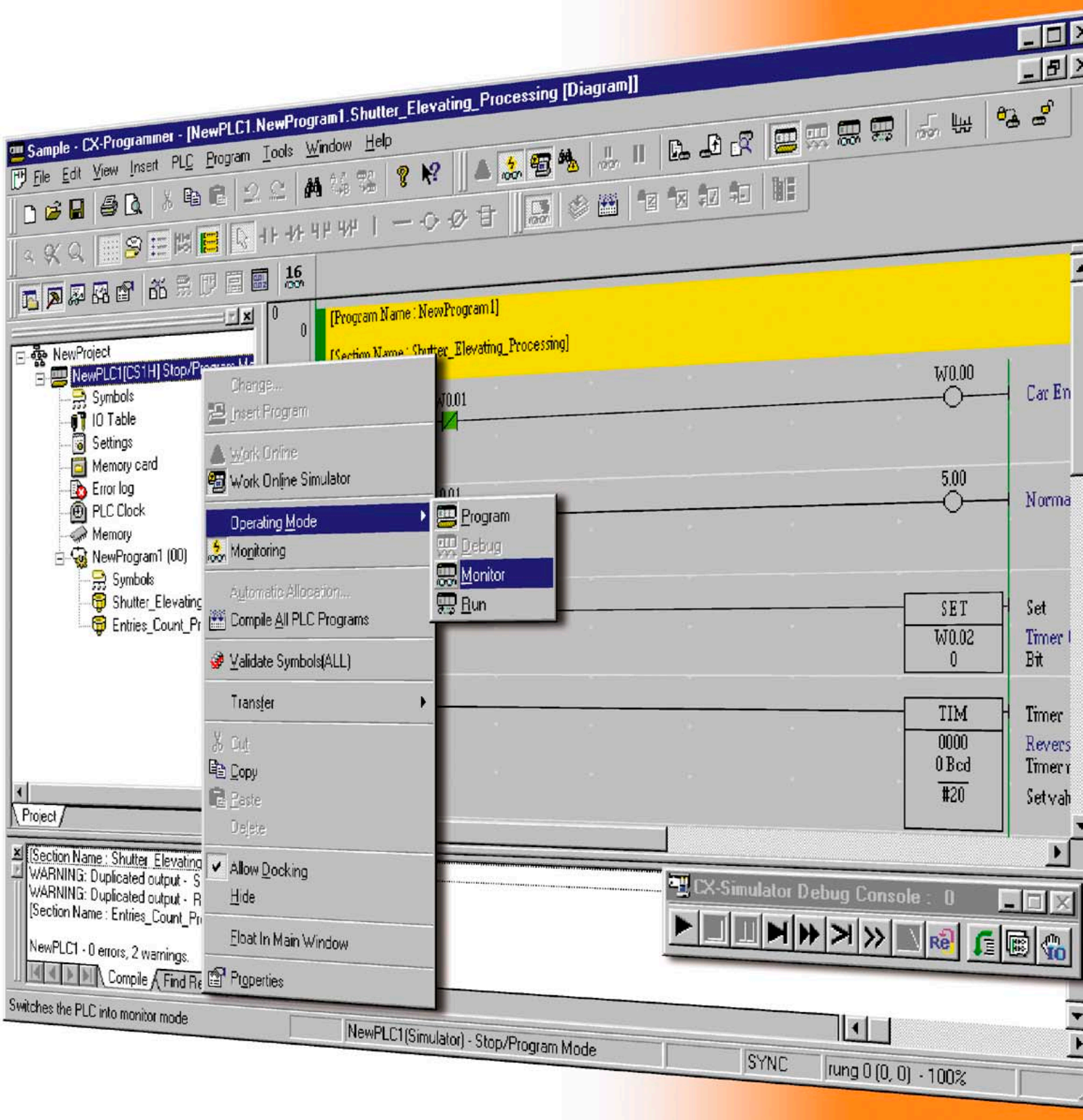
CX-Simulator

Introduction Guide



AUDIN - 8, avenue de la malle - 51370 Saint Brice Courcelles - Tel : 03.26.04.20.21 - Fax : 03.26.04.28.20 - Web : http: www.audin.fr - Email : info@audin.fr

WS02-SIMC1-E

CONTENTS

Chapter 1 Overview of CX-Simulator

Major Characteristics of CX-Simulator

Program execution, monitoring, debug without actual PLC

Monitoring of programs and IO memory present values is enabled. Moreover, normal debugs such as force on/off, differential monitoring, data trace and online edit are enabled from CX-Programmer. Also, any cyclic task can be started/stopped and interrupt tasks can be started in simulation.

Cycle time check without actual PLC

It is possible to check estimated cycle time (current, minimum, maximum, and mean values, servicing time *1) when the program is executed in an actual PLC in advance.

Program execution per step or scan, I/O brake condition settings

Debugs that cannot be realized in actual PLCs are provided.

- Step Run: Executes a program per instruction
- Scan Run: Executes a program per scan (cycle)
- I/O Break Condition Settings: Aborts execution when the conditions set in I/O memory are satisfied.
- Designation of the start and break points

Debugs in connection with display devices and serial communications devices *1

Regarding the serial communications port of a PC as the communications port of a PLC, it is possible to debug a program in combination with display devices or serial communications devices (barcode reader, ID sensor, etc.).

Display of the send messages of serial communications and network communications *1

It is possible to check the send messages issued by TXD instruction (communications port output), SEND instruction (network send), and CMND instruction (command send). It is useful for the debugs of serial communications and network communications.

*1: You need to start CX-Simulator from the Windows [Start] menu and set a virtual PLC. See Chapter 5 and the operation manual of CX-Simulator for the detailed operations.

Overview of CX-Simulator

CX-Simulator enables you to realize SYSMAC CS/CJ series CPU Units in your computer as a virtual PLC and operate (simulate) it equally as actual CPU Units. Combination use of CX-Simulator and CX-Programmer enables you to verify ladder program operation and cycle time in advance on a PC without an actual PLC.

Moreover, various debug functions of CX-Simulator make it possible to debug ladders, which used to be impossible by using an actual PLC only.

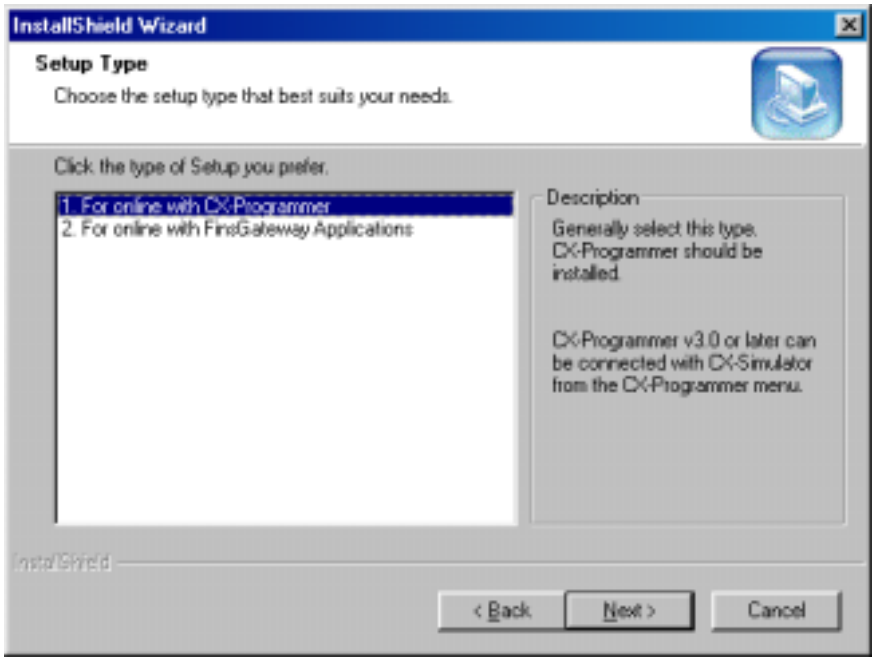
Relationship between CX-Simulator and CX-Programmer

CX-Simulator creates a virtual PLC on a virtual network in your PC. If you use the "Work Online Simulator" function of CX-Programmer Ver.3.0 or greater versions, CX-Simulator automatically starts up a virtual PLC of the current project's device type to open connection between CX-Programmer and the virtual PLC.

Before Installation

There are two setup types in CX-Simulator V1.3, but choose "1. For online with CX-Programmer" if CX-Programmer has already been installed in your computer.

CX-Simulator is automatically installed in the directory where CX-Programmer is installed.



Choose "2. For online with FinsGateway Applications" if you want to use a virtual PLC for the debug of an application using the FinsGateway network.

Chapter 2 Creation of Sample Program

This chapter explains basic functions such as programming and comment entry of a simple ladder by using CX-Programmer.

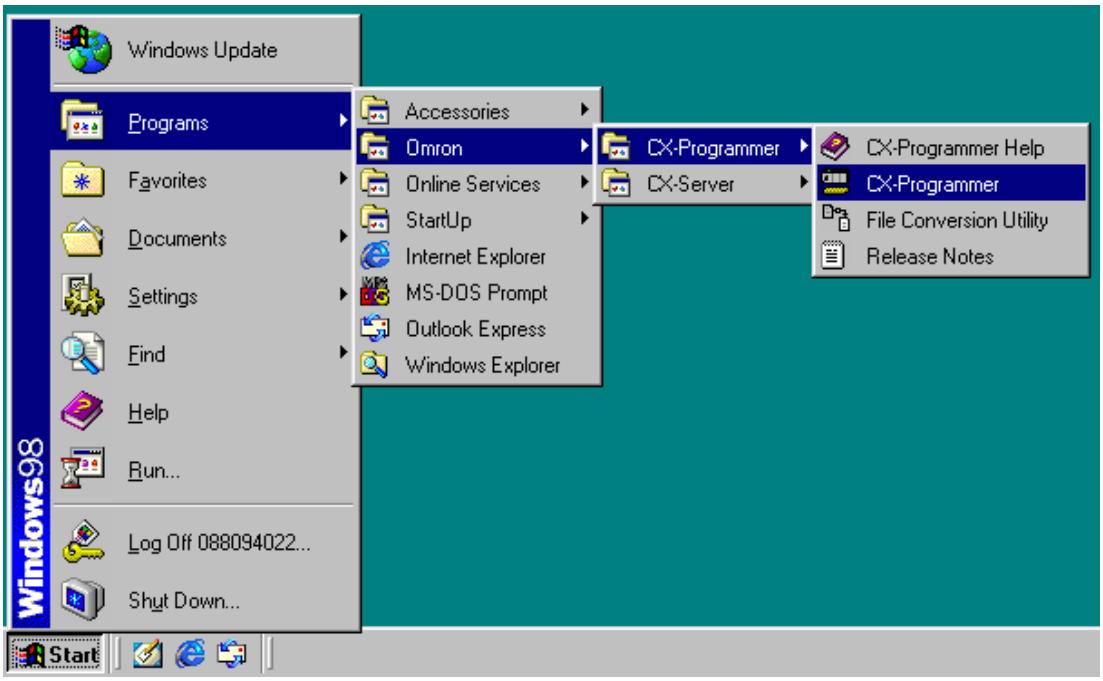
Here, a sample program "a program of car entry control by opening/closing shutters" is created as an example. This program is used to explain how to use the debug functions of CX-Simulator, which are mentioned after Chapter 3.

[Start]
↓
[Programs]
↓
[Omron]
↓
[CX-Programmer]
↓
[CX-Programmer]

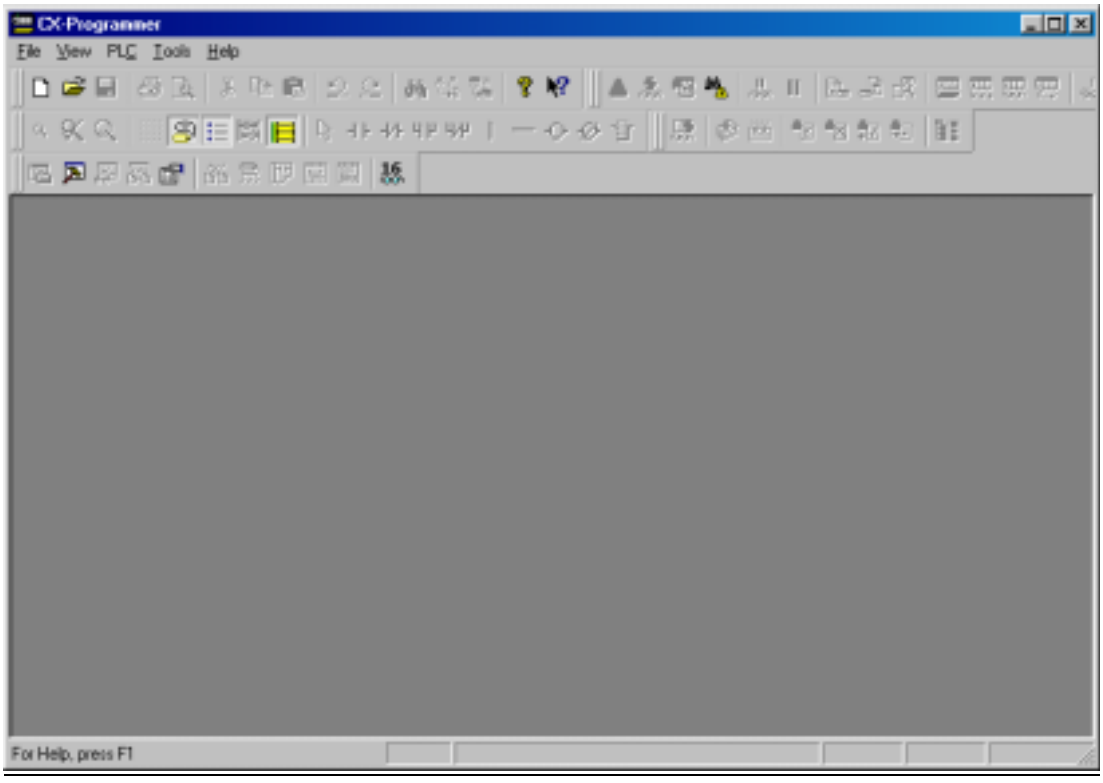
Or double-click the CX-Programmer icon.

2-1 Starting CX-Programmer

Start up CX-Programmer from Windows.



The initial screen of CX-Programmer shows up.

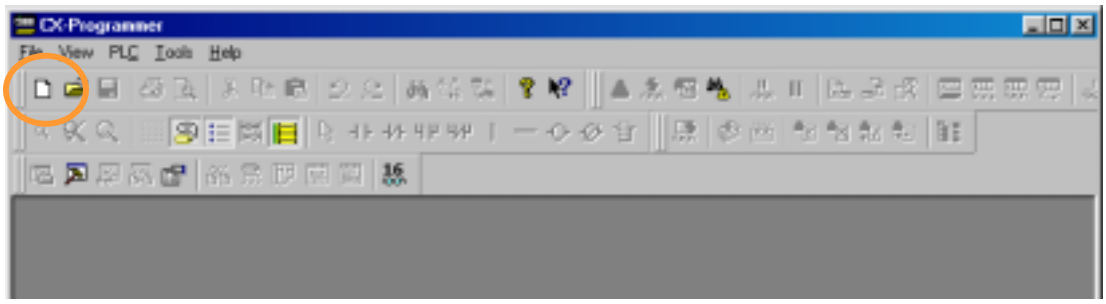


2-2 New Project Opening and Device Type Settings

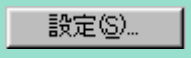
Click the toolbar button [New] in CX-Programmer.

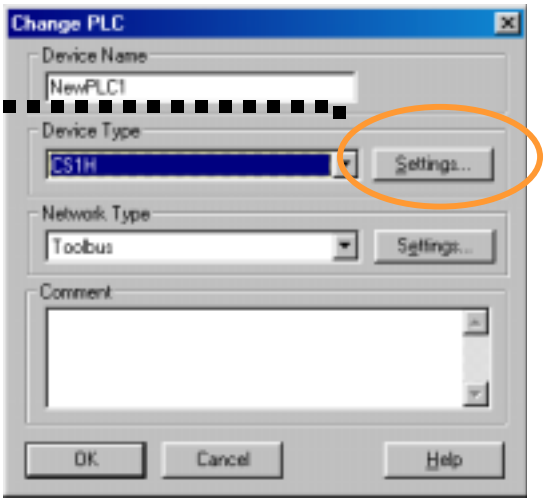
Set the device type to “CS1H” and the CPU type to “CPU63” in this example.

Click the left mouse button.



Click the left mouse button.





Click the left mouse button on the “Settings” button to show the [Device Type Settings] dialog.

The PLC type is set. In this example, set “CS1H”.

Click the left mouse button on the spin control button and select a CPU type.

The CPU type is set. In this example, set “CPU63”.

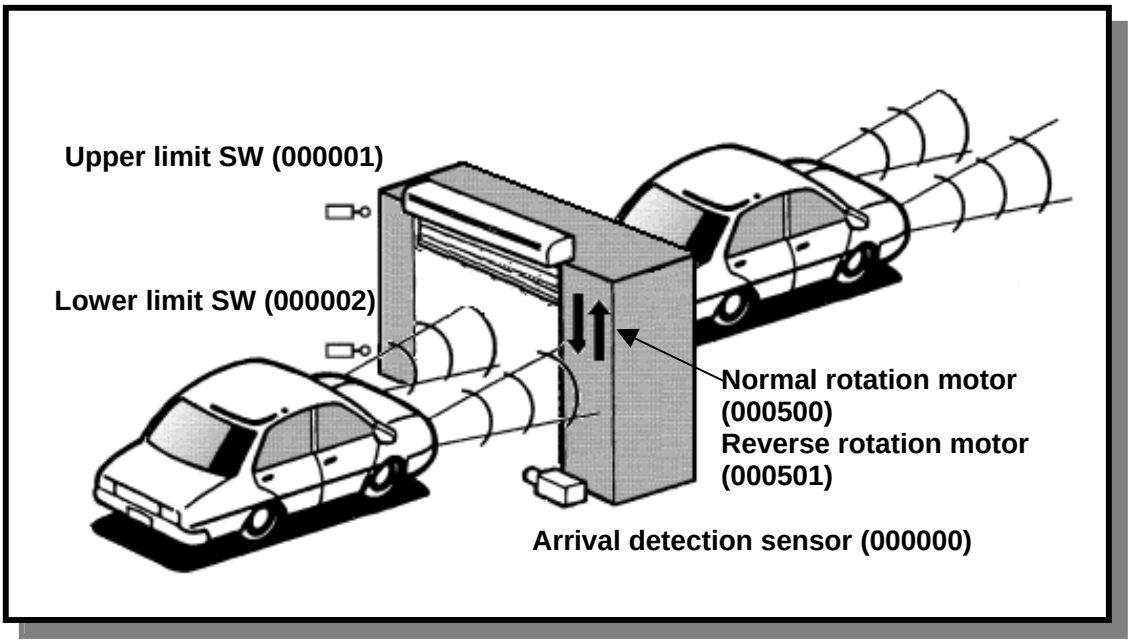
Here is the end of the setting. Now CX-Programmer is ready for the connection with CX-Simulator.

Click OK to decide the selected device type.

2-3 Creating Sample Program

The following is coding of a simple ladder program by using CX-Programmer.

This is a sample program of controlling car entry in a shed by opening/closing shutters. When a car approaches, the shutters automatically open; and in two seconds after the car passes the gate, the shutters close.



Operation Flow

(1) When a car enters, the arrival detection sensor (000000) is turned on.
(2) The normal rotation motor (000500) of the shutters is turned on.
(3) The shutters reach the upper limit in three seconds, the upper limit SW (000001) is turned on, and the shutters open.
(4) When the car passes the gate, the arrival detection sensor (000000) is turned off.
(5) In two seconds, the reverse rotation motor (000501) is turned on.
(6) In three seconds, the shutters reach the lower limit, the lower limit SW (000002) is turned on, and the shutters close.

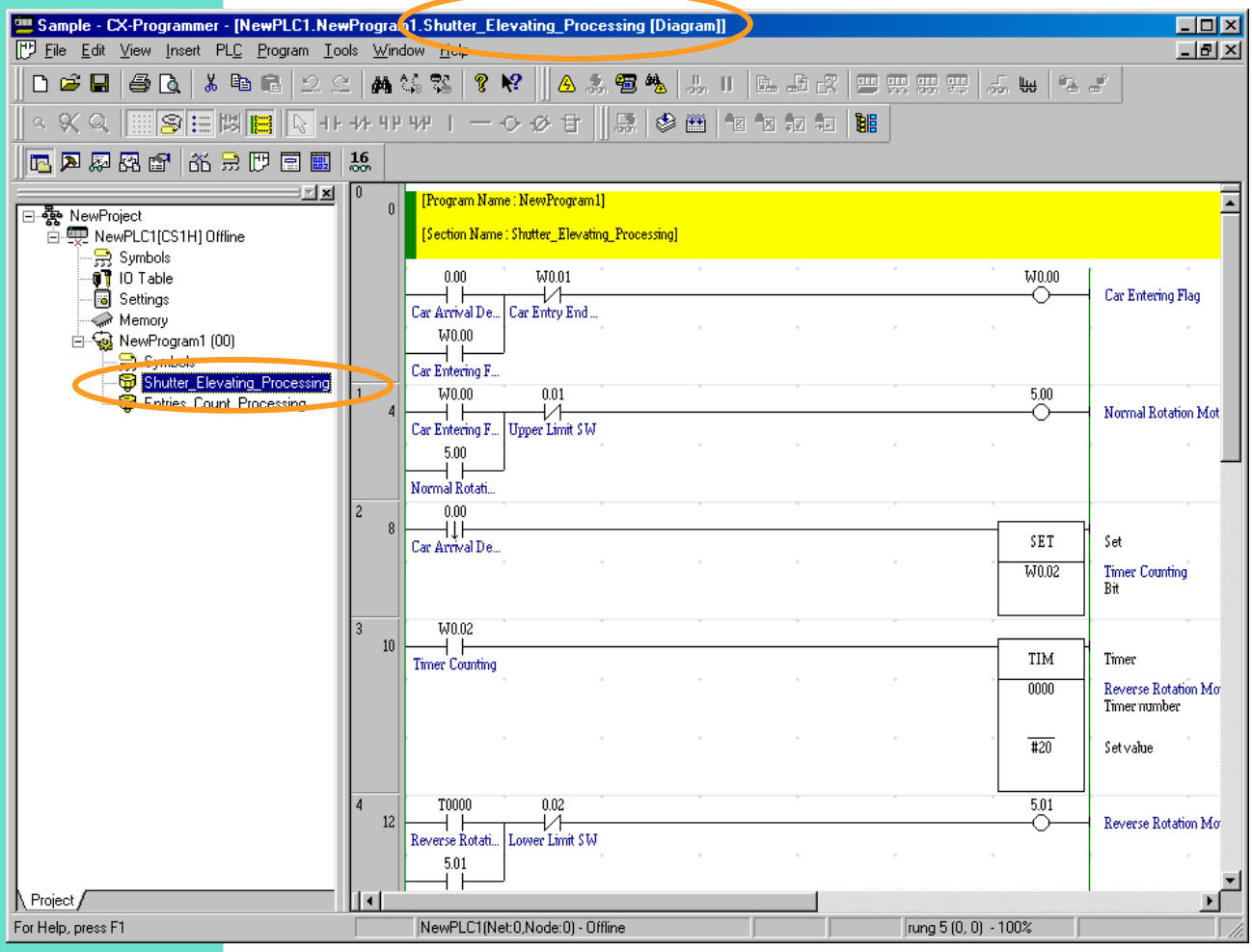
See CX-Programmer Introduction Guide or the operation manual of CX-Programmer for the detailed entry method of a ladder program.

Completed Sample Program

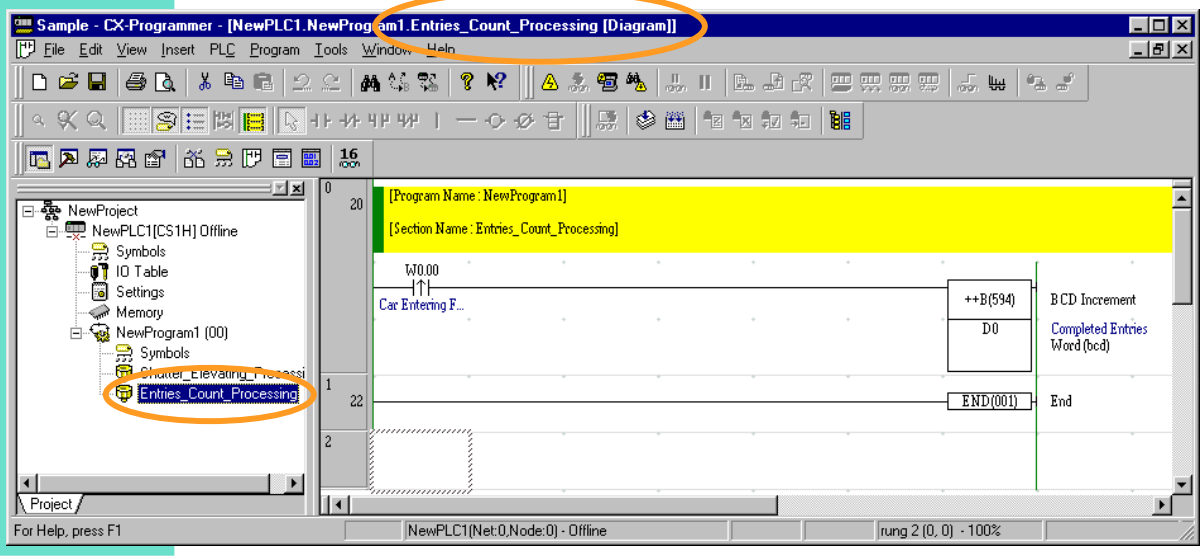
The program is modularized in two sections “Shutter elevating processing” and “Entries count processing” by the section function of CX-Programmer V2 or grater versions.

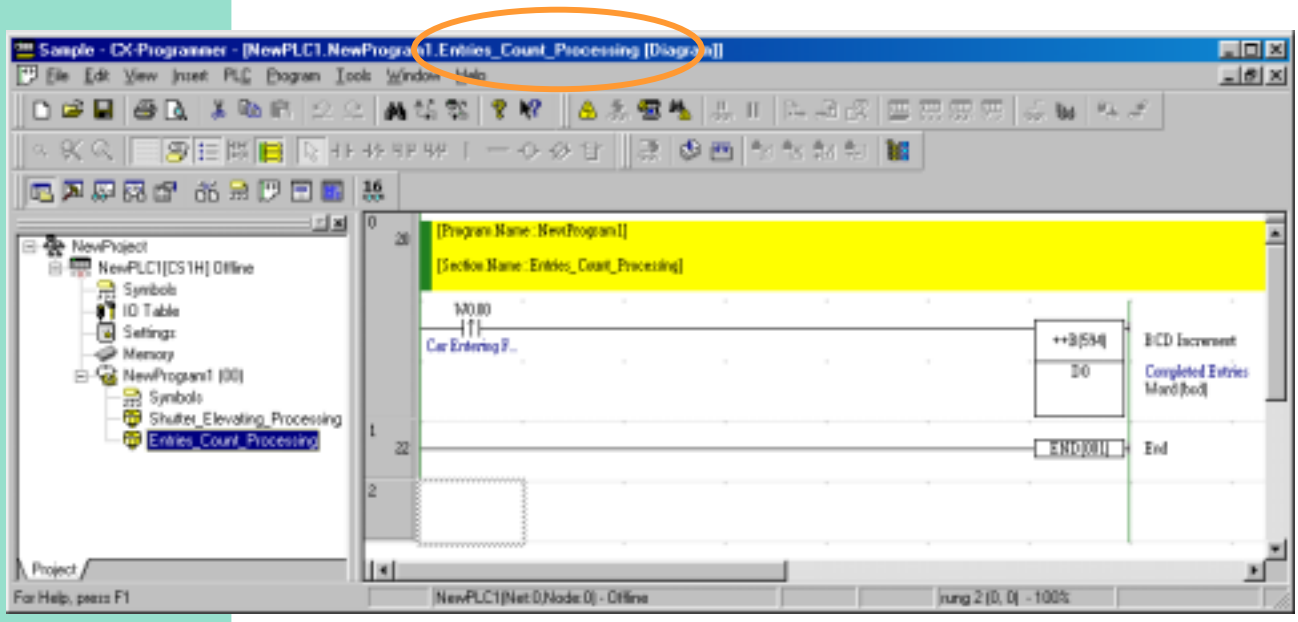
Since processing is simplified in the below program, it’s different from the actual program.

Shutter elevating processing



Entries count processing

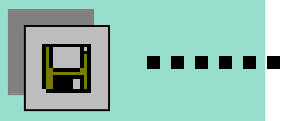




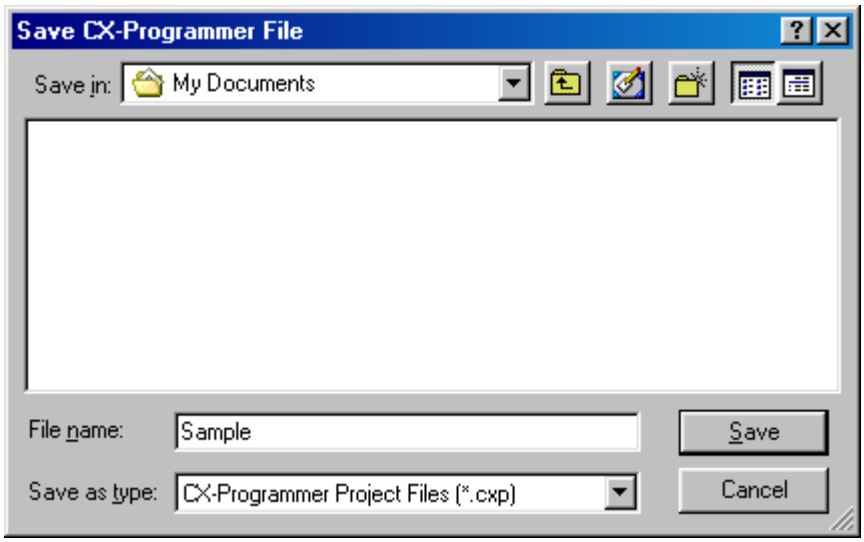
2-4 Saving Program

Save the created program.

Here, save the program as “sample”.



Or select [File] | [Save as] from the menu.



The program is saved.

This sample program is used in the following operations to explain CX-Simulator. Keep CX-Programmer open.

Chapter 3
Executing Program by CX-Simulator

This chapter explains how to operate CX-Programmer functions such as program transfer, PLC mode change, cycle time check, and ladder program monitoring, when CX-Programmer is connected to a CX-Simulator virtual PLC.

In addition, you can use CX-Programmer when connecting to a virtual PLC by the exactly same operation as when connecting to an actual PLC. For detailed operations, see the operation manual of CX-Programmer.

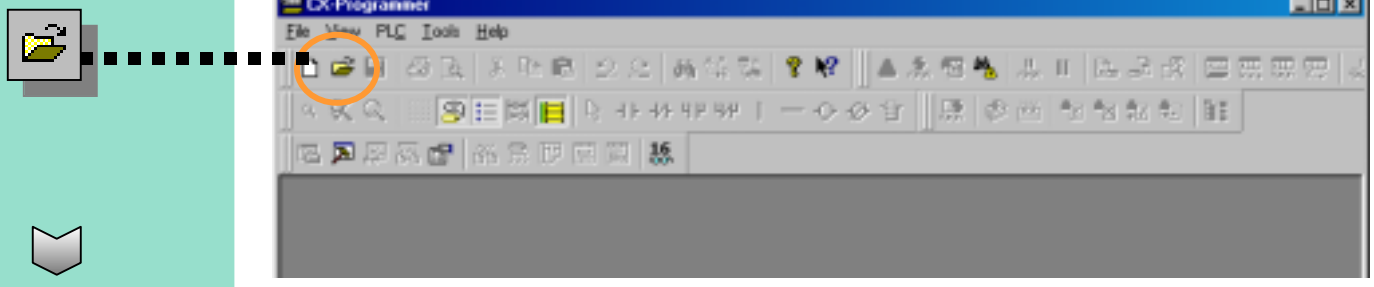
3-1 Connection of CX-Simulator Virtual PLC and CX-Programmer

Connect CX-Programmer with a CX-Simulator virtual PLC.

The sample program created in Chapter 2 is used for the explanation here.

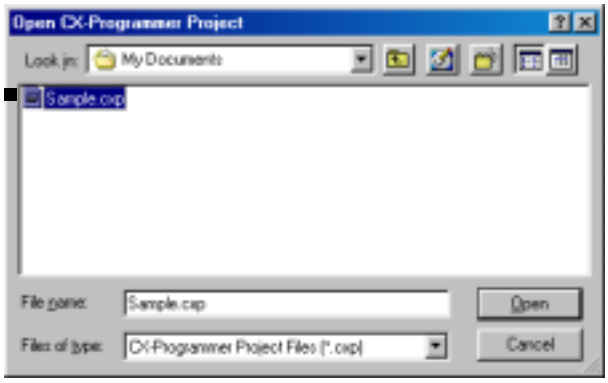
Starting CX-Programmer

Click the [Open] button from the toolbar of CX-Programmer.

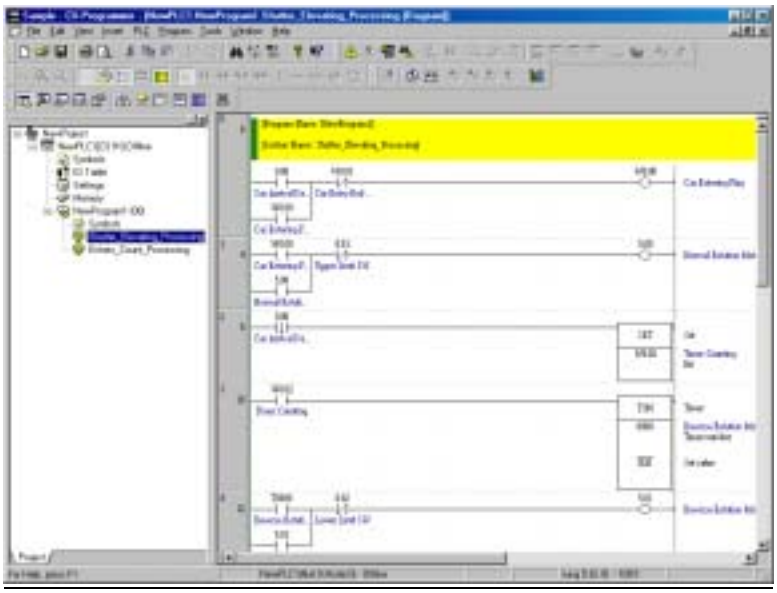


Select the sample program created in Chapter 2.

Select “Sample”.



Open

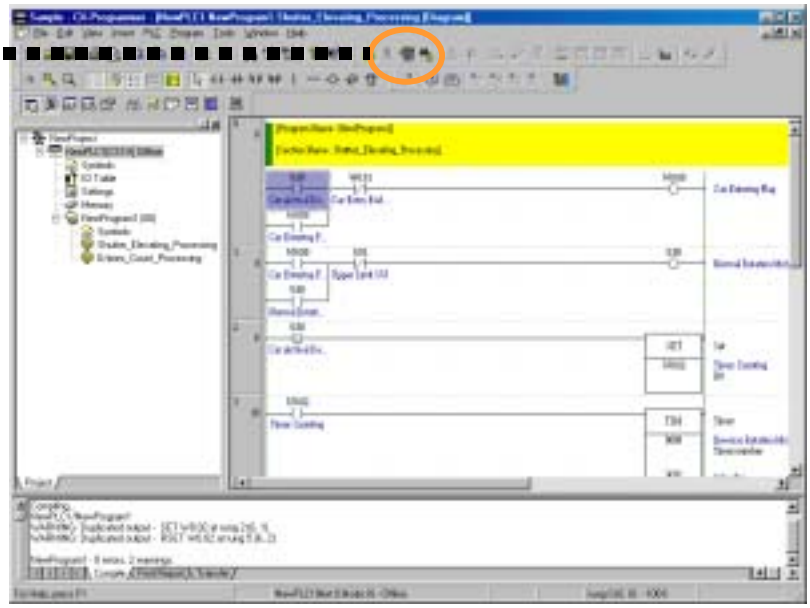


The sample program is loaded.

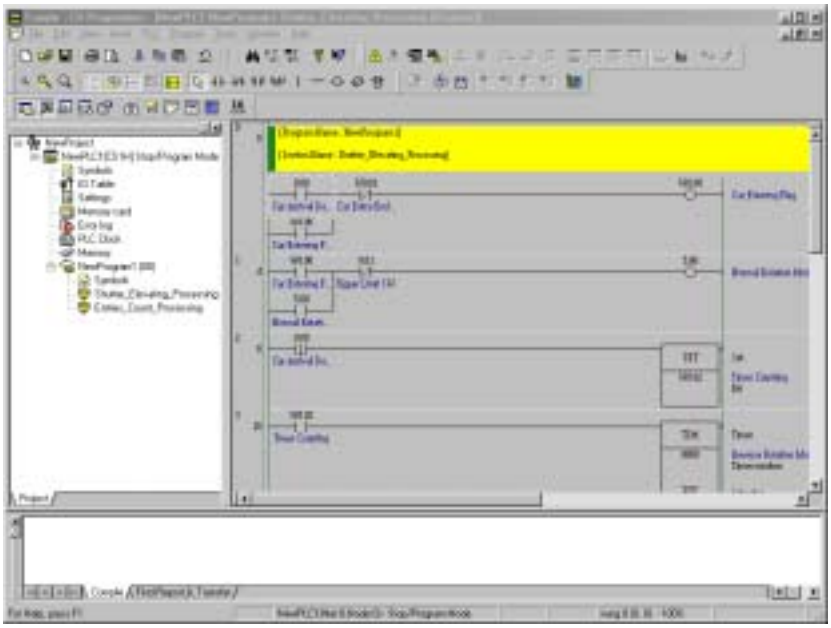
3-2 Program Transfer to Virtual PLC

Click the [Work Online Simulator] button.

Or select [PLC] | [Work Online Simulator] from the menu.



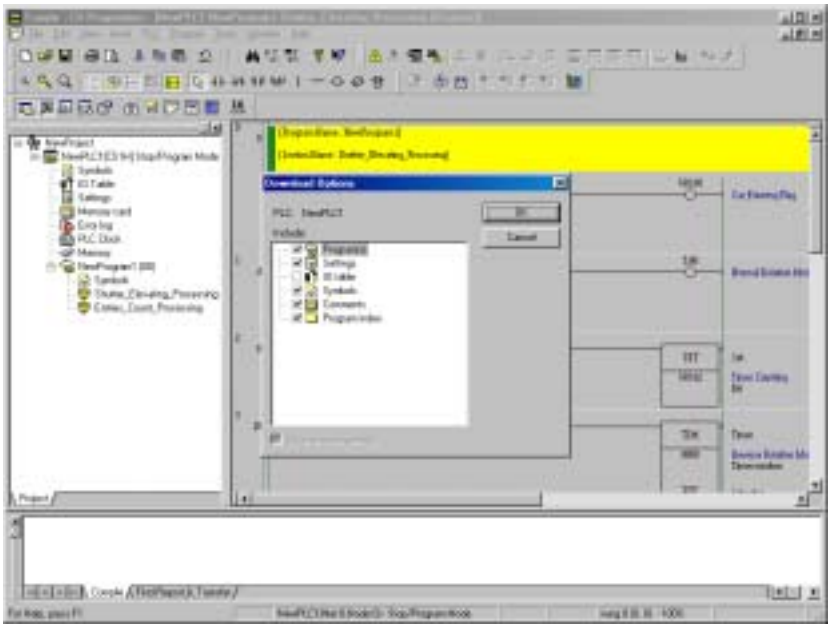
The background color of Ladder Window changes to gray and online connection is established.



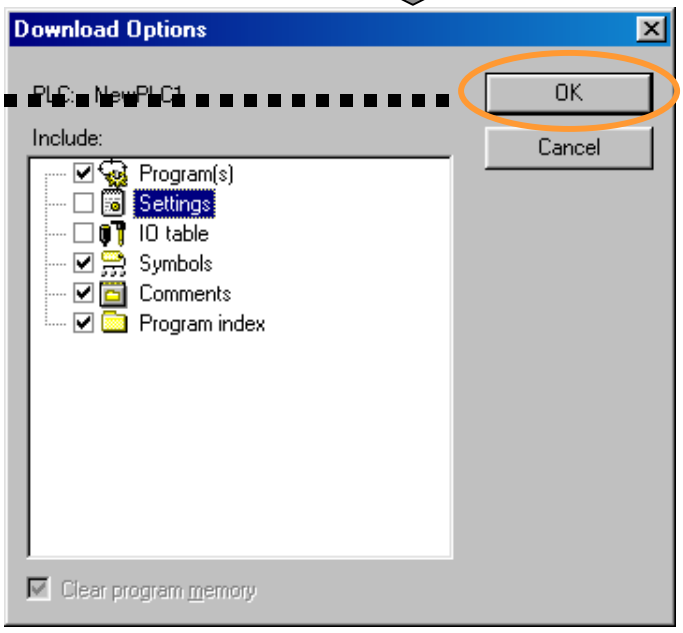
Now a CX-Simulator virtual PLC and CX-Programmer have been connected.

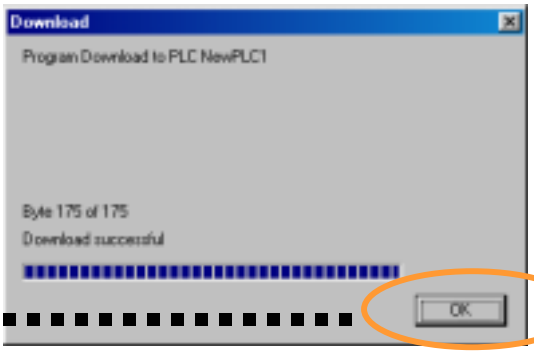
The Simulator Online function automatically shows the Transfer dialog after this and enables you to transfer the ladder program to the virtual PLC; however, you can change it to the setting of connection only in the option settings of CX-Programmer.

The Transfer dialog is automatically displayed.



OK





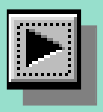
OK

3-3 Operating Mode Change of Virtual PLC

You can change the operating mode of a virtual PLC from the [CX-Simulator Debug Console] screen or CX-Programmer.

Operation from CX-Simulator Debug Console Tool

To start running a program (Monitor mode)



(Run: Monitor mode)

To stop running a program (Program mode)

(Stop: Program mode)

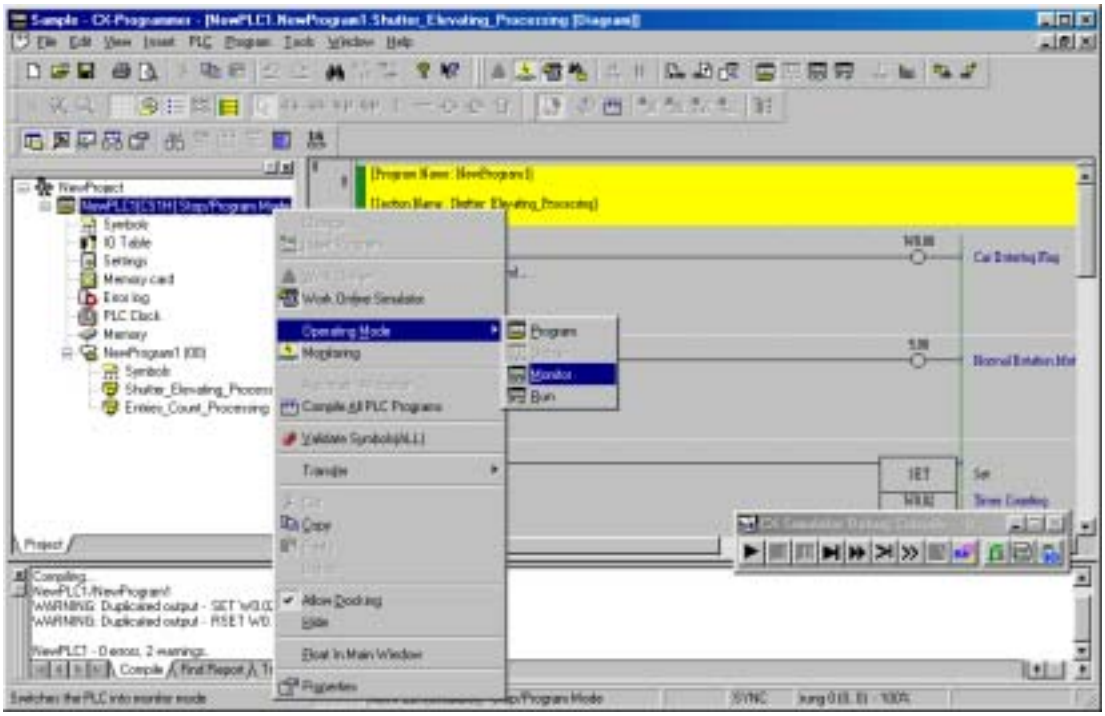
The number shown in the title bar of Debug Console Tool indicates the count number of the present cycle in the virtual PLC.

Operation in CX-Programmer

Click the right mouse button on [NewPLC1] and select [Operating Mode].

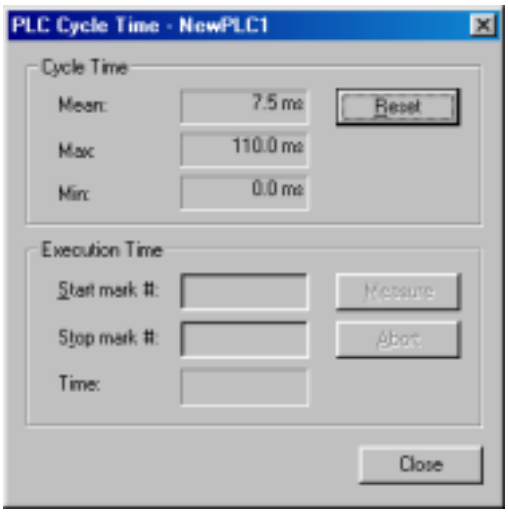
Or

Select [PLC] | [Operating Mode] from the menu.



3-4 Cycle Time Check in Virtual PLC

Change the operating mode of a virtual PLC to the Monitor mode.

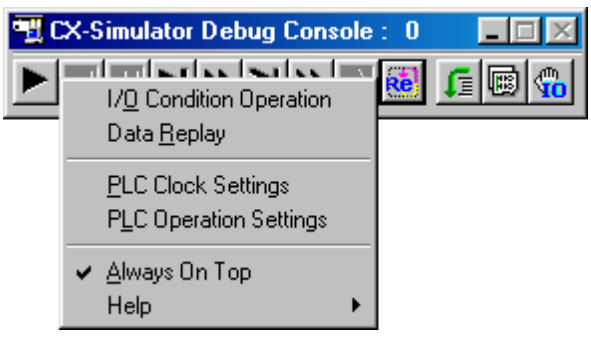


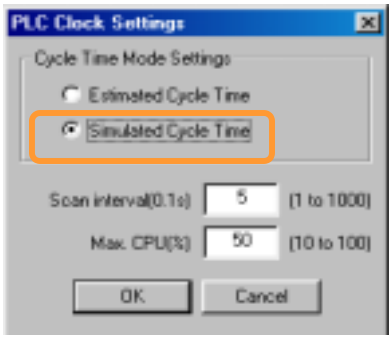
Select [PLC] | [Edit] | [Cycle Time] from the menu of CX-Programmer.

Estimated Cycle Time and Simulated Cycle Time

Set Estimated Cycle Time when you want to simulate with accuracy, for instance, when you want to operate data traced per scan as input. At this point, the time of one scan is enlarged, so the present values of instructions such as TIM proceed depending on your PC performance.

On the other hand, set Simulated Cycle Time, the accuracy is inferior to Estimated Cycle Time though, when you want to check the exterior operation of instructions such as TIM.



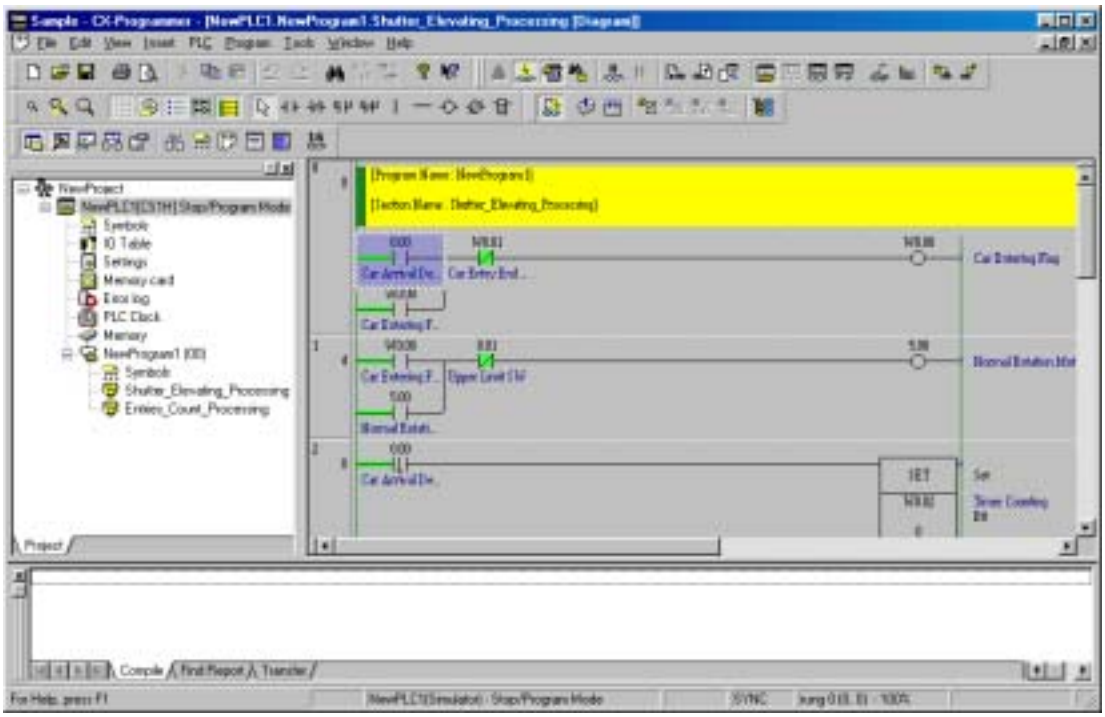


The default cycle time information in CX-Simulator is set to “Simulated Cycle Time” which is the time for running a program on an active PC. To check the estimated cycle time when running a program on an actual PLC, change “Cycle Time Mode Settings” from “Simulated Cycle Time” to “Estimated Cycle Time” in the PLC Clock Settings of CX-Simulator.

Set “Cycle Time Mode Settings” to “”Simulated Cycle Time” previous to the following operations.

3-5 Ladder Monitoring in Virtual PLC

Start monitoring of ladders in CX-Programmer.



Besides ladder monitoring, debug operations such as force on/of, differential monitoring, data trace and online edit are enabled on CX-Programmer as is the case with connecting to an actual PLC.

Chapter 4
Program Debug by CX-Simulator

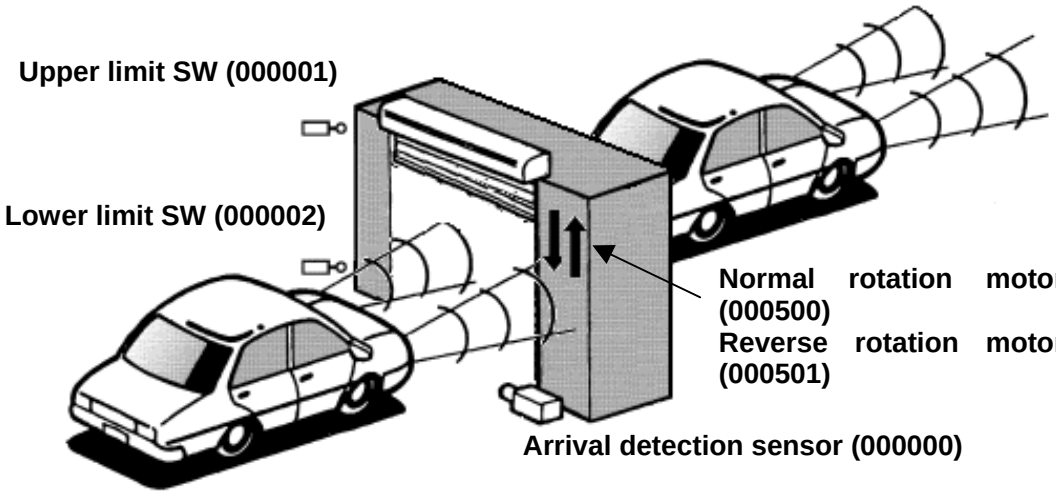
The debug operations of a ladder program by using the original functions of CX-Simulator, which are unavailable by actual PLCs, are explained in this chapter.

4-1 Debug with Sample Program

CX-Simulator has various program-debug functions. Not only ladder program execution by a virtual PLC instead of an actual PLC, but also the ON operation of a limit switch after a work moves **(I/O condition operation settings)**, the stop operation of program execution when a memory value becomes a designated value **(I/O break condition settings)**, program execution of optional range **(Start point/Break point settings)**, etc.

This section explains each function by using the sample program "Car entry control program by opening/closing a shutter" created in Chapter 2.

The below figure shows the image and operation flow of the sample program in Chapter 2:



Operation Flow

(1) When a car enters, the arrival detection sensor (000000) is turned on.
(2) The normal rotation motor (000500) of the shutter is turned on.
(3) **The shutter reaches the upper limit in three seconds**, the upper limit SW (000001) is turned on, and the shutter opens.
(4) When the car passes the gate, the arrival detection sensor (000000) is turned off.
(5) In two seconds, the reverse rotation motor (000501) is turned on.
(6) **In three seconds, the shutter reaches the lower limit**, the lower limit SW (000002) is turned on, and the shutter closes.

In the I/O condition operation settings, you can set the ON operations of the upper/lower limit switches by shutter elevating operations and the time until the limit switches are turned on, as virtual external input. In this example, the virtual external input that turns on the limit switches of the shutter in three seconds after the motors are turned on is designated.

4-2 Debug by Virtual External Input

CX-Simulator allows you to set external input; for example, the upper limit SW is turned on in three seconds after the shutter goes up (ON operation of the upper limit switch). How to set the I/O Condition tool of CX-Simulator is explained below.

4-2-1 Settings of Virtual External Input

In this example, the operations of two external input signals are set.

(1) The upper limit SW (000001) is turned on in three seconds after the normal rotation motor (000500) is turned on. (At this point, the lower limit SW (000002) is off.)

(2) The lower limit SW (000002) is turned on in three seconds after the reverse rotation motor (000501) is turned on. (At this point, the upper limit SW (000001) is actually off.)

No.	Run	Condition	Delay(ms)	Output
0	D	5.0=ON	3000	0.1=ON and 0.2=OFF
1	D	5.1=ON	3000	0.2=ON and 0.1=OFF

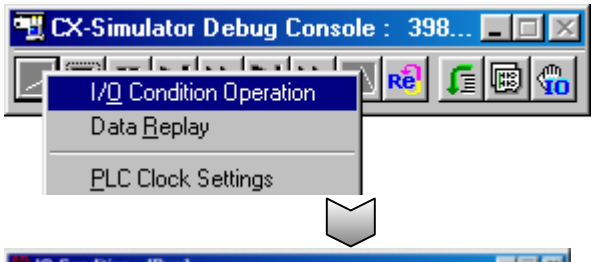
Setting in I/O Condition Tool

The following section explains how to set virtual external input by using I/O Condition Tool:

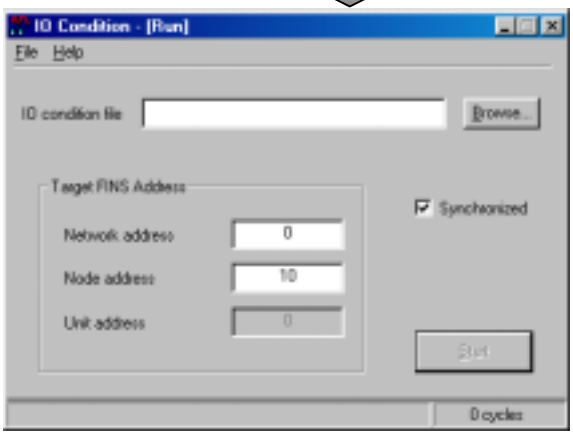
4-2-2 Starting I/O Condition Tool

Select [Replay] | [IO Condition] from the menu in the Debug Console screen.

Select [I/O Condition] from the Debug Console screen.

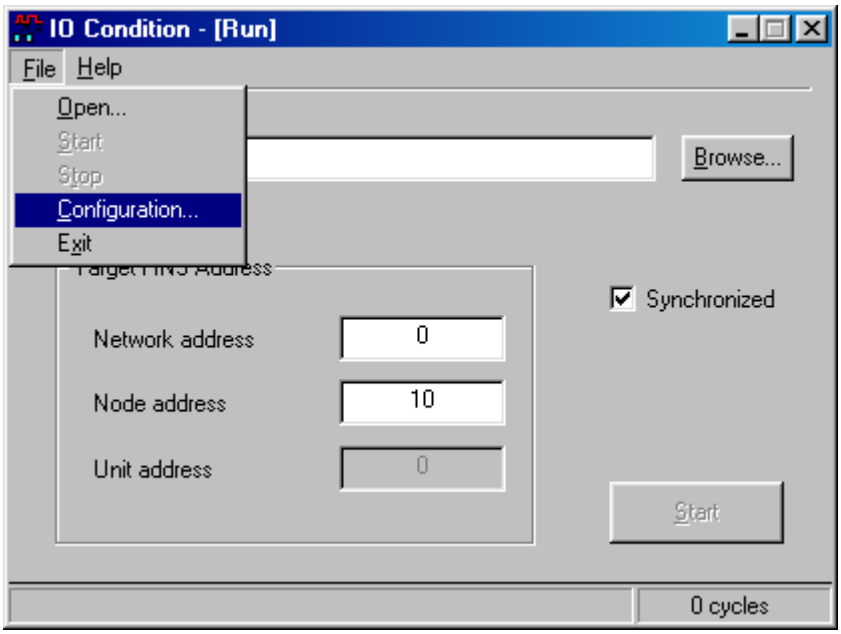


I/O Condition Tool starts up.

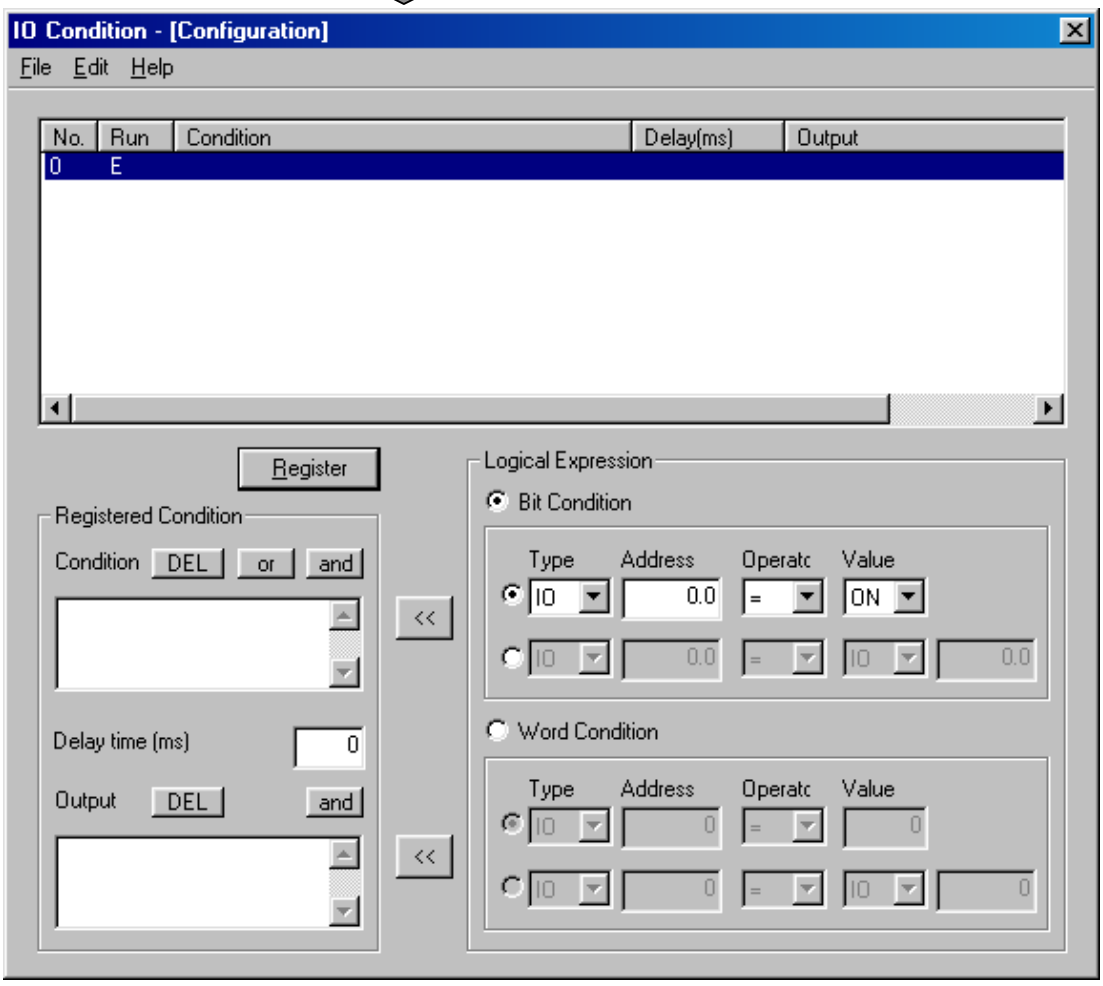


Select [File] | [Configuration] from the menu in the I/O Condition tool.

Select [File] | [Configuration] in the I/O Condition tool.



The [Configuration] screen of the I/O Condition tool is displayed.



4-2-3 Setting I/O Conditions

How to set I/O conditions is explained in the following example:

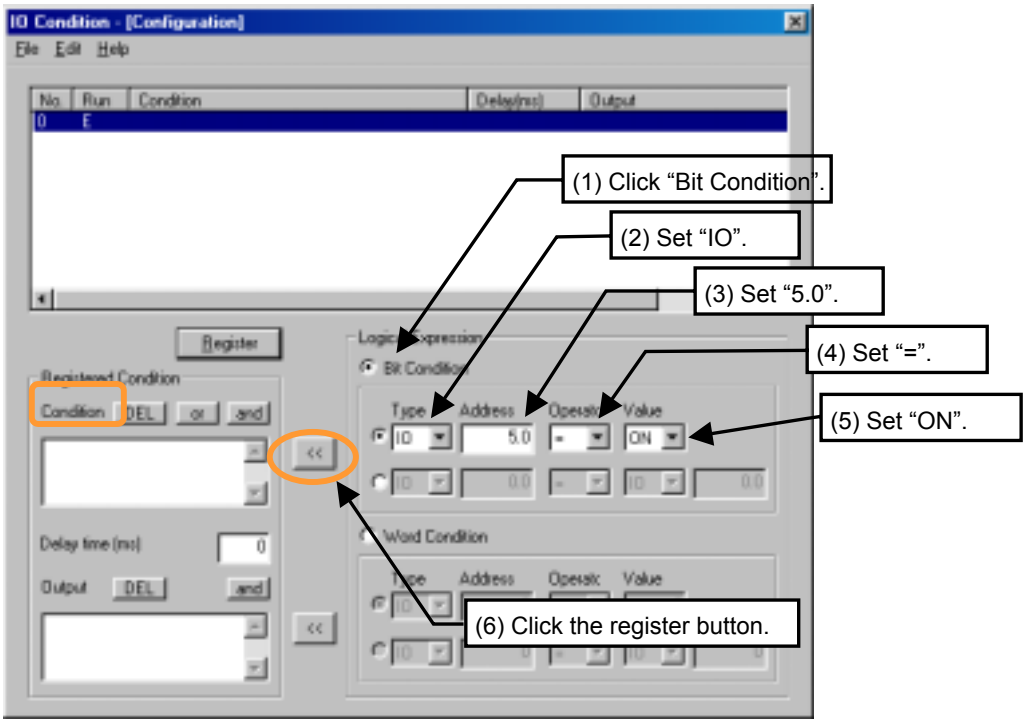
In three seconds after the normal rotation motor (000500) is turned on, the upper limit SW (000001) is turned on. (At this point, the lower limit SW (000002) is actually turned off at the same time.)

In this example, the following formulas are set:

- Condition: The normal rotation motor (000500) is turned on.
- Delay time: 3 seconds (3000ms)
- Output: The lower limit SW (000002) is turned off. In three seconds, the upper limit SW (000001) is turned on.

Condition Settings

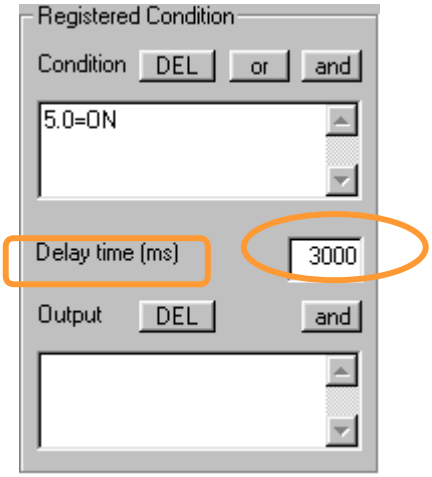
(1) Click "Bit Condition" in the "Logical Expression" field.
(2) Set "IO" to "Type".
(3) Set "5.0" to "Address".
(4) Set "=" to "Operator".
(5) Set "ON" to "Value".
(6) Click the register button << .



Delay Time Setting

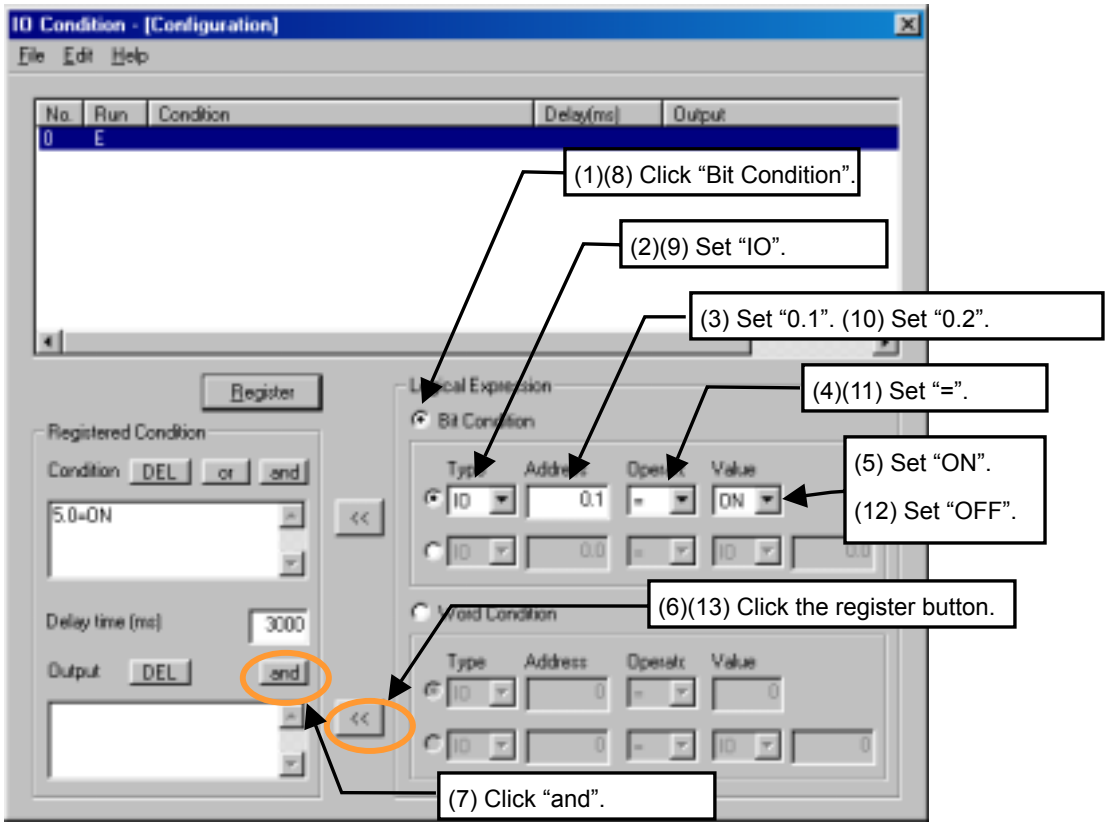
Enter 3000 (3 seconds) in the text box of "Delay time".

Enter 3000(ms) (=3 seconds) in the text box of "Delay time".

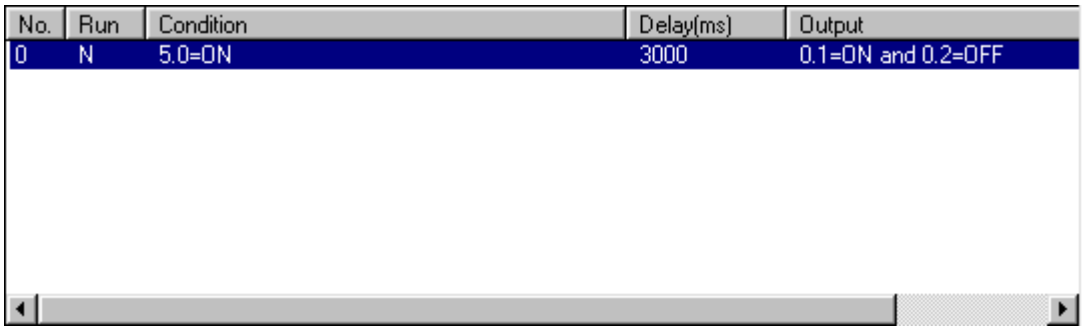


Output Settings

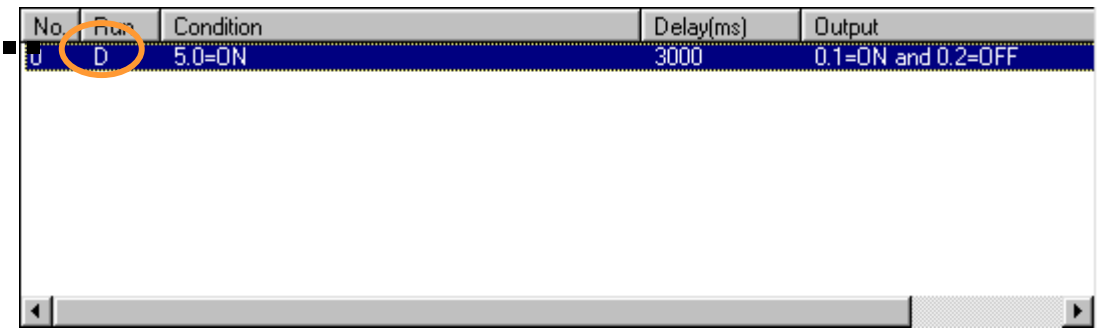
(1) Click “Bit Condition” in the “Logical Expression” field.
(2) Set “IO” to “Type”
(3) Set “0.1” to “Address”.
(4) Set “=” to “Operator”.
(5) Set “ON” to “Value”.
(6) Set the register button <<.
(7) Click the and button.
(8) Click “Bit Condition” in the “Logical Expression” field.
(9) Set “IO” to “Type”.
(10) Set “0.2” to “Address”.
(11) Set “=” to “Operator”.
(12) Set “OFF” to “Value”.
(13) Click the register button <<.



Register ■■■■■■ Click the “Register” button.



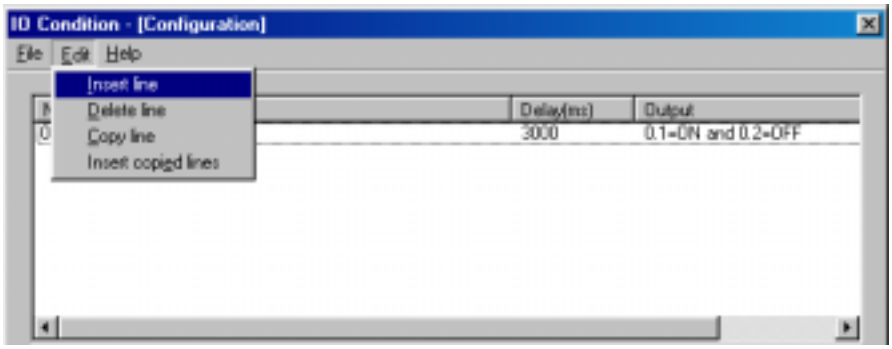
Double-click the registered formula ■■■■■■

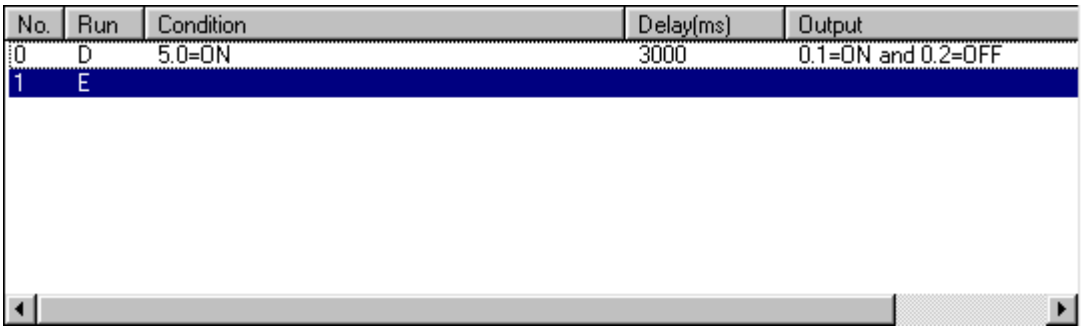


The “Run” column is changed from “N” to “D”. It means this formula will be executed.

Select [Edit] | [Insert line] from the menu in "I/O Condition – [Configuration]".

Add a new line to enter the second formula.





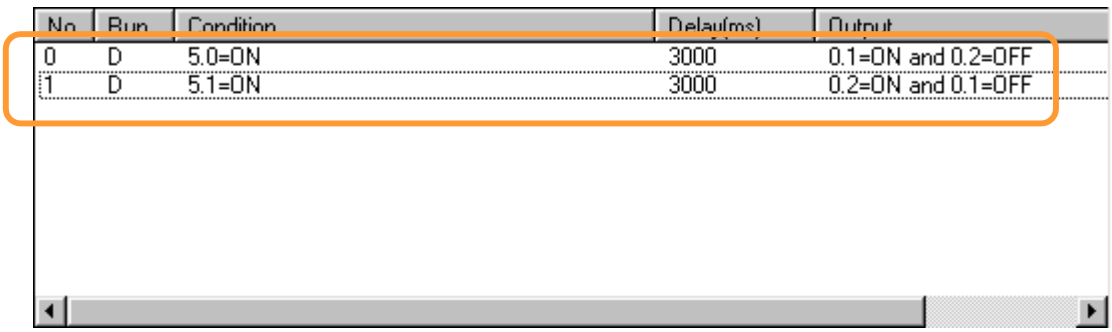
Like the registration operation of the first formula, set the followings.

The lower limit SW (000002) is turned on three seconds after the reverse rotation motor (000501) is turned on. (At this point, the upper limit SW (0000001) is actually turned off at the same time.)

In this example, the following formulas are set:

- Condition: The reverse rotation motor (000501) is turned on.
- Delay time: 3 seconds (3000ms)
- Output: The upper limit SW (000001) is turned off. In three seconds, the upper limit SW (000002) is turned on.

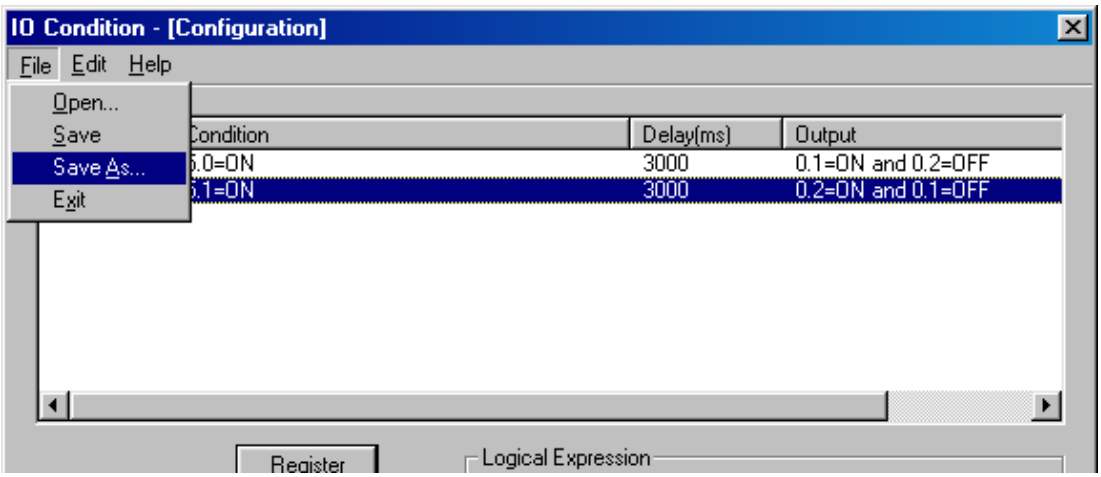
After the two formulas are registered, the set screen is displayed as follows:



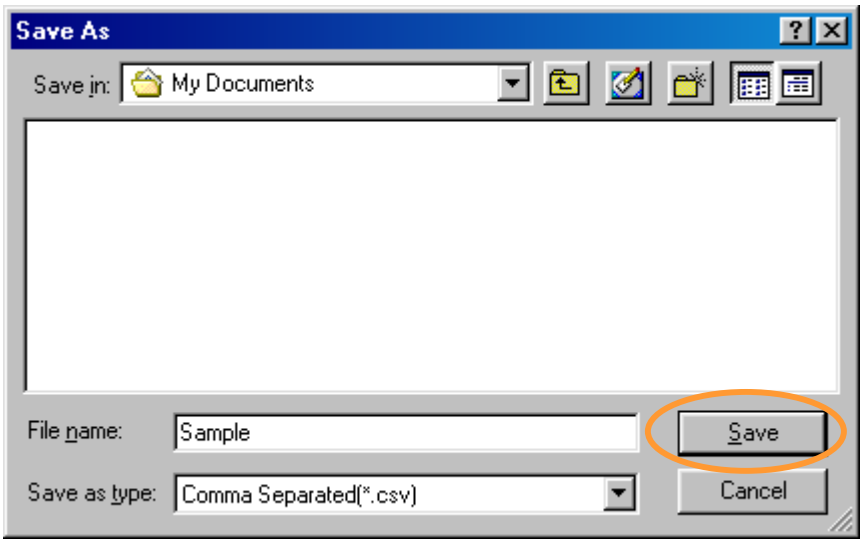
You can enter the conditions and output formulas in each column directly from the keyboard.

4-2-4 Saving and Ending I/O Condition Settings

Select [File] | [Save As] from the menu in "I/O Condition – [Configuration]".



Enter "sample" in "File name".



Select [File] | [Exit] from the menu in "I/O Condition – [Configuration]".

End I/O Condition [Configuration].

Prepare the external input debug by I/O Condition Configuration.

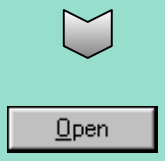
4-2-5 Debug by Virtual External Input

Execute the followings before starting debug:

- Load the sample program (Sample.cxp) created in Chapter 2 in CX-Programmer).
- Start the virtual PLC created in Chapter 3.
- Follow the procedure in Chapter 4 to connect CX-Programmer and the CX-Simulator virtual PLC.
- Start running the ladder of the virtual PLC.

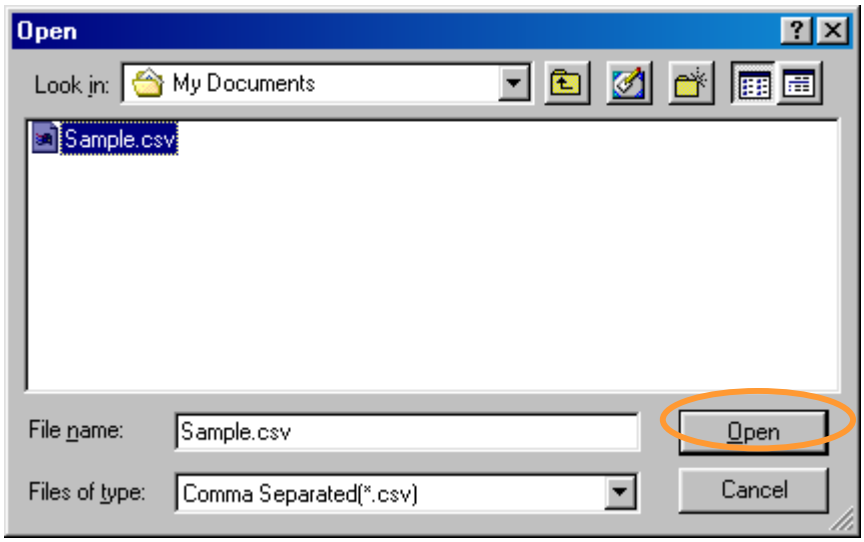
Now you are ready to start debugging.

Select [File] | [Open] from the menu in "I/O Configuration – [Run] " and select "Sample.csv".

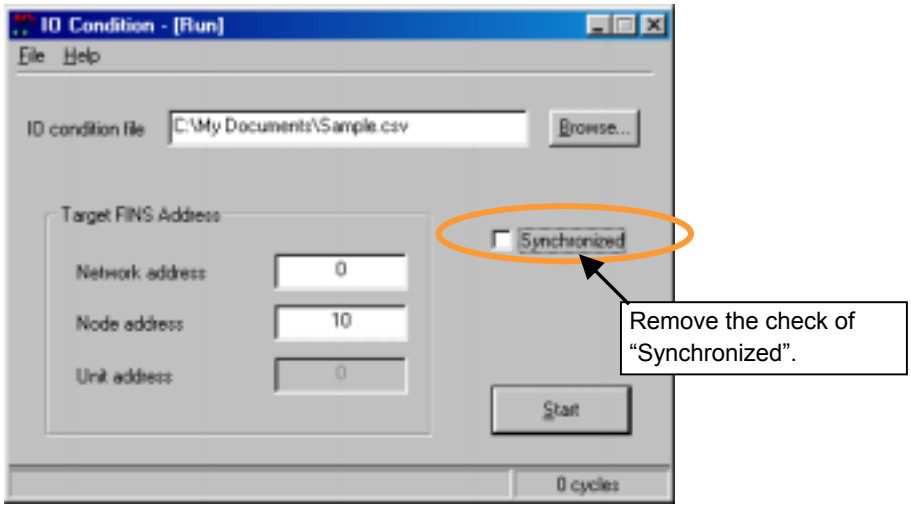


Execution of I/O Condition Tool

Load the file (Sample.csv) set in "I/O Condition [Configuration]".

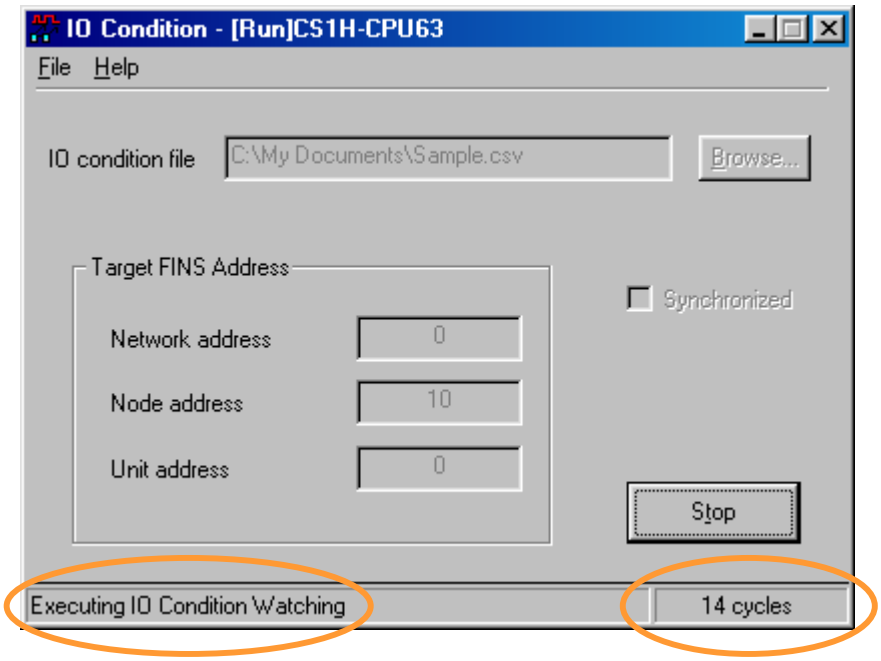


Click the check box of "Synchronized" to deactivate it.



Start

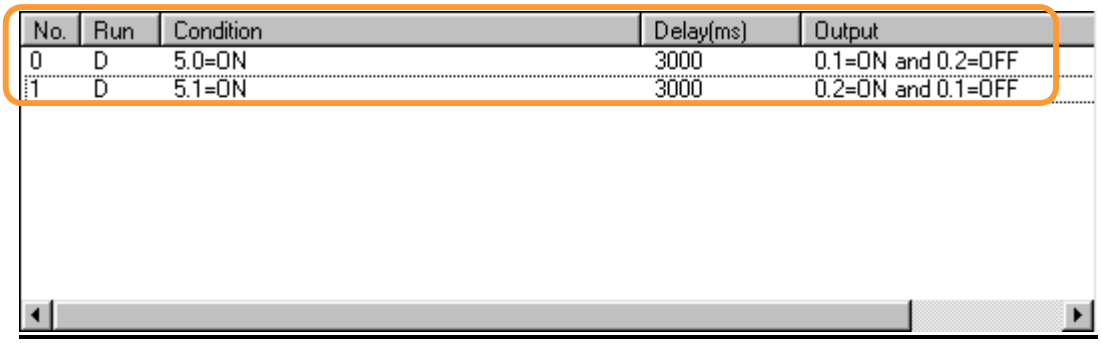
Click the [Start] button. I/O condition watching is started to check whether the set conditions are established.



After I/O condition watching is started, the screen shows the message ”Executing IO Condition Watching” and the number of cycles gradually increases.

If the ladder of a virtual PLC has not started running yet, the screen shows the message as follows. In this case, start running the ladder of the virtual PLC.

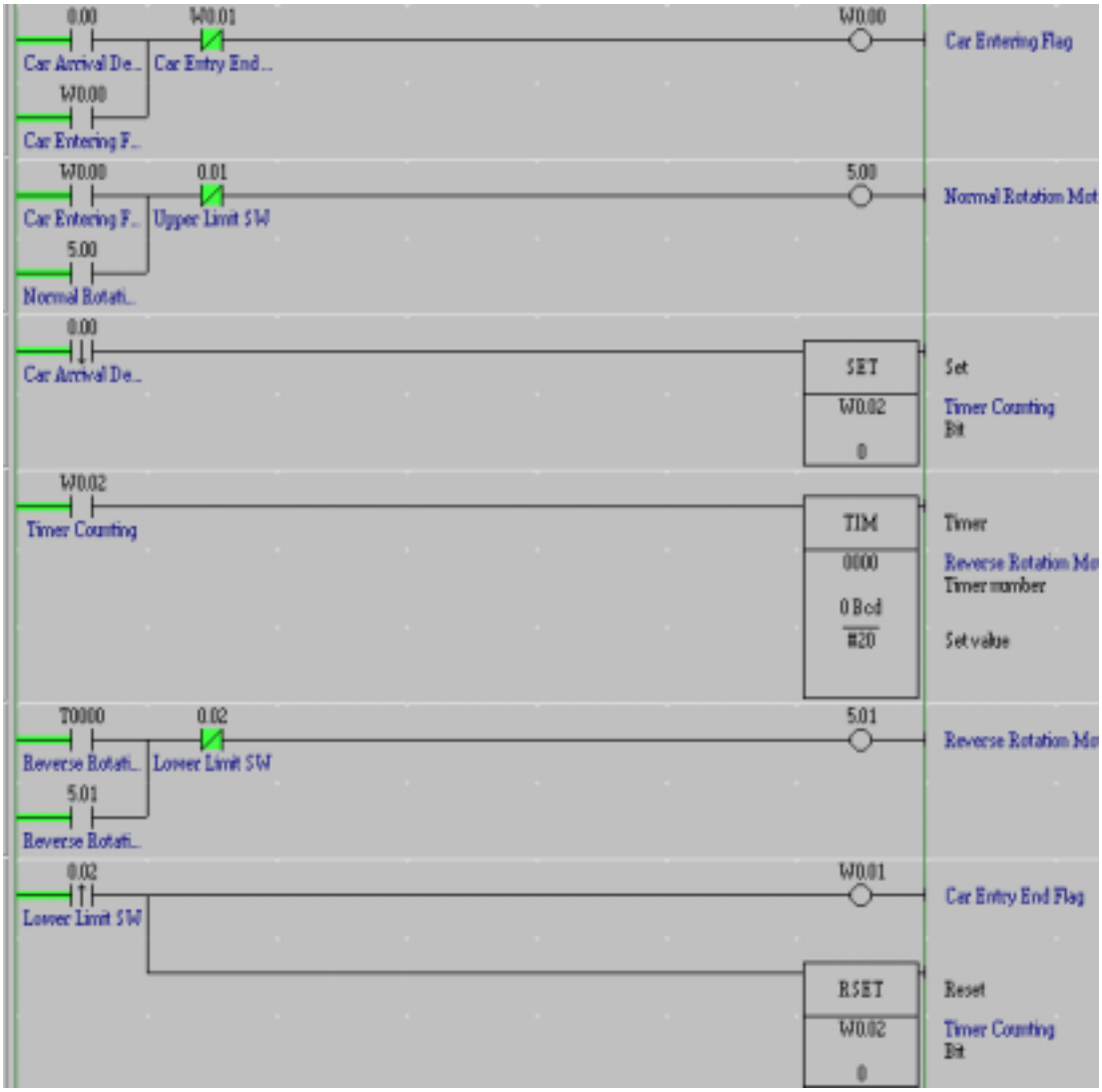
By the above procedure, the following two formulas are established and the output operation after the establishment is started.



No.	Run	Condition	Delay(ms)	Output
0	D	5.0=ON	3000	0.1=ON and 0.2=OFF
1	D	5.1=ON	3000	0.2=ON and 0.1=OFF

Show “Shutter Elevating Processing” of the ladder of CX-Programmer and start monitoring.

Or select [PLC] | [Monitor] | [Monitoring] from the menu in CX-Programmer.

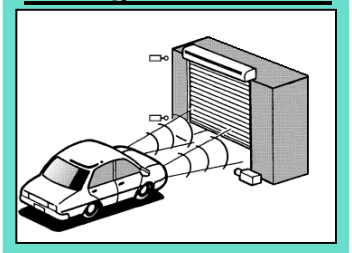


By the procedures so far, the ladder is running and the condition set in IO Condition Configuration is monitored by the virtual PLC.
After this, pseudo input of car arrival and gate passing is executed and debug of a ladder program is executed.

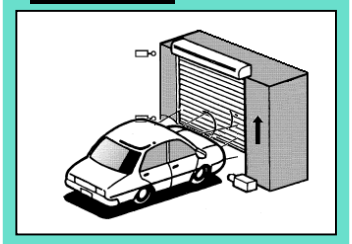
4-2-6 Program Debug Operation Using Virtual External Input

Check the following processing flow in order.

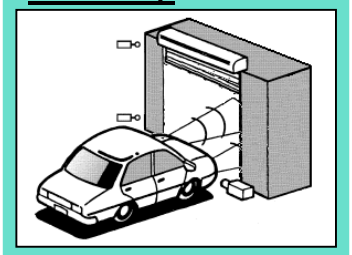
Waiting for car arrival



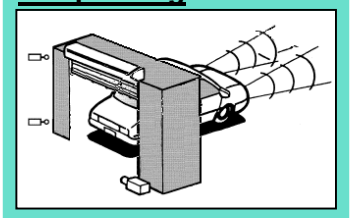
Car arrival



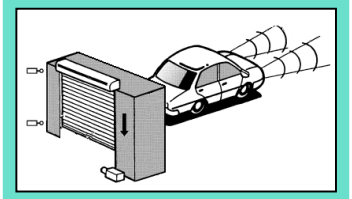
Shutter up



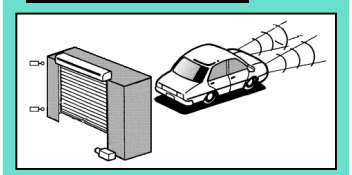
Car passing



Shutter down



End of car entry



(1) Waiting for car arrival

↓

(2) Car arrival

The arrival detection sensor (000000) is set from CX-Programmer.

↓

(3) Start of shutter up

↓

(4) After 3 seconds

↓

(5) End of shutter up

The normal rotation motor (000500) is turned on by the conditions and output formulas set in IO Condition Configuration, and in 3 seconds (after the shutter is completely pulled up), the upper limit SW is turned on.

↓

(6) Car passing

The arrival detection sensor (000000) is reset from CX-Programmer.

↓

(7) After 2 seconds

The ladder program counts 2 seconds after car passing.

↓

(8) Start of shutter down

↓

(9) After 3 seconds

↓

(10) End of shutter down

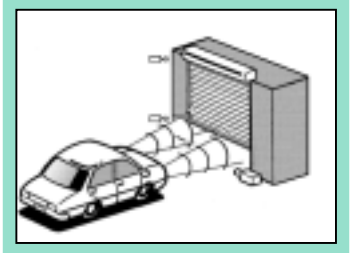
The reverse rotation motor (000501) is turned on by the conditions and output formulas set in IO Condition Configuration, and in 3 seconds (after the shutter is completely brought down), the lower limit SW is turned on.

↓

(11) End of car entry

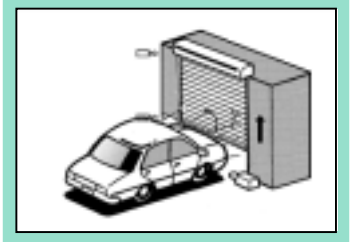
The ladder program counts the number of car entries.

Waiting for car arrival



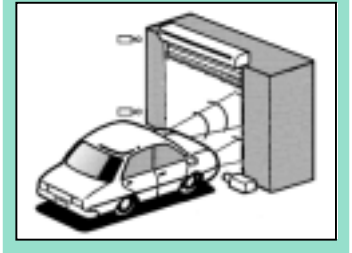
After clicking the contact 0.00, click the right button of the mouse to select [Set] | [On].

Car arrival



Setting on the arrival detection sensor (0.00) turns on the normal rotation motor (5.00).

Shutter up

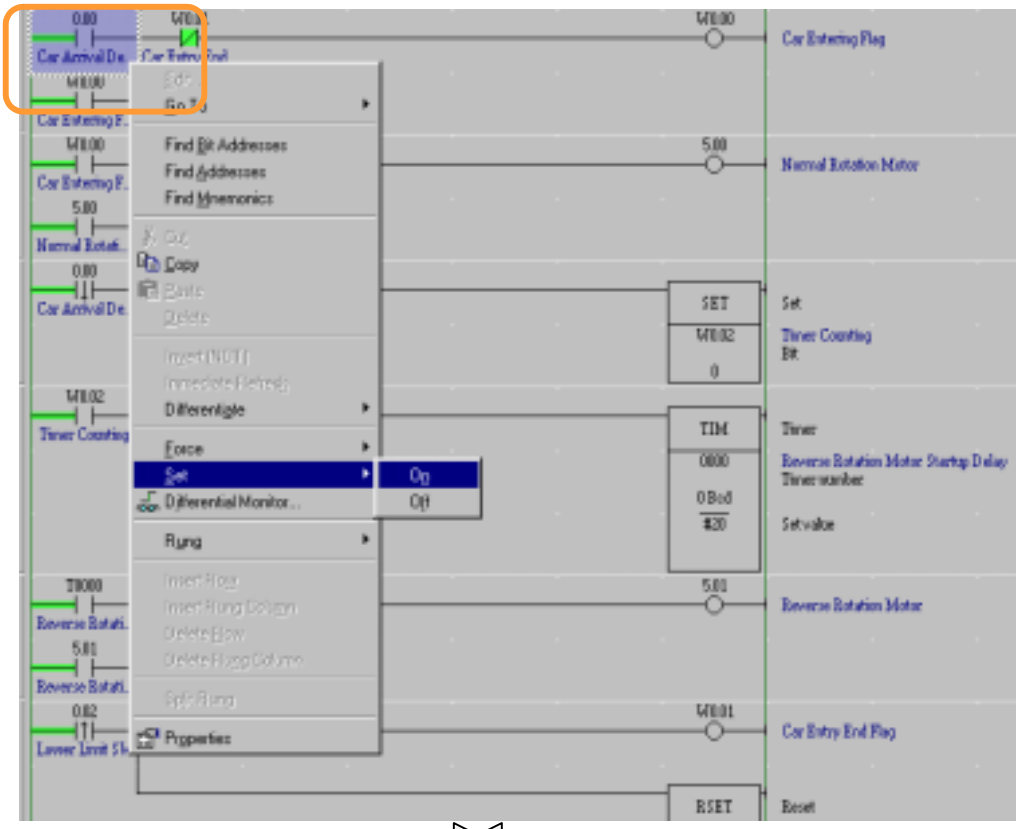


Operation Check by Pseudo Input of Car Arrival

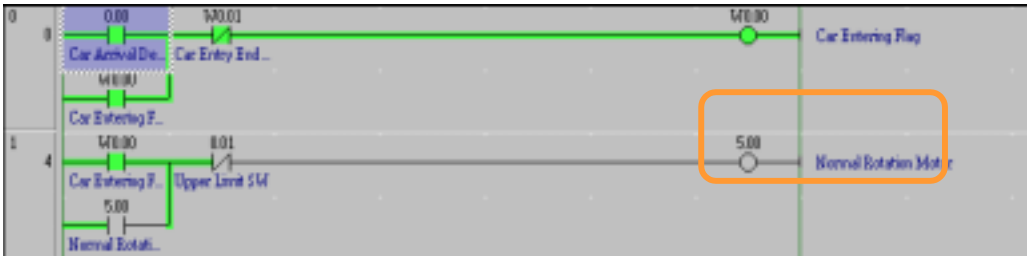
After this, program debug is proceeded according to the processing flow.

First, use the Set On function of CX-Programmer for simulating the waiting for car arrival status through the car arrival status.

Select [Set | On] on the arrival detection sensor (000000) to simulate car arrival.



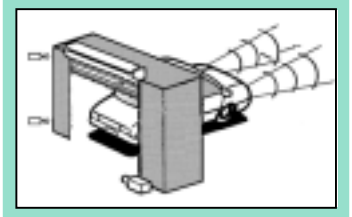
After car arrival (Set on the arrival detection sensor), the normal rotation motor is turned on.



If the motor isn't turned on, once switch the operation mode of the virtual PLC to Program Mode, and repeat the same operation.

In three seconds, the shutter turns on the upper limit SW and the normal rotation motor is turned off. **(Turn on the upper limit SW by the condition/output formulas set in I/O Condition Configuration)**

Car passing



After click the bit 0.00, click the right-mouse button and select [Set] | [Off].

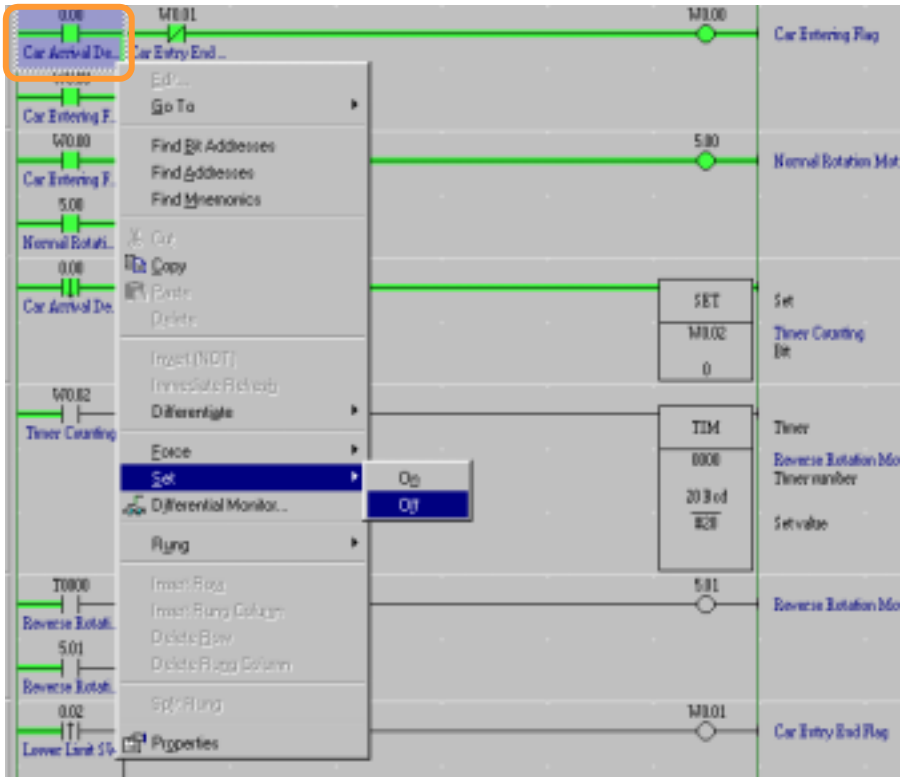
Setting off the arrival detection sensor (0.00) turns on the timer calculating flag (W0.02).

Timer starts calculating.

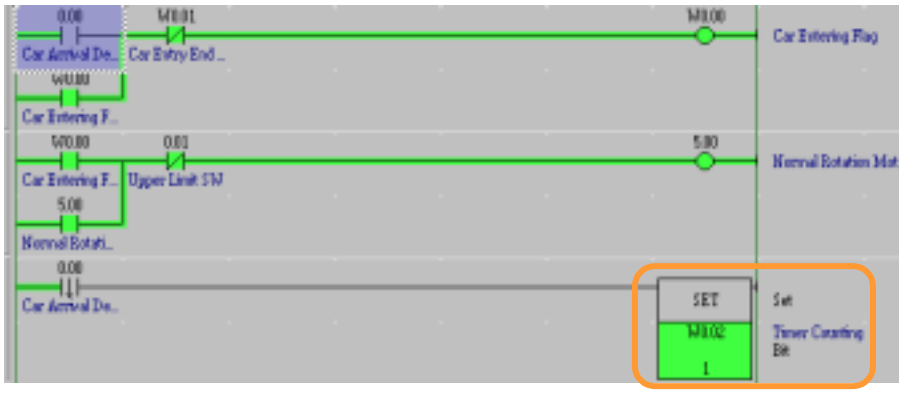
Operation Check by Pseudo Input of Car Arrival

Next, use the Set Off function of CX-Programmer for simulating the car passing status.

Set off the arrival detection sensor (000000) to simulate the car passing status.



After car passing (Set off the arrival detection sensor), the Timer Calculating flag is turned on.



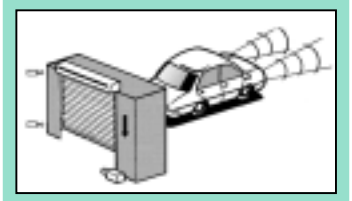
After the timer completes two-second calculation, the reverse rotation motor (5.01) is turned on.

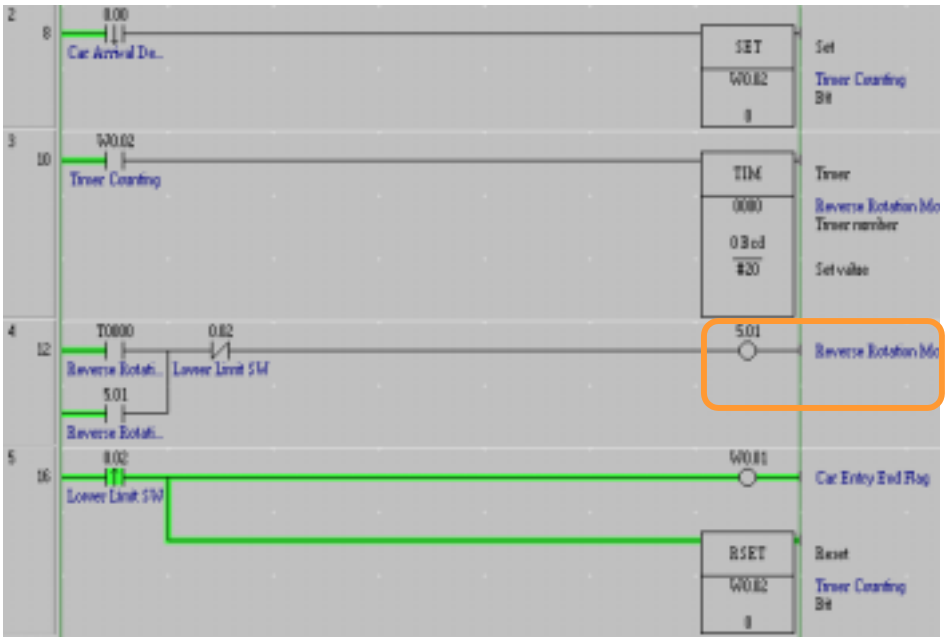
After the timer calculates two seconds, the timer flag is turned on, and the reverse rotation motor is turned on.

In three seconds, the shutter turns on the lower limit switch and the reverse rotation motor is turned off.

(The lower limit SW is turned on according to the condition/output formulas set in Set I/O Condition.)

Shutter down





The following operation makes it possible to debug a series of operation from car arrival to car passing.

The Set On/Off functions of CX-Programmer enable you to simulate the car arrival and passing statuses as explained above.
Moreover, setting I/O conditions enables you to automatically generate the shutter elevating operations.

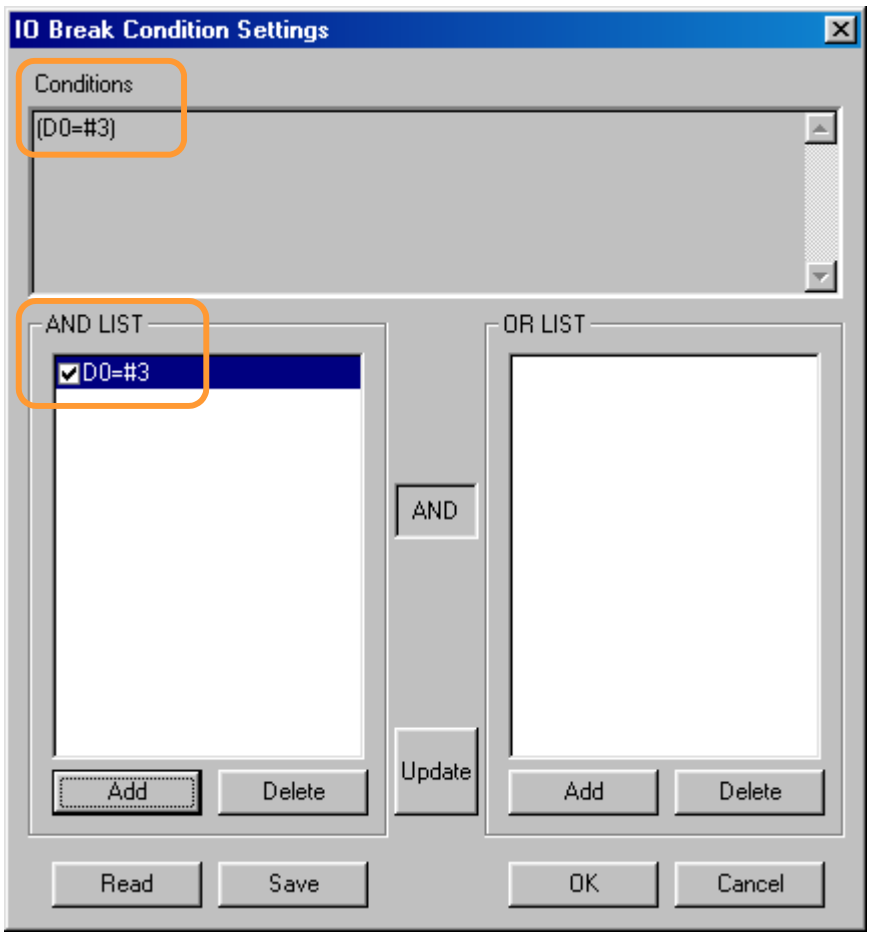
4-3 Debug by IO Break Condition Settings

CX-Simulator allows you to stop a program, for instance, when the number of entered cars reaches to a certain number. This is a function for break when I/O memory status is monitored and the set conditions are satisfied.

4-3-1 Setting Contents of IO Break Condition

Here, the way to stop the program by using IO Break Condition when the number of entered cars turns three is explained.

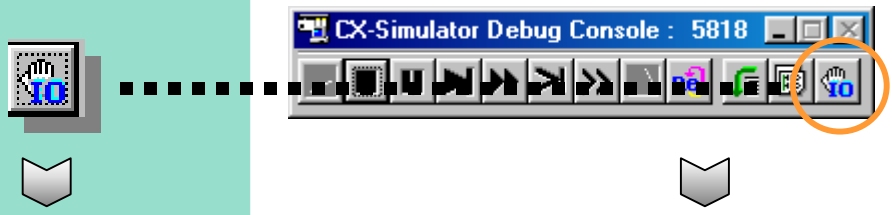
D0 (0 word of data memory) is set as the storing memory for the number of completed entries.



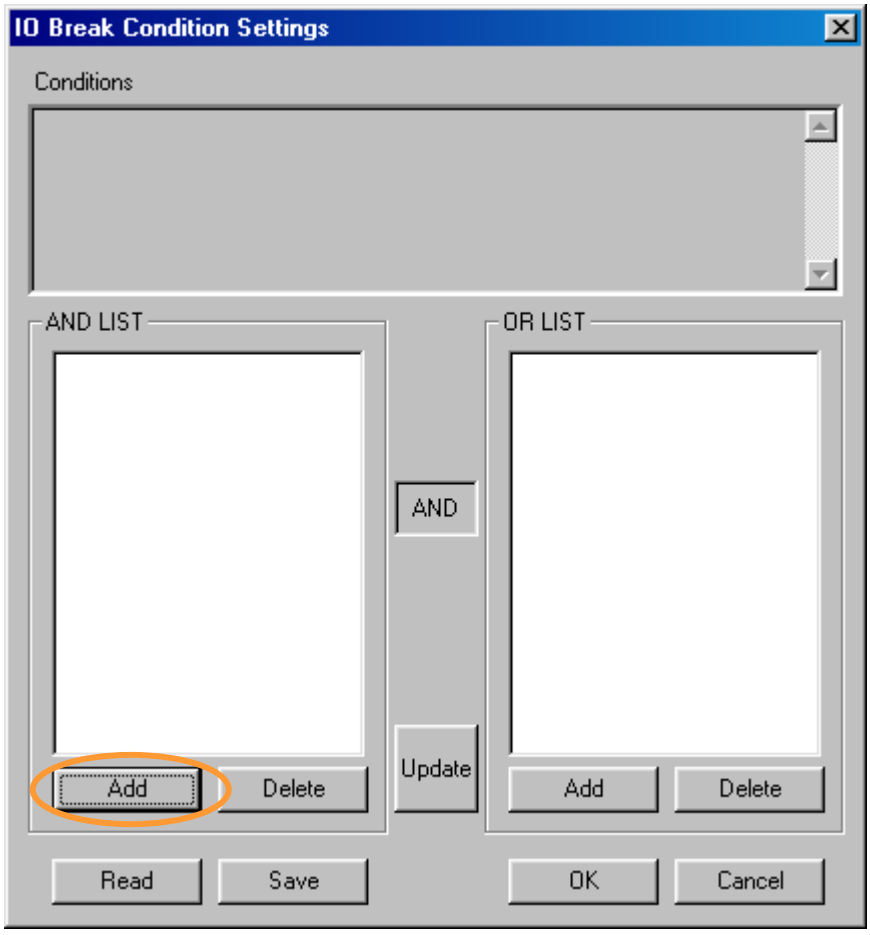
Settings in IO Break Condition Settings

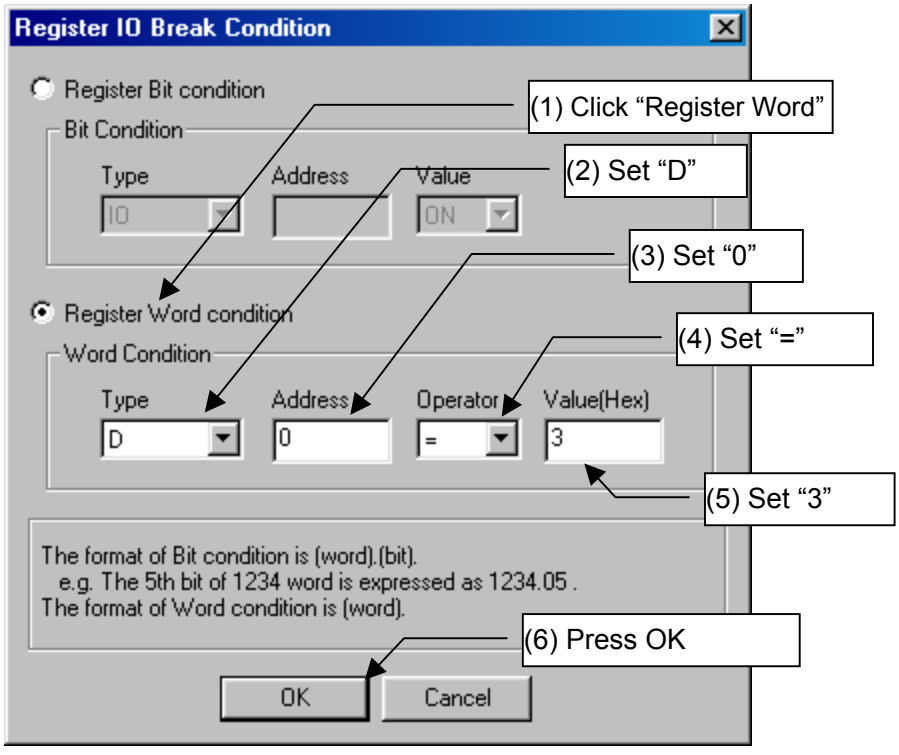
4-3-2 Setting IO Break Conditions

Start IO Break Condition Settings from Debug Console.



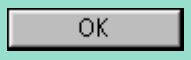
Add



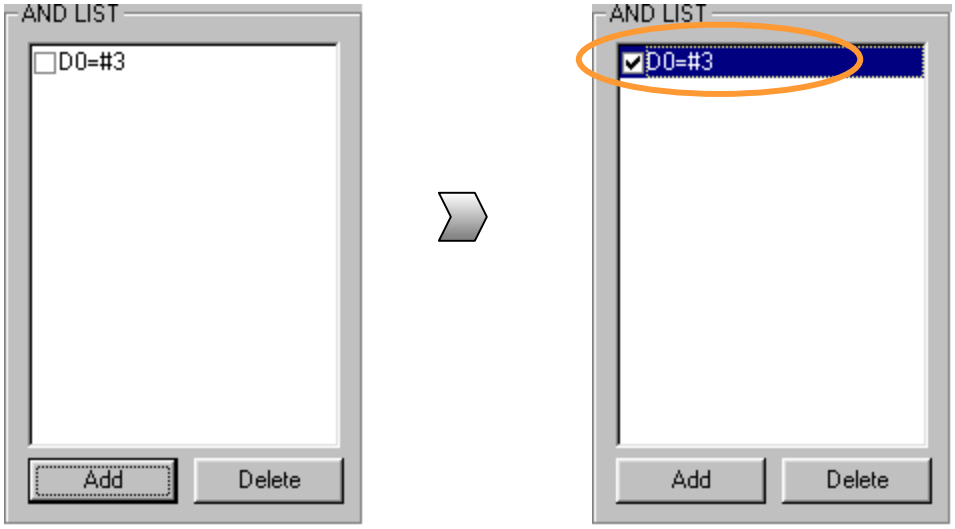


(1) Click “Register Word Condition”.
(2) Set “D” to “Type”.
(3) Set “0” to “Address”.
(4) Set “=” to “Operator”.
(5) Set “3” to “Value”.
(6) Press OK.

Check the formula registered in the list (Activate the formula).



The setting is registered in a list (AND LIST).

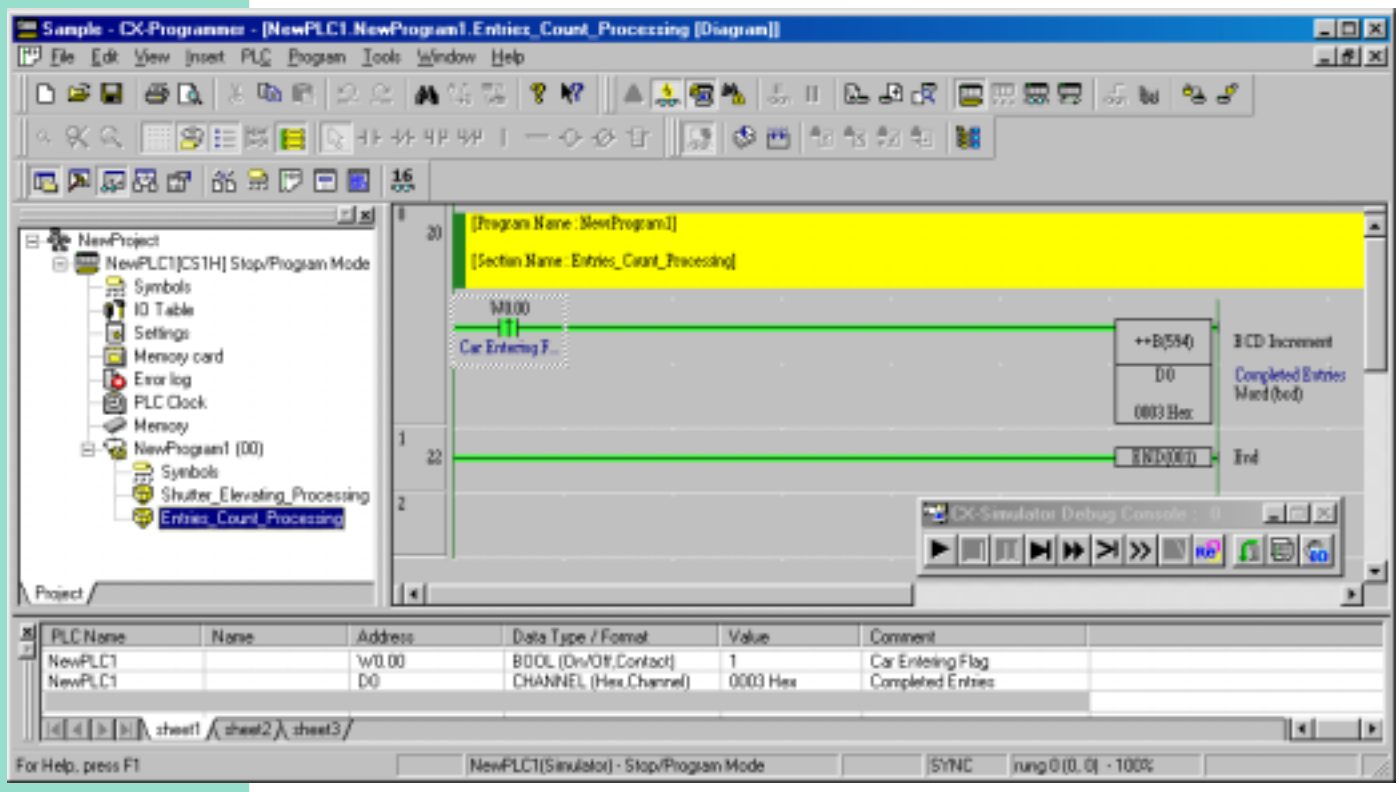


Now, the program is aborted when the number of completed entries (the value of D0) reaches to three (Hex: #0003).

Setting IO Break Condition makes it possible to abort the execution of a program when a certain address value becomes a specified value.
It enables you to check the IO memory status when the conditions are satisfied.
Using this function together with the Watch Window or PLC Memory function of CX-Programmer allows you to check all IO memory statuses with the program

4-3-3 Example of Program Debug Operation by Using IO Break Condition

Double-click the sample program “Entries Count Process“ and start monitoring in advance.



To display Watch Window, press Alt **and** 3 **together.**

Simulate car arrival and passing by using the Set On/Off functions of CX-Programmer. Here, use Watch Window (IO Monitor function).

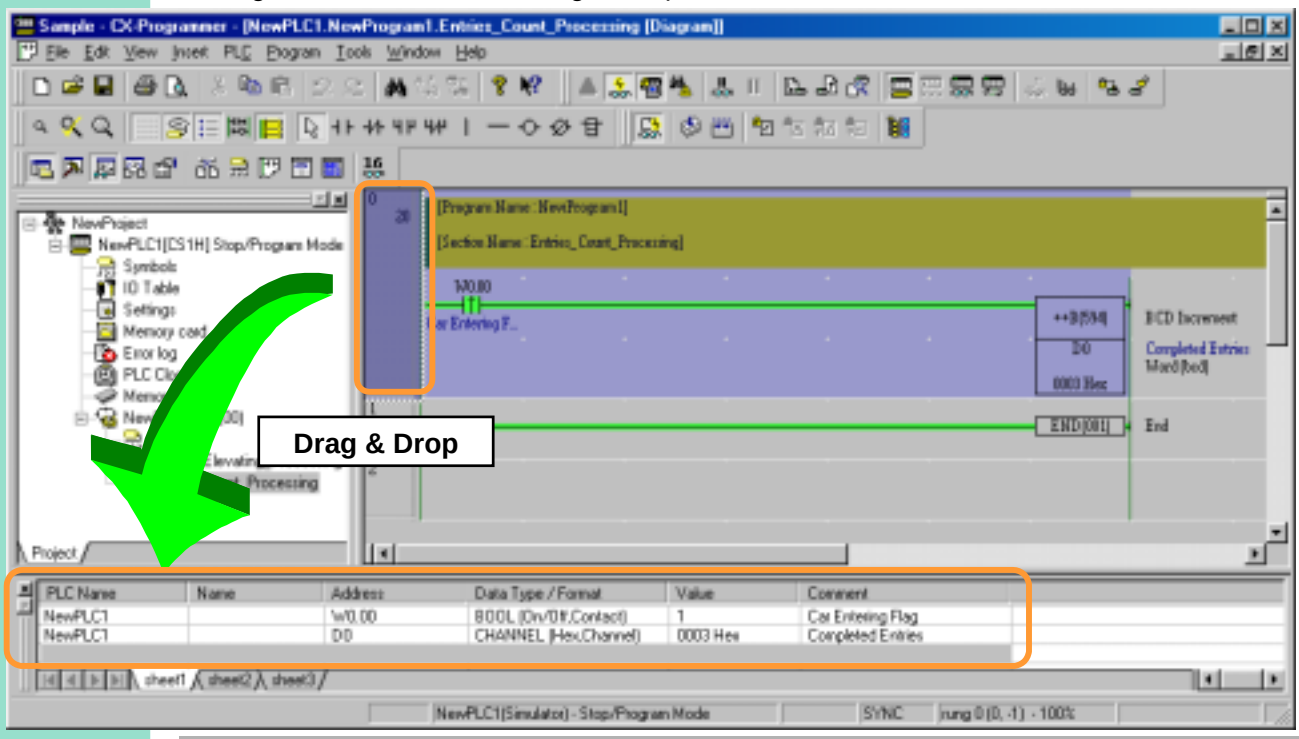
Display Watch Window in advance.

Run the virtual PLC in advance.

Register Addresses in Watch Window

Since the number of car entries is counted in the entry count (D0) by rise of the car entering flag (W0.00) in this sample program, enter these two addresses in Watch Window.

To register addresses, use Drag & Drop function from Ladder Window.

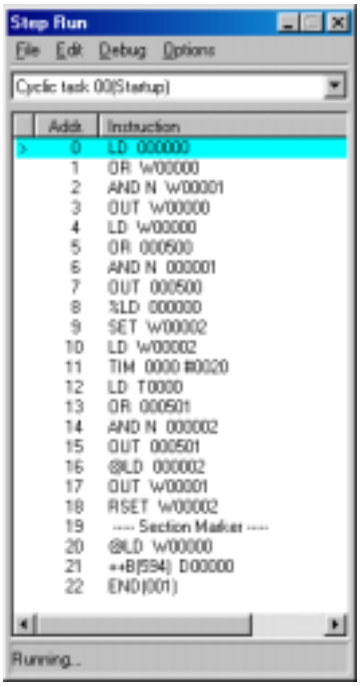


It is possible to register addresses to be monitored in Watch Window easily by Drag & Drop operation from Ladder Window. Not only the registration per rung but also per contact, coil, and advanced instruction is enabled by the same operation.
Also, Set On/Off and Change Value operations are enabled to the registered addresses easily.

Display the view of Step Run in advance.

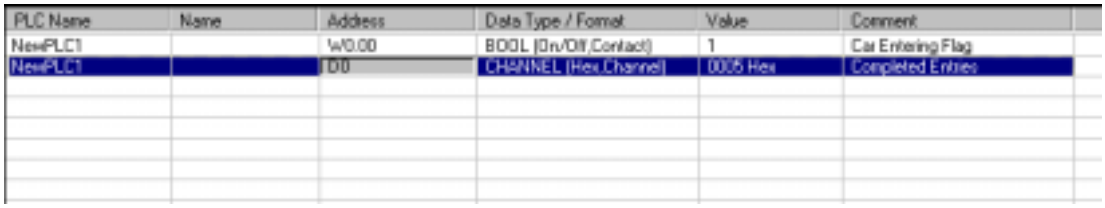
(The status is currently displayed as “Running”.)

Click the icon in Debug Console.

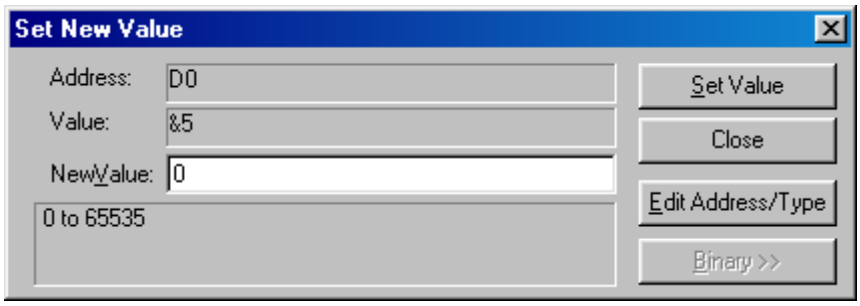


Double-click D0 on Watch Window. (Or, select D0 and press the Return key.)

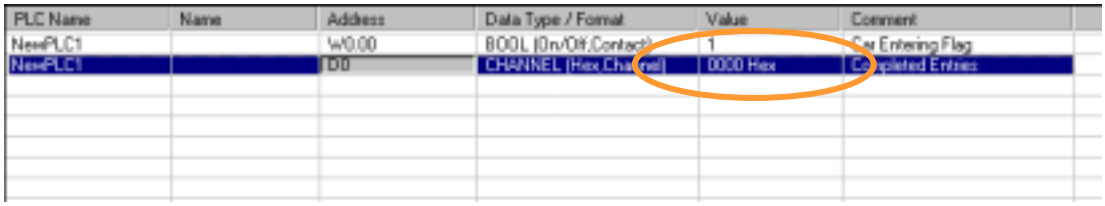
First, initialize the value of the number of completed entries (D0) (change the value to 0).



Enter “0” in “Value”, and press the Return key.



The value of D0 is changed to “0000 Hex”.



Double-click W0.00 on Watch Window.

Enter “1” (Set) in “Value” and press the Return key.

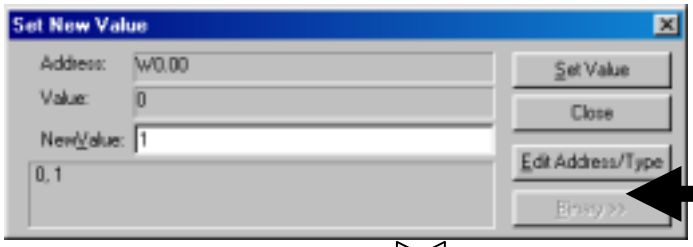
The number of completed entries (D0) is incremented to “0001 Hex”.

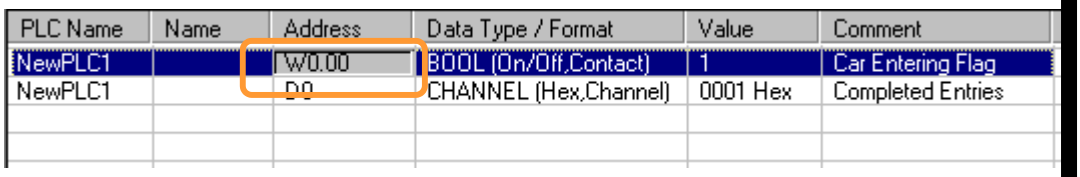
Double-click W0.00 in Watch Window.

Enter “0” (Set Off) and press the Return key.

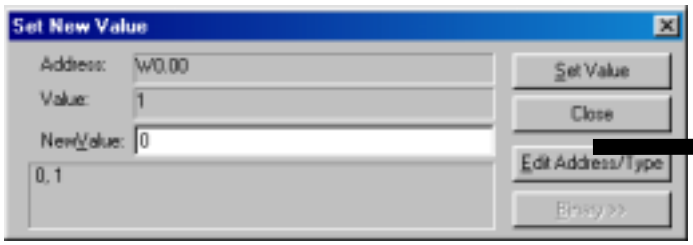
Pause by IO Break (Pseudo Input of Car Arrival/Car Passing)

Set W0.00 on for simulating the car arrival operation.



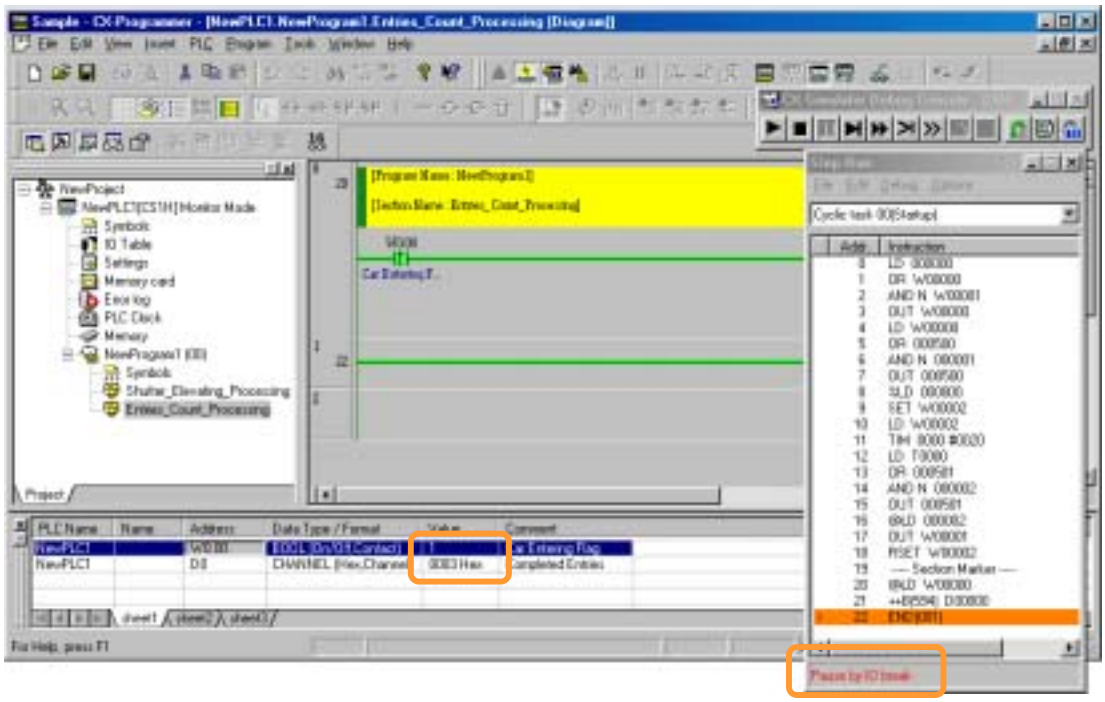


Set W0.00 off for simulating the car passing operation.



Repeat until the value of D0 turns “0003”.

When the number of completed entries (D0) turns “0003”, the message shown bottom of the Step Rung view is changed to “Pause by IO break”, and the program execution is aborted.

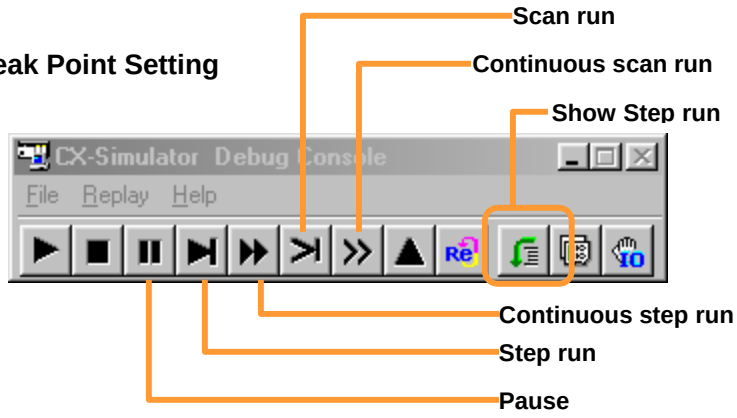


4-4 Other Debug Functions

Running Program in the Unit of Step and Scan, Break Point Setting

If “Step run” is used, for instance, it is possible to inspect a sequence calling subroutine processing or check the values in progress from FOR to NEXT (Fig 1).

Also, when a break point is specified and “Scan run” is executed, it is possible to execute the program until the break point and then abort it. It enables you to restart the program execution from any line (Fig 2).



Step run: Executes a program per instruction.
Continuous step run: Executes instructions one by one automatically.
Pause: Aborts continuous step running temporarily.
Scan run: Executes one scan and then aborts.
Continuous scan run: Repeats the operation of one scan.

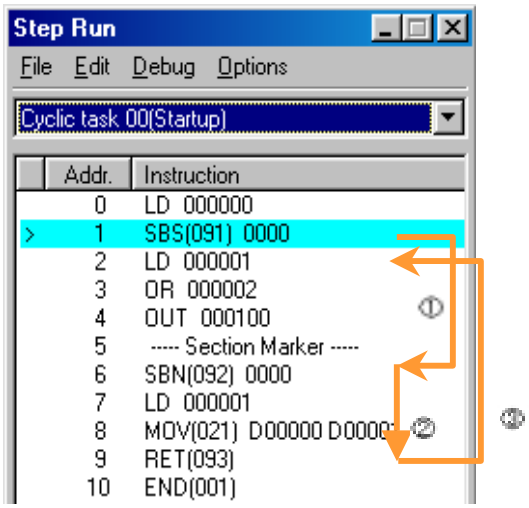


Fig 1. Debug example by step running

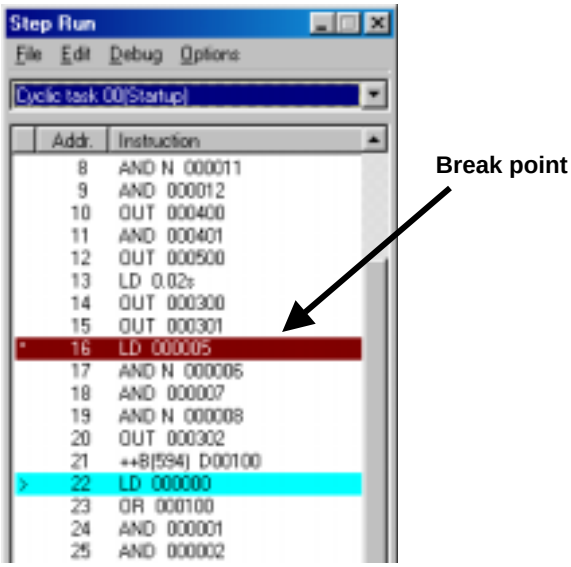


Fig 2. Debug example of break point

Chapter 5
Startup from CX-Simulator Menu

This chapter explains how to start and end CX-Simulator and how to set CX-Simulator for creating a virtual PLC in your PC.

Creating a virtual PLC from the CX-Simulator menu enables you to use the following functions. See the operation manual of CX-Simulator for the detailed operations.

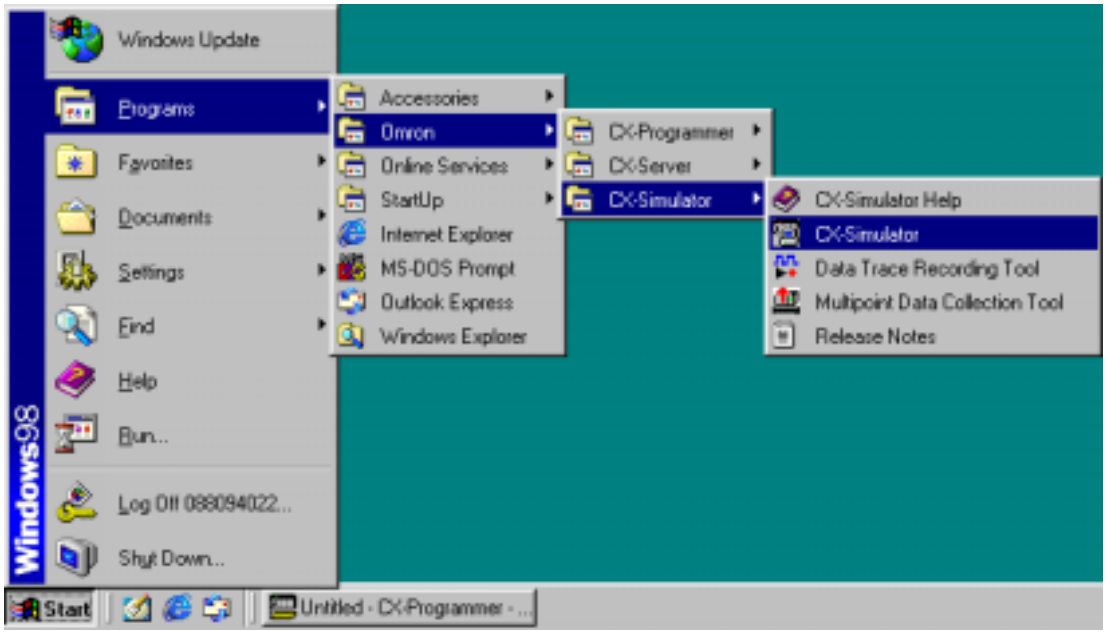
- **Serial communications for connecting with PT**
- **Network communications by network communications instructions**
- **Measurement of I/O refresh time with an I/O unit registered**
- **Display of message instructions or network communications instructions on your PC screen**
- **Record of the communications log of FINS Commands sent/received by a virtual PLC**

[Start]
↓
[Programs]
↓
[Omron]
↓
[CX-Simulator]
↓
[CX-Simulator]

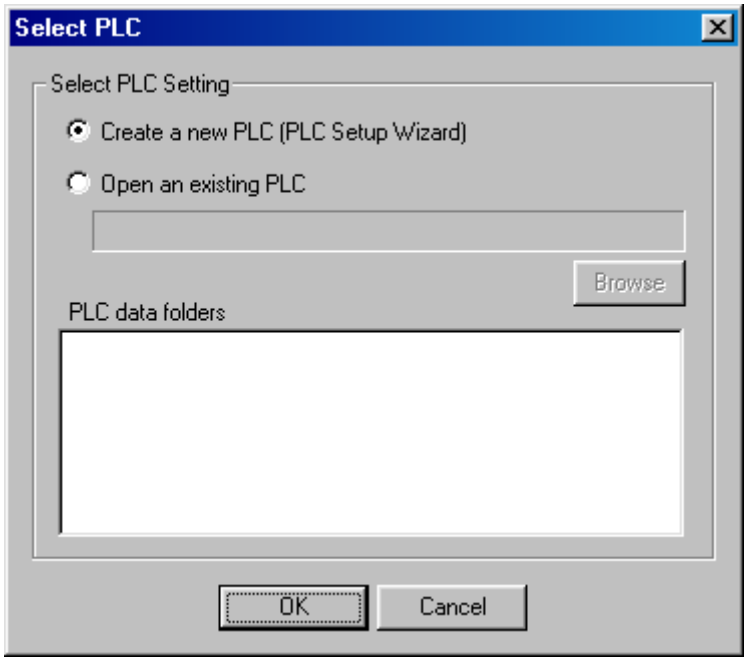
Or double-click the CX-Simulator icon.

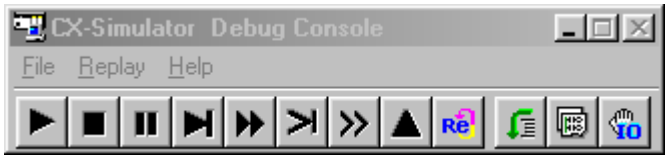
3-1 Starting CX-Simulator

Start up CX-Simulator from Windows.



The [Select PLC] and [CX-Simulator Debug Console] screens show up.

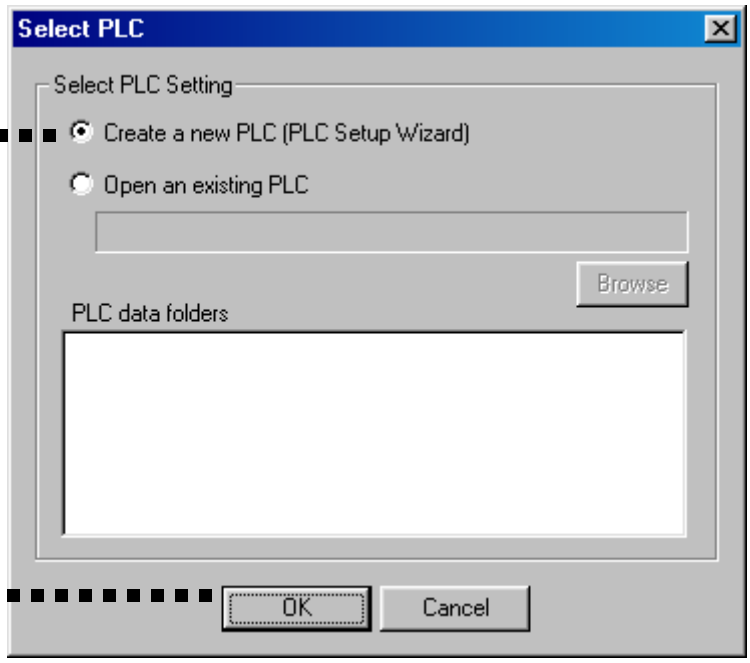




5-2 Creating Virtual PLC

Create a virtual PLC according to the contents of the [Select PLC] screen.

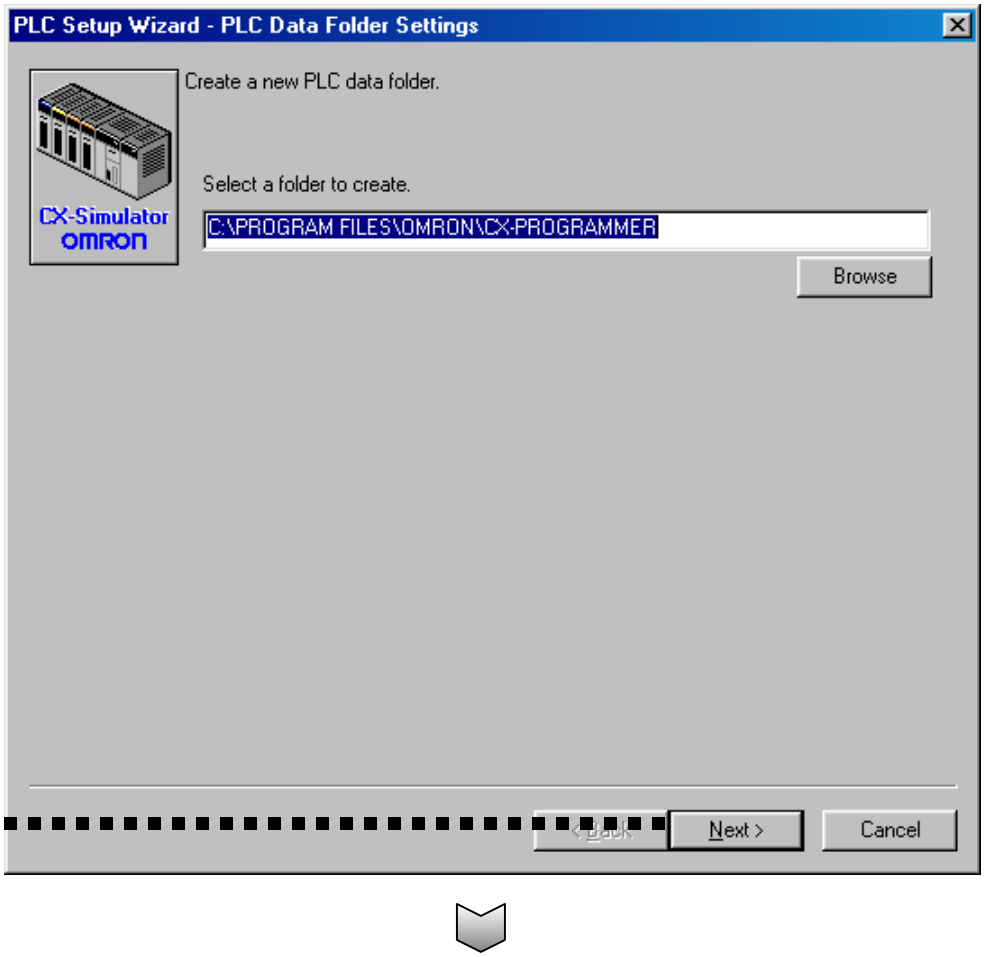
Check [Create a new PLC].



OK

Select a directory to create data.

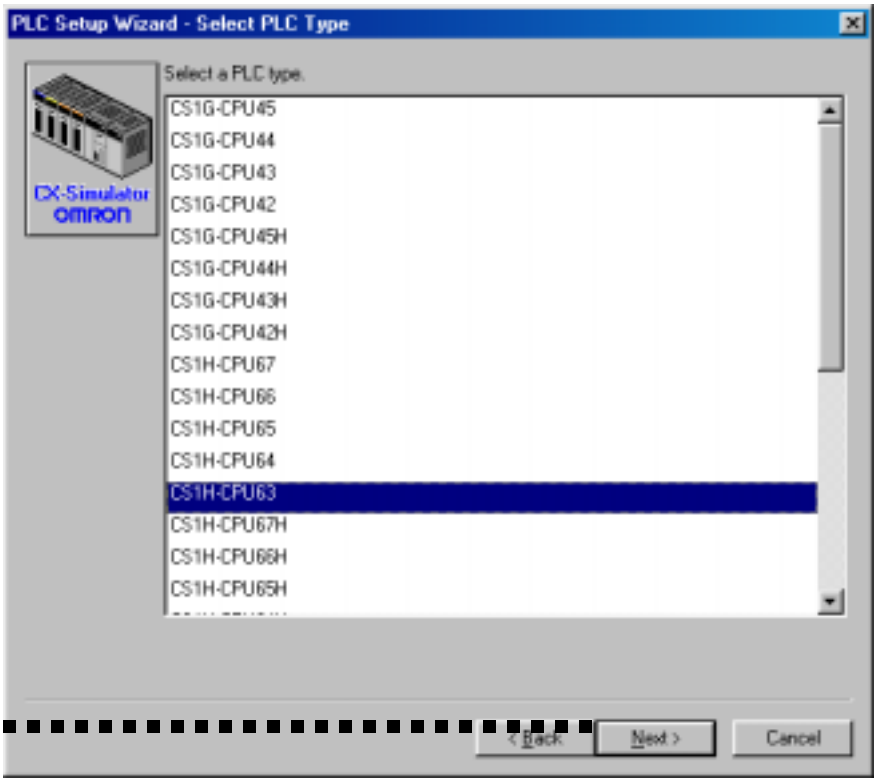
Select a directory to save data. Use the default directory this time.



Next >

Select a PLC type.

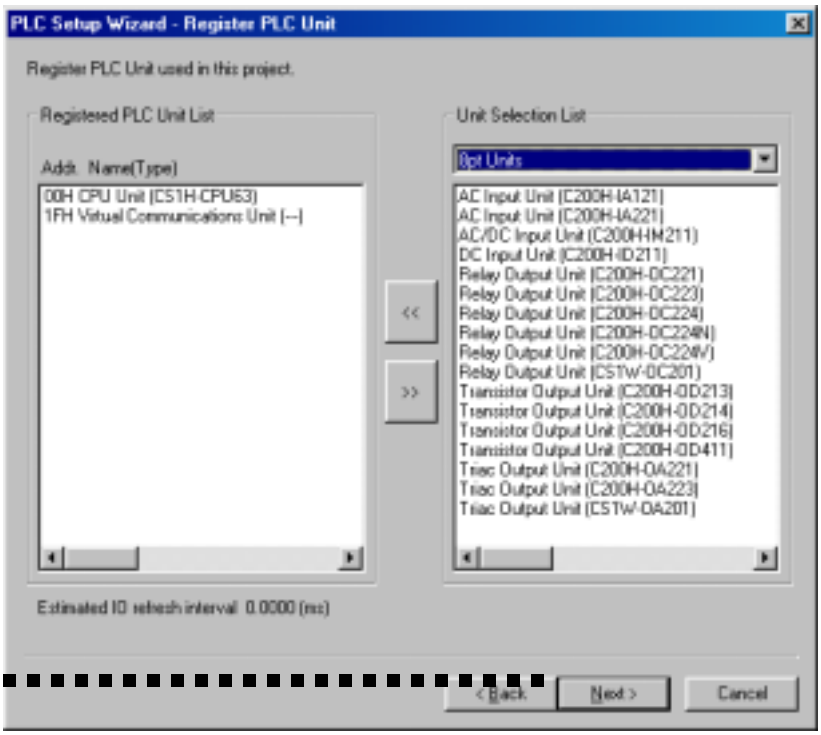
Select a CPU unit type. Select “CS1H-CPU63” this time.



Next >

Register PLC Unit.

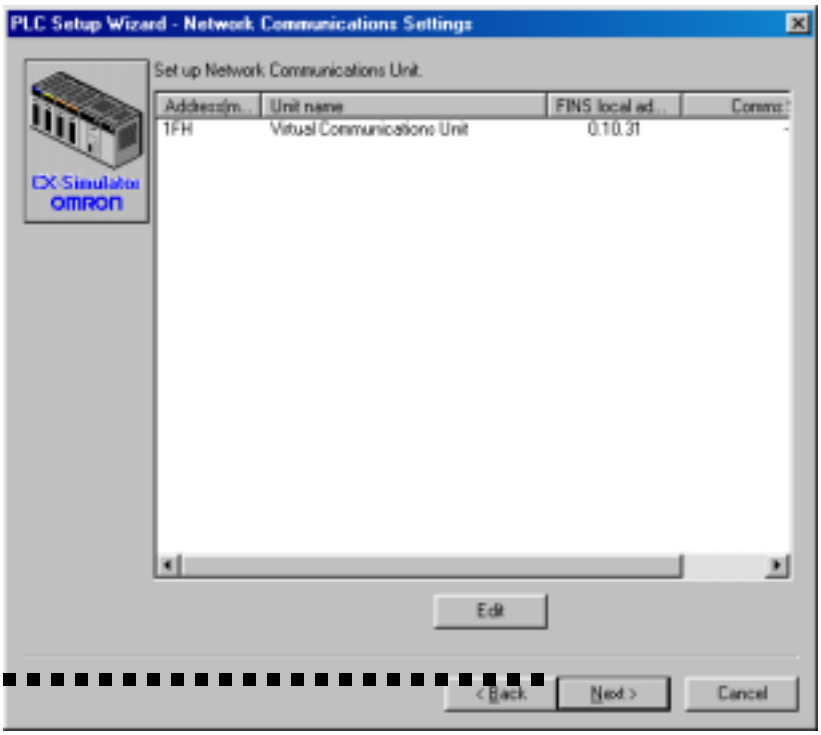
You can register IO Units and Special IO Units. Register the units to be used in the actual case, and you’ll improve the accuracy of the advance verification of cycle time. Do not register any unit in this example.



Next >

Set network communications.

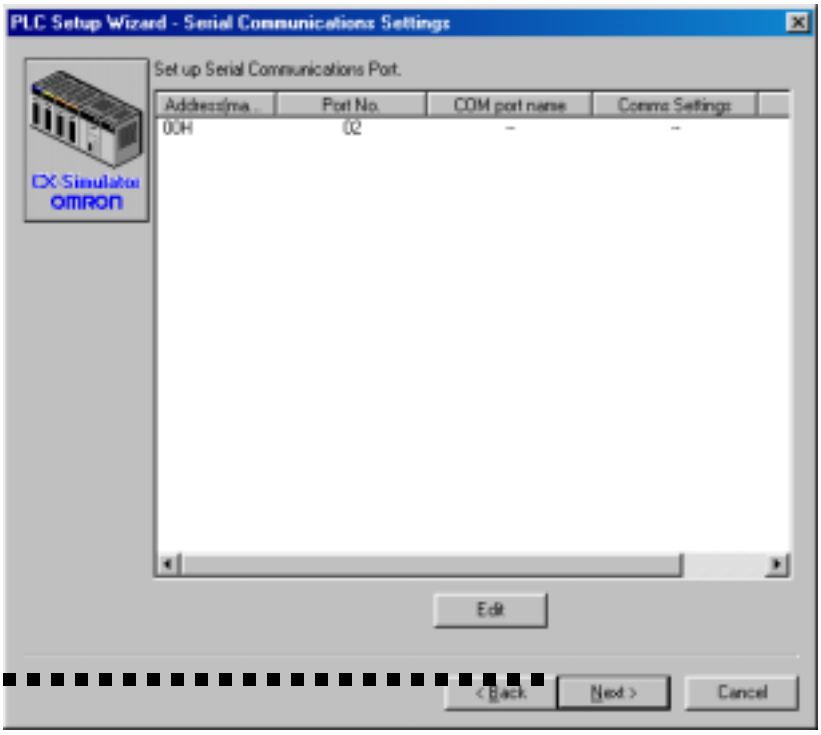
Network settings for connecting to a virtual PLC. Normally, do not change.



Next >

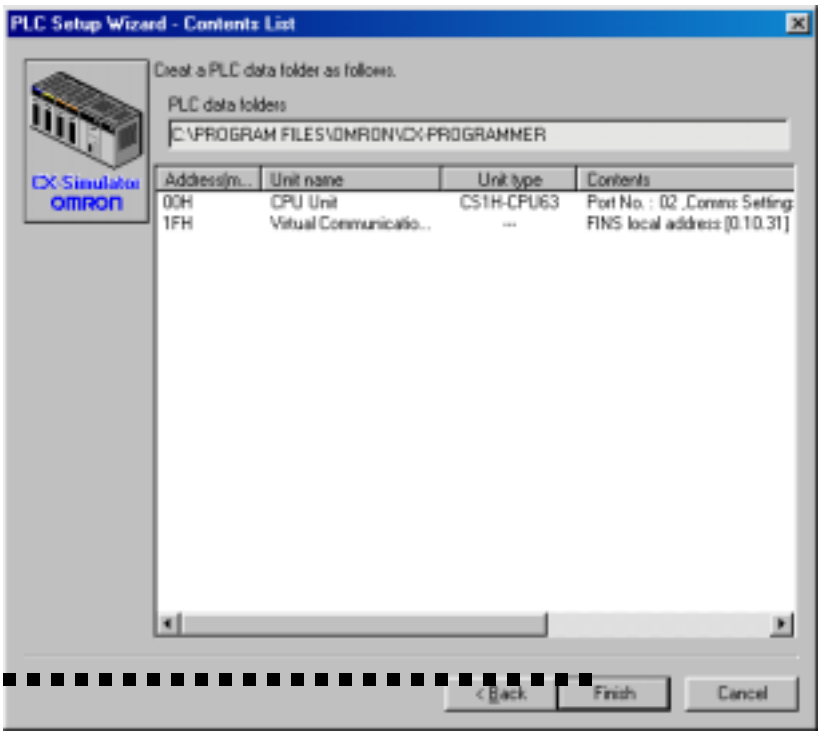
Set serial communications.

Regarding the serial port of you PC as the built-in serial port of CPU Unit, you can connect with external equipment such as Programmable Terminal. Do not set anything this time.

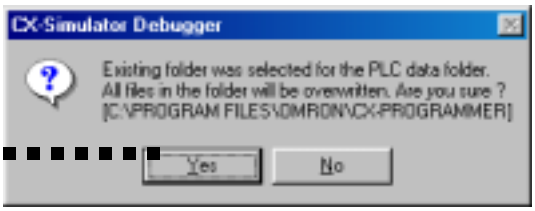


Next >

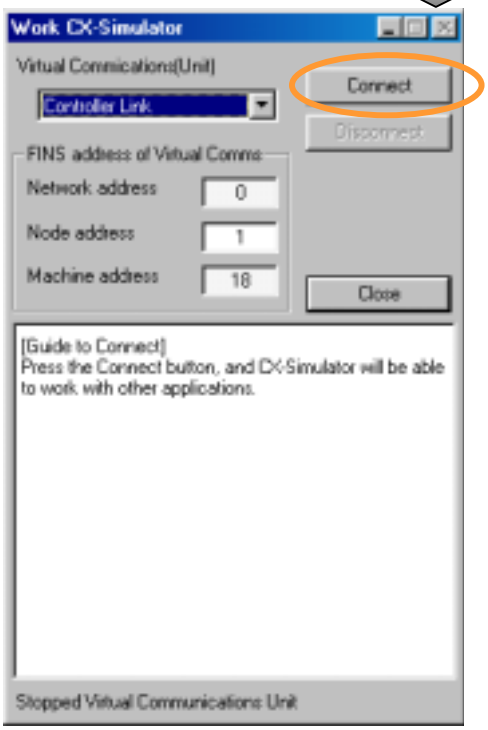
Contents List



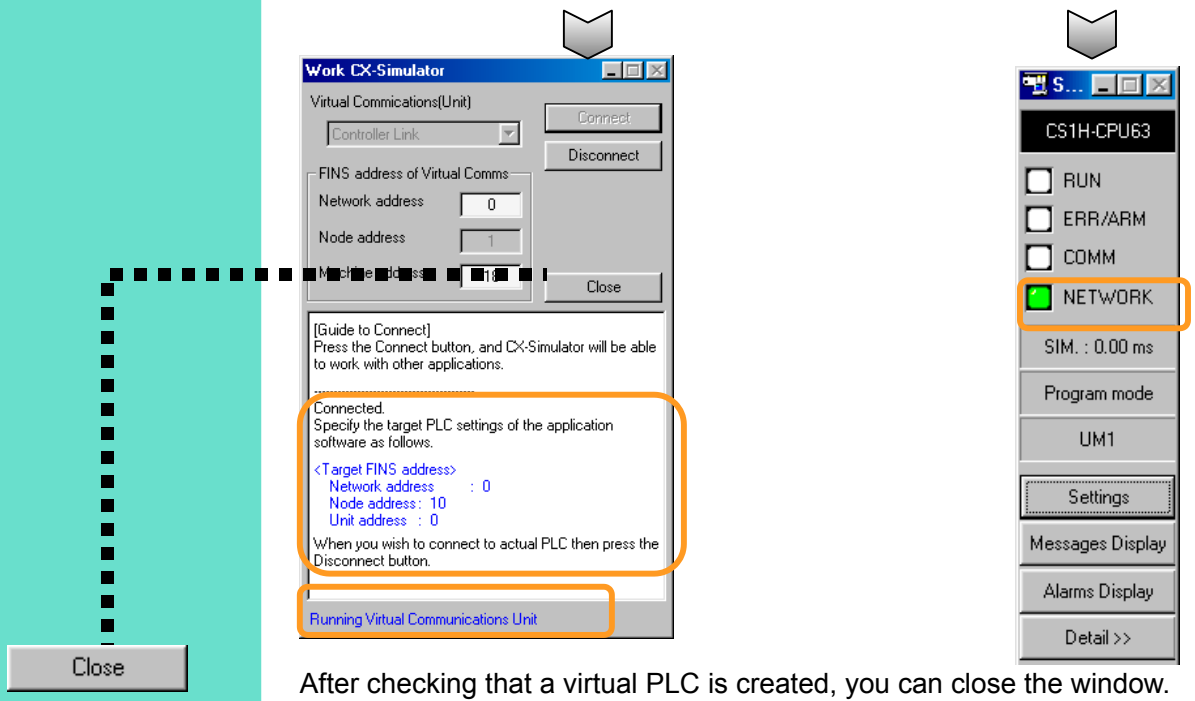
Finish



Yes



Connect



After checking that a virtual PLC is created, you can close the window.

It is the end of creating a virtual PLC.

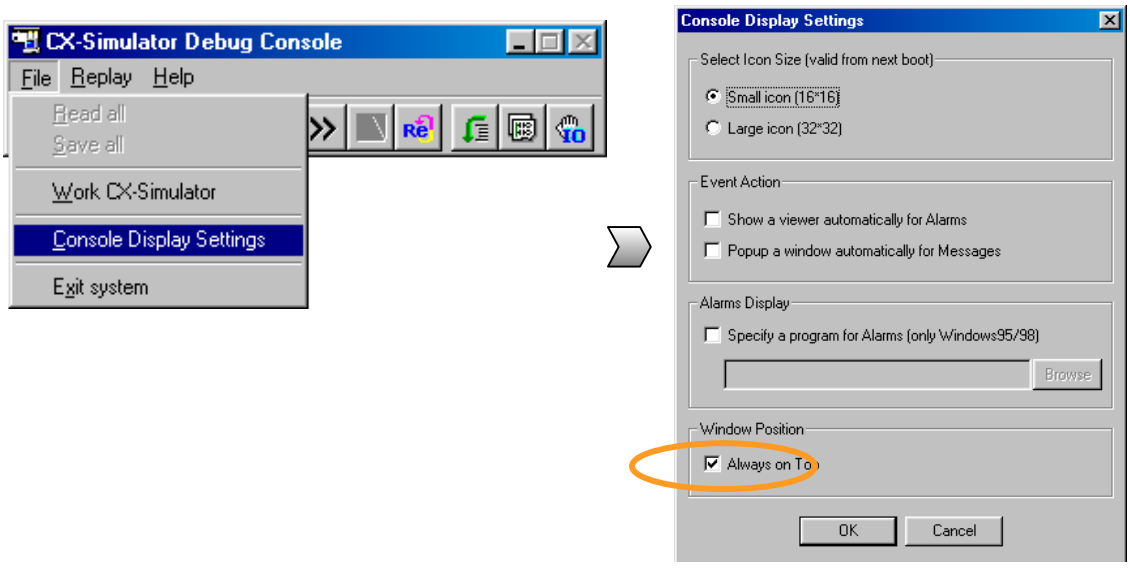
Now the virtual PLC CS1H-CPU63 exists in your PC.

After a virtual PLC is created, [NETWORK] in [Status Settings] (above right) starts blinking.

The virtual PLC is created in your PC with the network address 1 and the node address 10. You can connect CX-Programmer and the virtual PLC by the exactly same procedure as the actual PLC.To connect to the virtual PLC from CX-Programmer, set the above addresses to the destination PLC.

CX-Simulator Console Display Settings

You can set display settings. Check [Always on Top] for your convenience.



Select [File] | [Work CX-Simulator] on the [Debug Console] screen.

5-3 Saving/Ending CX-Simulator

Take the following procedure to save the data of a set virtual PLC and to end CX-Simulator.