transition precedence can be changed if desired.

The procedure for changing the transition precedence is detailed below.

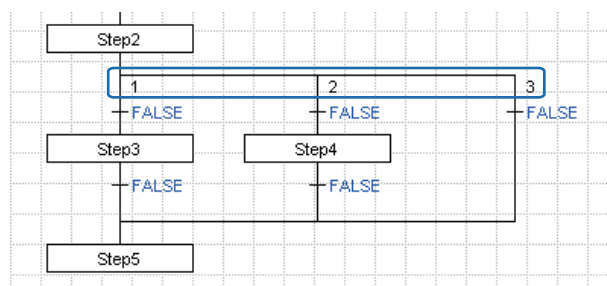


Change the transition precedence of diverged branches.

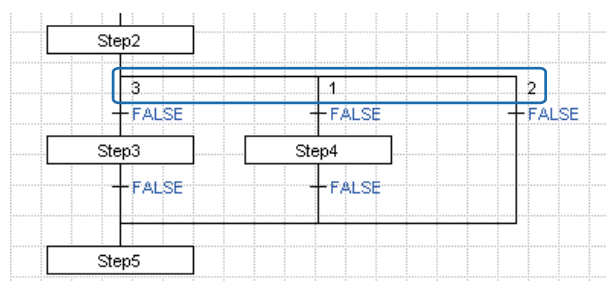
- 1 Right-click the divergence line, and select **Use Default Transition Precedence** from the pop-up menu.



The SFC chart will be labeled with numbers representing the transition precedence.



- 2 Double-click the numbers, type the desired order, and press the **Enter** key. The transition precedence will be changed.



A-3 Online Editing

Programs can be corrected while the PLC is running.

Note

Online editing cannot be performed when using the simulator.

A-3-1 Transferring Programs to PLC

In preparation for online editing, connect the computer and the PLC, and transfer the program from the computer to the PLC.

Normally, online editing is performed on a program uploaded from a running PLC.

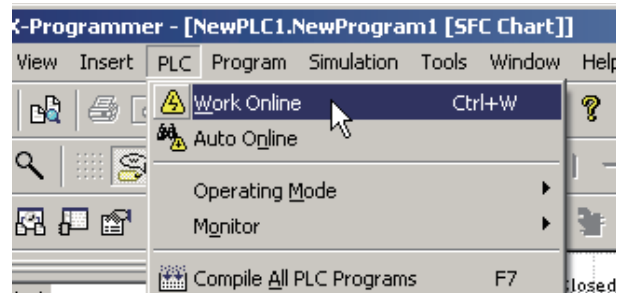
1 Establish an online connection.

- ① Physically connect the computer to the PLC in advance.

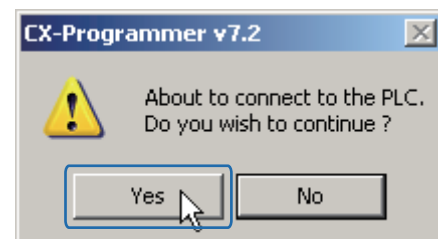
Info

For details on establishing an online connection with the PLC, refer to *CX-Programmer Operation Manual*.

- ② Select **PLC - Work Online**.
A confirmation dialog box will be displayed.

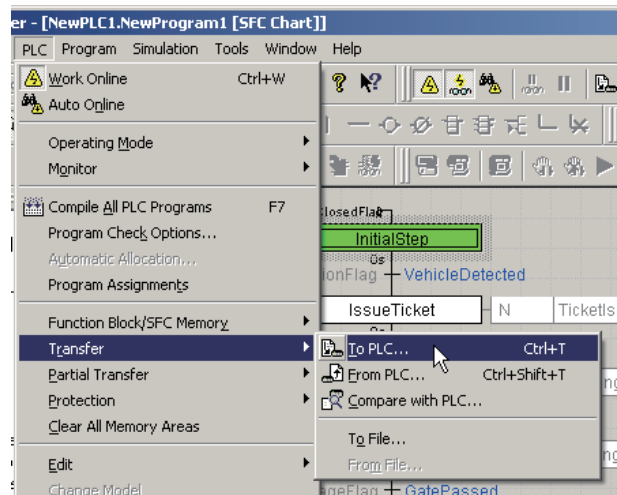


- ③ Click **Yes**.
An online connection will be established between the computer and PLC.

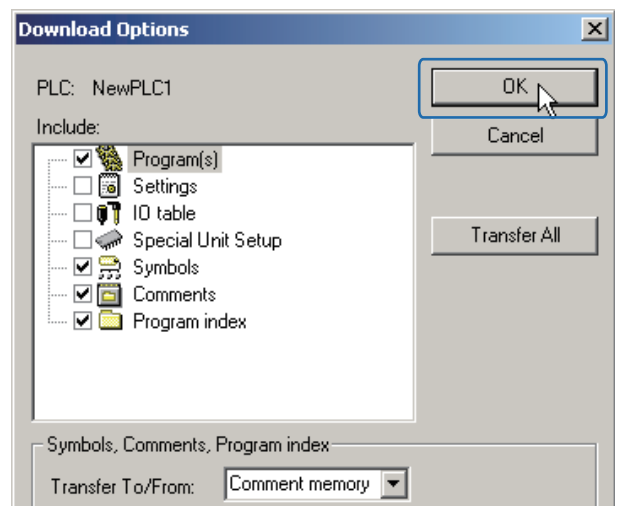


2 Transfer the program.

- ① Select **PLC - Transfer - To PLC**.
The **Download Options** dialog box will be displayed.



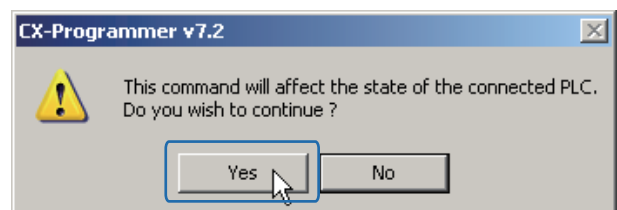
- ② Click **OK**.
The **Download** dialog box will be displayed.
When the data transfer is completed, the **Download successful** message will be displayed.



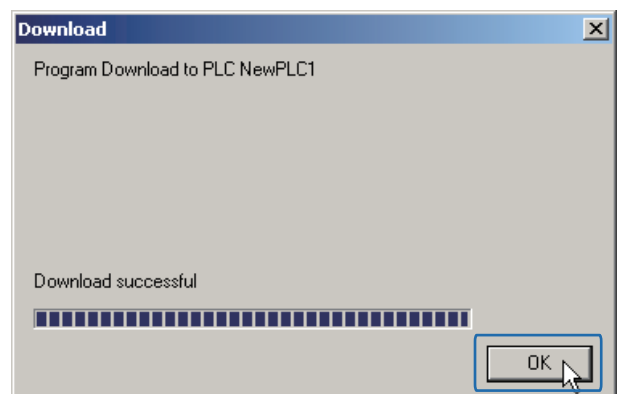
- ③ Click **Yes**.
The dialog box will be closed.

Info

If the PLC is not in Program mode, a dialog box will be displayed to confirm switching of the operating mode.
Click **Yes**.



- ④ Click **OK**.
The program transfer is now completed.



A-3-2 Online Editing

Load and correct the program, currently in use by the PLC.

Note

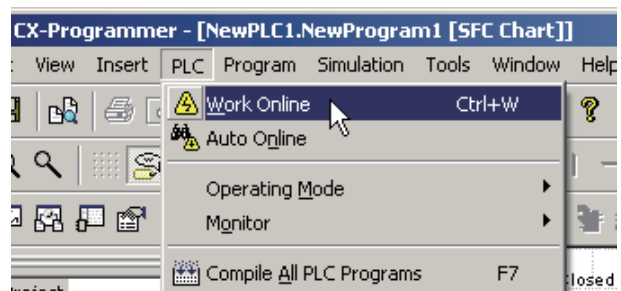
The programs used in this manual are provided for educational purposes only, to aid in understanding how CX-Programmer works.

When designing a program for actual use, be sure to make considerations for safety in terms of hardware devices and control methods.

1 Establish an online connection.

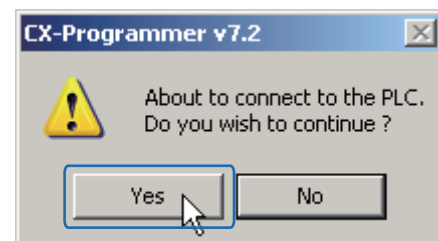
① Select **PLC - Work Online**.

A confirmation dialog box will be displayed.



② Click **Yes**.

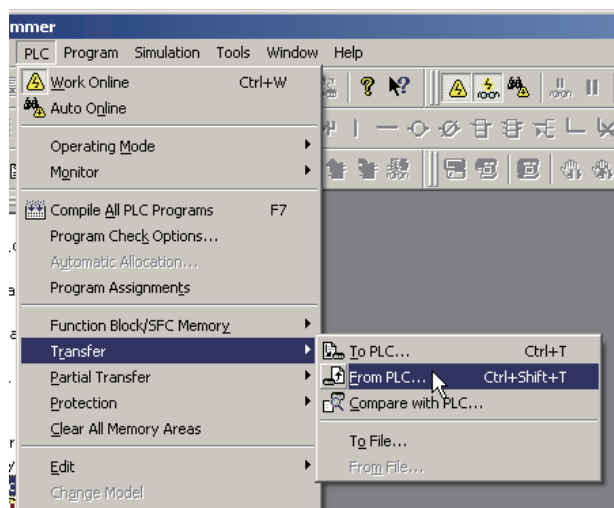
An online connection will be established between the computer and PLC.



2 Load the program.

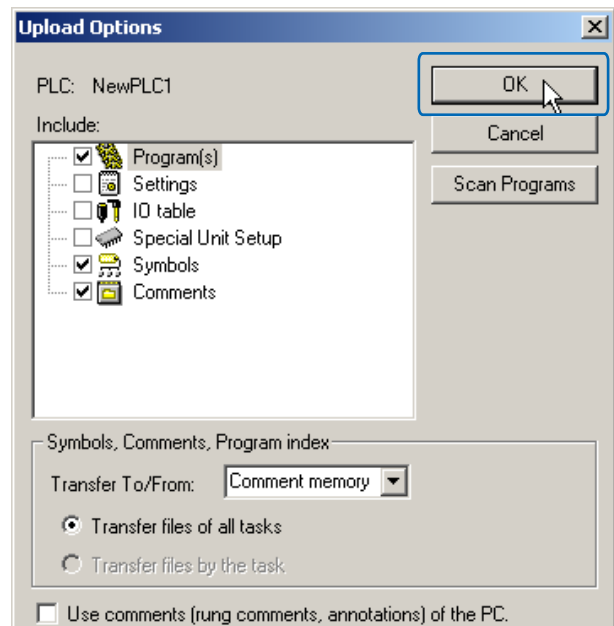
① Select **PLC - Transfer - From PLC**.

The **Upload Options** dialog box will be displayed.

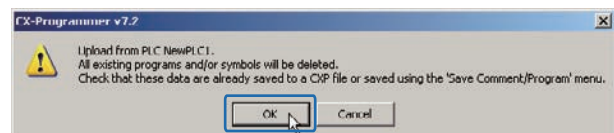


② Click **OK**.

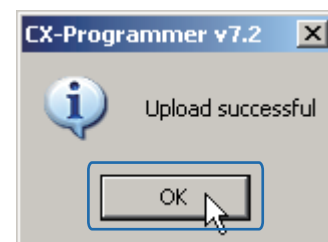
A dialog box will be displayed, confirming the deletion of programs currently being edited in CX-Programmer.

③ Click **OK**.

The **Upload** dialog box will be displayed. When the data transfer is completed, the **Upload successful** message will be displayed.

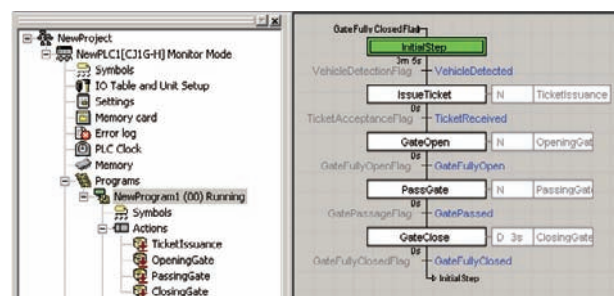
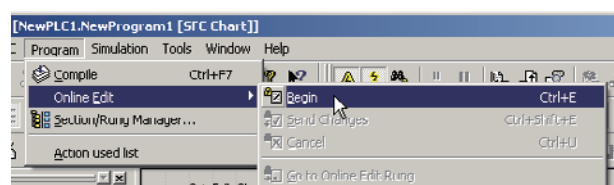
④ Click **OK**.

The program transfer is now completed.

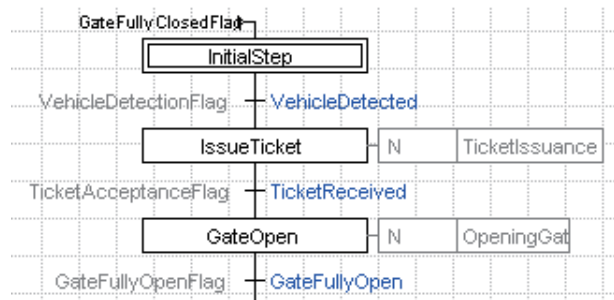


3 Begin online editing.

① Display the SFC chart. Switch to Monitor mode.

② Select **Program - Online Edit - Begin**.

The SFC editor will become un-greyled, and the program becomes editable.



③ Edit the program.

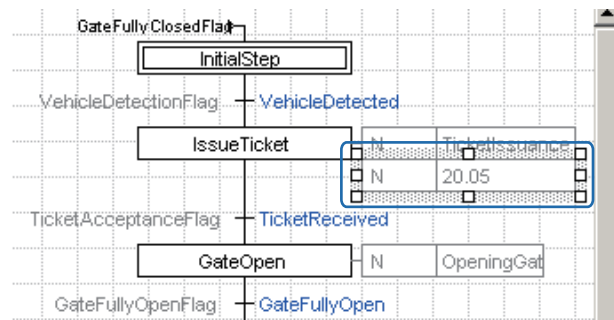
Info

To edit an action program or a transition program, first display the program editor by double-clicking the target program within the project window, and then begin online editing.

When adding a new action program or transition program to the SFC chart, you will need to perform online editing in two sessions.

1st session: Add the new action program or transition program to the project workspace.

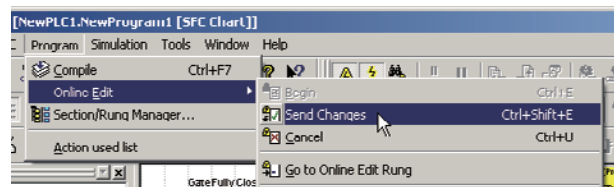
2nd session: Add the program added in the 1st session to the SFC chart.



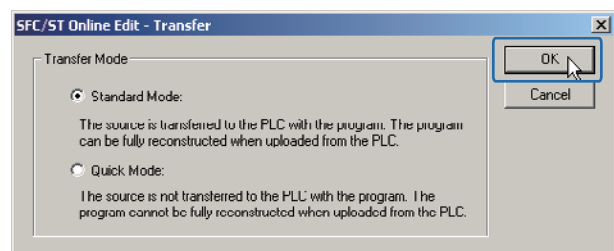
4 Transfer the changes.

① Select **Program - Online Edit - Send Changes**.

The **SFC/ST Online Edit - Transfer** dialog box will be displayed.



② Click **OK**.

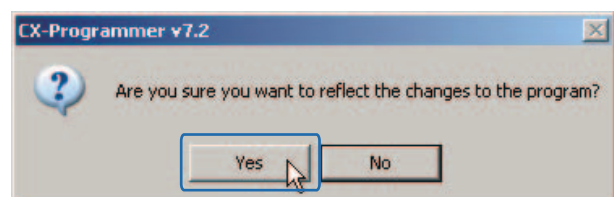


③ Click **Yes**.

The modified program will be transferred to the PLC. End the online editing session.

Info

To discard the changes made during online editing, or to end the online editing session, select **Program - Online Edit - Cancel**. The online editing session will be terminated without transferring the program to the PLC.



Memo

A

Appendix

Index

A

Action 1-2, 1-4, 2-28
 Action Block 1-4, 2-28
 Action Flow 2-5
 Action Program 1-3, 2-8, 2-28, A-2
 Action Qualifier(AQ) 1-4, 2-30, A-3
 Active Step 1-5

C

Compile 2-24
 Connector 2-16, 2-23
 Convergence 1-5, A-4
 Creating SFC A-4

D

Device Type 2-9
 Divergence 1-5, A-6

F

Force-Set 2-26, 2-32

I

Inactive Step 1-5
 Initial Step 2-13

J

Jump 2-22

L

Loop 2-23

M

Main Window 2-9, A-1
 Monitor Mode 2-26

N

Network Type 2-10

O

Online Connection 2-25

P

Program Error 2-24
 Project 2-9
 Project Workspace A-2

S

Set 2-27
 SFC Chart 1-3, 2-6, 2-13
 SFC Editor 1-3, 2-12, A-1
 SFC Operation 1-4
 SFC Overview 1-1
 SFC Program 2-11, 2-13

Simulator 2-25
 Simultaneous Sequence Convergence 1-4, A-5
 Simultaneous Sequence Divergence 1-4, A-4
 Starting CX-Programmer 2-9
 Step 1-4, 2-13, A-3
 Subchart A-2
 Symbols A-2

T

Task A-2
 Transition 1-2, 1-4, 2-13, 2-18
 Transition Condition 1-1, 1-2, 1-5, 2-7
 Transition Program 1-3, 2-7, A-2

OMRON Corporation**Industrial Automation Company****Control Devices Division H.Q.****PLC Division**Shiokoji Horikawa, Shimogyo-ku,
Kyoto, 600-8530 Japan

Tel: (81) 75-344-7084/Fax: (81) 75-344-7149

Regional Headquarters**OMRON EUROPE B.V.**Wegalaan 67-69-2132 JD Hoofddorp
The Netherlands

Tel: (31)2356-81-300/Fax: (31)2356-81-388

OMRON Industrial Automation Global: www.ia.omron.com**OMRON ELECTRONICS LLC**One Commerce Drive Schaumburg,
IL 60173-5302 U.S.A.

Tel: (1) 847-843-7900/Fax: (1) 847-843-7787

OMRON ASIA PACIFIC PTE. LTD.No. 438A Alexandra Road # 05-05/08 (Lobby 2),
Alexandra Technopark, Singapore 119967

Tel: (65) 6835-3011/Fax: (65) 6835-2711

OMRON (CHINA) CO., LTD.Room 2211, Bank of China Tower,
200 Yin Cheng Zhong Road,

PuDong New Area, Shanghai, 200120, China

Tel: (86) 21-5037-2222/Fax: (86) 21-5037-2200

Authorized Distributor:In the interest of product improvement,
specifications are subject to change without notice.**Cat. No. R149-E1-03**