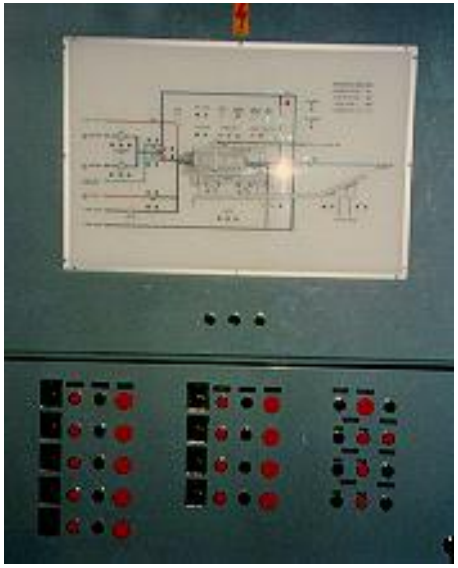
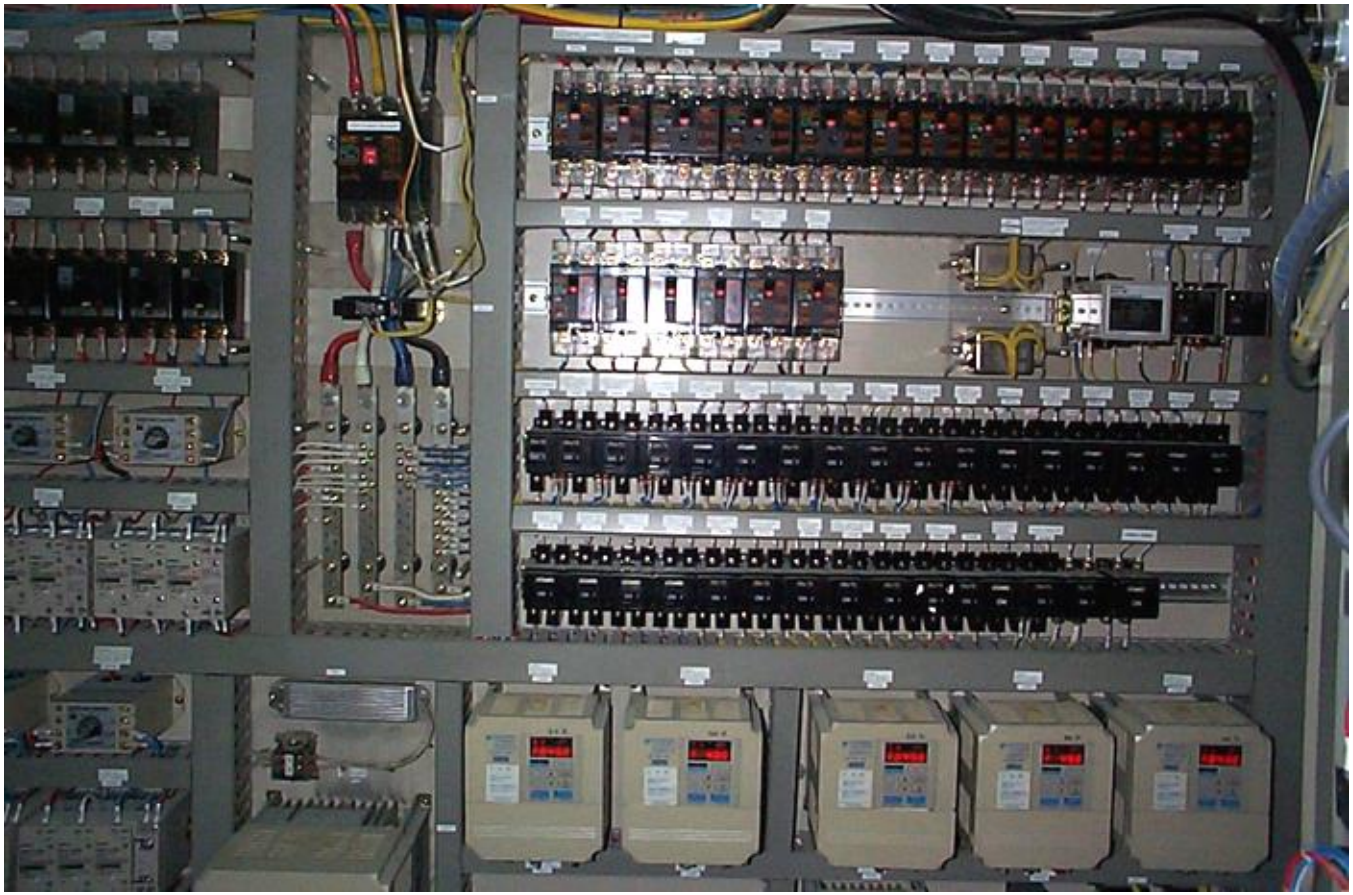
# Introduction to Control System (Control Panel)



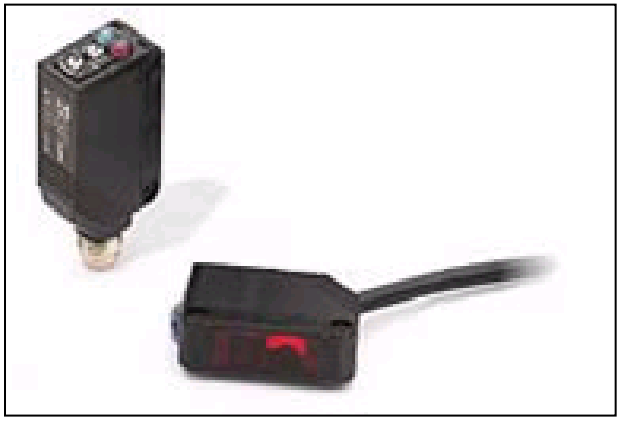
Start & Stop Operation

# What is inside the control panel?



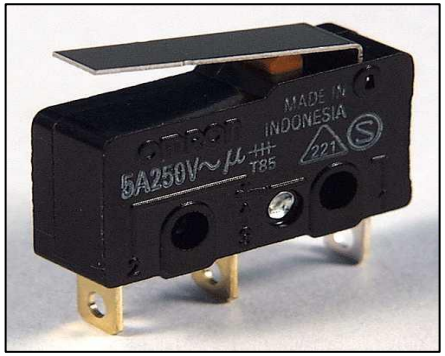
Contactors, relays, timers & counters
in the control panel

# Input Devices



Photoelectric Switch

Rotary Encoder

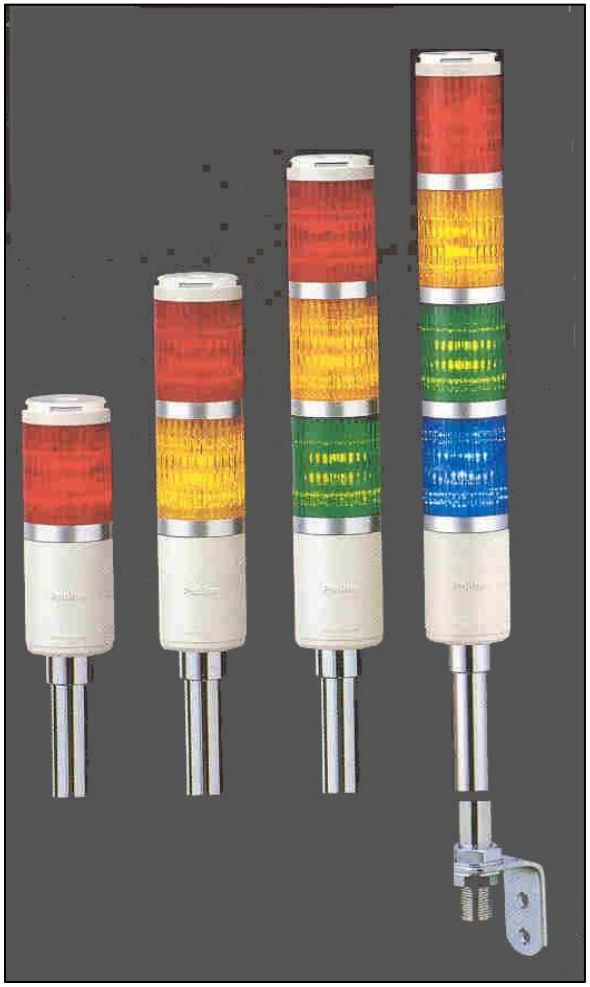



Limit Switch

Proximity Switch

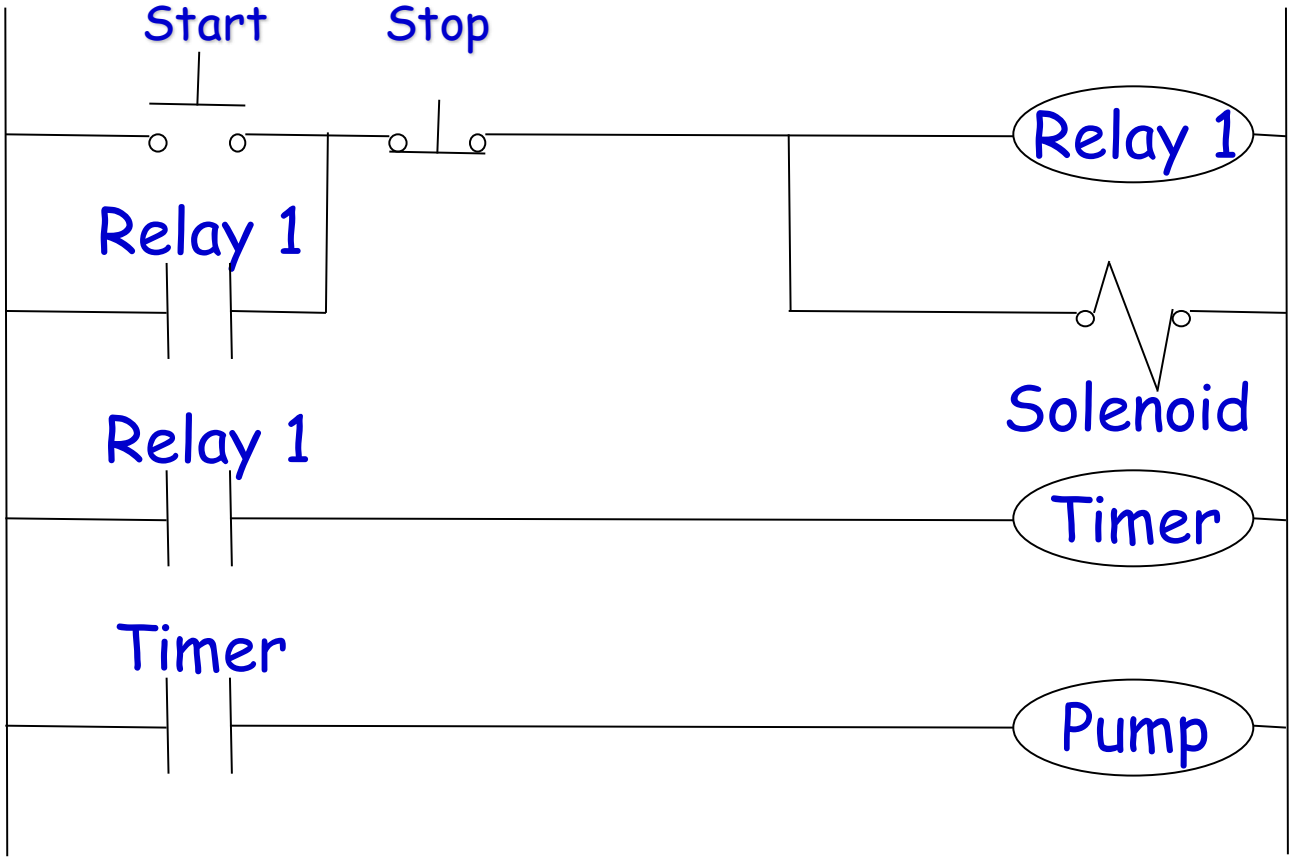
# Output Devices


**7-segment Display Unit**

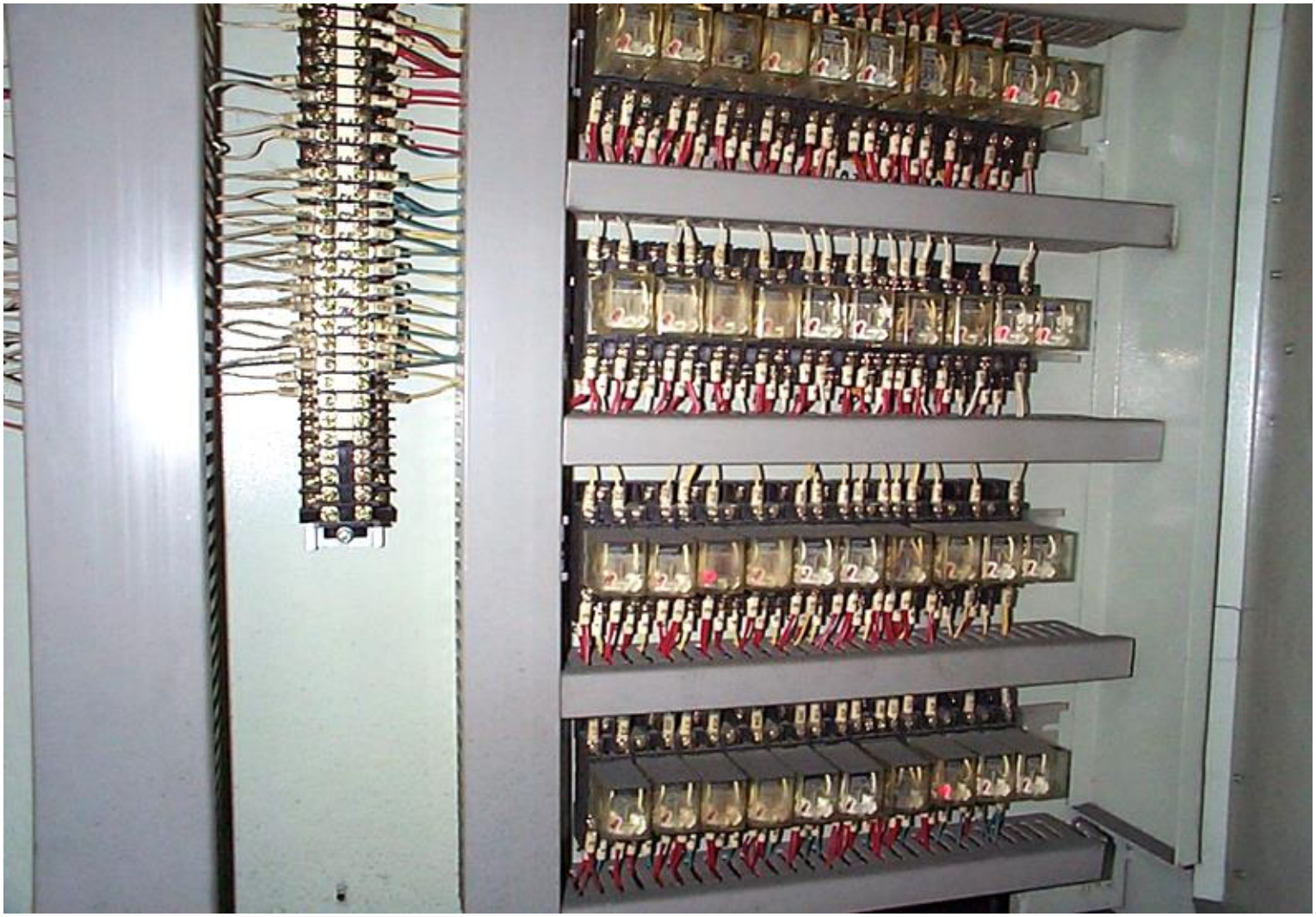


**Tower Light**

# Conventional Hardwire Logic Circuit



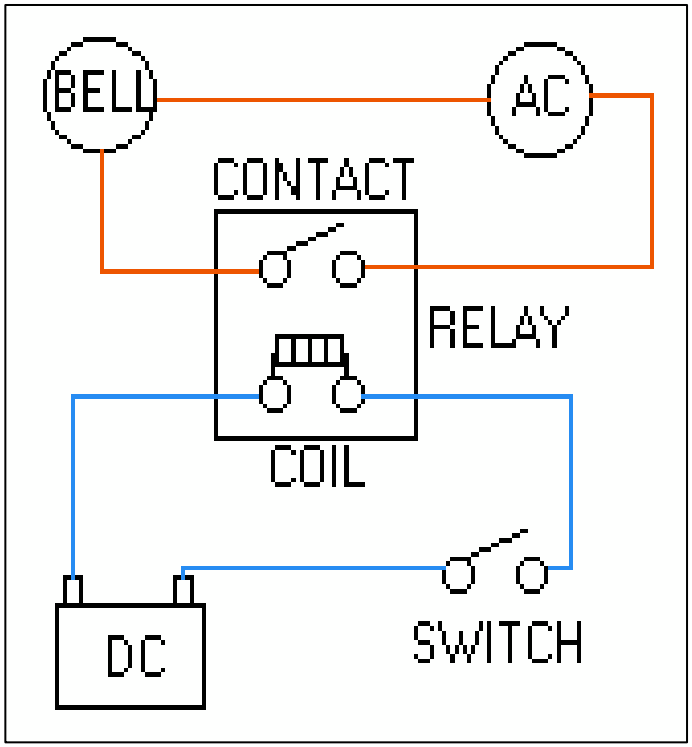
# Conventional Relay Control Panel



# Introduction: What is a PLC?

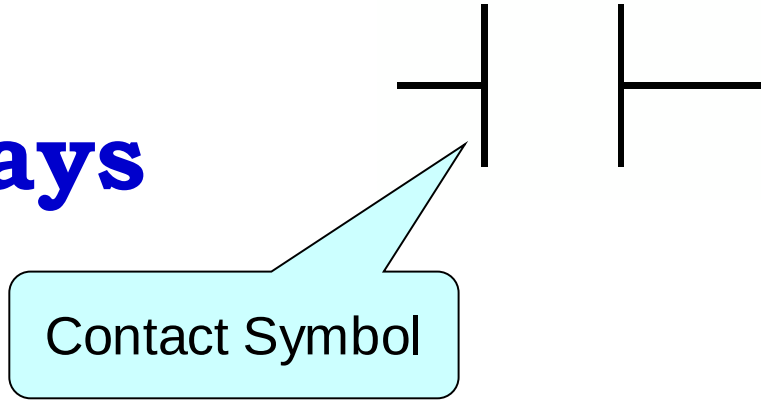
- A PLC is a device that was invented to replace the necessary sequential relay circuits for machine control.
- The PLC works by monitor its inputs and depending upon their state, turning on/off its outputs.
- The user enters a program, usually via software, that gives the desired results.

# Relay




- Relay is an electromagnetic switch.
- Applying voltage to the coil will generate magnetic field.
- This magnetic field sucks the contacts of the relay in, causing them to make a connection.
- These contacts can be considered as a switch.

# Replacing Relays

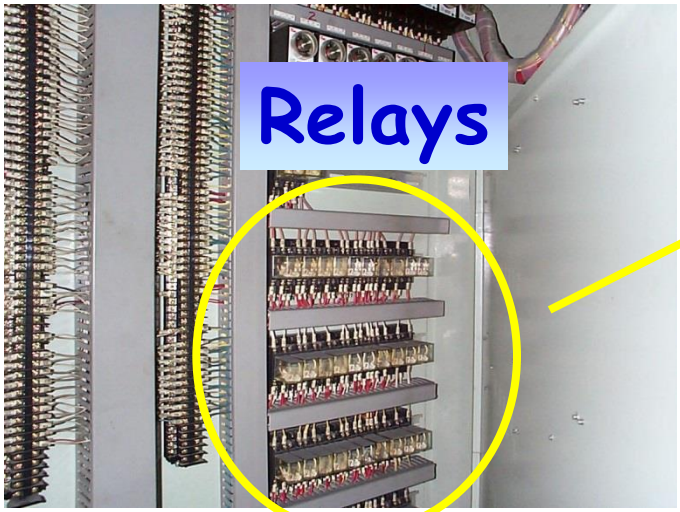


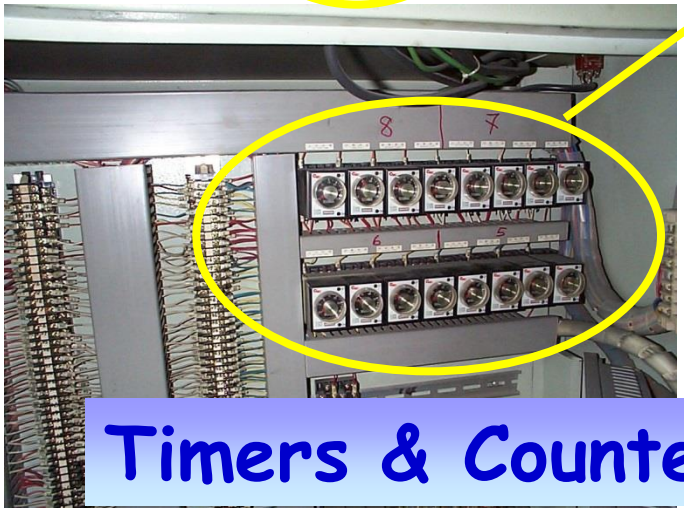

- The main purpose of PLC is to replace relays.
- Relay based control system is designed using **Ladder diagram**.
- CX-Programmer is the software used to convert ladder diagram into PLC mnemonic code.
- PLCs use input, output coil and contact instead of switch, relay, motor etc.
- Bus bar is two vertical bars common to all ladder diagram.

# PLC History

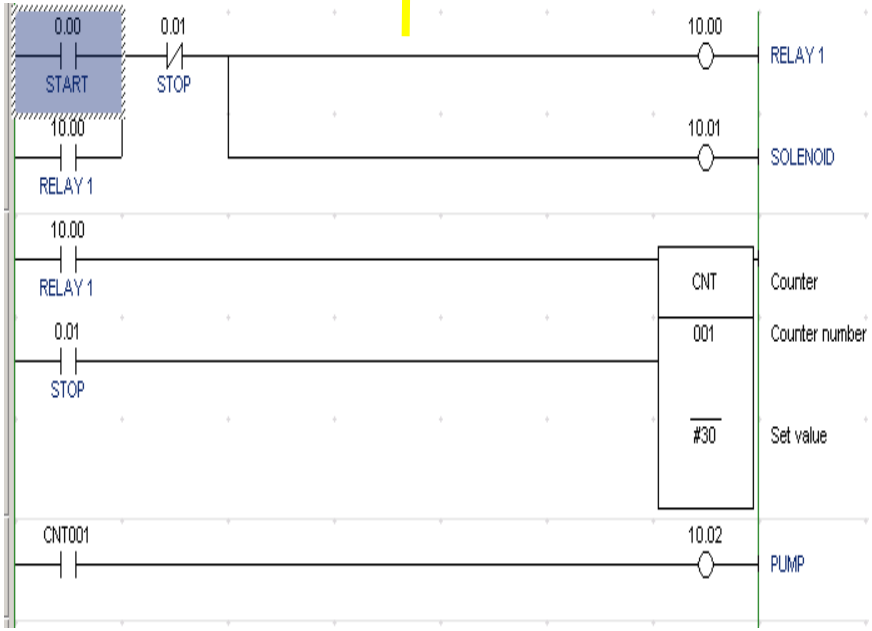
- In the late 1960's PLCs were first introduced to replace complicated relay based control systems.
- In 1970's, the dominant PLC technologies were sequencer state-machi- es. Communications abilities began to introduced for PLCs.
- In 1980's, an attempt to standardize communications with Manufacturing Automation Protocol(MAP).
- In 1990's, an attempt to standardize PLC programming languages under one international standard: IEC 1131-3.

# 1. Sequence Control



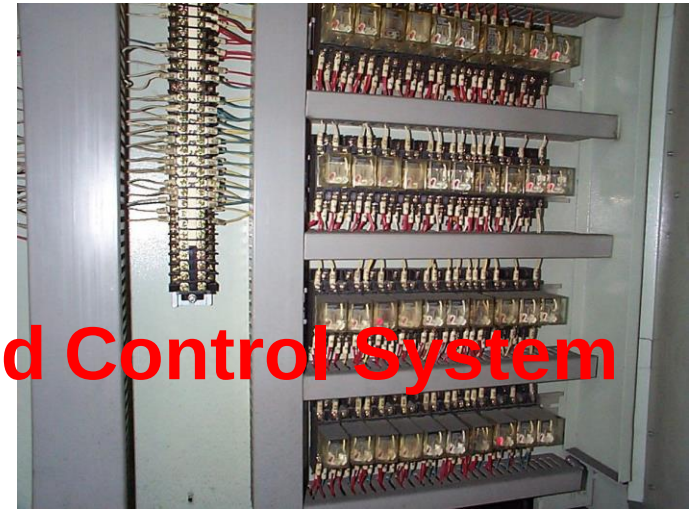




# Positioning Control with PLC

# The advantages of Using PLC

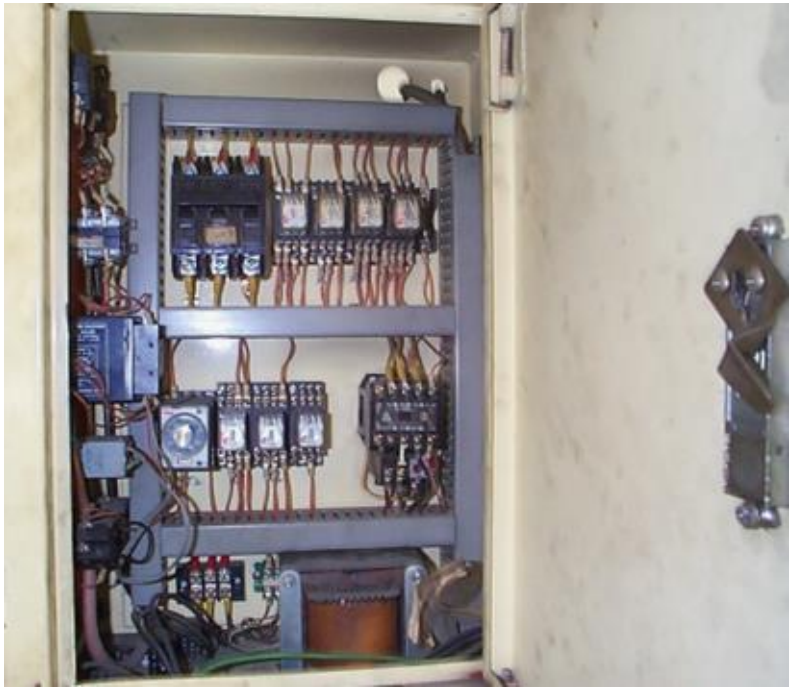
- Smaller physical size than hard-wire solutions.
- Easier and faster to make changes.
- Diagnostics are centrally available.
- Applications can be immediately documented.
- Applications can be duplicated faster and less expensively.
- Fast project implementation time
- Low operating cost
- Increase productivity



**PLC Vs Relay Based Control System**

# Control Device

Wired Logic



Specific Purpose

Control sequence is fixed for one design

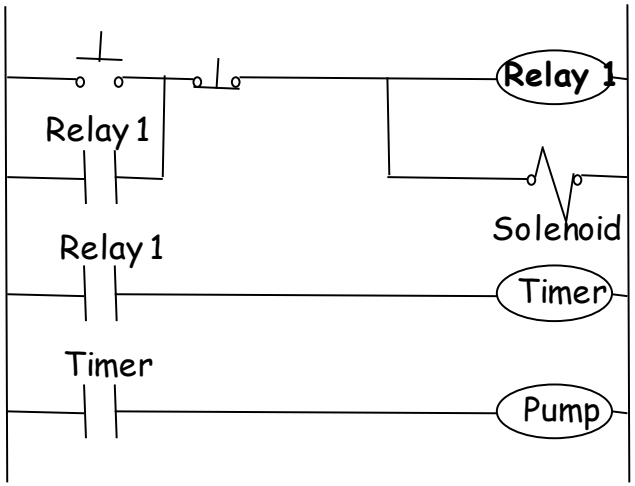
PLC

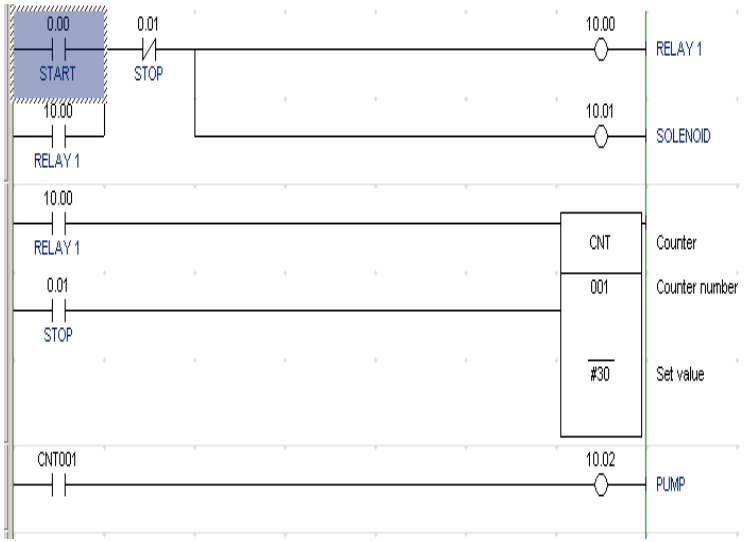
General Purpose

Control sequence can be changed from time to time without rewire

# Modification

## An example for hard wire logic




**Mnemonic Code**

| | | |
|---|---|---|
| LD | | START |
| OR | | RELAY 1 |
| AND | NOT | STOP |
| OUT | | RELAY1 |
| OUT | | SOLENOID |
| LD | | RELAY 1 |
| TIM | | 000 |
| | #0020 | |
| LD | TIM | 000 |
| OUT | | PUMP |

For PLC:-

Modification by only to change the ladder program.

For hardwired logic:-

Modification by rewired all the things.

# Delivery Period

Several days

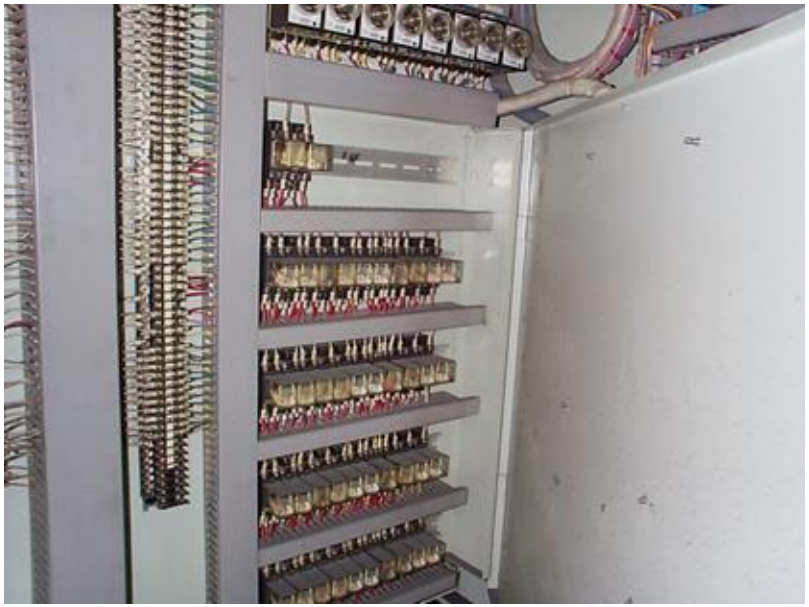
Wired Logic

Almost Immediate

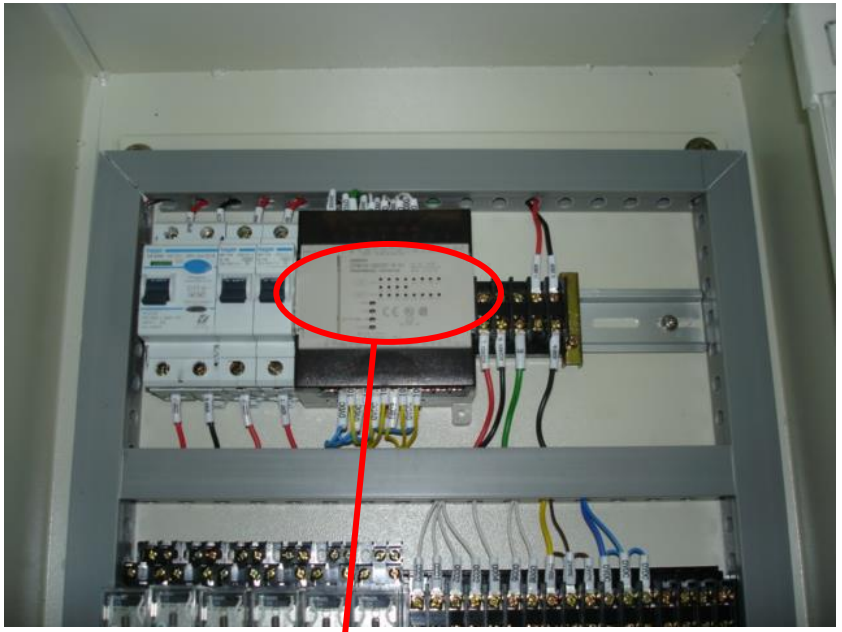
PLC

# Maintenance

Difficult



Easy



LED indicator

How ?

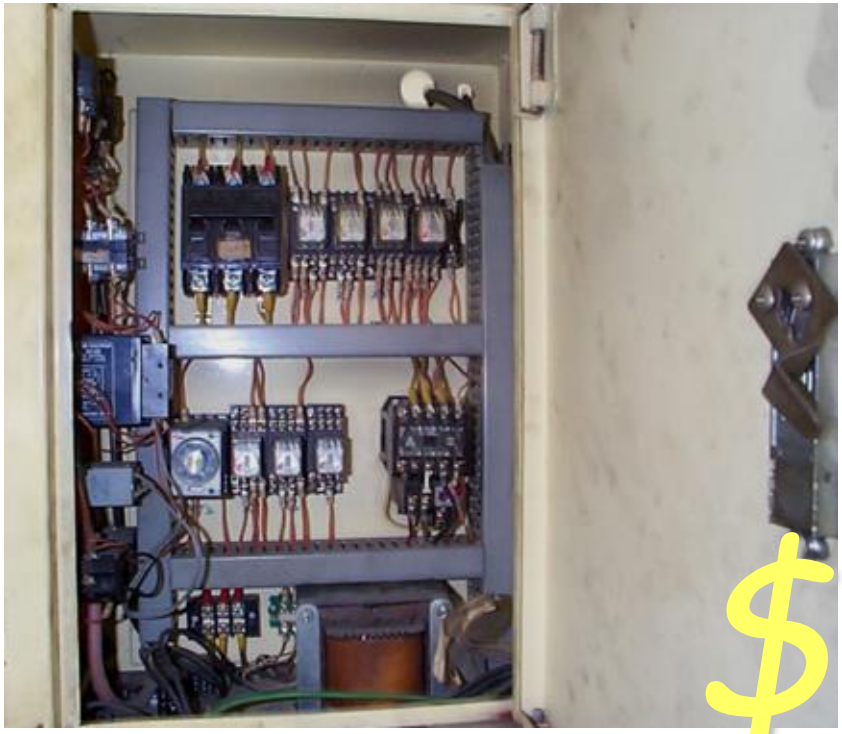
Wired Logic

Input/output connector can be remove easily

PLC

# Economic Efficiency

Advantage on small scale



Wired Logic

$$$

Advantage on small, medium and large

PLC

# Power Consumption

High

> 2 Amp

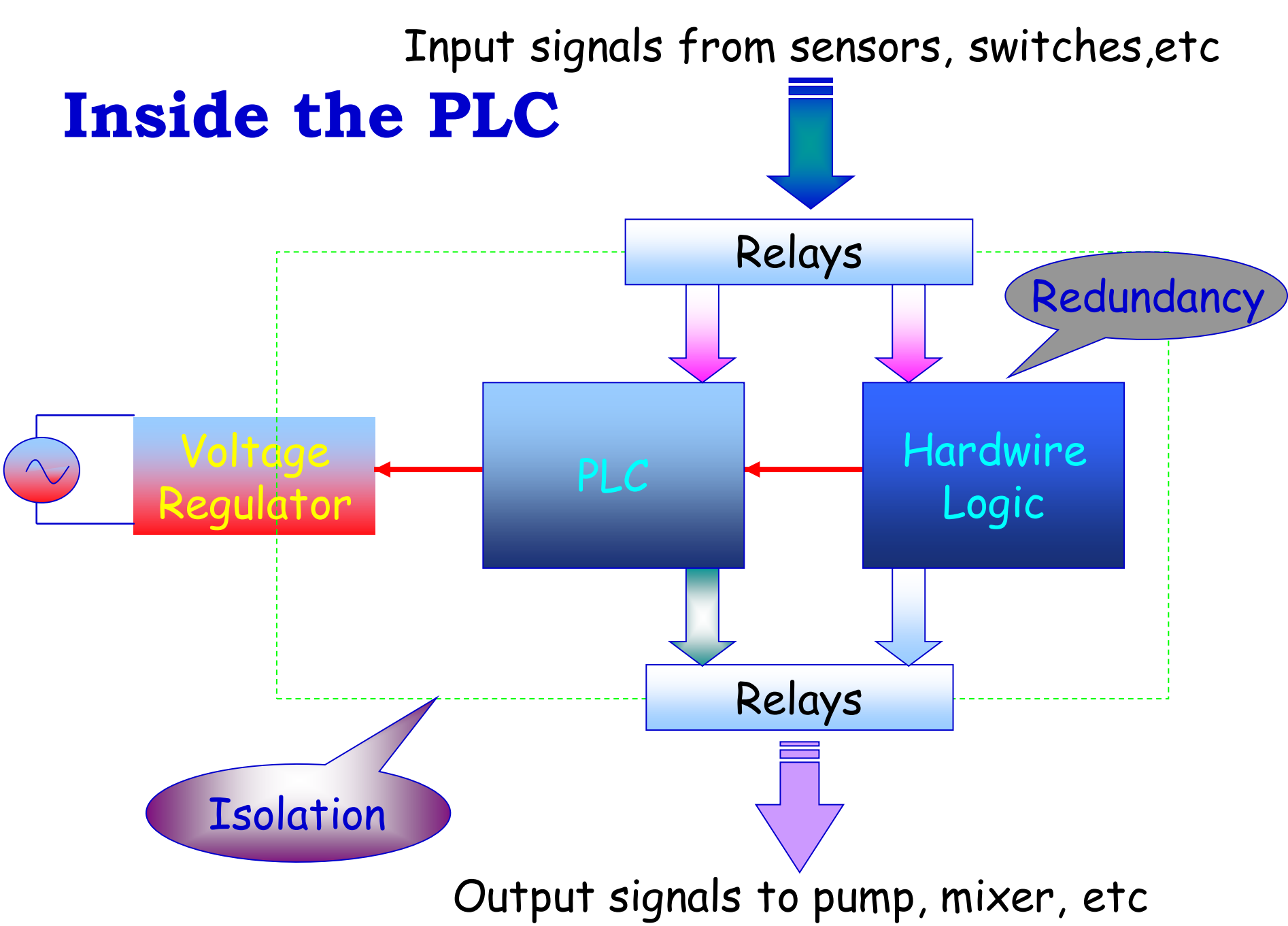
Wired Logic

Low

0.2 Amp

PLC

# Inside the PLC



# Inside the PLC

- **Input Relays** - (contacts)
  - connected to the outside world
  - physically exist and receive signals from switches, sensors, etc..
- **Internal Relays** - (contacts)
  - do not receive signals from the outside world nor do they physically exist
  - simulated relays and are what enables a PLC to eliminate external relays
  - also some special relays that are dedicated to performing only one task

# Inside the PLC

- **Counters**
  - do not physically exist
  - simulated counters and they can be programmed to count pulses
  - limited in their counting speed
  - high-speed counters that are hardware based
- **Timers**
  - do not physically exist
  - on-delay and off reset
  - increments vary from 10ms through 100ms

# Inside the PLC

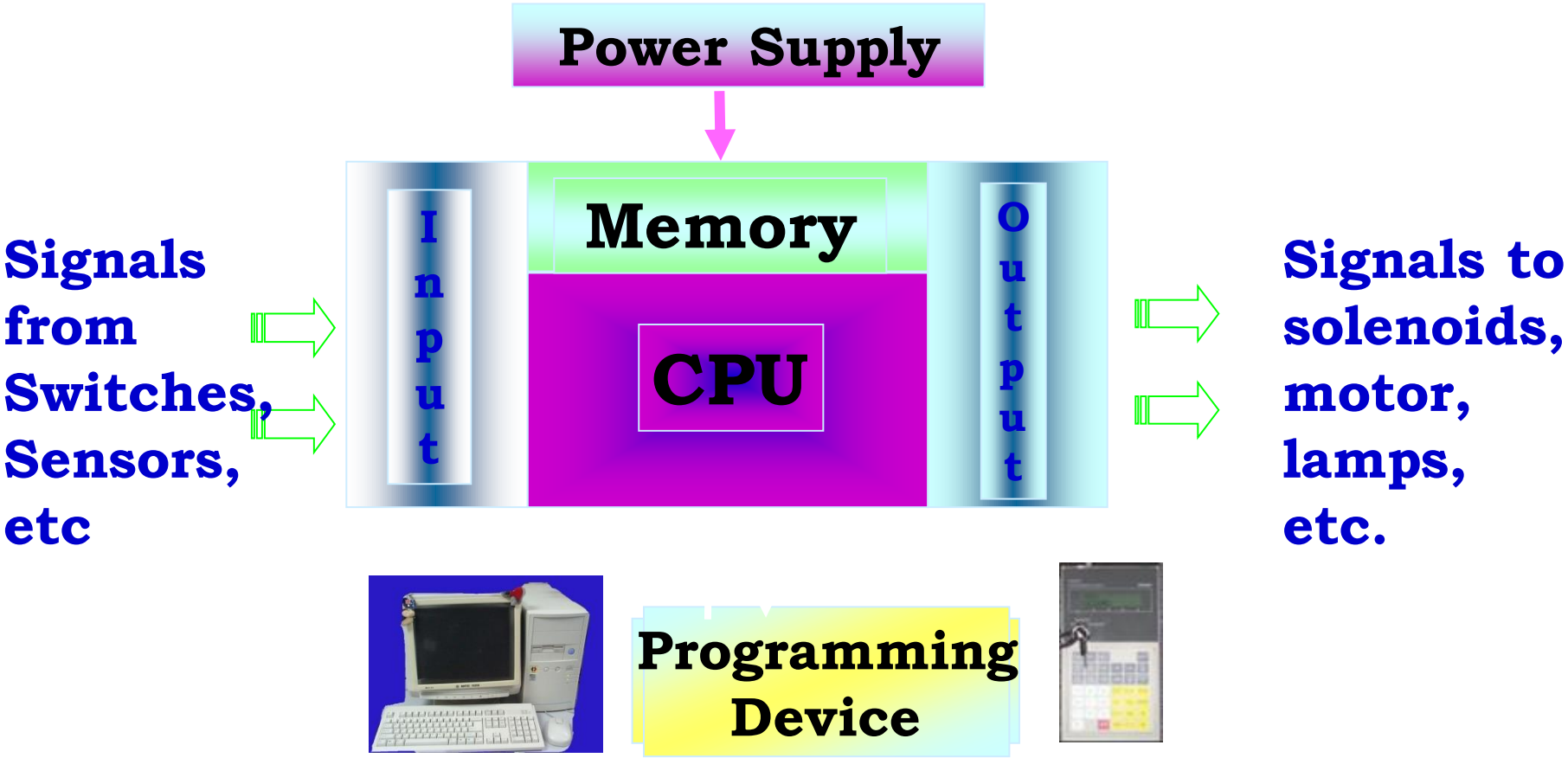
- **Output Relays** (coils)
  - connected to the outside world
  - physically exist and send on/off signals to solenoids, lights, etc
  - can be transistors, relays, or triacs depending upon the model chosen
- **Data Memory**
  - registers assigned to store data for math or data manipulation
  - can also typically be used to store data when power is removed from the PLC

# Types of Memory

- ROM: Read Only Memory
- RAM: Temporary storage area, need battery backup
- EPROM: Erasable Programmable Read Only Memory
- EEPROM: Electrical Erasable Programmable Read Only Memory

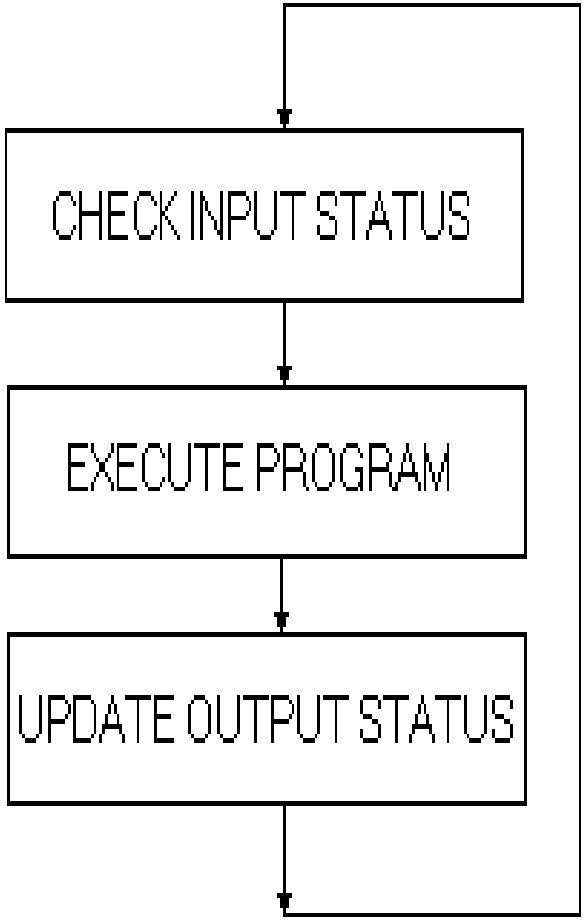
♦ Note :Battery is used for backup the data in the RAM in PLC when Power Off

# How PLC Work?



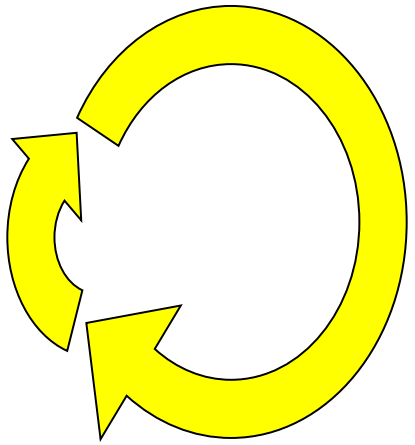

A PLC monitors inputs, makes decisions based on its program, and controls outputs to automate a process or machine.

# How PLC Work?

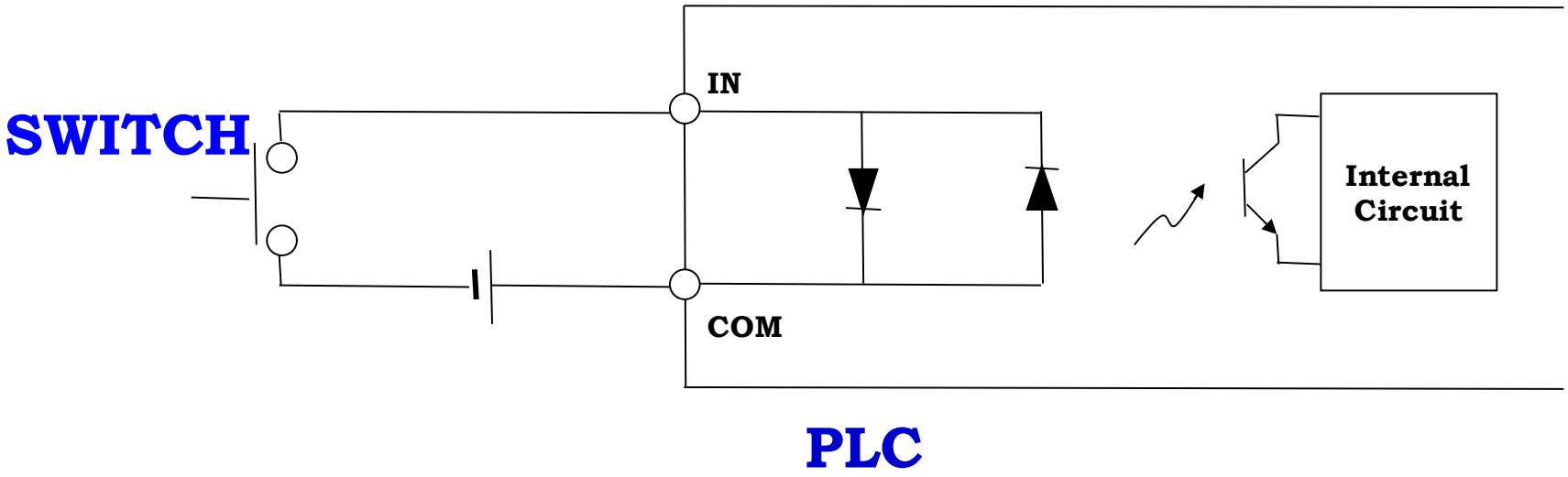


# Response Time (Scan Time)

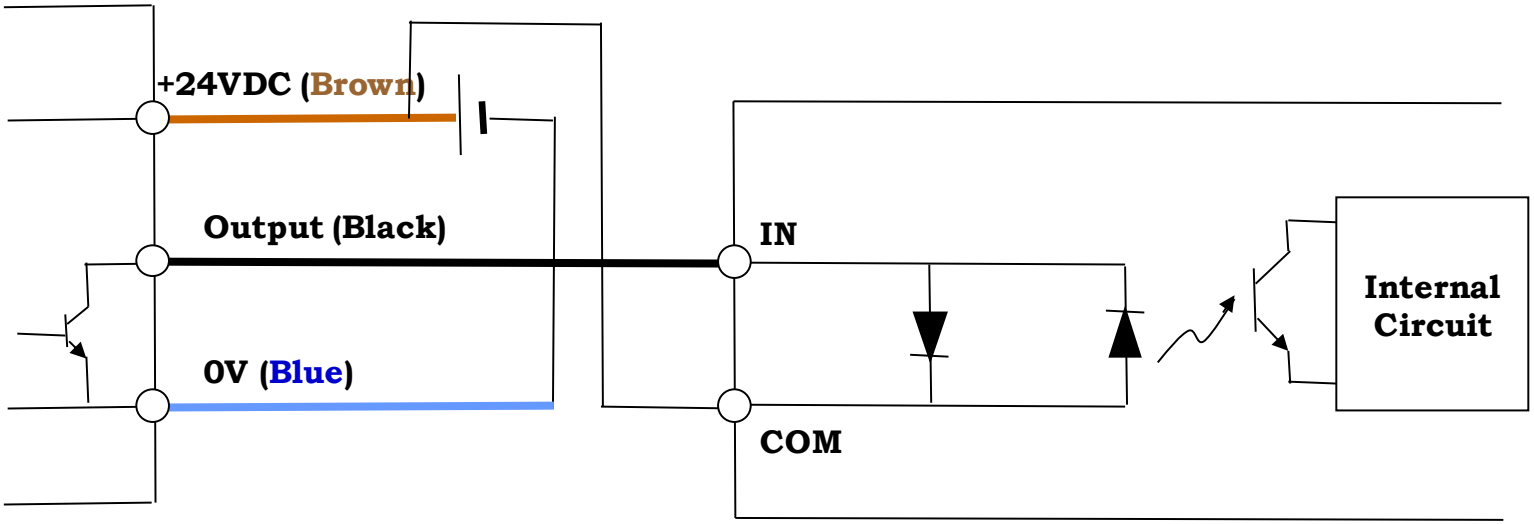
- Just like our brains, the PLC takes a certain amount of time to react to changes
- Total PLC Response Time is a summation of
  - Input Response Time
  - Program Execution Time
  - Output Response Time
- Factor Influence Scan Time
  1) CPU speed
  2) Length/Size of program
  3) Number of I/O
  4) Remote I/O sub-system



# Input Wiring (Switch & PLC)



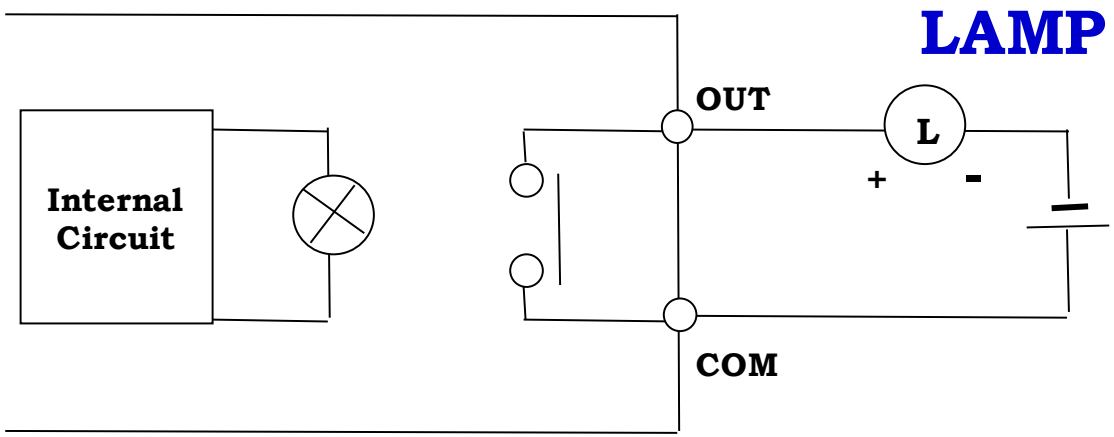
# Input Wiring (Sensor & PLC)



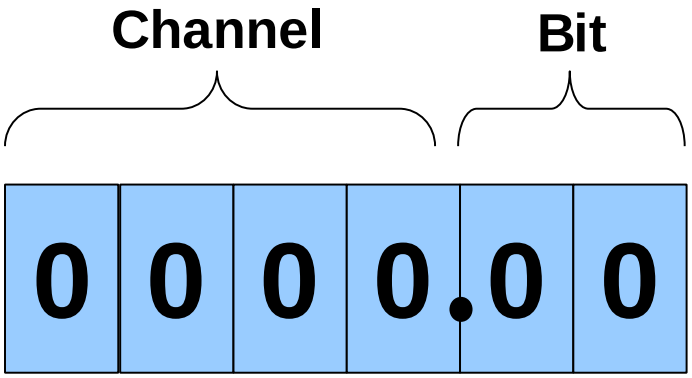

**24VDC NPN SENSOR**

**PLC**

# Output Wiring (Lamp & PLC)



# PLC I/O Addressing




1 Channel / 1 Word = 16 Bits

i.e.: Input: 000.00, 000.01, 000.02, .....000.14, 000.15,
Output: 100.00, 100.01, .....100.15

# CPU Status Indicators

| Indicator | Status | Meaning |
|---|---|---|
| PWR (Green) | ON<br>OFF | Power supply to PLC<br>Power not supply to PLC |
| RUN (Green) | ON<br>OFF | RUN or MONITOR mode<br>PROGRAM mode or Fatal Error occurred |
| ERR/ALM (Red) | ON<br>Flash<br>OFF | Fatal Error (PLC operation stops)<br>Non-fatal Error (PLC operation continue)<br>CPU unit operate normally |
| COMM (Green) | ON<br>OFF | Data being transfer<br>Data not being transfer |
| PRPHL (Green) | ON<br>OFF | Data being transfer via Peripheral Port<br>Data not being transfer via Peripheral Port |
| BKUP (Yellow) | ON<br>OFF | Access Backup Memory<br>Not access Backup Memory |
| INH (Yellow) | ON<br>OFF | Output OFF Bit turn On<br>Output OFF Bit turn Off |

# PLC Programming Tools

## CX-Programmer Software

# PLC Operating Mode

1) PROGRAM Mode...

- Programs are not executed.

- Used for inserting or modification existing program.

2) MONITOR Mode...

- Programs are executed.

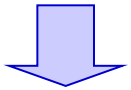
- Online editing, and changes to present value in I/O memory, are enabled in this mode.

3) RUN Mode...

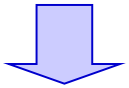
- Programs are executed. (No changes can be made)

# History of Omron Software Tools for PLC Programming

LSS, Ladder Support Software (DOS)

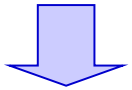


SSS, Sysmac Support Software (DOS)



SYSWIN Ver 3.4

(16 bit Windows Application)



CX-Programmer

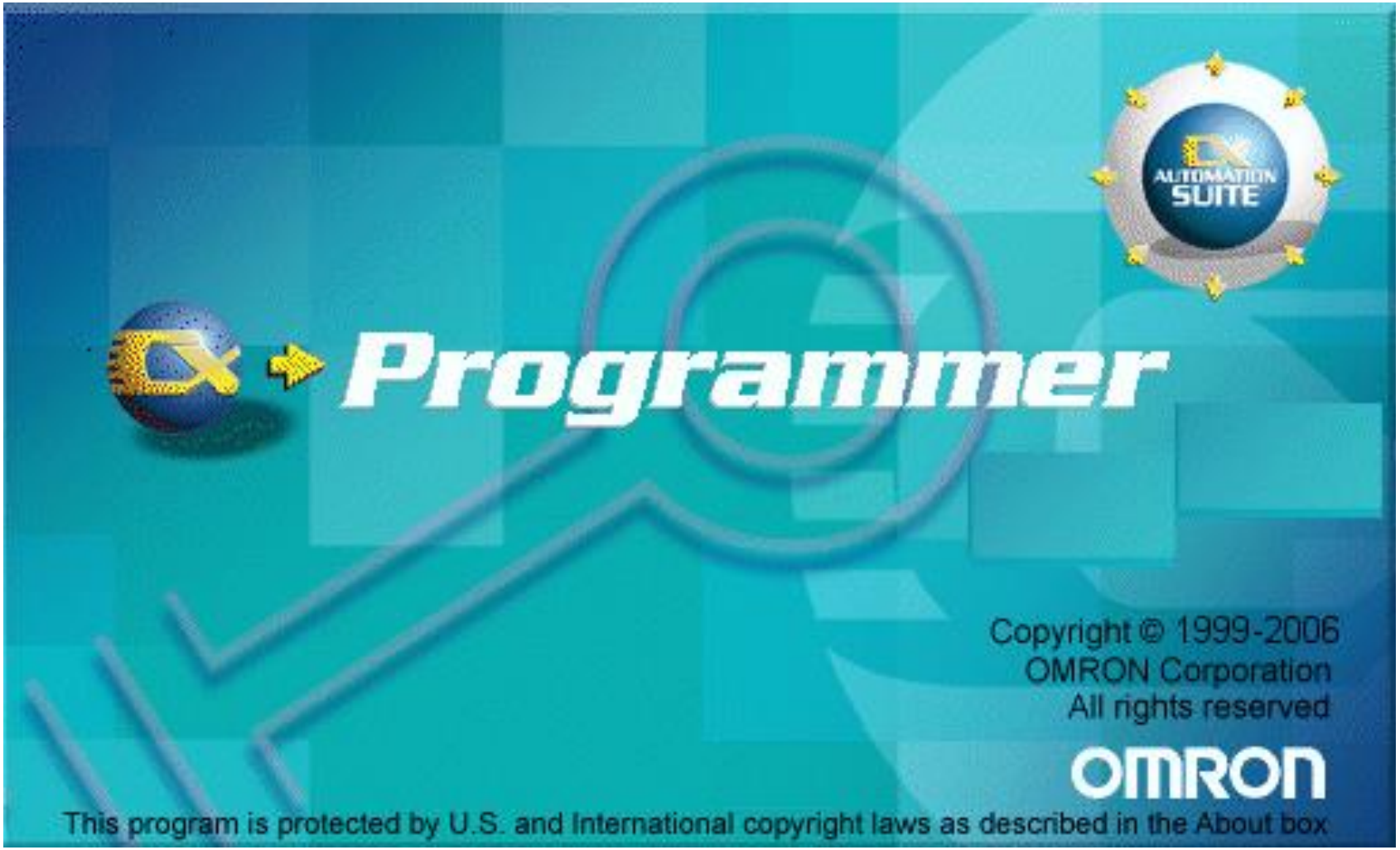
(32 bit Windows Application)

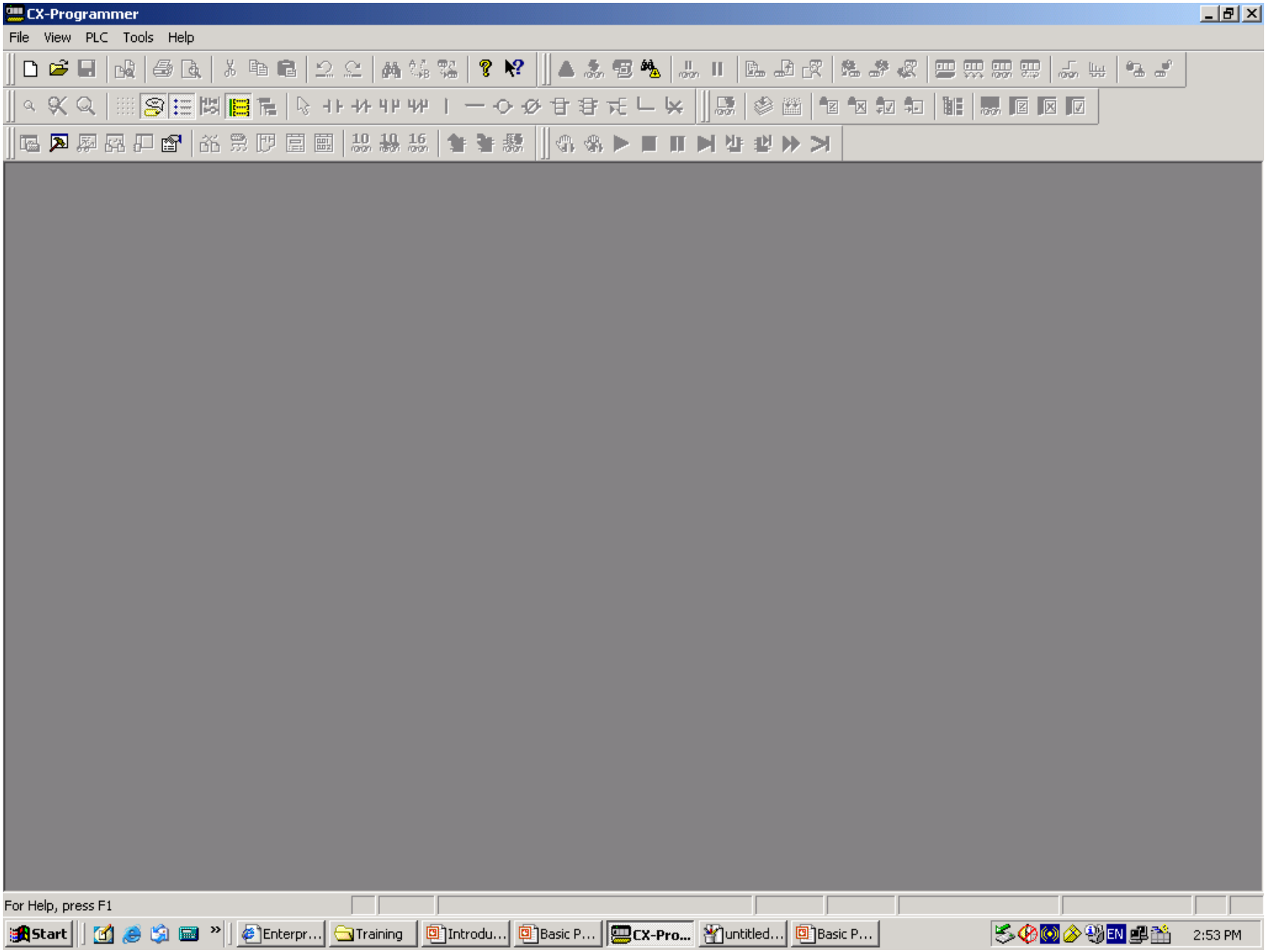
*Now CX-Programmer is included in CX-One*

# CX-Programmer Features

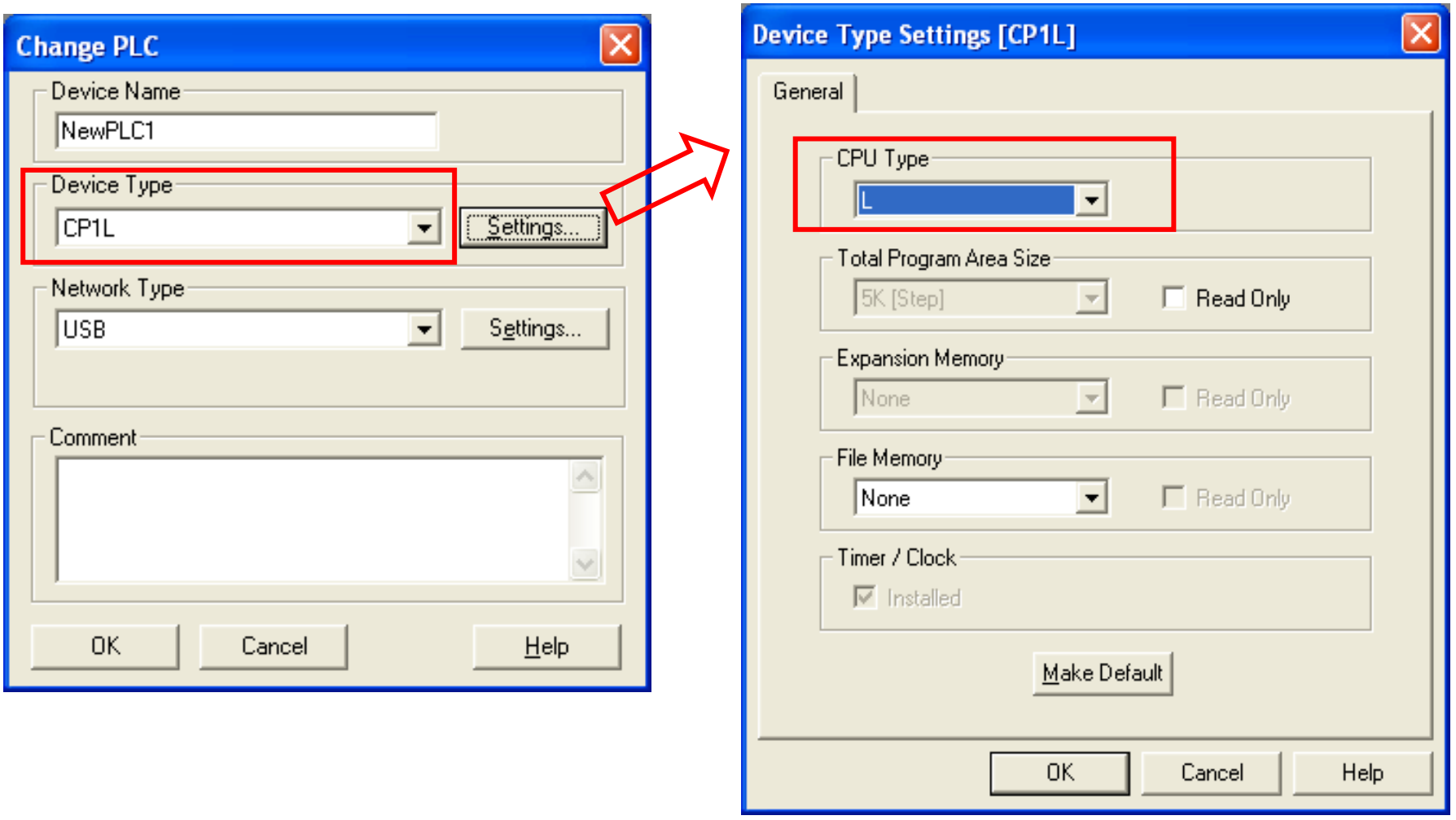
- ✯ Programming in Ladder, Mnemonic Code or Function Block
- ✯ Multiple window display
- ✯ On-line editing and monitoring
- ✯ Extensive I/O and comment documentation
- ✯ Communications support for ToolBus, SYSMACWAY, Ethernet, USB, etc.

# Get Started

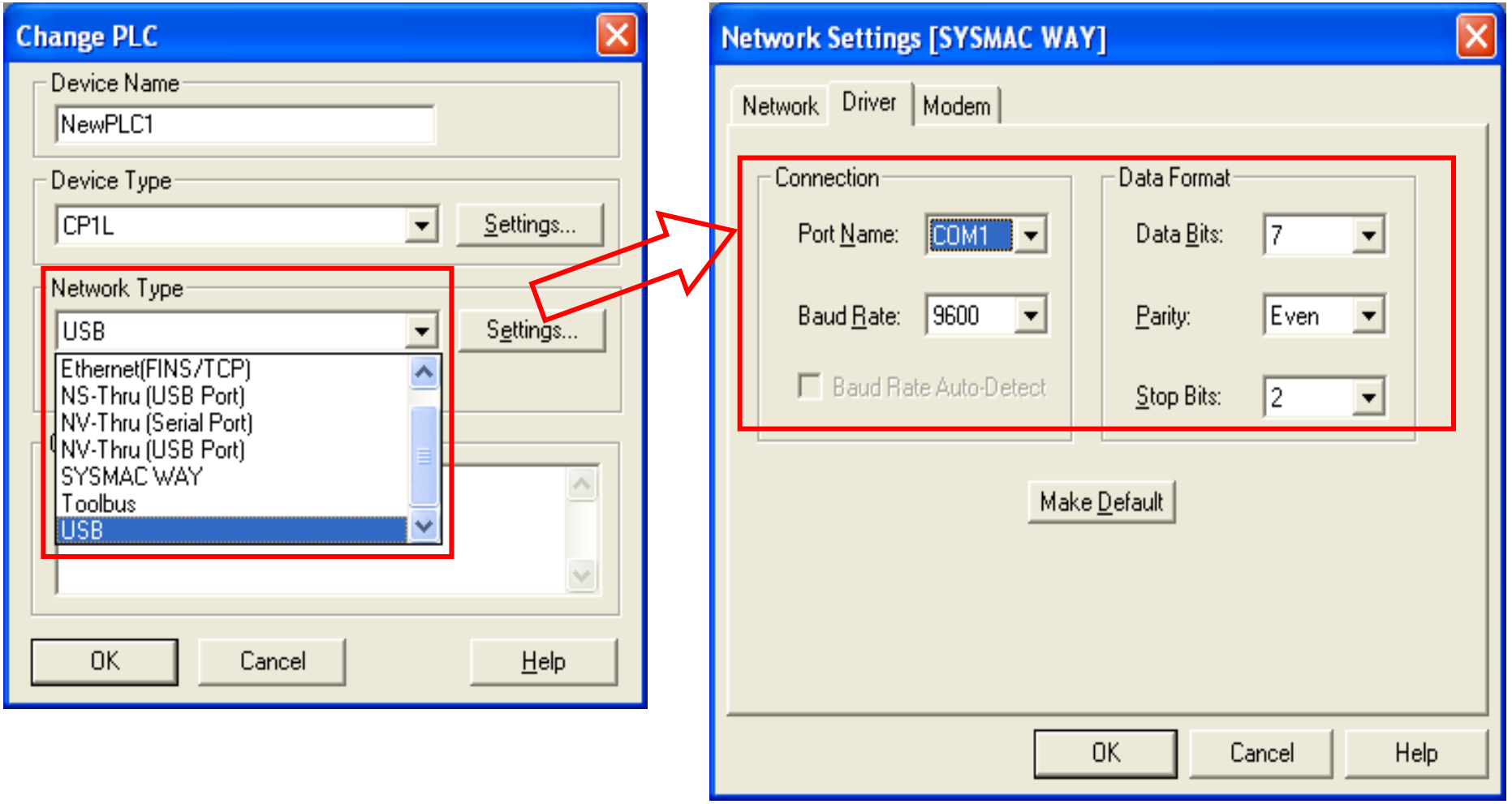

AUTOMATION SUITE
Programmer
Copyright © 1999-2006
OMRON Corporation
All rights reserved
OMRON
This program is protected by U.S. and International copyright laws as described in the About box

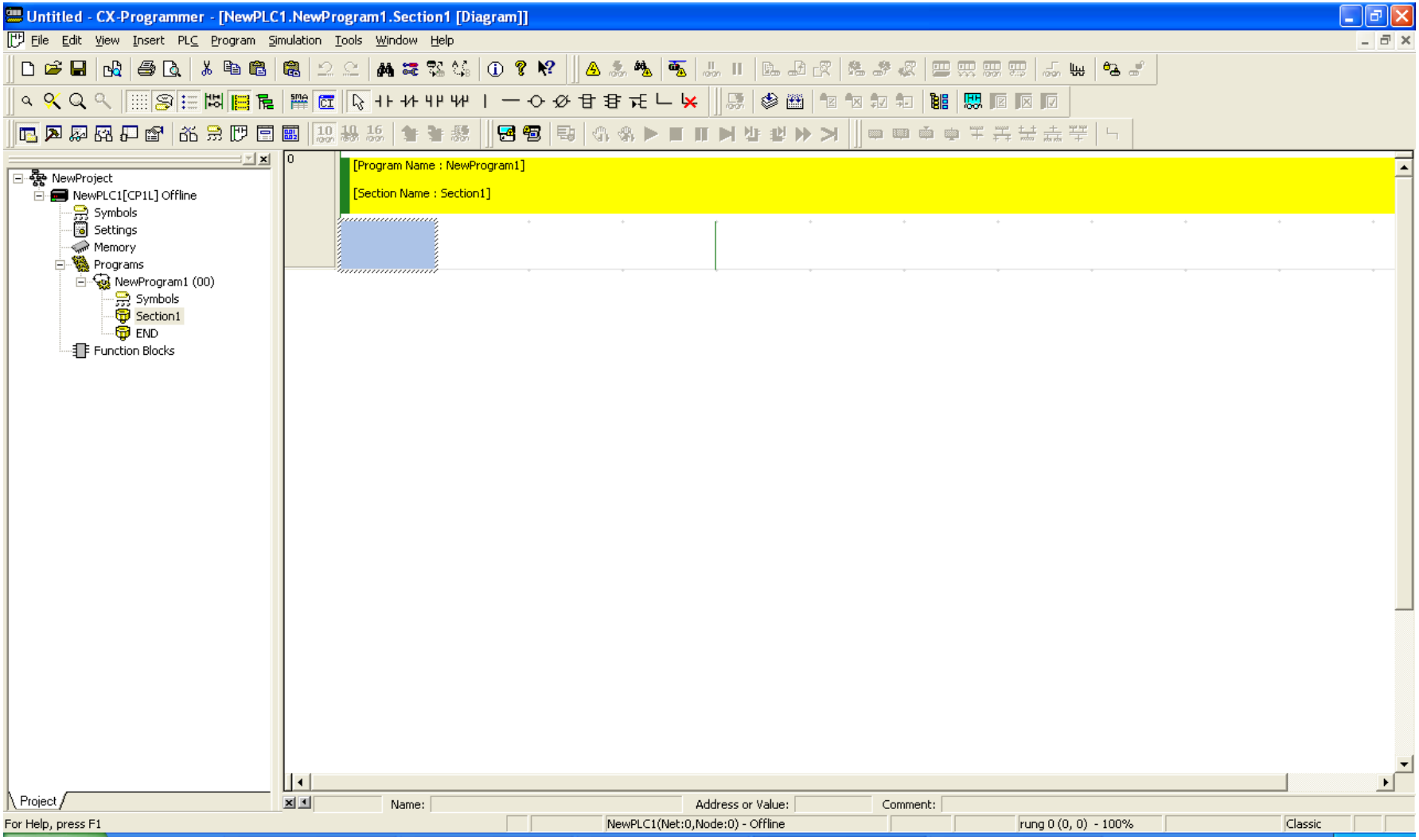

CX-Programmer
File View PLC Tools Help
For Help, press F1
Start
Enterpr...
Training
Introdu...
Basic P...
CX-Pro...
untitled...
Basic P...
EN
2:53 PM

# Set PLC Model & CPU Type



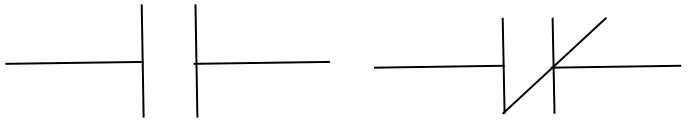
# Set Communication Interface




Untitled - CX-Programmer - [NewPLC1.NewProgram1.Section1 [Diagram]]
File Edit View Insert PLC Program Simulation Tools Window Help
NewProject
NewPLC1[CP1L] Offline
Symbols
Settings
Memory
Programs
NewProgram1 (00)
Symbols
Section1
END
Function Blocks
Project
0
[Program Name : NewProgram1]
[Section Name : Section1]
Name:
Address or Value:
Comment:
For Help, press F1
NewPLC1(Net:0,Node:0) - Offline
rung 0 (0, 0) - 100%
Classic

# Simple Circuit for First Trial

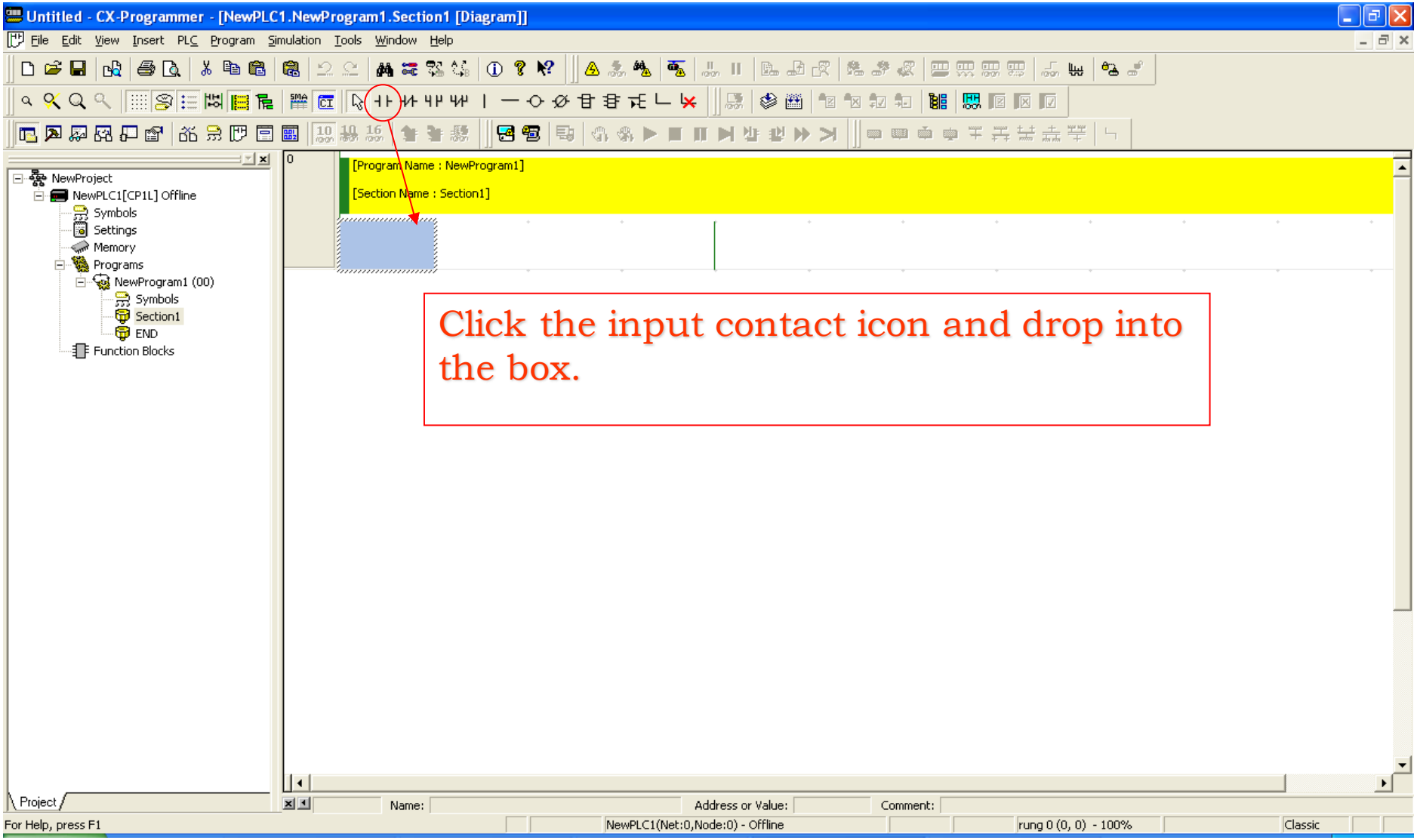
# LD / LD NOT

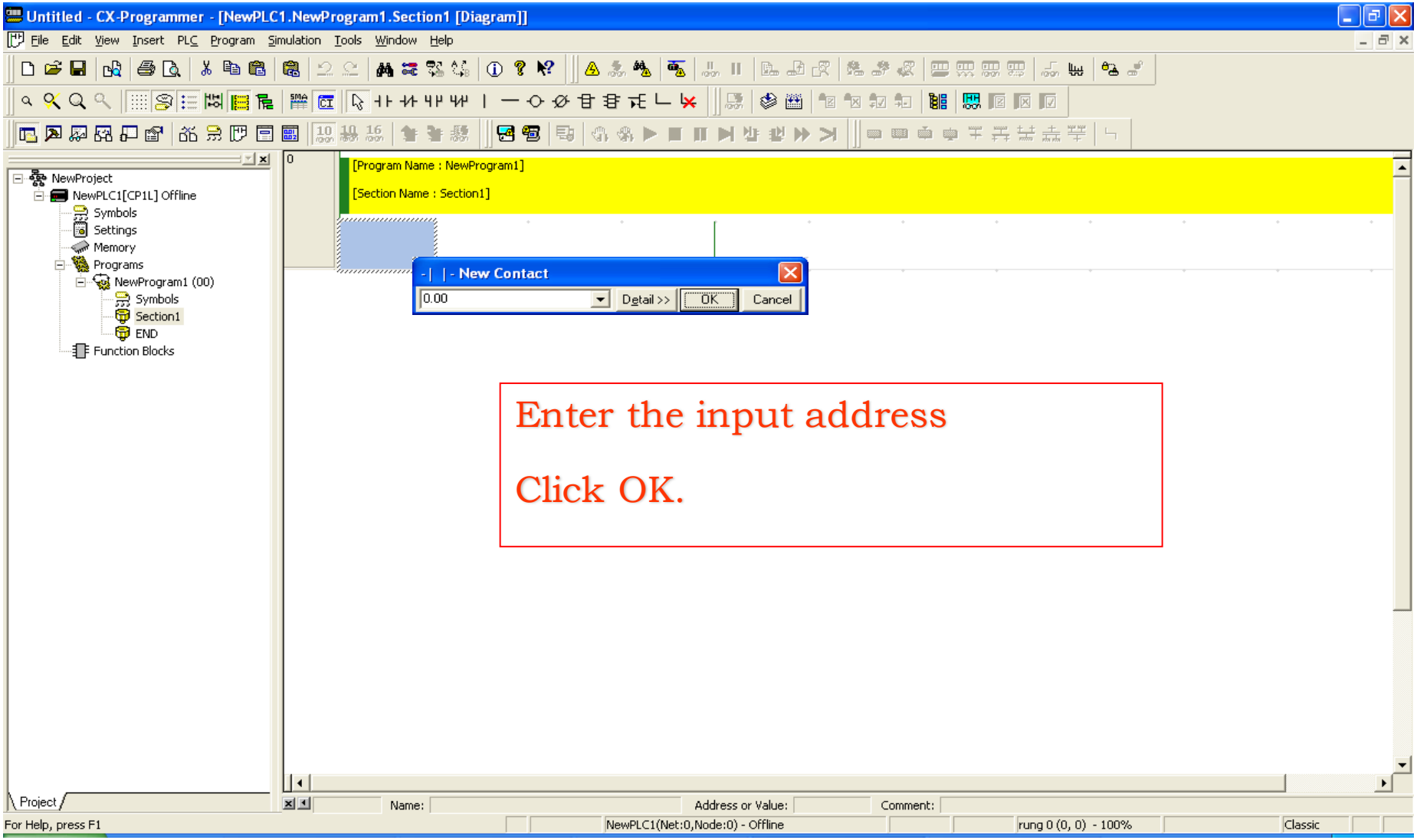


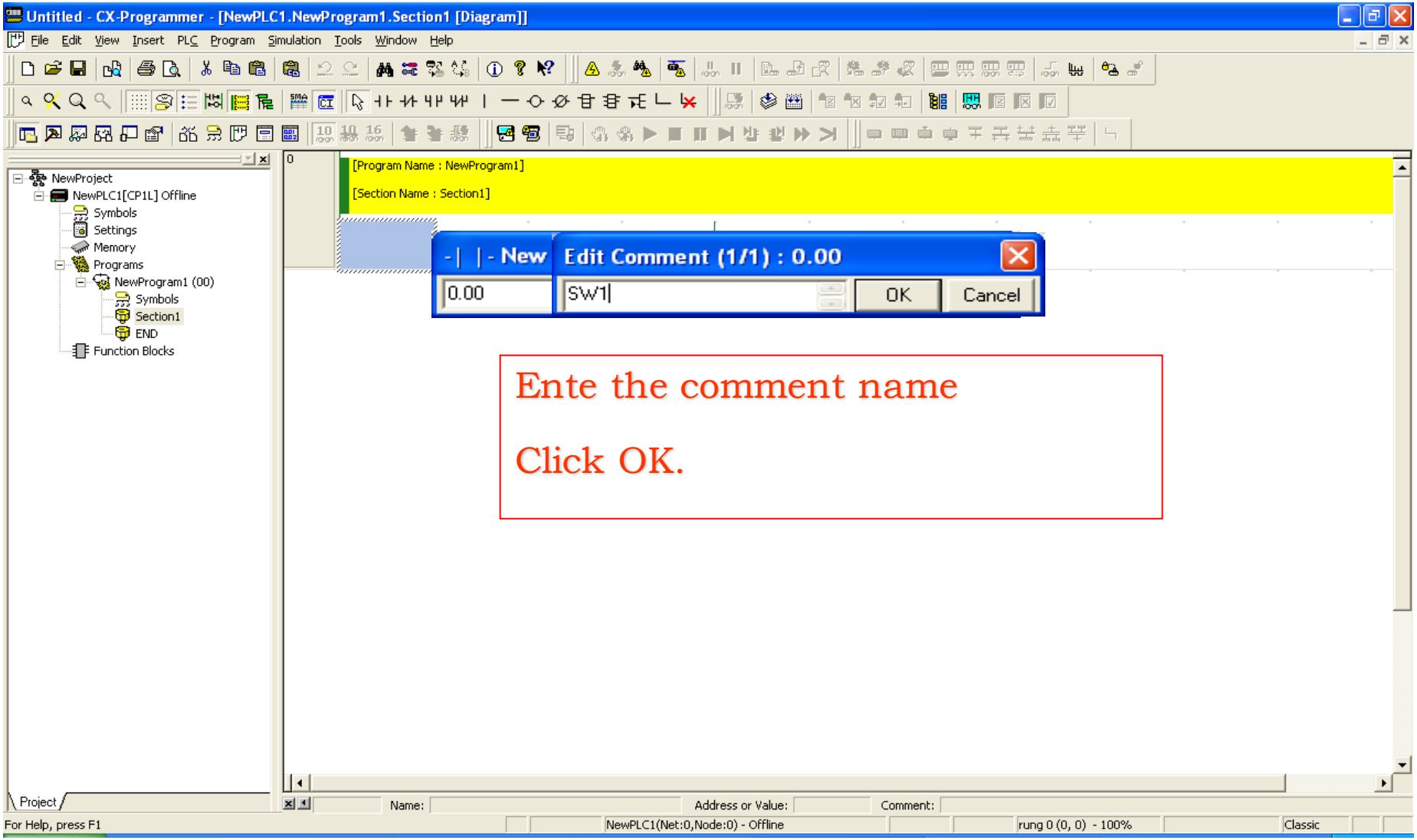
- The first condition that starts any logic block within a ladder diagram corresponds to a
  LD or LD NOT instruction.

- LD - Normally Open contact, examine if On

- LD NOT - Normally Close contact, examine if Off


Untitled - CX-Programmer - [NewPLC1.NewProgram1.Section1 [Diagram]]
File Edit View Insert PLC Program Simulation Tools Window Help
NewProject
NewPLC1[CP1L] Offline
Symbols
Settings
Memory
Programs
NewProgram1 (00)
Symbols
Section1
END
Function Blocks
Project
0
[Program Name : NewProgram1]
[Section Name : Section1]
Click the input contact icon and drop into the box.
Name:
Address or Value:
Comment:
For Help, press F1
NewPLC1(Net:0,Node:0) - Offline
rung 0 (0, 0) - 100%
Classic

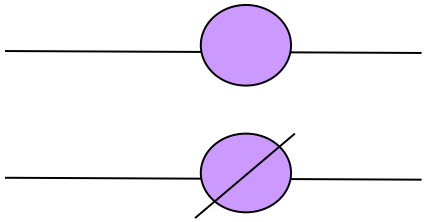

Untitled - CX-Programmer - [NewPLC1.NewProgram1.Section1 [Diagram]]
File Edit View Insert PLC Program Simulation Tools Window Help
NewProject
NewPLC1[CP1L] Offline
Symbols
Settings
Memory
Programs
NewProgram1 (00)
Symbols
Section1
END
Function Blocks
Project
0
[Program Name : NewProgram1]
[Section Name : Section1]
- | | - New Contact
0.00
Detail >>
OK
Cancel
Enter the input address
Click OK.
Name:
Address or Value:
Comment:
For Help, press F1
NewPLC1(Net:0,Node:0) - Offline
rung 0 (0, 0) - 100%
Classic



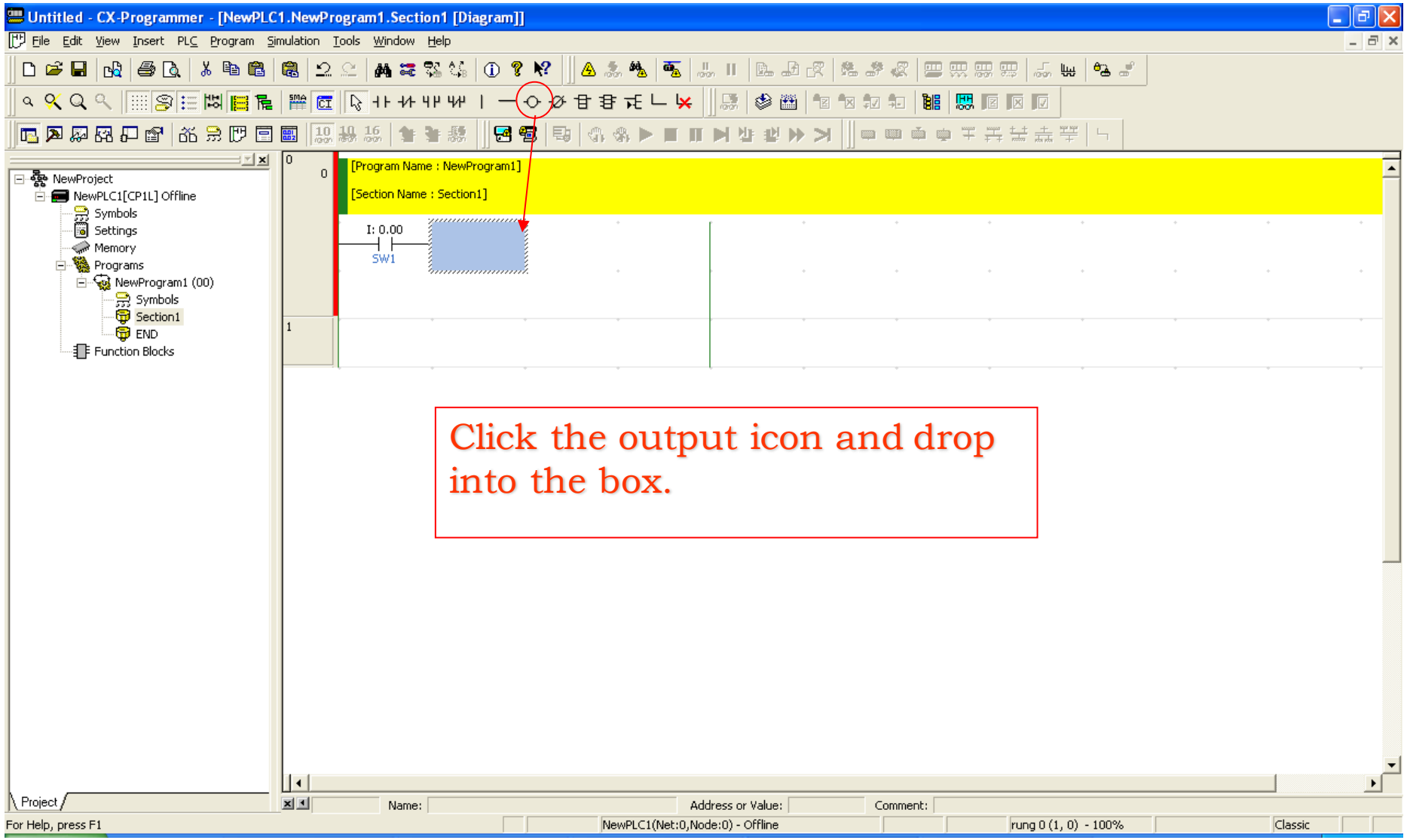

Ente the comment name

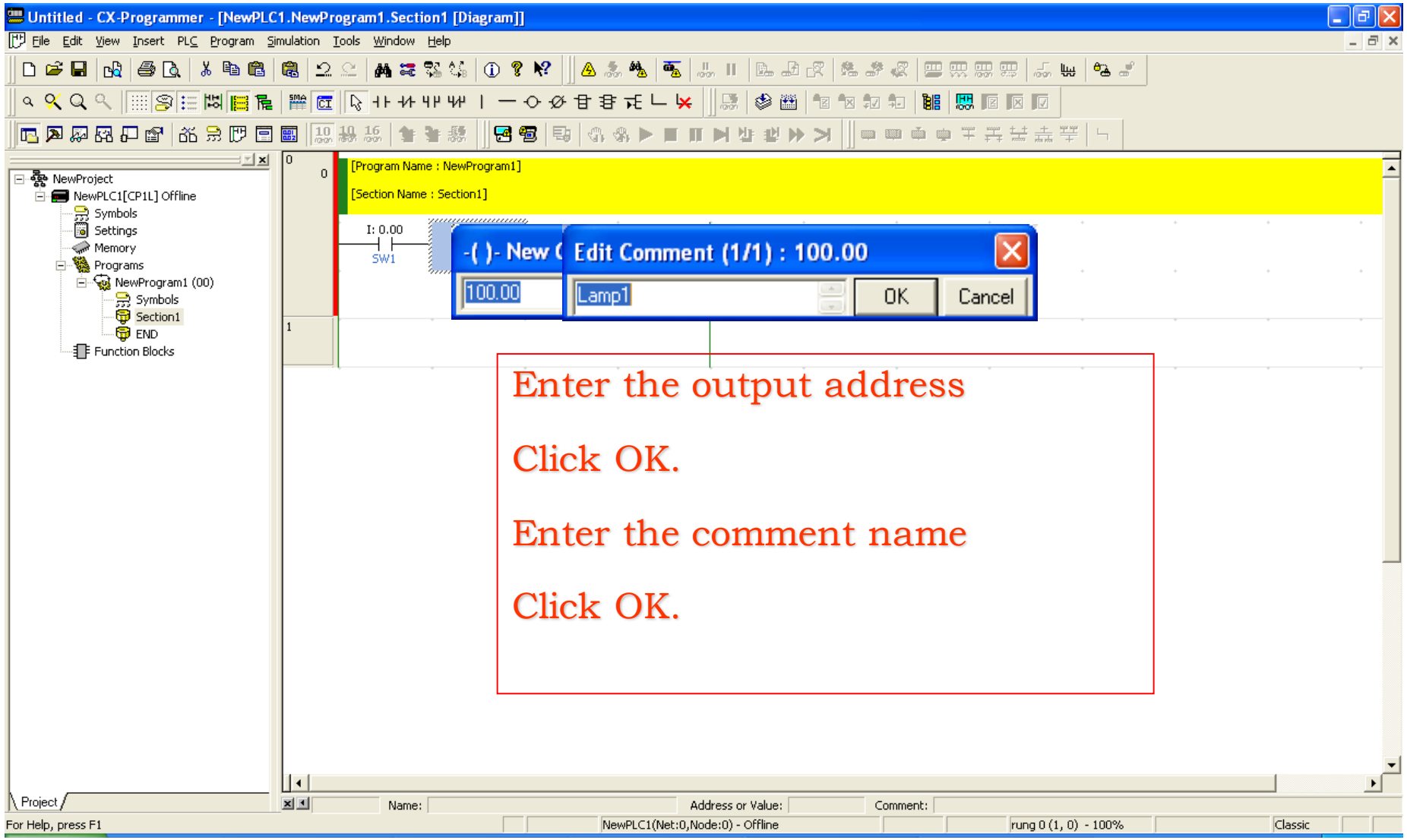
Click OK.

# OUT / OUT NOT



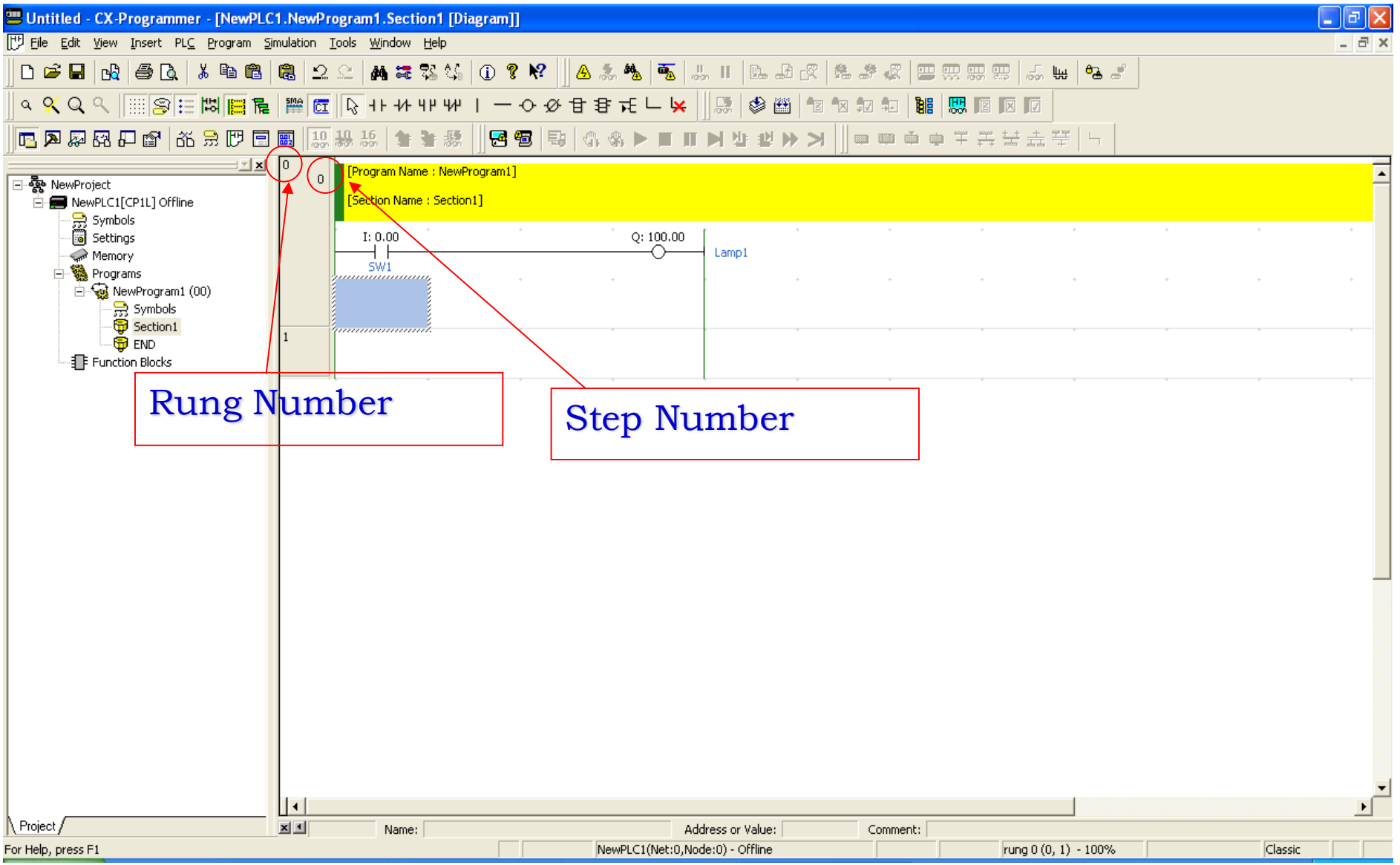
- The OUT instruction is like a relay coil.
- When there is a path of True instruction preceding this on the ladder rung, it will also be True.
- When the instruction is True it is physically On.
- We can think of this instruction as a normally open output.
- This instruction can be used for internal coils and external outputs.
- The OUT NOT instruction is the reverse logic of OUT instruction.


Untitled - CX-Programmer - [NewPLC1.NewProgram1.Section1 [Diagram]]
File Edit View Insert PLC Program Simulation Tools Window Help
NewProject
NewPLC1[CP1L] Offline
Symbols
Settings
Memory
Programs
NewProgram1 (00)
Symbols
Section1
END
Function Blocks
[Program Name : NewProgram1]
[Section Name : Section1]
I: 0.00
SW1
Click the output icon and drop into the box.
Project
Name:
Address or Value:
Comment:
For Help, press F1
NewPLC1(Net:0,Node:0) - Offline
rung 0 (1, 0) - 100%
Classic

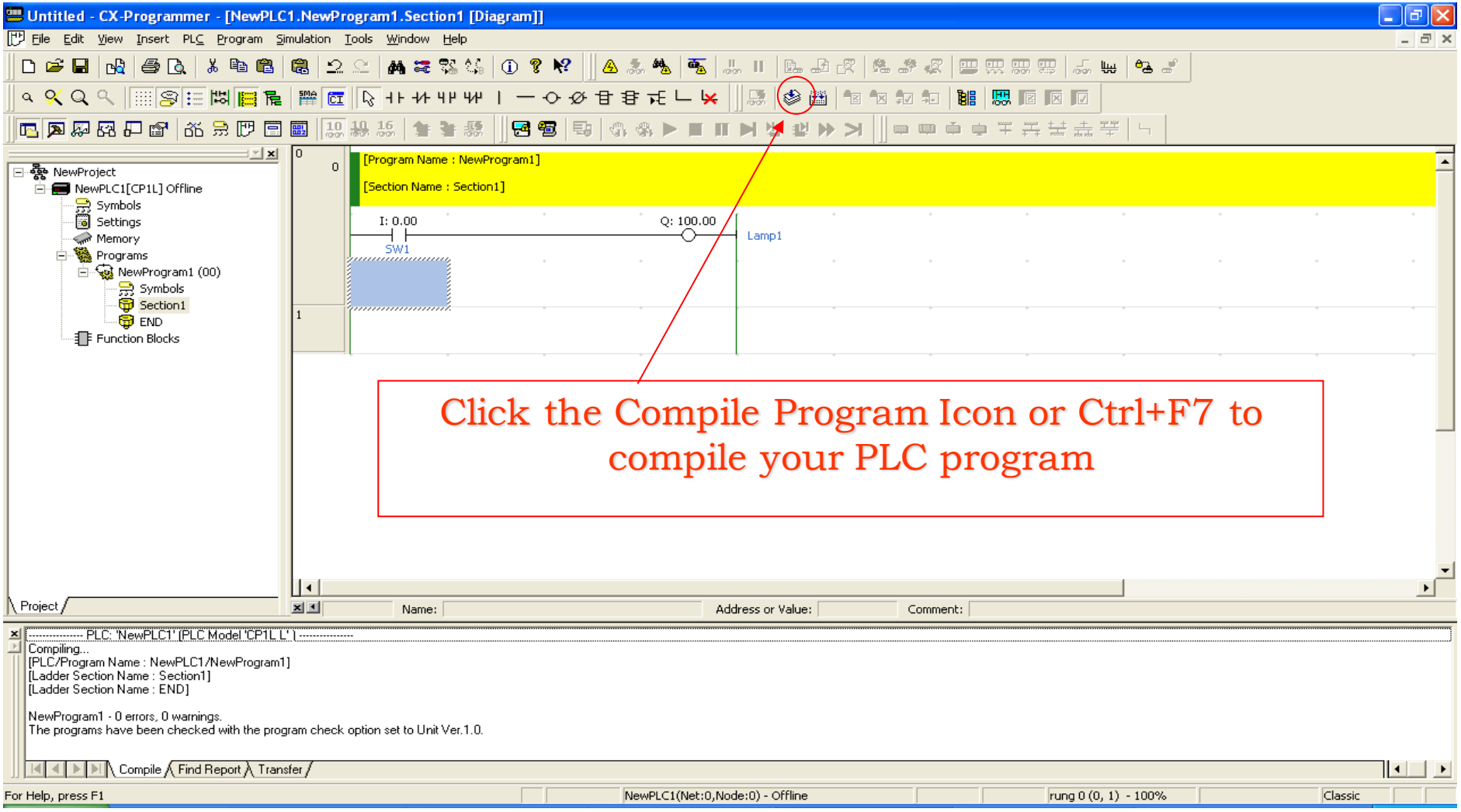

Untitled - CX-Programmer - [NewPLC1.NewProgram1.Section1 [Diagram]]
File Edit View Insert PLC Program Simulation Tools Window Help
NewProject
NewPLC1[CP1L] Offline
Symbols
Settings
Memory
Programs
NewProgram1 (00)
Symbols
Section1
END
Function Blocks
[Program Name : NewProgram1]
[Section Name : Section1]
I: 0.00
SW1
-( )- New
Edit Comment (1/1) : 100.00
100.00
Lamp1
OK
Cancel
Enter the output address
Click OK.
Enter the comment name
Click OK.
Project
Name:
Address or Value:
Comment:
For Help, press F1
NewPLC1(Net:0,Node:0) - Offline
rung 0 (1, 0) - 100%
Classic

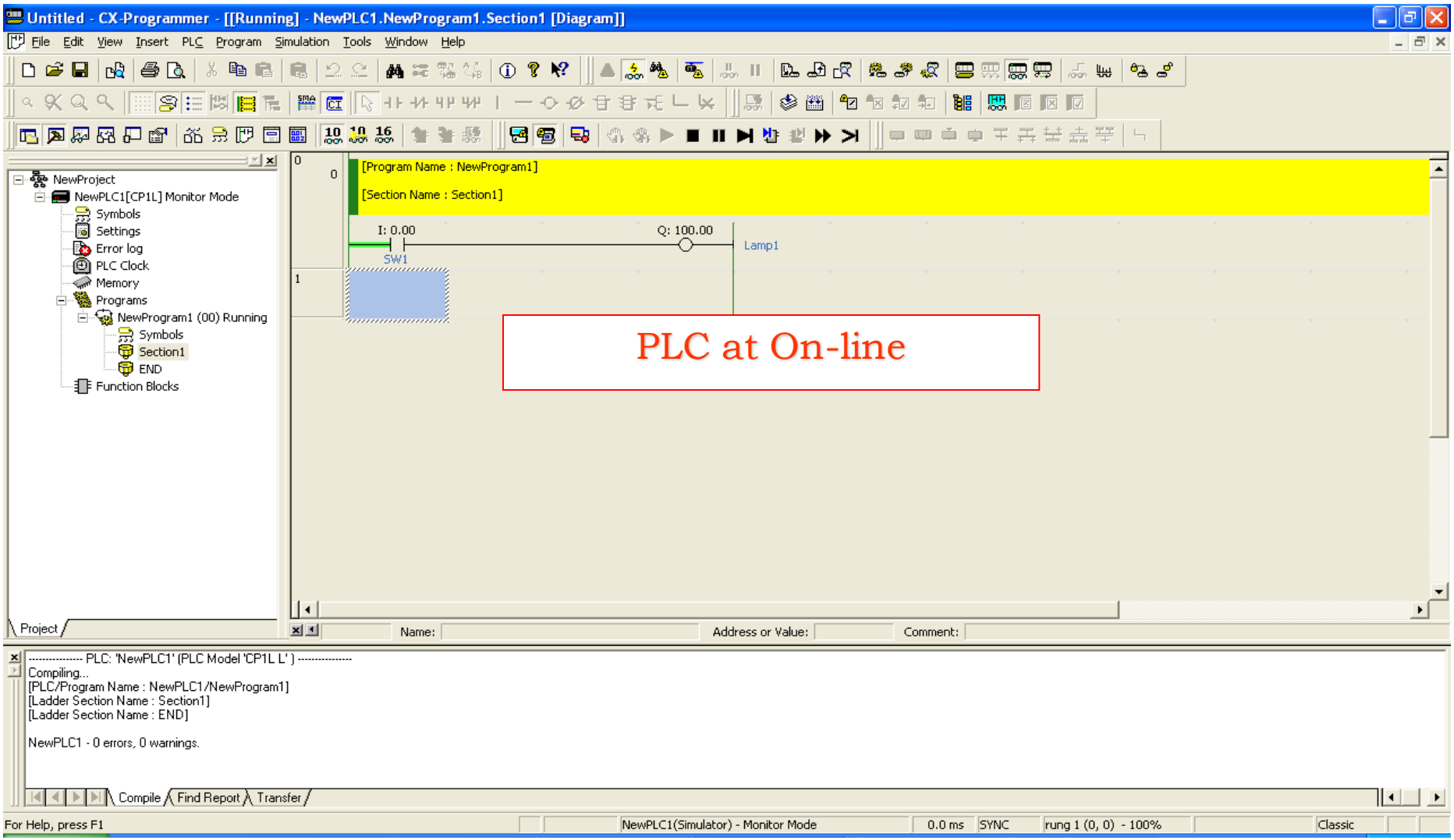
# END (01)

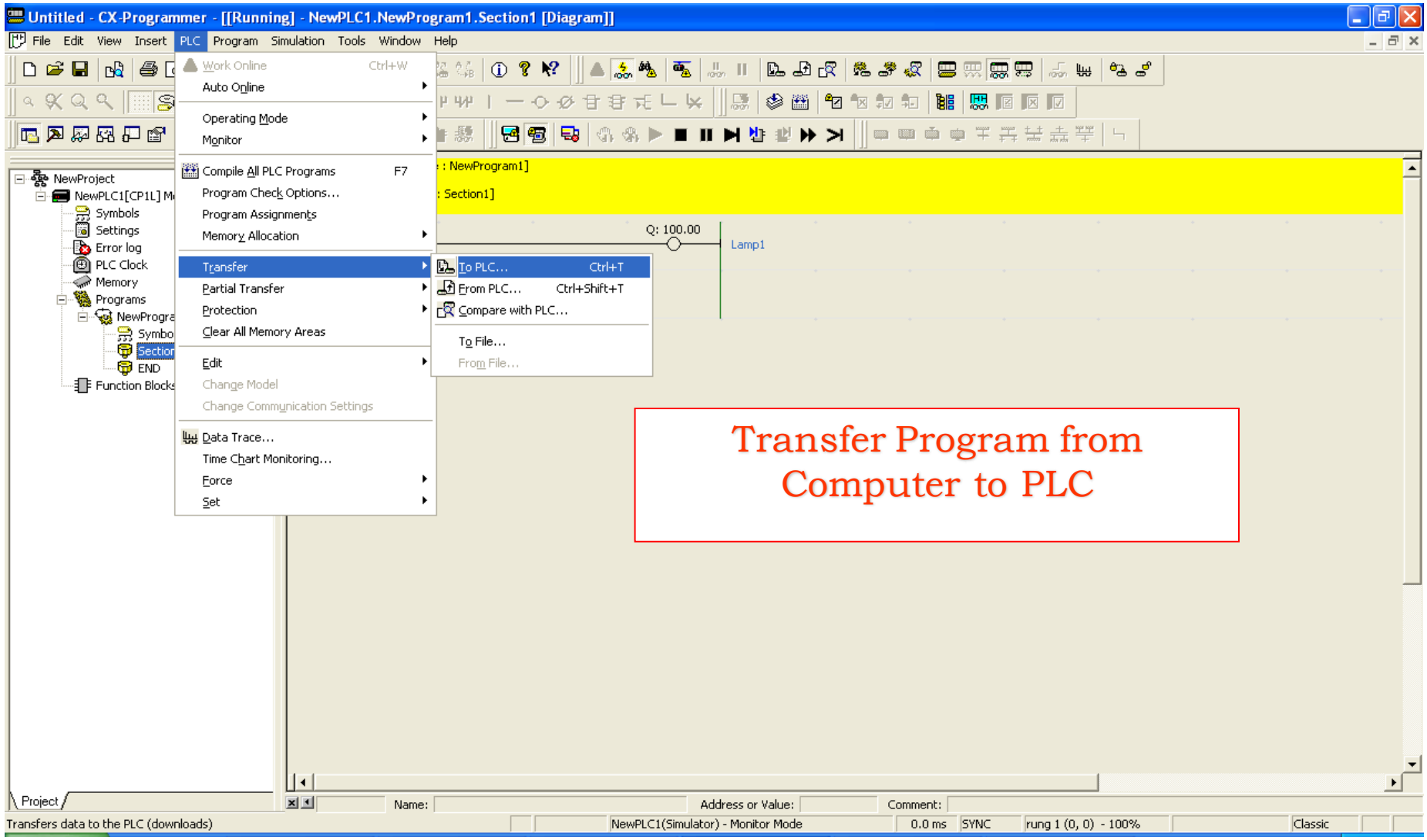
- The last instruction required to complete a simple program is the END instruction.
- When the CPU scan the program, it executes all instructions up to the first END instruction before returning to the beginning of the program and begin the execution again.

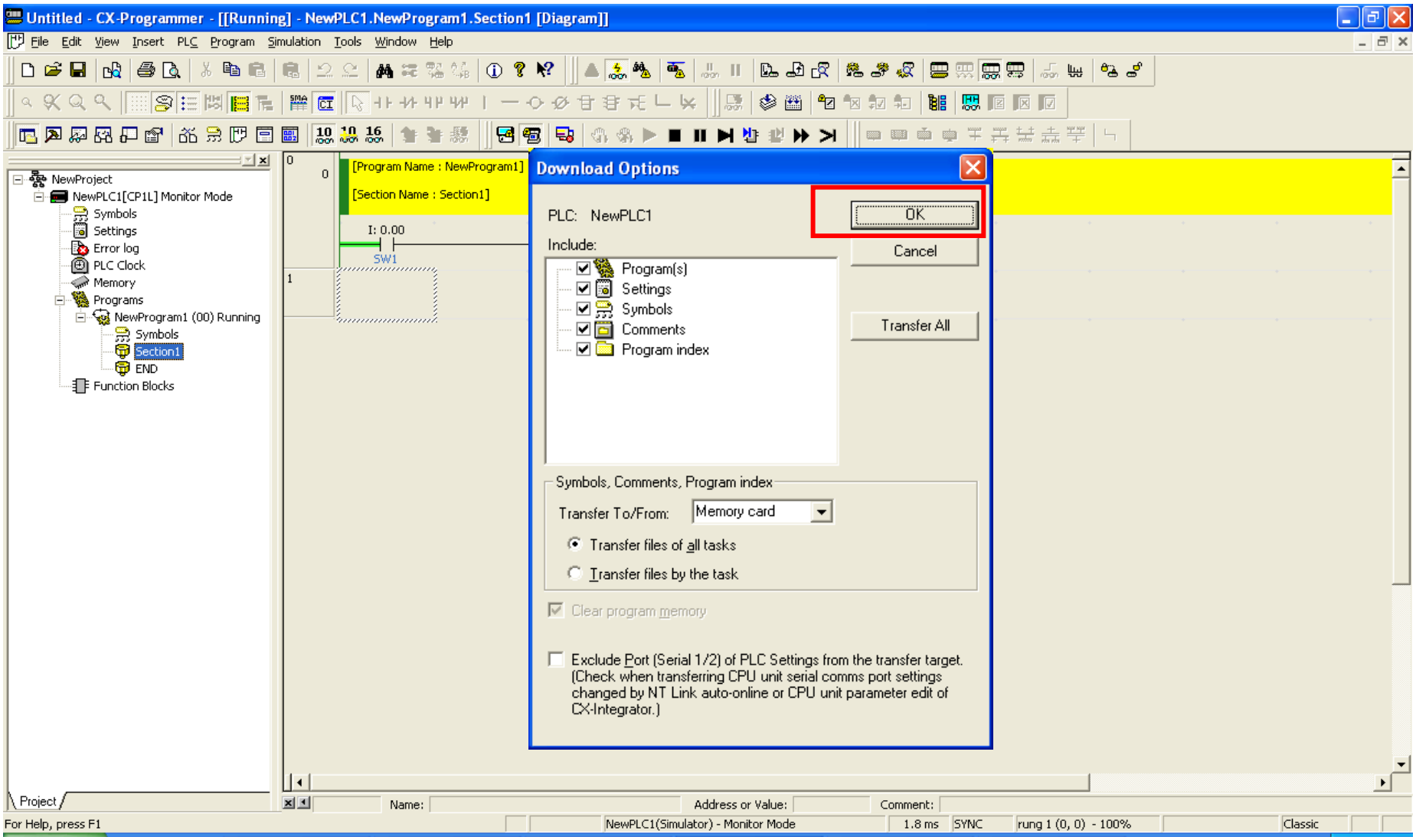

Untitled - CX-Programmer - [NewPLC1.NewProgram1.Section1 [Diagram]]
File Edit View Insert PLC Program Simulation Tools Window Help
NewProject
NewPLC1[CP1L] Offline
Symbols
Settings
Memory
Programs
NewProgram1 (00)
Symbols
Section1
END
Function Blocks
0
0
[Program Name : NewProgram1]
[Section Name : Section1]
I: 0.00
SW1
Q: 100.00
Lamp1
1
Rung Number
Step Number
Project
Name:
Address or Value:
Comment:
For Help, press F1
NewPLC1(Net:0,Node:0) - Offline
rung 0 (0, 1) - 100%
Classic

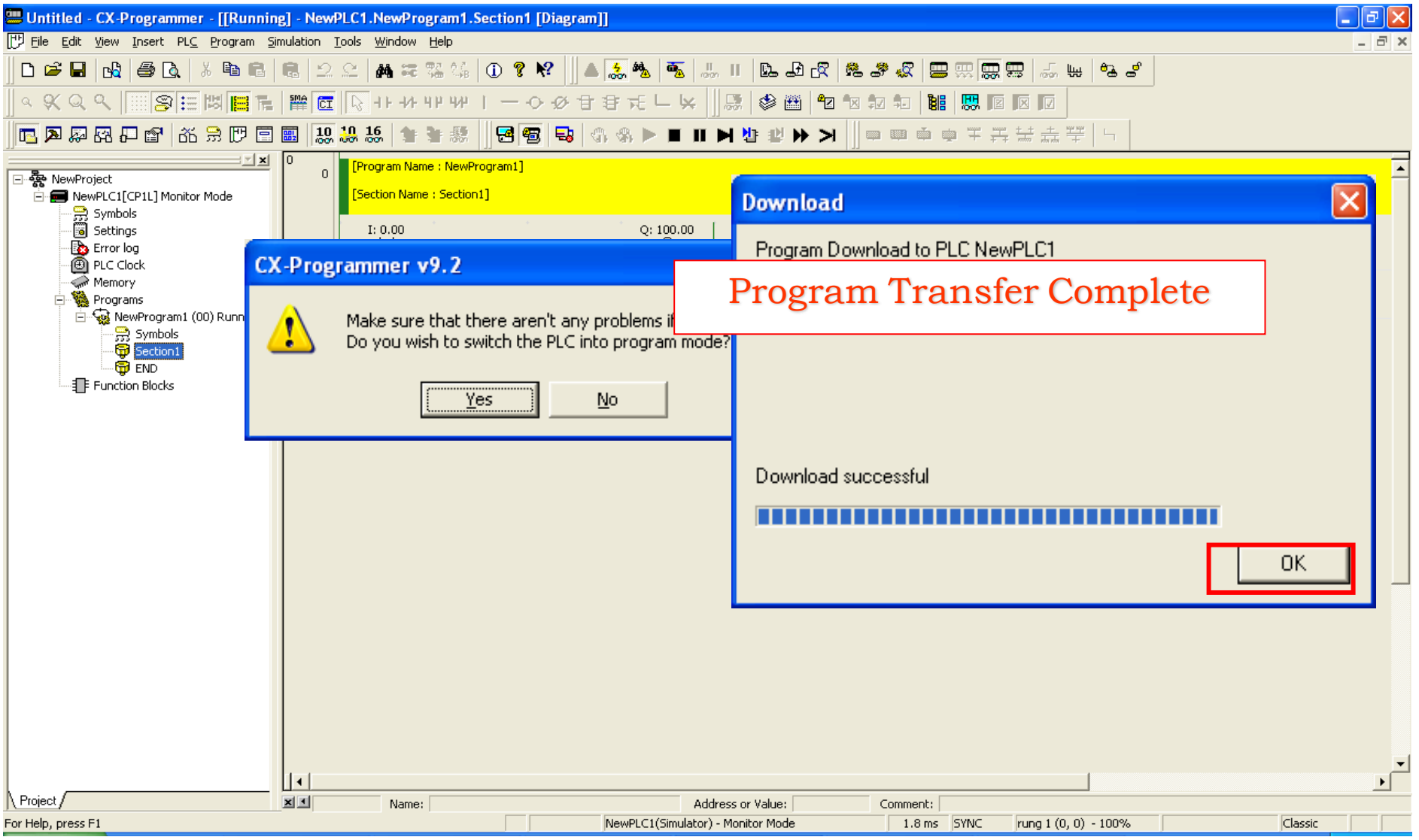
# A simple PLC program completed!

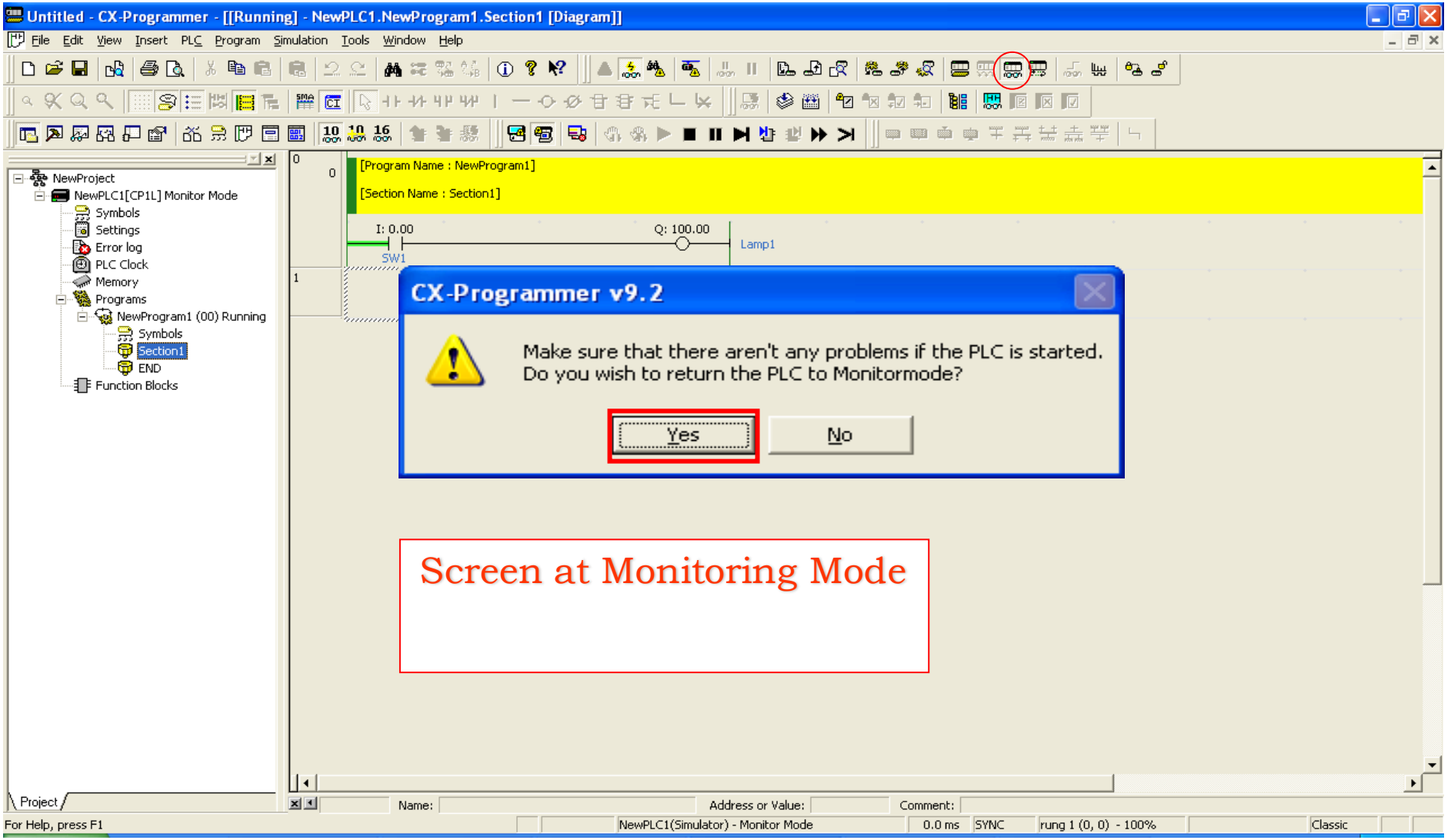

Untitled - CX-Programmer - [NewPLC1.NewProgram1.Section1 [Diagram]]
File Edit View Insert PLC Program Simulation Tools Window Help
NewProject
NewPLC1[CP1L] Offline
Symbols
Settings
Memory
Programs
NewProgram1 (00)
Symbols
Section1
END
Function Blocks
[Program Name : NewProgram1]
[Section Name : Section1]
I: 0.00
SW1
Q: 100.00
Lamp1
Click the Compile Program Icon or Ctrl+F7 to compile your PLC program
Name:
Address or Value:
Comment:
PLC: 'NewPLC1' [PLC Model 'CP1L L']
Compiling...
[PLC/Program Name : NewPLC1/NewProgram1]
[Ladder Section Name : Section1]
[Ladder Section Name : END]
NewProgram1 - 0 errors, 0 warnings.
The programs have been checked with the program check option set to Unit Ver.1.0.
Compile
Find Report
Transfer
For Help, press F1
NewPLC1(Net:0,Node:0) - Offline
rung 0 (0, 1) - 100%
Classic


Untitled - CX-Programmer - [[Running] - NewPLC1.NewProgram1.Section1 [Diagram]]
File Edit View Insert PLC Program Simulation Tools Window Help
NewProject
NewPLC1[CP1L] Monitor Mode
Symbols
Settings
Error log
PLC Clock
Memory
Programs
NewProgram1 (00) Running
Symbols
Section1
END
Function Blocks
Project
[Program Name : NewProgram1]
[Section Name : Section1]
I: 0.00
SW1
Q: 100.00
Lamp1
PLC at On-line
Name:
Address or Value:
Comment:
---------- PLC: 'NewPLC1' (PLC Model 'CP1L L' ) ----------
Compiling...
[PLC/Program Name : NewPLC1/NewProgram1]
[Ladder Section Name : Section1]
[Ladder Section Name : END]
NewPLC1 - 0 errors, 0 warnings.
Compile
Find Report
Transfer
For Help, press F1
NewPLC1(Simulator) - Monitor Mode
0.0 ms
SYNC
rung 1 (0, 0) - 100%
Classic


Untitled - CX-Programmer - [[Running] - NewPLC1.NewProgram1.Section1 [Diagram]]
File Edit View Insert PLC Program Simulation Tools Window Help
Work Online Ctrl+W
Auto Online
Operating Mode
Monitor
Compile All PLC Programs F7
Program Check Options...
Program Assignments
Memory Allocation
Transfer
Partial Transfer
Protection
Clear All Memory Areas
Edit
Change Model
Change Communication Settings
Data Trace...
Time Chart Monitoring...
Force
Set
To PLC... Ctrl+T
From PLC... Ctrl+Shift+T
Compare with PLC...
To File...
From File...
NewProject
NewPLC1[CP1L]
Symbols
Settings
Error log
PLC Clock
Memory
Programs
Function Blocks
: NewProgram1]
: Section1]
Q: 100.00
Lamp1
Transfer Program from Computer to PLC
Project
Name:
Address or Value:
Comment:
Transfers data to the PLC (downloads)
NewPLC1(Simulator) - Monitor Mode
0.0 ms
SYNC
rung 1 (0, 0) - 100%
Classic


Untitled - CX-Programmer - [[Running] - NewPLC1.NewProgram1.Section1 [Diagram]]
File Edit View Insert PLC Program Simulation Tools Window Help
NewProject
NewPLC1[CP1L] Monitor Mode
Symbols
Settings
Error log
PLC Clock
Memory
Programs
NewProgram1 (00) Running
Symbols
Section1
END
Function Blocks
Project
[Program Name : NewProgram1]
[Section Name : Section1]
I: 0.00
SW1
Download Options
PLC: NewPLC1
Include:
Program(s)
Settings
Symbols
Comments
Program index
OK
Cancel
Transfer All
Symbols, Comments, Program index
Transfer To/From:
Memory card
Transfer files of all tasks
Transfer files by the task
Clear program memory
Exclude Port (Serial 1/2) of PLC Settings from the transfer target. (Check when transferring CPU unit serial comms port settings changed by NT Link auto-online or CPU unit parameter edit of CX-Integrator.)
Name:
Address or Value:
Comment:
For Help, press F1
NewPLC1(Simulator) - Monitor Mode
1.8 ms
SYNC
rung 1 (0, 0) - 100%
Classic


Untitled - CX-Programmer - [[Running] - NewPLC1.NewProgram1.Section1 [Diagram]]
File Edit View Insert PLC Program Simulation Tools Window Help
NewProject
NewPLC1[CP1L] Monitor Mode
Symbols
Settings
Error log
PLC Clock
Memory
Programs
NewProgram1 (00) Runn
Symbols
Section1
END
Function Blocks
[Program Name : NewProgram1]
[Section Name : Section1]
I: 0.00
Q: 100.00
CX-Programmer v9.2
Make sure that there aren't any problems i
Do you wish to switch the PLC into program mode?
Yes
No
Download
Program Download to PLC NewPLC1
Program Transfer Complete
Download successful
OK
Project
Name:
Address or Value:
Comment:
For Help, press F1
NewPLC1(Simulator) - Monitor Mode
1.8 ms
SYNC
rung 1 (0, 0) - 100%
Classic


Untitled - CX-Programmer - [[Running] - NewPLC1.NewProgram1.Section1 [Diagram]]
File Edit View Insert PLC Program Simulation Tools Window Help
NewProject
NewPLC1[CP1L] Monitor Mode
Symbols
Settings
Error log
PLC Clock
Memory
Programs
NewProgram1 (00) Running
Symbols
Section1
END
Function Blocks
[Program Name : NewProgram1]
[Section Name : Section1]
I: 0.00
SW1
Q: 100.00
Lamp1
CX-Programmer v9.2
Make sure that there aren't any problems if the PLC is started.
Do you wish to return the PLC to Monitormode?
Yes
No
Screen at Monitoring Mode
Project
Name:
Address or Value:
Comment:
For Help, press F1
NewPLC1(Simulator) - Monitor Mode
0.0 ms
SYNC
rung 1 (0, 0) - 100%
Classic

# AND / AND NOT

- When 2 or more conditions lie in series on the same instruction line, the first one correspond to a LD or LD NOT instruction, and the rest of the conditions, to AND or AND NOT instructions.
- AND - taking the logical AND of the execution condition
- AND NOT - taking the logical AND between its execution condition and the inverse of its operand bit.

# OR / OR NOT

- When 2 or more conditions lie on separate instruction line running in parallel and joining together, the first one correspond to a LD or LD NOT instruction, and the rest of the conditions correspond to OR or OR NOT instructions.
- OR - taking the logical OR of the execution condition.
- OR NOT - taking the logical OR between its execution condition and the inverse of its operand bit.

# AND LD & OR LD Instructions

## AND LD

- The AND LD instruction logically AND the execution conditions produced by two logic blocks.

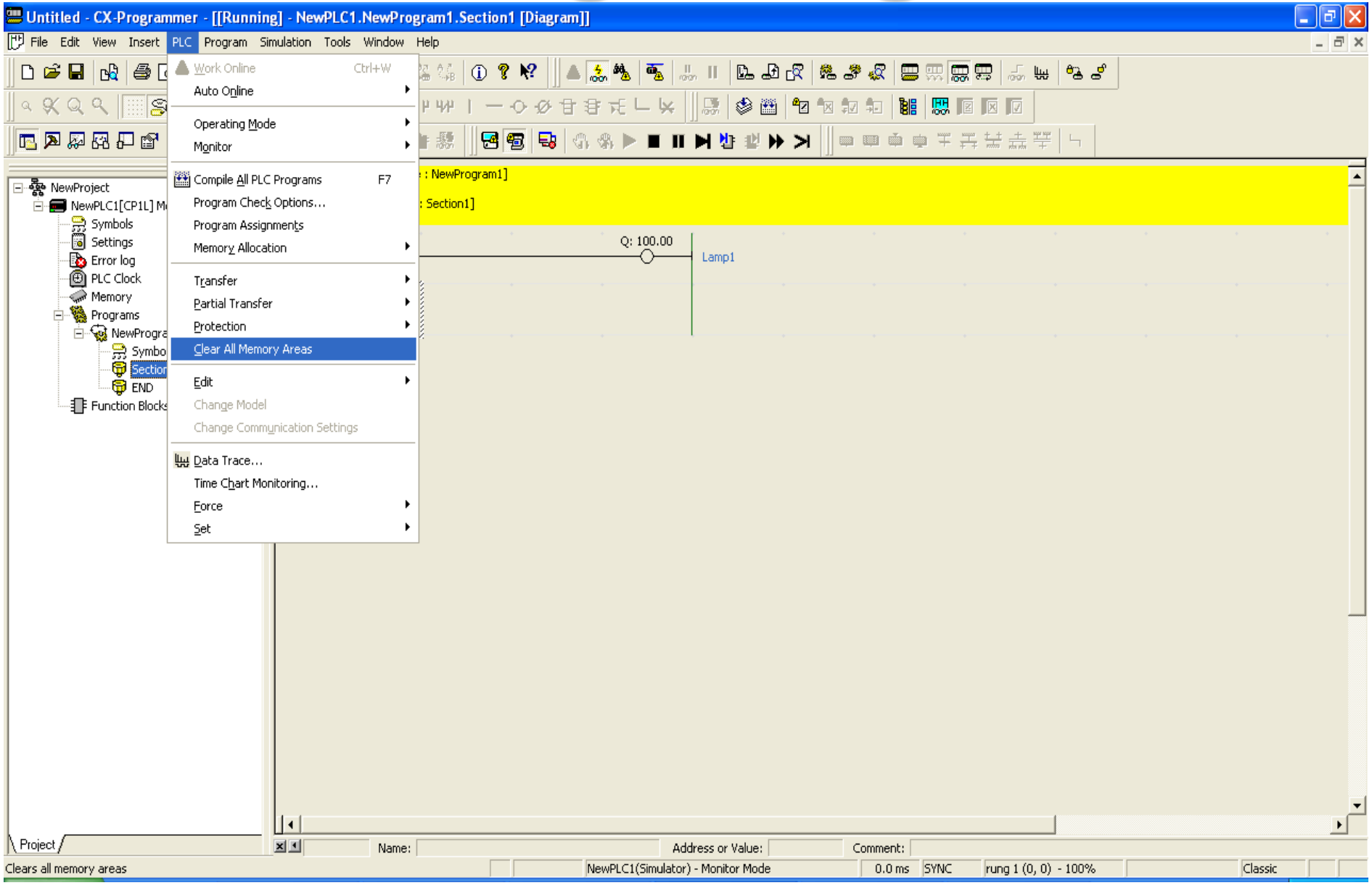
## OR LD

- The OR LD instruction logically OR the execution conditions produced by two logic blocks.

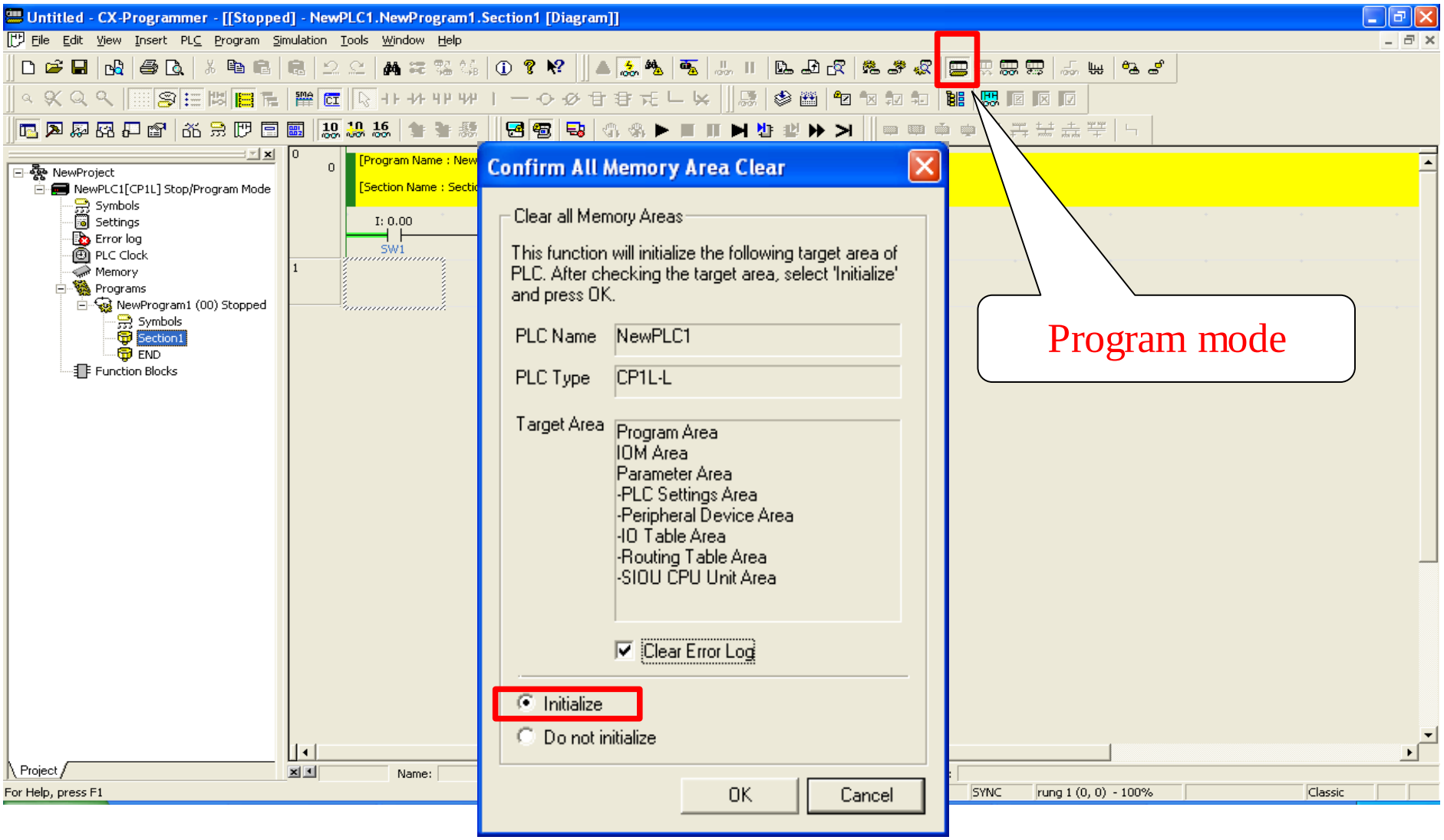
# Memory Area Functions

| Data | | Function |
|---|---|---|
| IR | Internal Relay Area | Bits are allocated to external I/O terminals. Some can be used free within program |
| IO | I/O Relay Area | |
| WR | Work Relay Area | Not allocated to external I/O terminal. Freely use within program |
| SR | Special Relay Area | Serve for specific function such as flags & control bits |
| TR | Temporary Relay Area | Contain bits that record the ON/OFF status of program branches |
| AR | Auxiliary Relay Area | Serve for specific functions such as flags & control bits |
| LR | Link Relay Area | Used for data links (Serial PLC Link) with another PLC |
| | PLC Link Area | |
| TC | Timer/Counter Area | Same numbers are used for timers & counters in CPM1A) |
| DM | Data Memory Area | Word values are retained when power Off or operating mode switching |
| HR | Holding Relay Area | Bit status are retained when power Off or operating mode switching |

# Clearing All Program



# Clearing All Program



# Various Relays

# I/O Relay (IR)

## I/O Relay

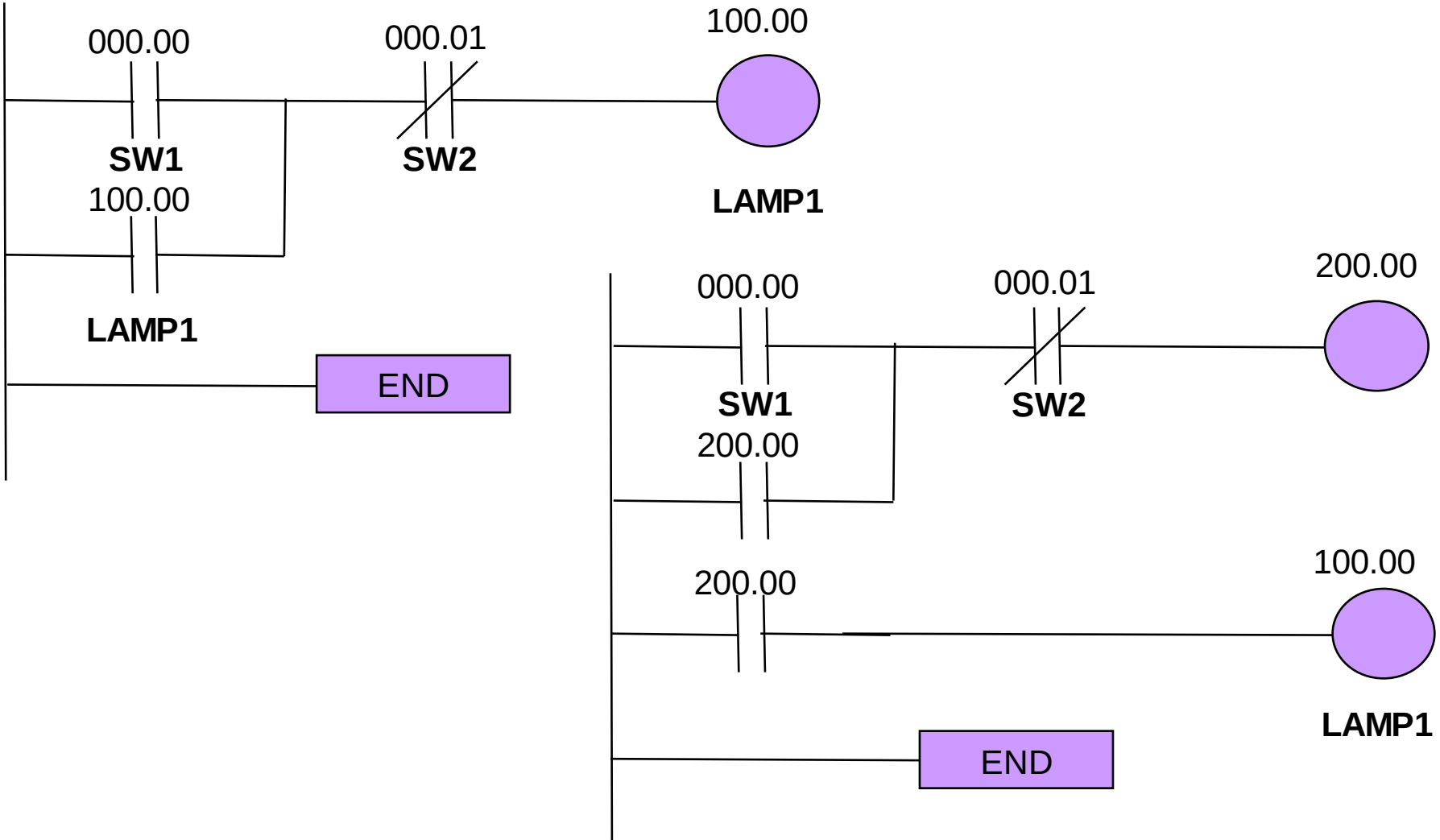
- Corresponds to external input terminals and output terminals
- e.g.: 000.00 to 000.15 (Input relay)
- e.g.: 100.00 to 100.15 (Output relay)

## Internal Relay

- Can be used freely within the program, they cannot be used for I/O exchange with external I/O terminals.

# IR Application Example

## Self-Holding / Latching Circuit



# Condition & Clock Pulse Flags

- These flags include the Arithmetic Flags, such as the Error Flag

  and Equals Flag, which indicate the results of instruction

  execution.

- The Condition Flags are specified with symbols, such as P_CY

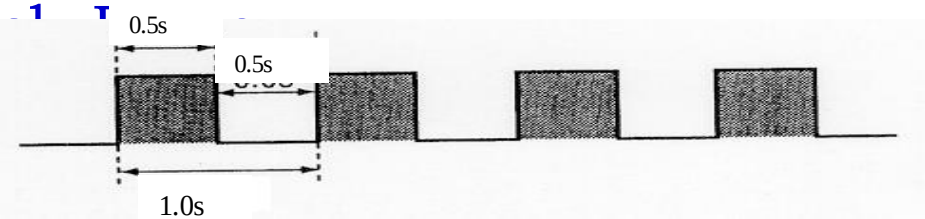
  and P_ER, rather than addresses.

- e.g.: P_On – Always On Flag

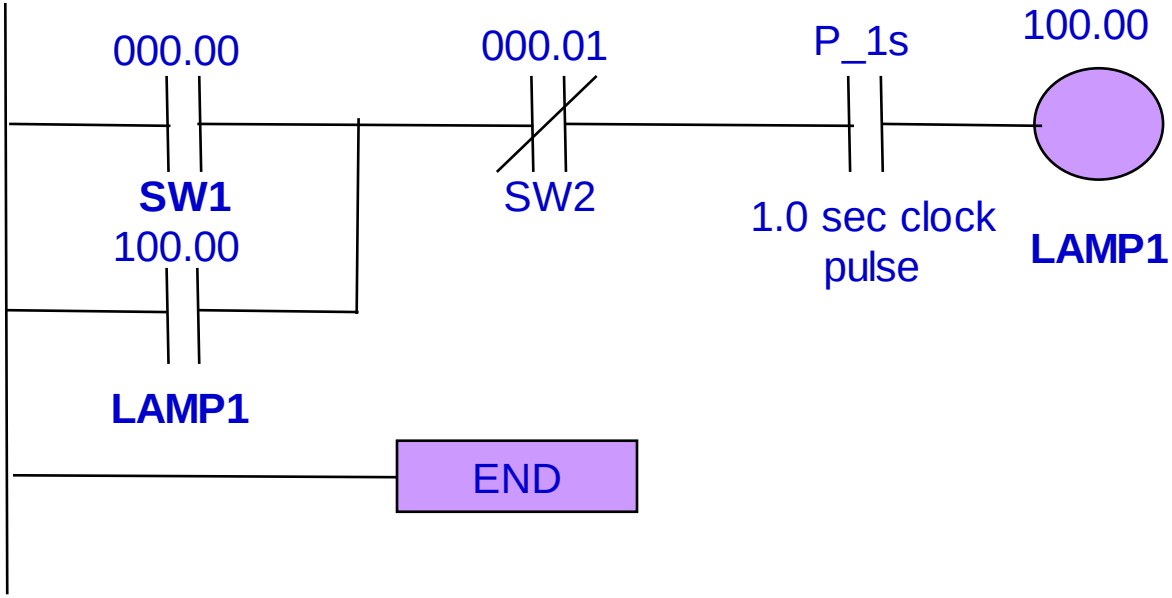
  P_Off – Always Off Flag

  P_1s – 1 second Clo

**P_1s (1s clock pulse)**



# Special Flags Application Example

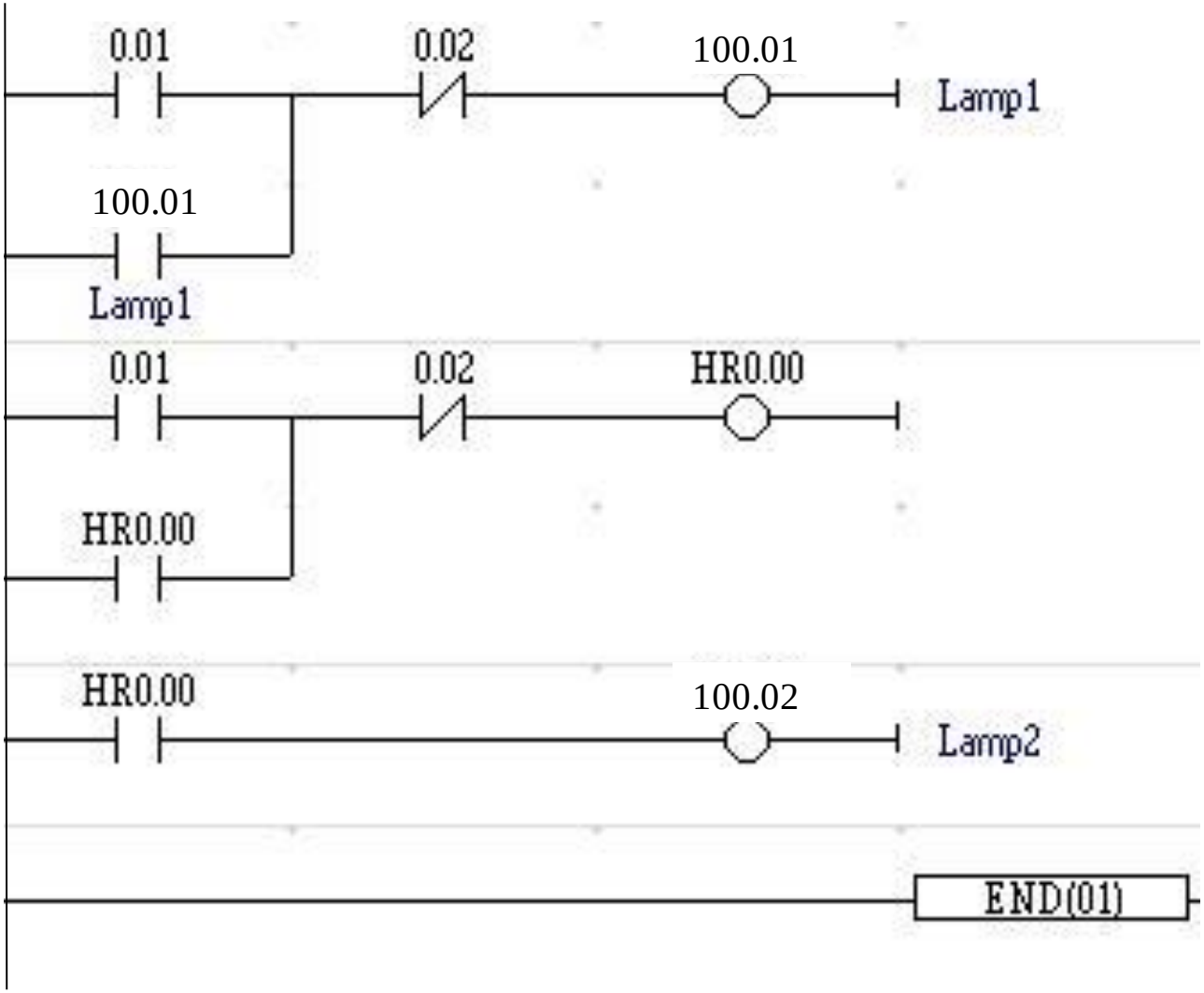



*** Lamp 1 will starts flashing/blinking after SW1 was being trigger.**

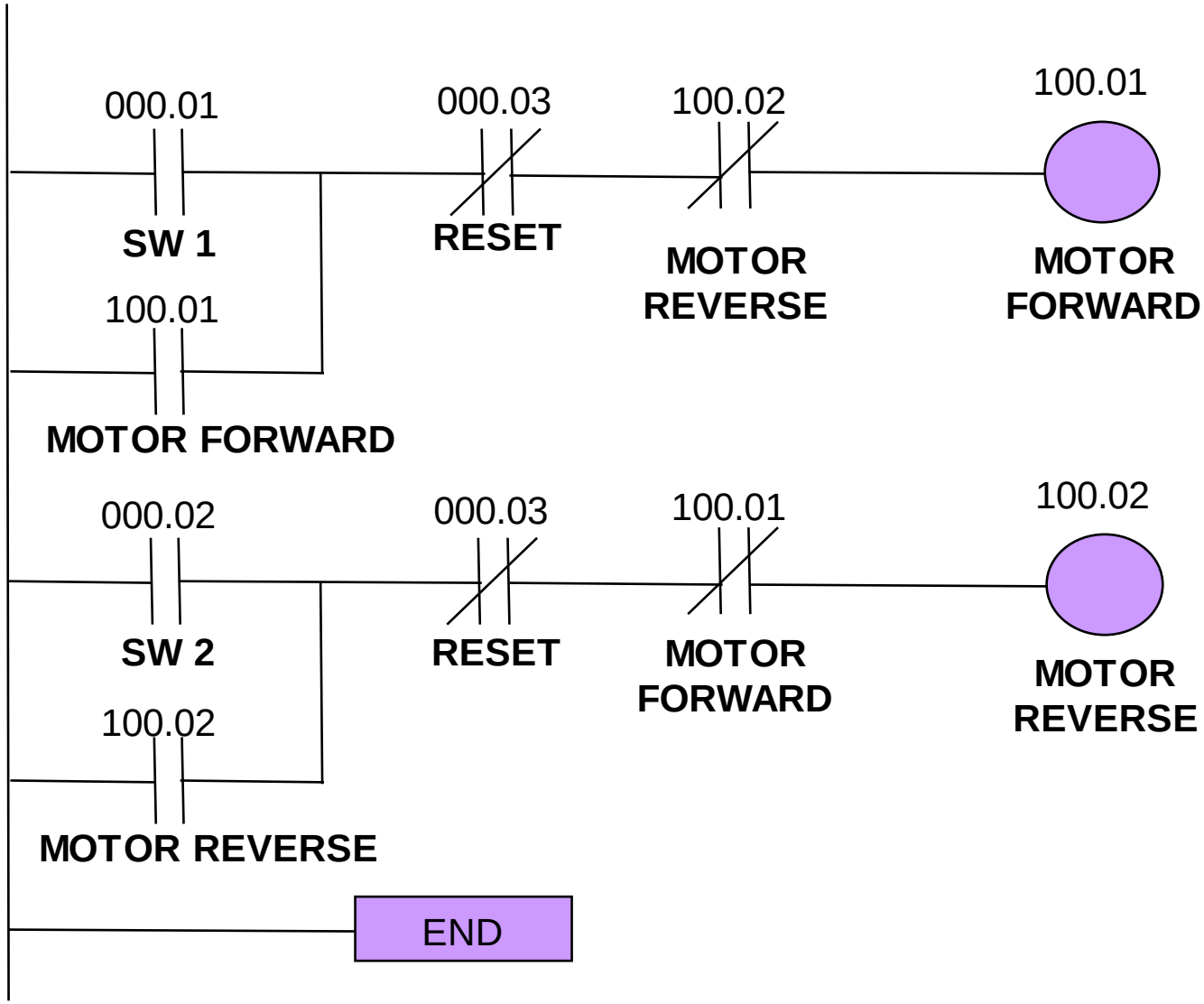
# Holding Relay (HR)

- Maintains the ON/OFF state even when the power is cut off.

# HR Application Example

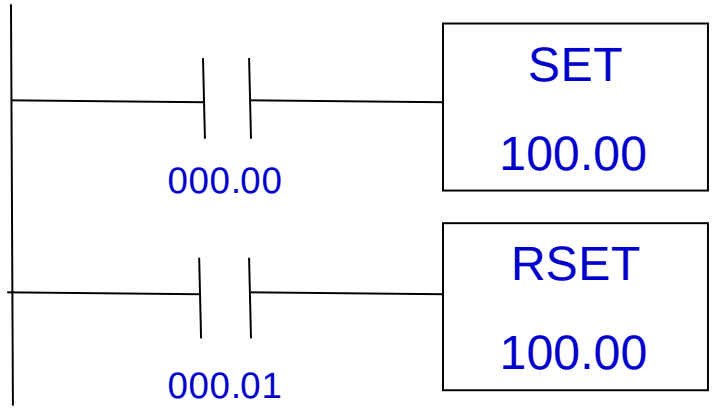


# INTERLOCK CIRCUIT



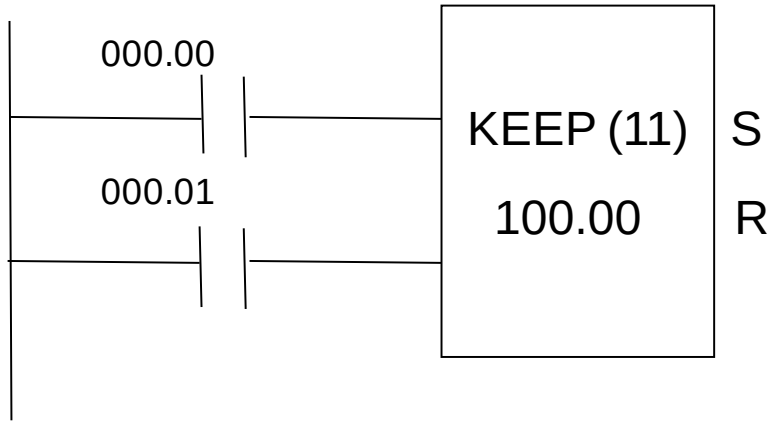
# Latch Instruction – SET, RSET

- Used momentary switches to latch and reset output.



# Latch Instruction – KEEP (11)

- Used momentary switches to latch and reset output.
- Operates like a latching relay that is set by S and reset by R.
- Remains ON or OFF state of a bit until one of its two inputs sets or reset.

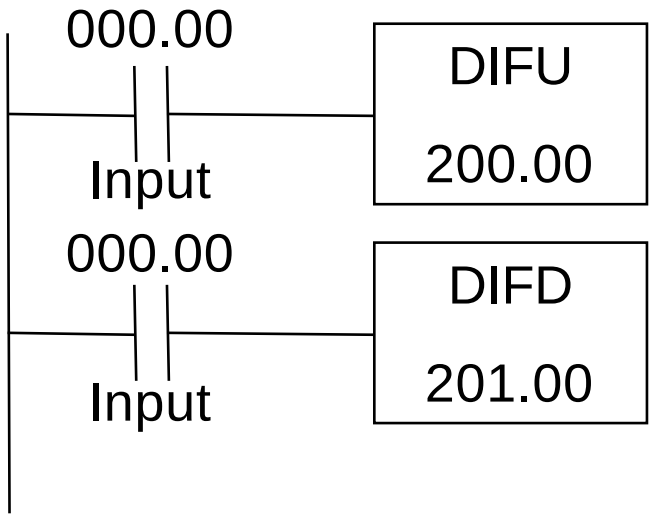


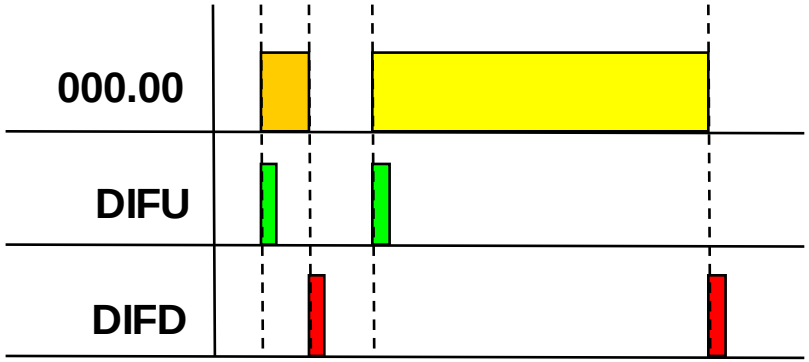

```
00  LD          000.00
01  LD          000.01
02  KEEP (11)   100.00
03  END (01)
```

# Differentiation Up & Down DIFU (13) and DIFD (14)

- DIFU turns its output ON when it detects an OFF > ON transition in its input signal.
- DIFD turns its output ON when it detects an ON > OFF transition in its input signal.

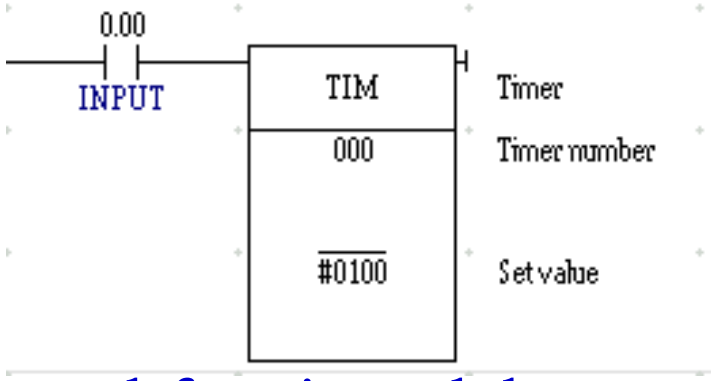
# One Shot-DIFU (13) and DIFD (14)




DIFU and DIFD only turns an output ON for only one scan/pulse.

# Timer (TIM)



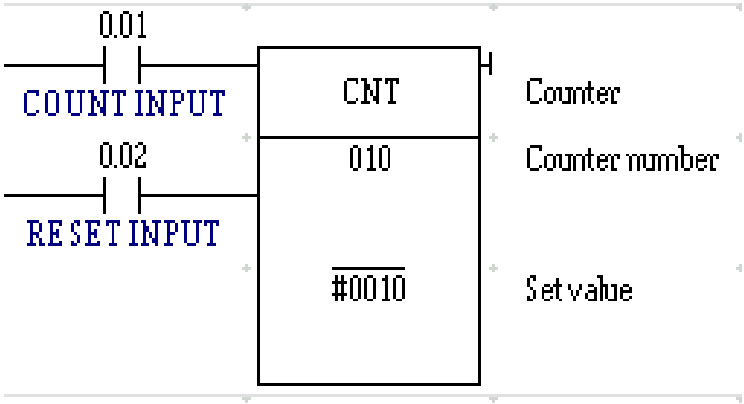

- Timer is an instruction normally used for time delay. It can be ON delay or OFF delay.
- The timer is based on a decrement method.
- On-Delay-Timer after our sensor (input) turns on we wait x-seconds before activating a solenoid valve (output).

** Timer number for CPM1A, T000 to T127

…Must not coincide with counter number

- Set value range from #0000 ~ #9999 (0 ~ 999.9 sec)
- If the start input becomes OFF or the power is turned OFF during operations before time up (current value #0000), the current value will be reset to the value set.

# Counter (CNT)



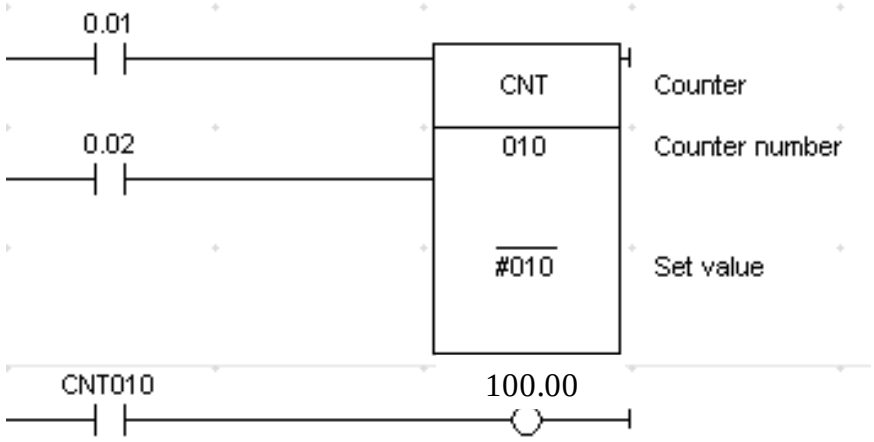

- CNT is a preset decrement counter.
- It decrements one count every time an input signal goes from OFF to ON.
- The counter must be programmed with a count input, a reset input, a counter number and set value.
- Set value range from #0000 ~ #9999

** Counter number for CPM1A, C000 to C127

…Must not coincide with timer number

- Its current value will not be reset even if the PC power supply is turned OFF. It will preserve the count until then.

# COUNTER




| | | |
|---|---|---|
| LD | | 000.01 |
| LD | | 000.02 |
| CNT | | 010 |
| | #010 | |
| LD | | CNT01 |
| OUT | | 100.00 |
| END | | (01) |