U90 Ladder Software Manual

Version 3.50, 6/03

Table Of Contents

Welcome to U90 Ladder

U90 Ladder is the software tool used to create applications for the M90 PLC. After you plan the control task, use U90 Ladder to write, debug, and download the PLC control and HMI applications into the M90.

Program Editors

The program editors are where you create and edit both the PLC and HMI aspects of your project application.

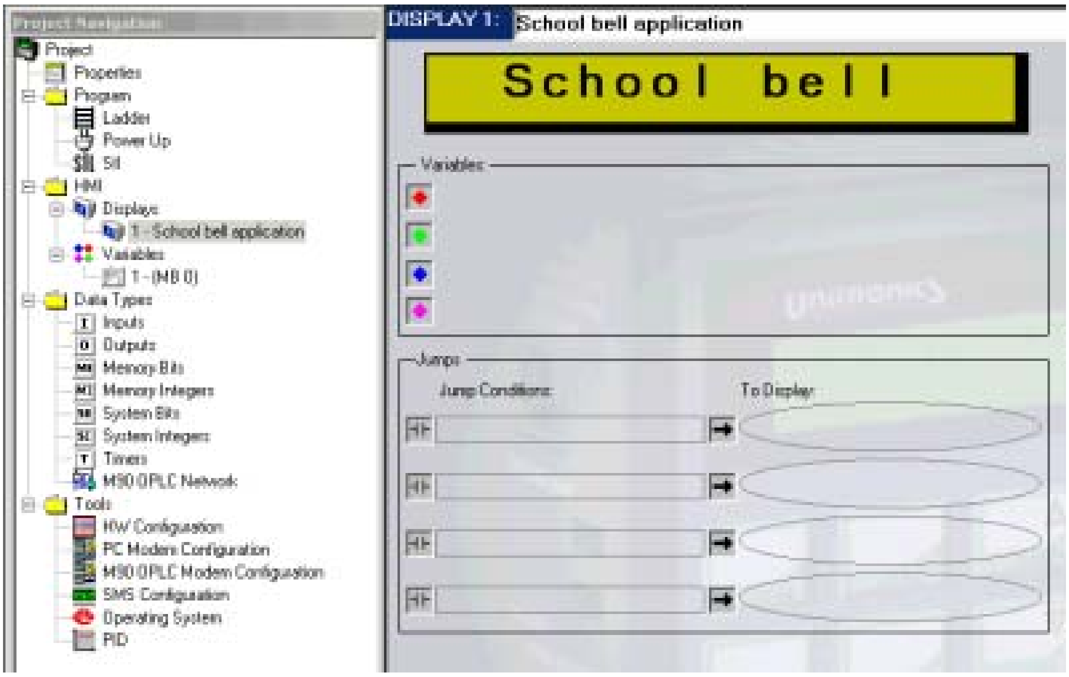
There are 3 editors:

1. Ladder
2. Display
3. Variable

The Ladder Editor is a program editor. The Display and Variable Editors are HMI editors. Each editor is operated through a different window. You switch between editors via the Standard Toolbar buttons or in the Project Navigator.

Project Navigation Tree

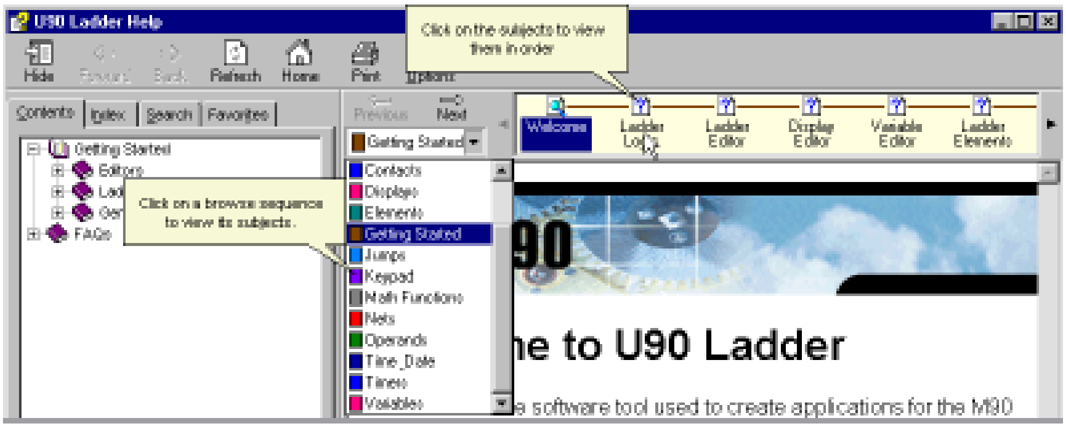
The Project Navigation Tree allows you to navigate easily between components of your program, data types and tools of the U90 Ladder.



Browse Sequences

This U90 Help version contains selected browse sequences. These sequences are arranged by topic to help you make the most out of U90's on-line help. The subject of each sequence appears in the Browse pull-down menu as shown below. The subjects in a browse sequence are arranged from general to most specific.

Below, the browser sequence shown is Getting Started.



Click on the subjects to view them in order

Click on a browse sequence to view its subjects.

e software tool used to create applications for the M90

Printing Documentation

All of the topics in this help file are contained in the U90 Ladder Software Manual. This manual can be viewed or printed from the software CD.

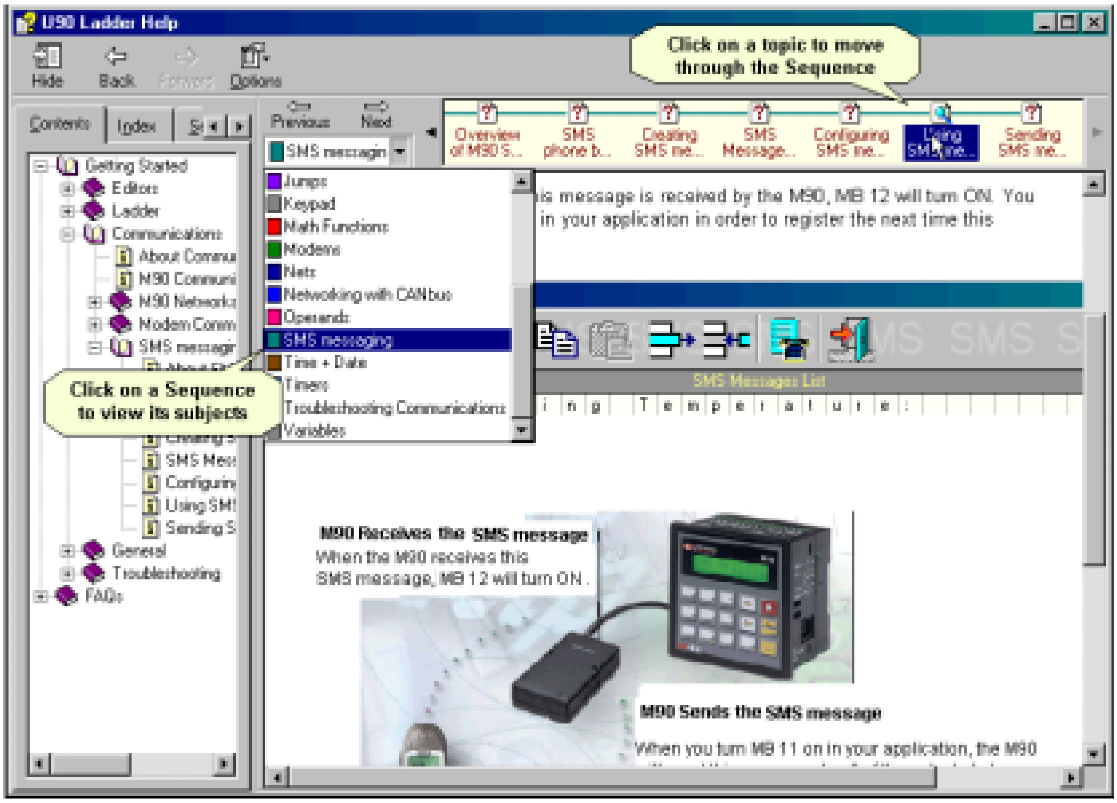
Interface Language

U90 Ladder can be run in a variety of languages, by selecting Language from the View menu, and then clicking on the desired language.

Help

Use the Help browser to learn how to use U90 Ladder effectively. Topics in a browse sequence are classed according to subject. This enables you to see related topics without conducting a search.

The browse sequence shown below is **SMS messaging.**



Click on a topic to move through the Sequence

Click on a Sequence to view its subjects

is message is received by the M90, MB 12 will turn ON. You in your application in order to register the next time this

M90 Receives the SMS message

When the M90 receives this SMS message, MB 12 will turn ON.

M90 Sends the SMS message

When you turn MB 11 on in your application, the M90

Getting Started

Opening a new project

To open a new project:

- On the Project menu , click **New**.
- On the Standard toolbar, select **New** .
- On the keyboard, press **Ctrl** + **N**

Opening a project

To open an existing project

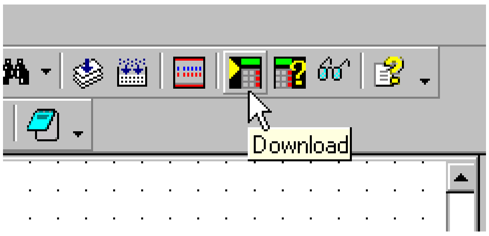
- On the Project menu, click **Open**. The Open dialog box appears. Select the file you want to open.
- On the Standard toolbar, select **Open** . The Open dialog box appears. Select the file you want to open.
- Ctrl + O. The Open dialog box appears. Select the file you want to open.

Downloading a Project

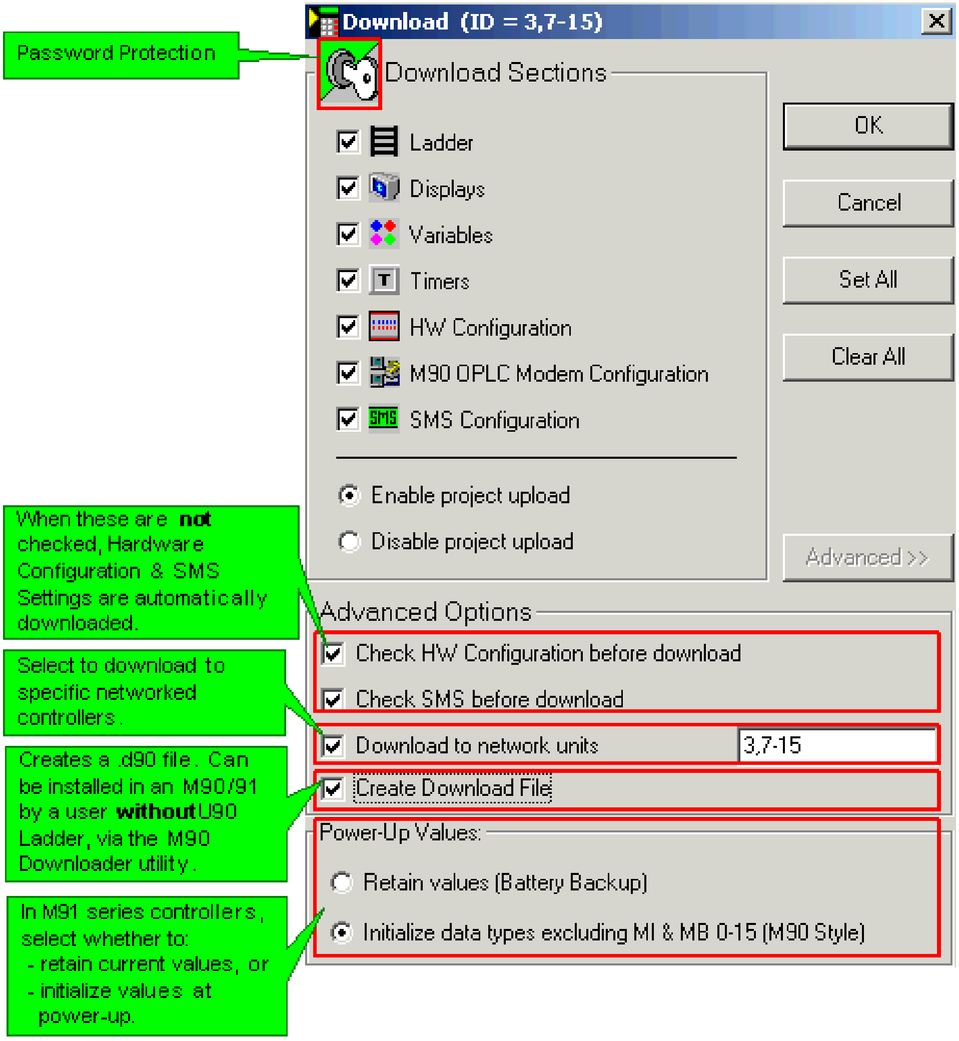
The Download process transfers your project from the PC to the controller.

To download a project to a controller:

1. Click the **Download** icon on the Standard toolbar.



2. The Download Window opens with Download Sections. Those sections which have yet to be downloaded to a controller will be selected. If you have made no changes in the project since the last download, you have to select the Download Sections manually. Click OK.



Download (ID = 3,7-15)

Password Protection

Download Sections

- ☑ Ladder
- ☑ Displays
- ☑ Variables
- ☑ Timers
- ☑ HW Configuration
- ☑ M90 OPLC Modem Configuration
- ☑ SMS Configuration

● Enable project upload
○ Disable project upload

Advanced Options

- ☑ Check HW Configuration before download
- ☑ Check SMS before download
- ☑ Download to network units 3,7-15
- ☑ Create Download File

Power-Up Values:

○ Retain values (Battery Backup)
● Initialize data types excluding MI & MB 0-15 (M90 Style)

When these are **not** checked, Hardware Configuration & SMS Settings are automatically downloaded.

Select to download to specific networked controllers.

Creates a .d90 file. Can be installed in an M90/91 by a user **without** U90 Ladder, via the M90 Downloader utility.

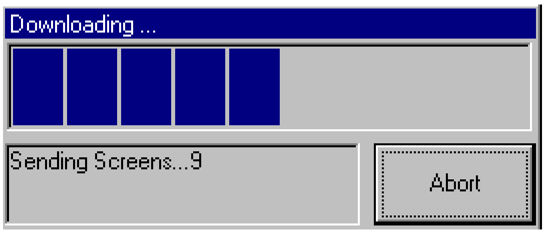
In M91 series controllers, select whether to:
- retain current values, or
- initialize values at power-up.

The key at the top tells you if the project is password protected. If so, the password will have to be supplied at upload.

Note Ladder Image and Project Symbols option. If you do not select this option, the Ladder program cannot be uploaded to a PC for editing. You only be able to view the uploaded program in STL. To enable the Ladder program to upload from the M90 into a PC, select this option.

Note the different Power-up value (Battery Backup) options.

3. The Downloading Progress window opens. This window closes when download is complete.



Downloading ...

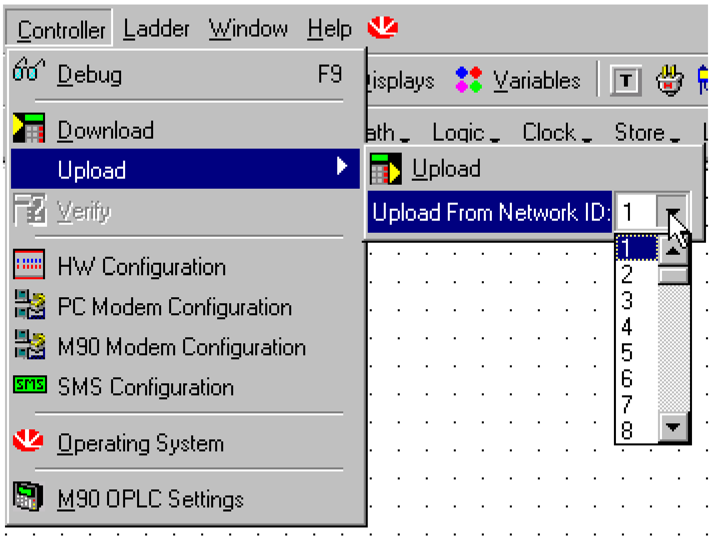
Sending Screens...9

Uploading a Project

1. Select Upload from the Controller menu.

2. Two new options are displayed: Upload, and Upload from Network ID.
3. Upload from:

   - a stand-alone M90 by clicking on the Upload button
   - from a specific M90 on a network by selecting the M90's ID number as shown below.



4. All sections of the project in the M90 will upload.

Note that if the program is protected by a password, you must supply this password in order to upload.

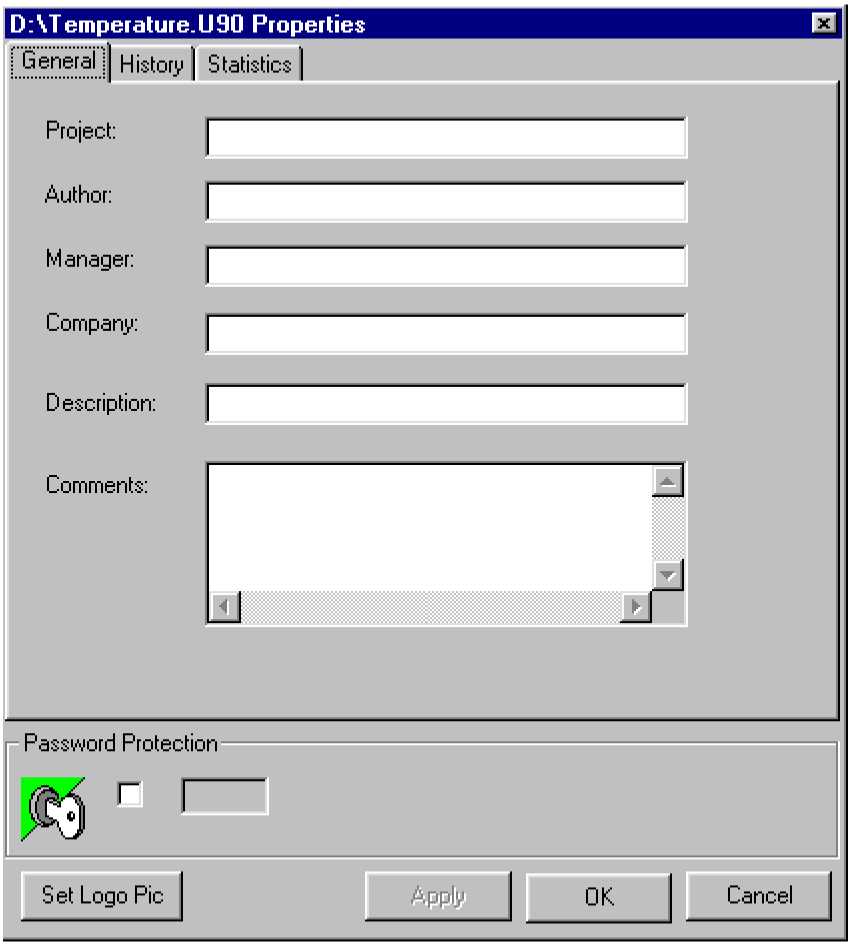
Project Properties

Project properties include:

- General information, including password protection
- History
- Statistics

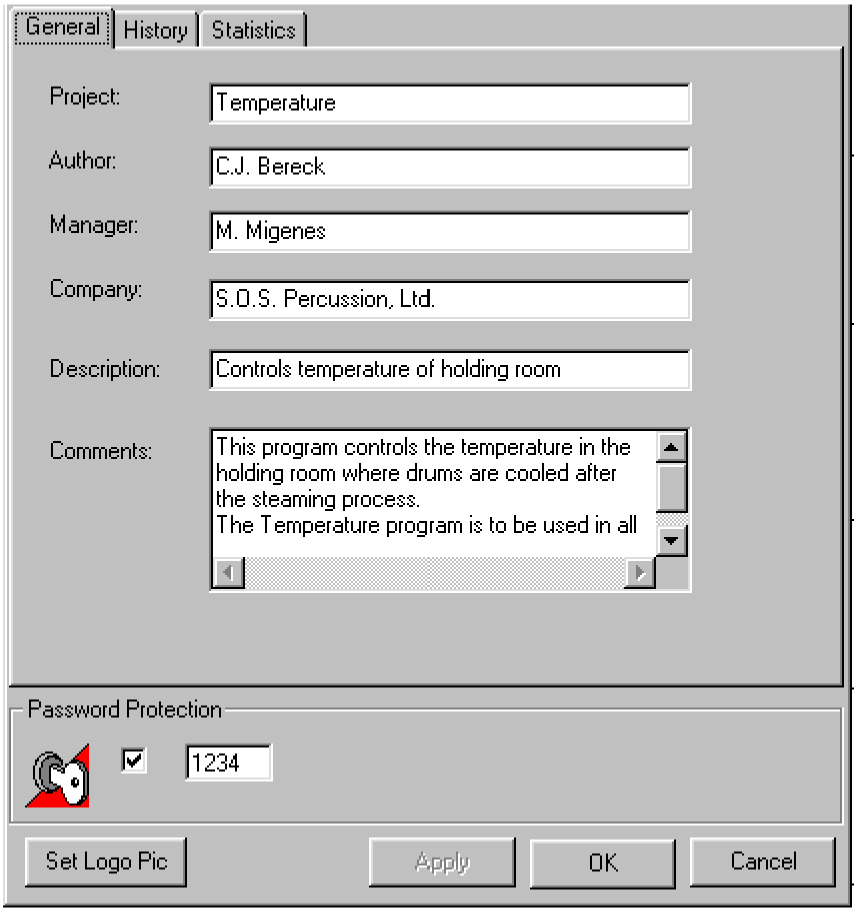
To access program properties

1. Display project properties by selecting Properties from the Project menu. The project Properties box opens.
2. Select property topics by clicking the tabs.



General

When you select General, the fields are blank. You enter all of the project information manually. An example is shown below.



Project: Temperature

Author: C.J. Bereck

Manager: M. Migenes

Company: S.O.S. Percussion, Ltd.

Description: Controls temperature of holding room

Comments: This program controls the temperature in the holding room where drums are cooled after the steaming process.
The Temperature program is to be used in all

Password Protection

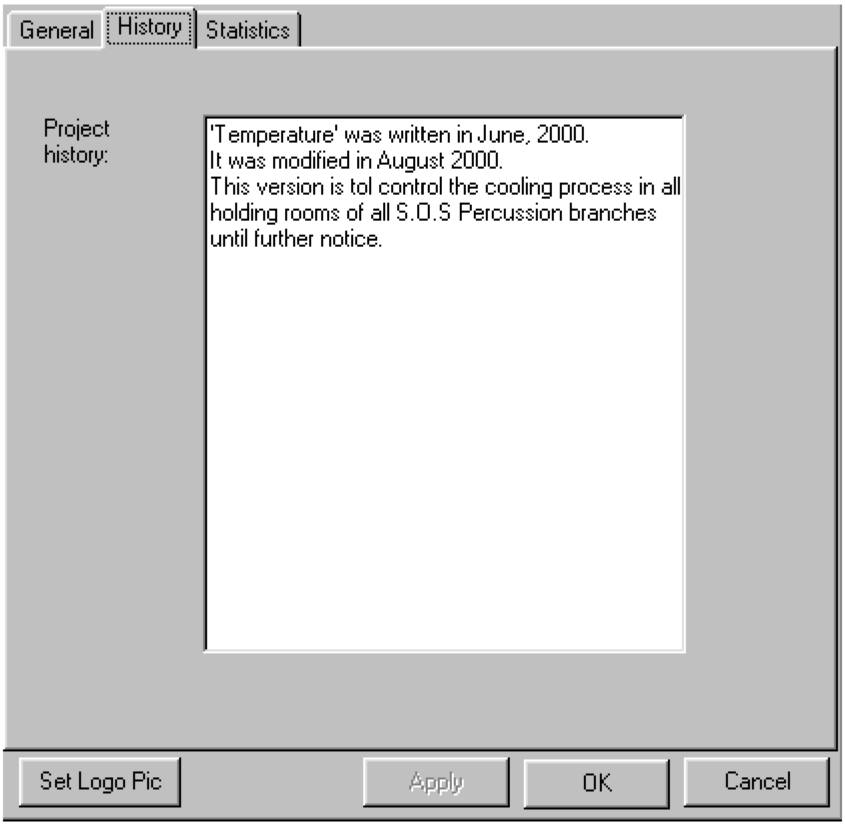
☑ 1234

Password

You can apply password protection to your program. This will prevent anyone who does not have the password from uploading the program from the M90.

History

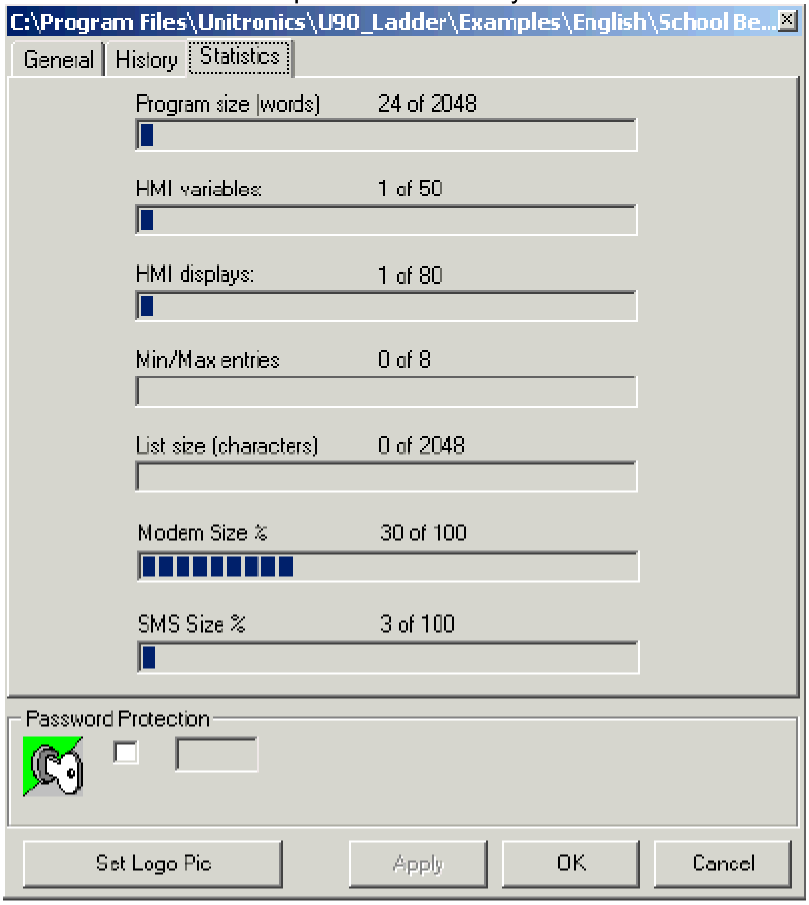
When you first open History, the field is blank. Enter the desired text as shown below.



Project history:

'Temperature' was written in June, 2000.
It was modified in August 2000.
This version is tol control the cooling process in all holding rooms of all S.O.S Percussion branches until further notice.

Statistics
When you open Statistics, the progress bars show how much of the project's available space is in use. The statistics update automatically.


C:\Program Files\Unitronics\U90_Ladder\Examples\English\School Be...
General
History
Statistics

Program size (words) 24 of 2048

HMI variables 1 of 50

HMI displays: 1 of 80

Min/Max entries 0 of 8

List size (characters) 0 of 2048

Modem Size % 30 of 100

SMS Size % 3 of 100

Password Protection

☐

Set Logo Pic
Apply
OK
Cancel

Set Logo Pic

You can also import your company's logo into your project. Then, when you print sections of your project, the logo will be printed at the top of each page.

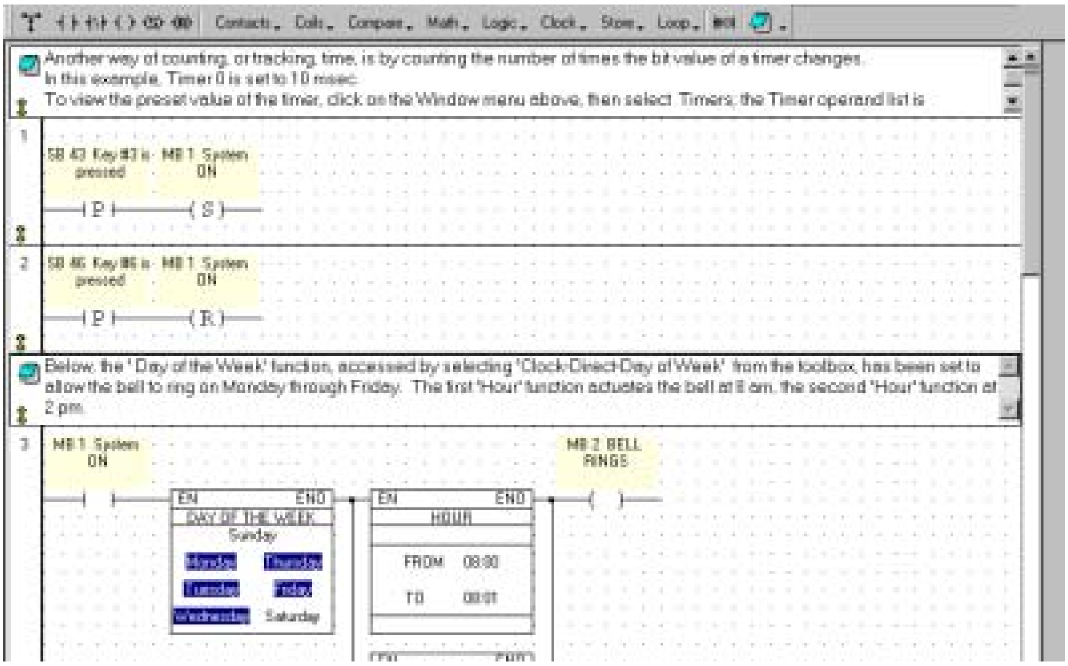
Ladder Editor

Use the Ladder Editor to create the Ladder diagram that will form the backbone of your project application. Ladder diagrams are composed of contacts, coils and function block elements. Power flows from left to right in a Ladder diagram.

Use the Ladder Editor to:

- Place and connect Ladder program elements.
- Apply Compare, Math, Logic, Clock and Loop functions.
- Place Comments on Ladder nets.

Ladder Editor view:



Another way of counting, or tracking time, is by counting the number of times the bit value of a timer changes.
In this example, Timer 0 is set to 10 msec.
To view the preset value of the timer, click on the Window menu above, then select Timers; the Timer operand list is

Below, the 'Day of the Week' function, accessed by selecting 'Clock-Direct-Day of Week' from the toolbox, has been set to allow the bell to ring on Monday through Friday. The first 'Hour' function actuates the bell at 8 am, the second 'Hour' function at 2 pm.

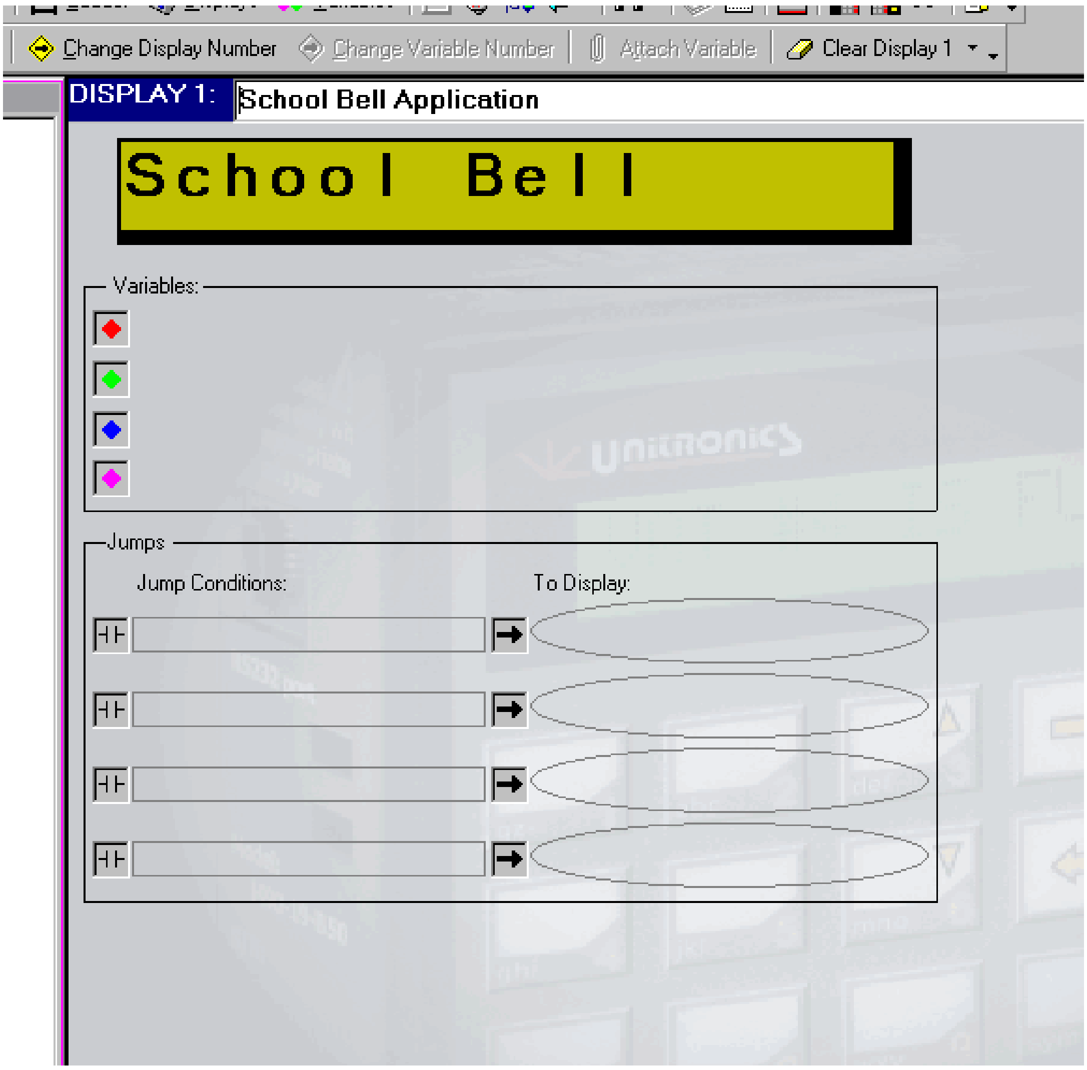
Using the HMI Display Editor

Use this editor to create your HMI application for customizing the M90 operating panel functions to the control task.

Use the Display Editor to:

1. Create text displays that will appear on the M90 LCD. You can create up to 80 displays.
2. Link display text to a variable. You can define up to 50 variables.
3. Configure links, or up to four jumps, to a display.
4. Format the M90 LCD variable display.

Display Editor view:



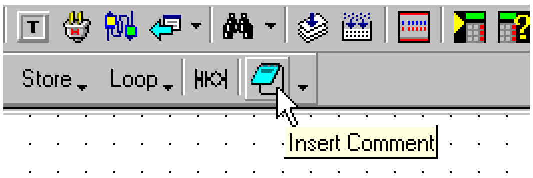
Comments Tool

You can insert comments into the Ladder Editor to label different parts of your program. Comments can be written in Notepad and added later to the project using **Cut** and **Paste** functions.

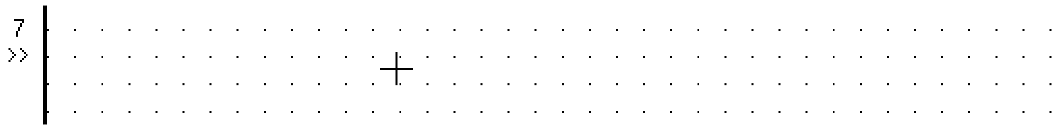
These Comments are 'internal' comments for the programmer(s). The Comments are not downloaded to or displayed on the controller.

To insert comments:

1. On the Ladder toolbar, click **Insert Comment** icon .

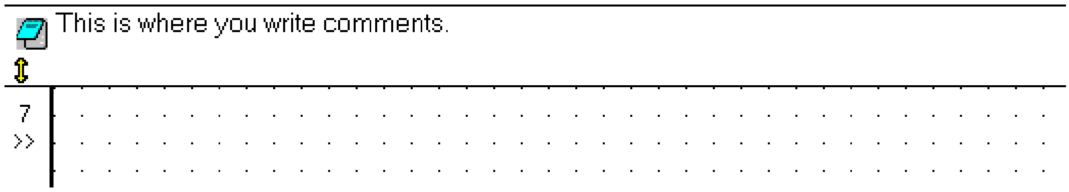


2. Move your cursor to the net in which you wish to insert a comment and click.



3. The Comment will appear above the net.

4. Type in your comments.

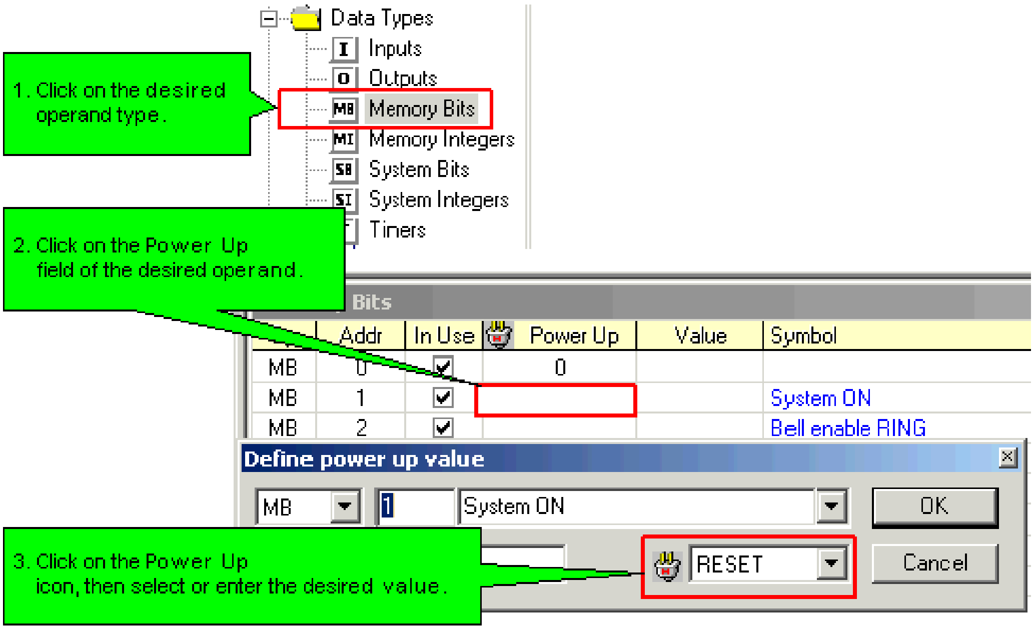

This is where you write comments.

The length and content of your comments will have no effect on your project. They are not downloaded to the controller and do not affect the memory or word size of a project.

Power-up

You can assign Power Up values to most Data Types. These values are written into the operand by the program when the controller is turned on. Outputs, MBs, SBs can be set or reset; integer values can be written into MIs and SIs.

You can assign Power Up values when you place an element into a net, or by opening a Data Type list as shown below.



	Addr	In Use	Power Up	Value	Symbol
MB	0	☑	0		
MB	1	☑			System ON
MB	2	☑			Bell enable RING

1. Click on the desired operand type.

2. Click on the Power Up field of the desired operand.

3. Click on the Power Up icon, then select or enter the desired value.

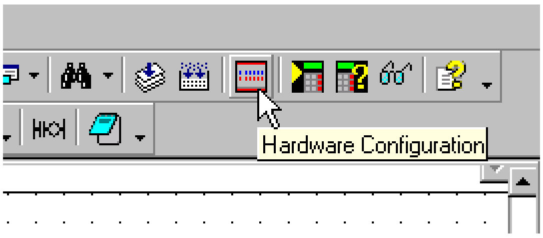
Hardware Configuration

Hardware Configuration

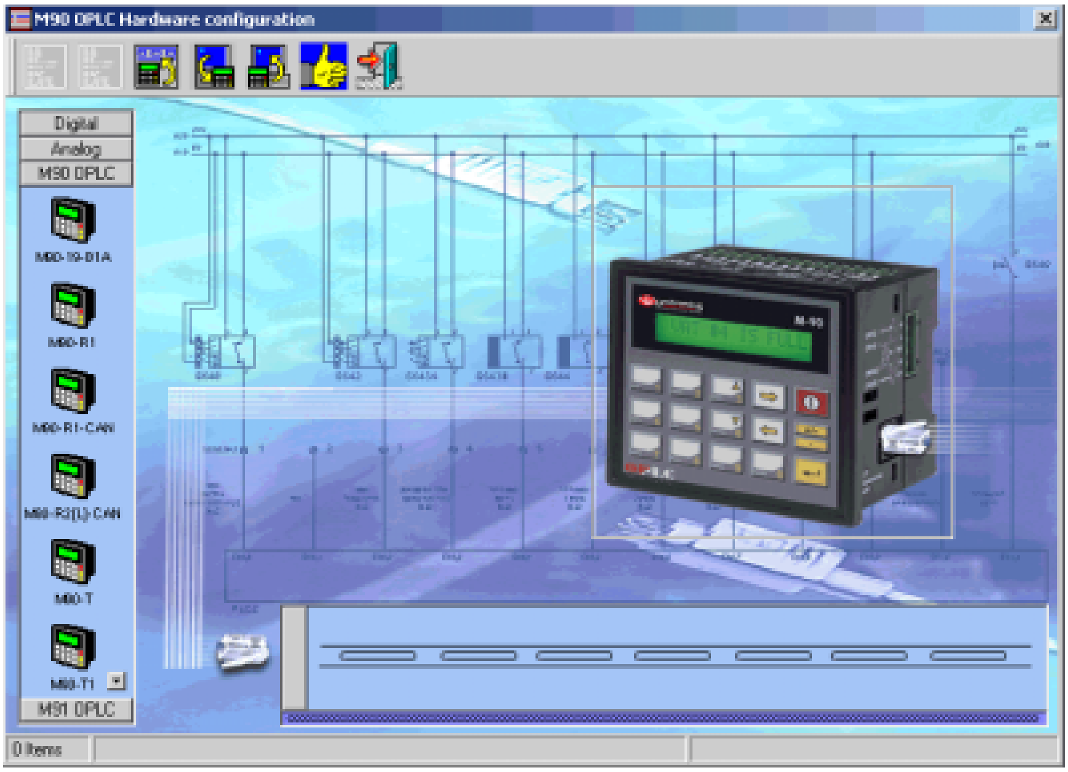
Hardware configuration enables you to select controllers from both the M90 and M91 series.

Selecting the Controller Model

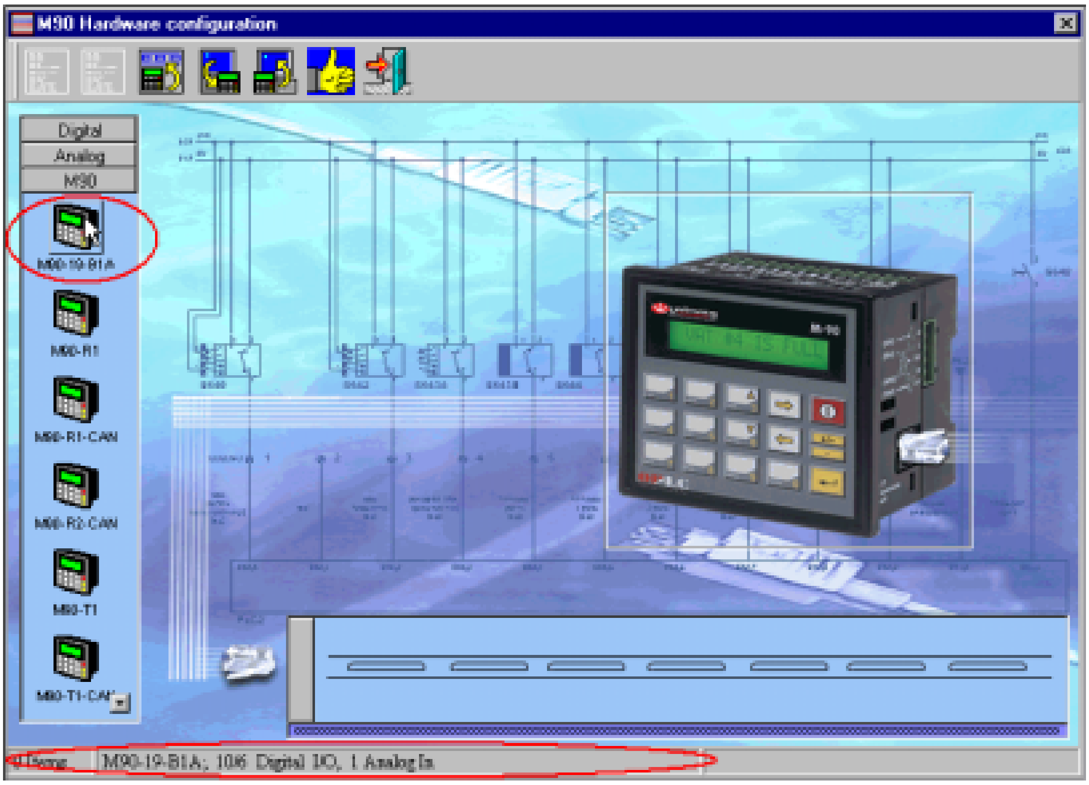
1. Click **Hardware Configuration** on the Standard toolbar.



2. The **M90 Hardware Configuration** window opens.



3. Select the appropriate M90 model.



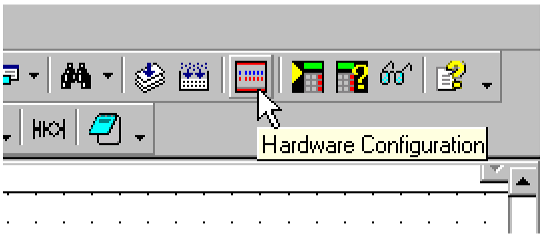
M90-19-B1A; 10/6 Digital I/O, 1 Analog In

Configuring an Analog Input

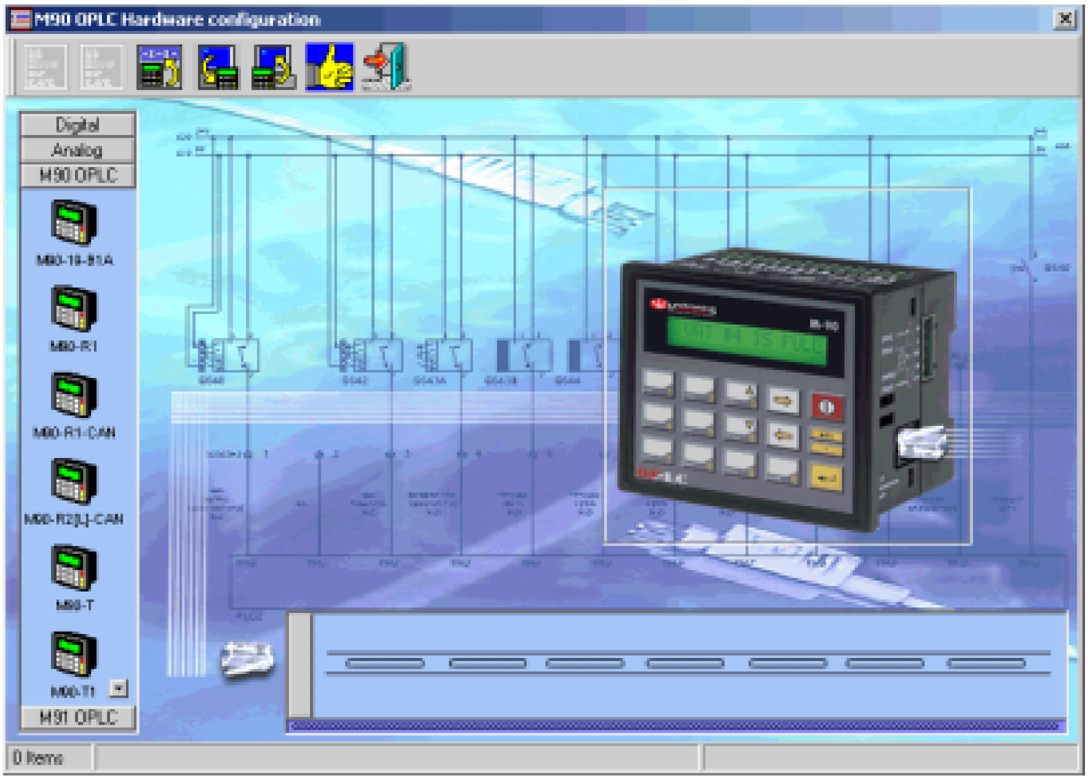
M90

To attach an Analog Input to an MI:

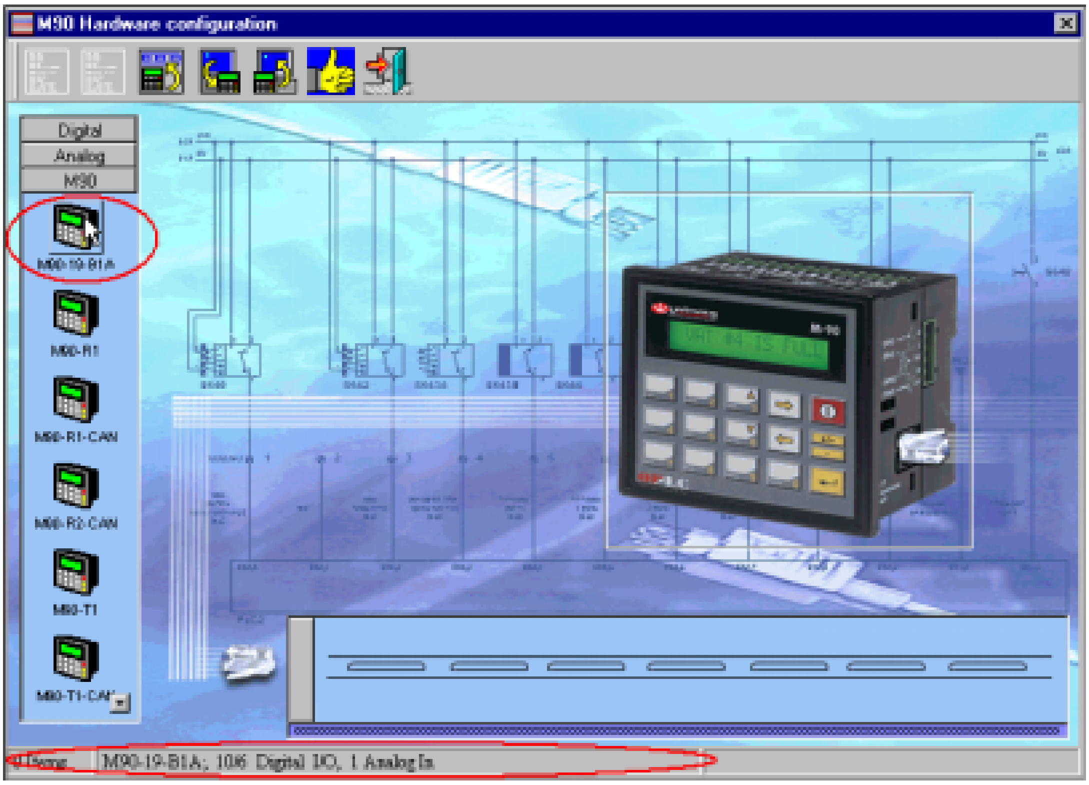
1. Click **Hardware Configuration** on the Standard toolbar.



2. The **M90 Hardware Configuration** window opens.



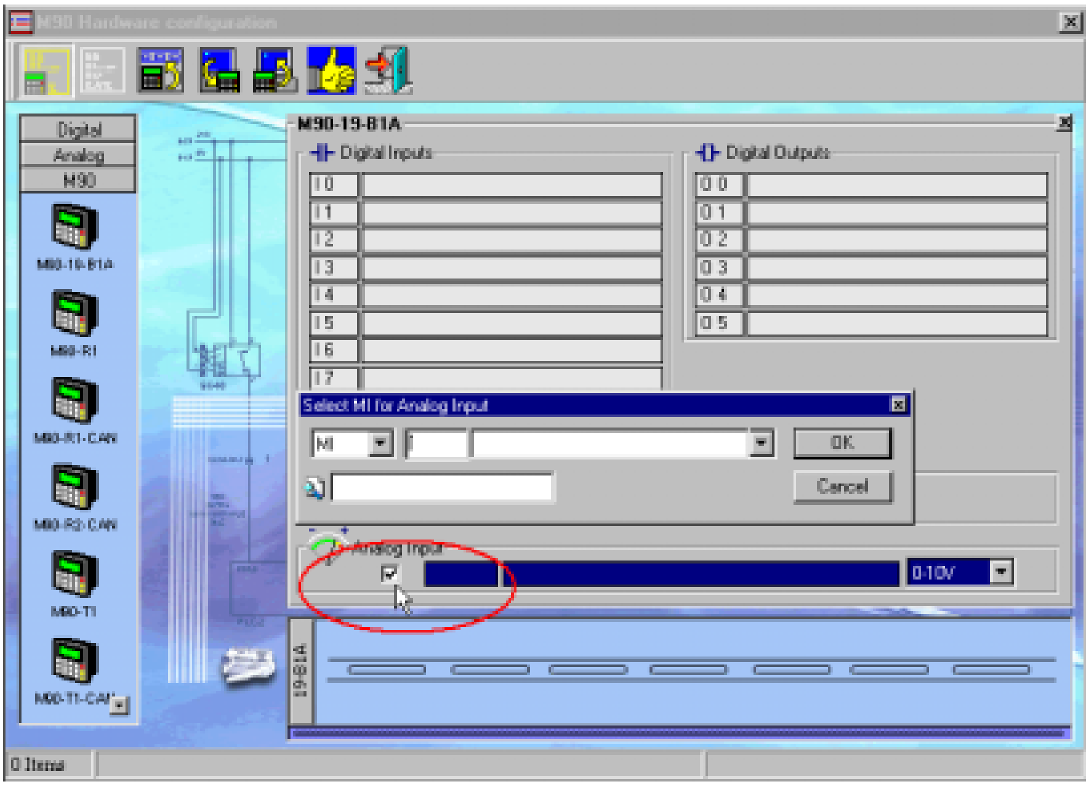
3. Click the appropriate M90 model.



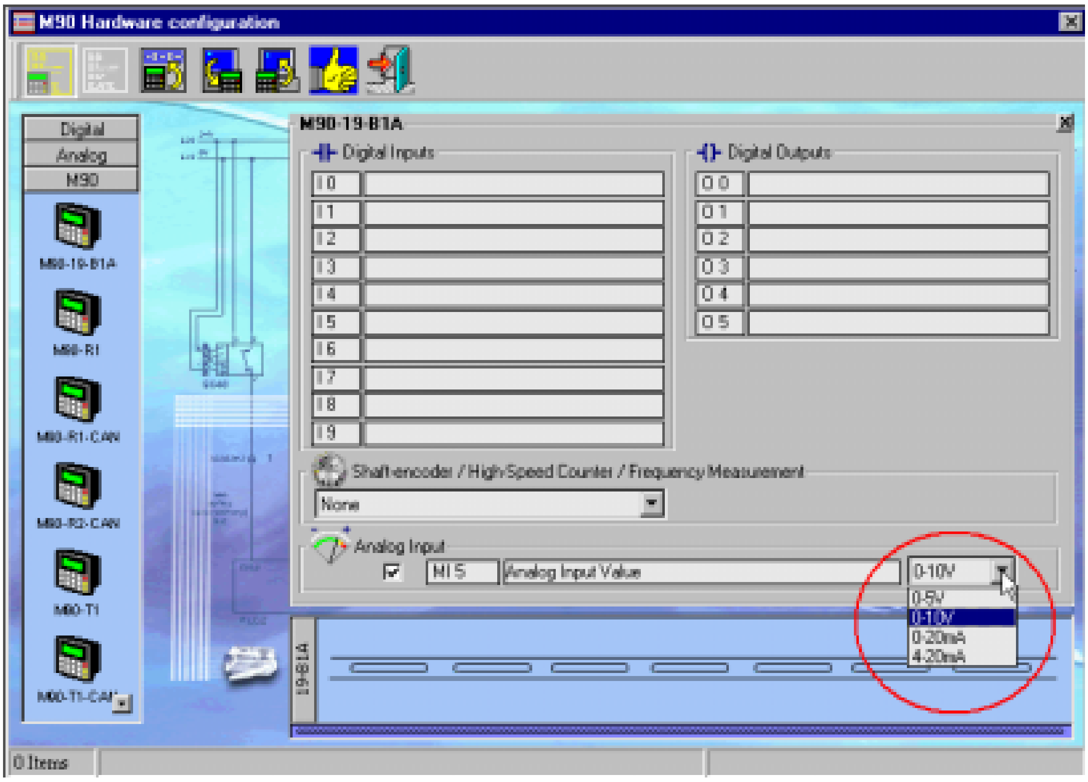
M90-19-B1A; 10/6 Digital I/O, 1 Analog In

4. The I/O options for that model are displayed.

5. Check the **Analog Input** check box. The **Select MI for Analog Input** window opens.

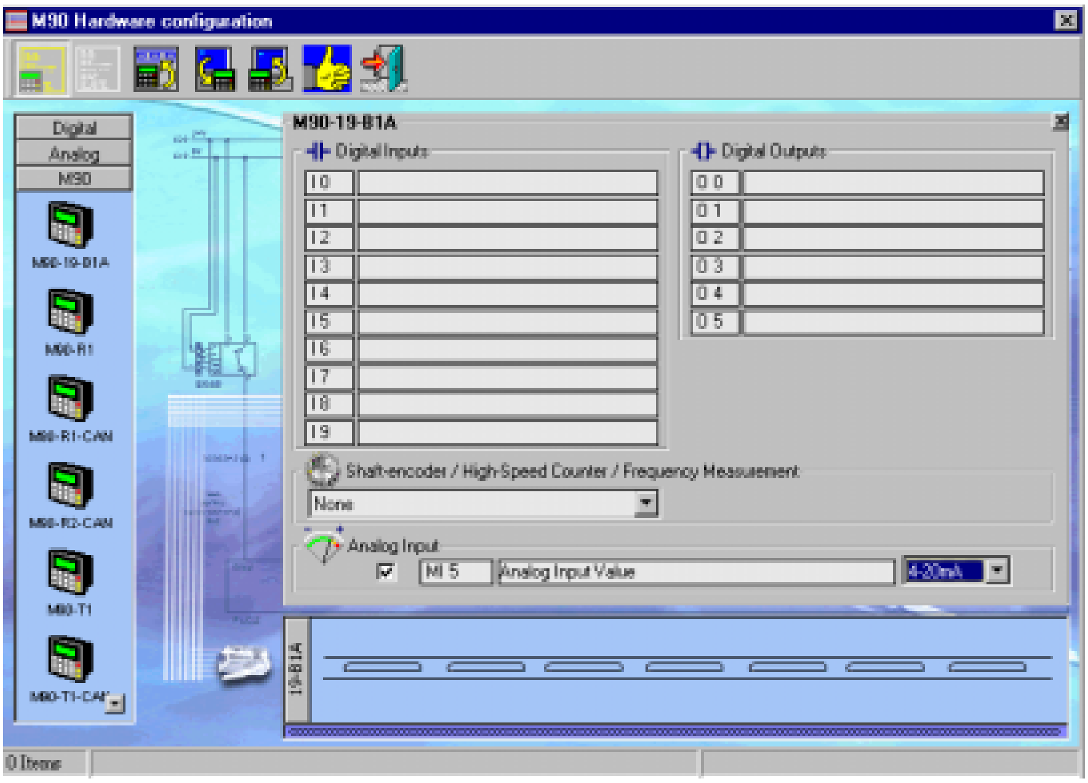


6. Enter the desired Address and Symbol of the MI Operand. Select the Analog Input type from the drop-down menu.



Analog Input ☑ MI 5 Analog Input Value 0-10V

7. The M90 Hardware Configuration window now appears with the new Analog Input configuration.



M90 Hardware configuration

Digital
Analog
M90

M90-19-B1A
M90-R1
M90-R1-CAN
M90-R2-CAN
M90-T1
M90-T1-CAN

M90-19-B1A

Digital Inputs

I 0
I 1
I 2
I 3
I 4
I 5
I 6
I 7
I 8
I 9

Digital Outputs

O 0
O 1
O 2
O 3
O 4
O 5

Shaft-encoder / High-Speed Counter / Frequency Measurement

None

Analog Input

☑ MI 5 Analog Input Value 4-20mA

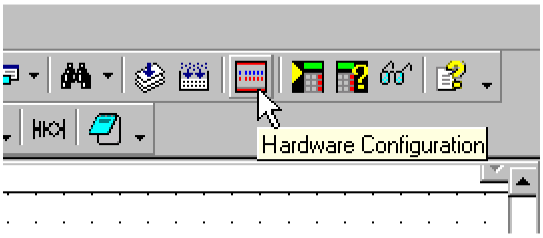
19-B1A

0 Items

M91

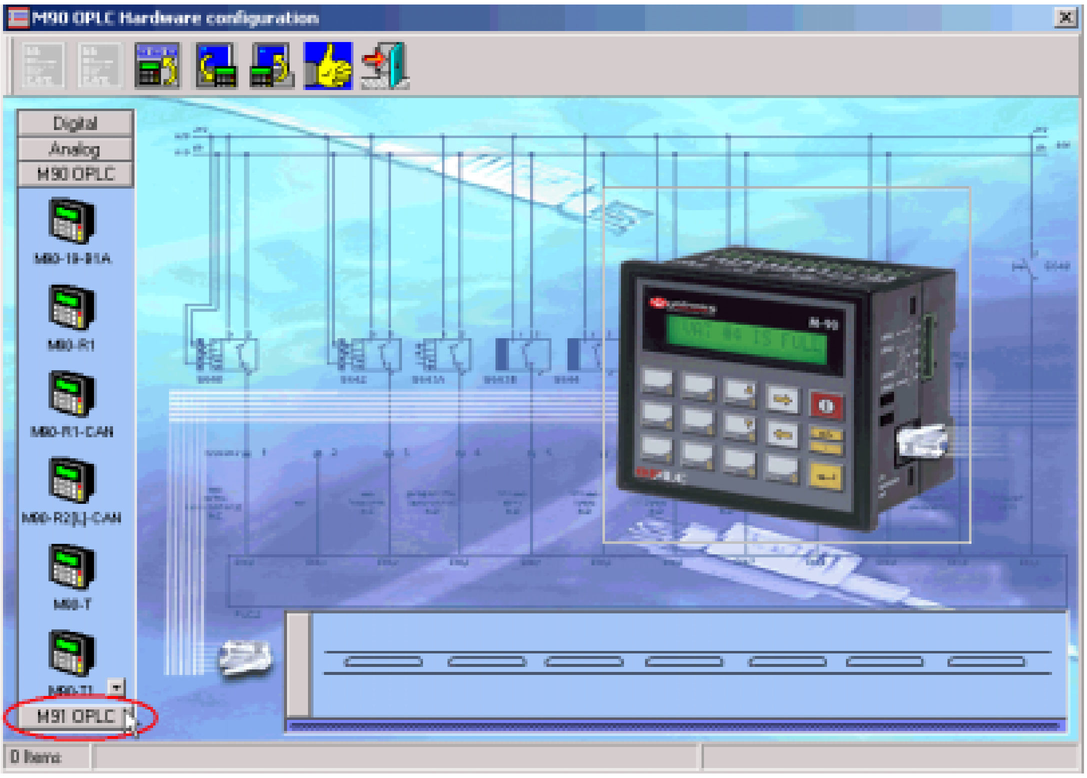
To attach an Analog Input to an MI:

1. Click **Hardware Configuration** on the Standard toolbar.



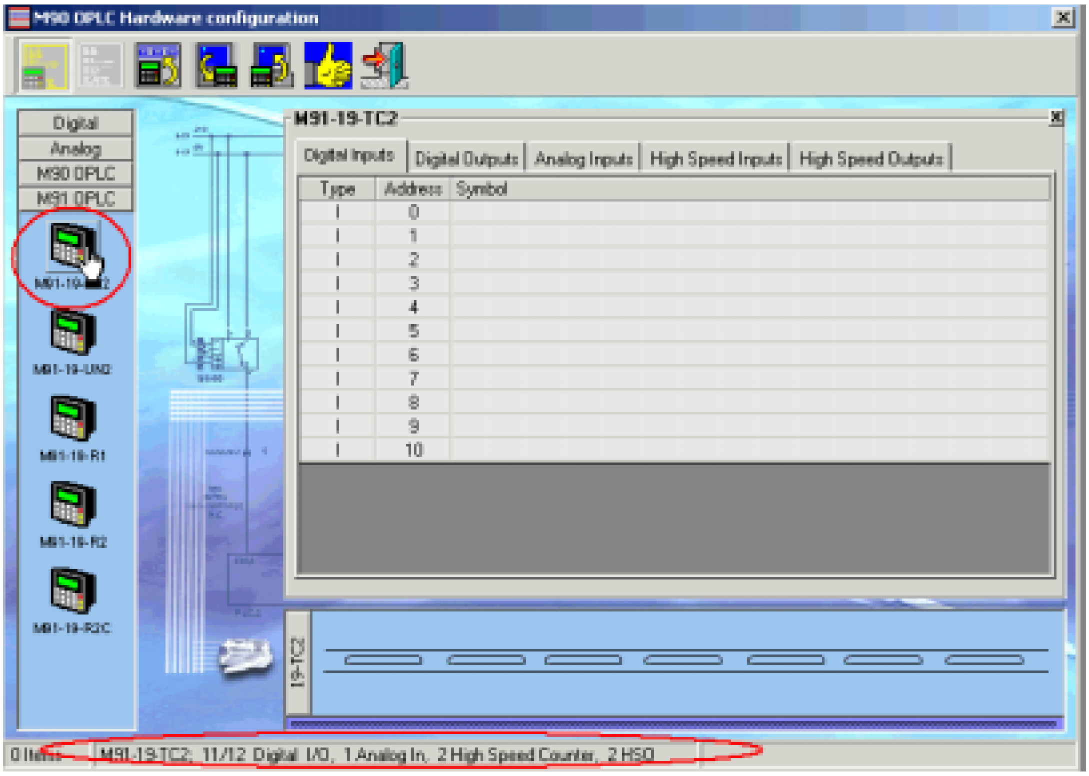
2. The **M90 Hardware Configuration** window opens.

3. Click on the M91 bar.



3. Select the appropriate M91 model; the model's I/O options are displayed.

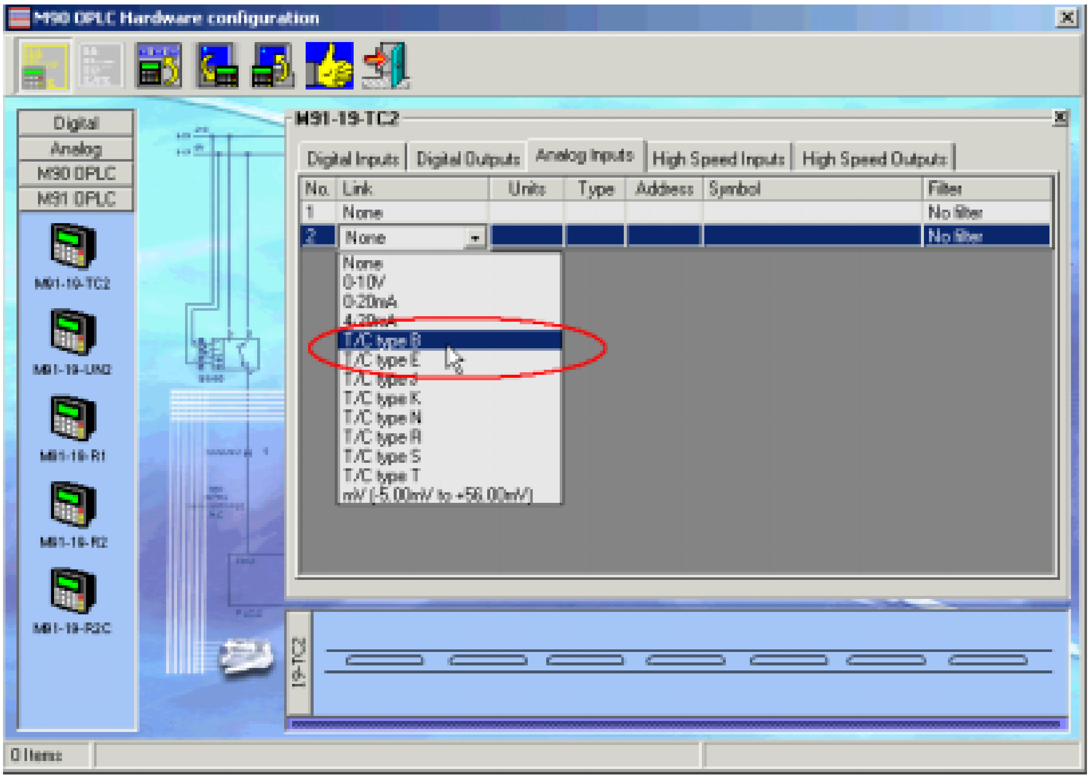
4. Click on the Analog Inputs tab.



Type	Address	Symbol
I	0	
I	1	
I	2	
I	3	
I	4	
I	5	
I	6	
I	7	
I	8	
I	9	
I	10	

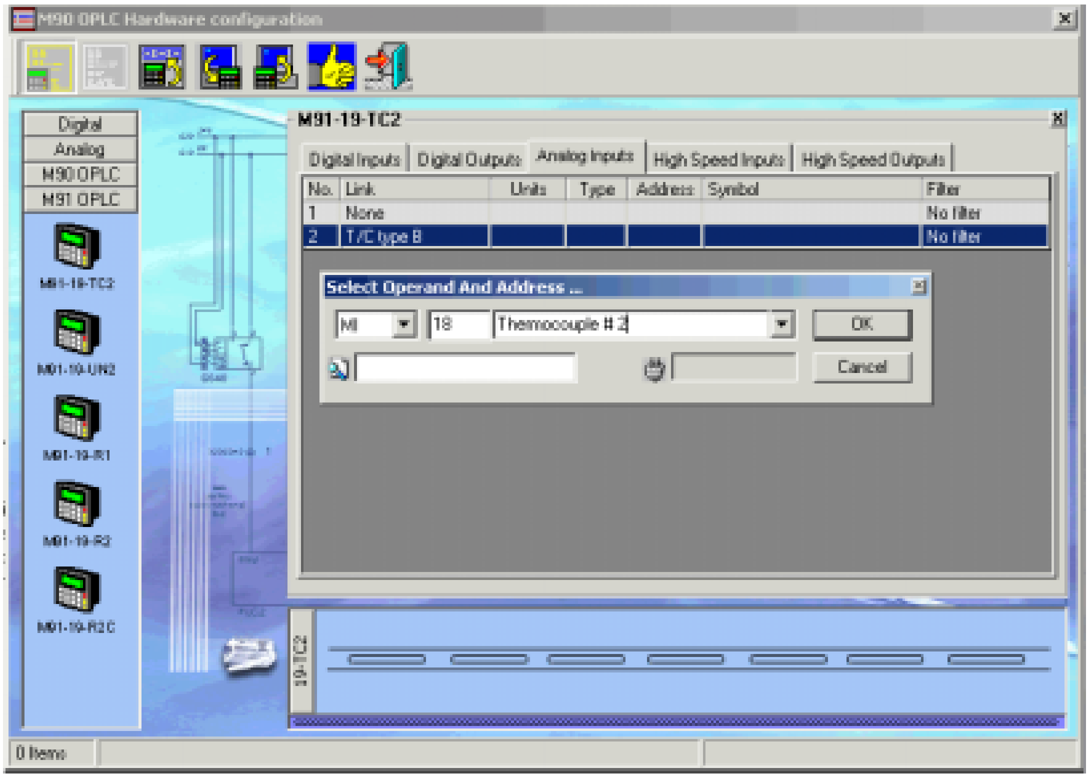
M91-19-TC2: 11/12 Digital I/O, 1 Analog In, 2 High Speed Counter, 2 HSO

5. Click the Link field, then select the desired type of input. The **Select MI for Analog Input** window opens.



No.	Link	Units	Type	Address	Symbol	Filter
1	None					No filter
2	None					No filter

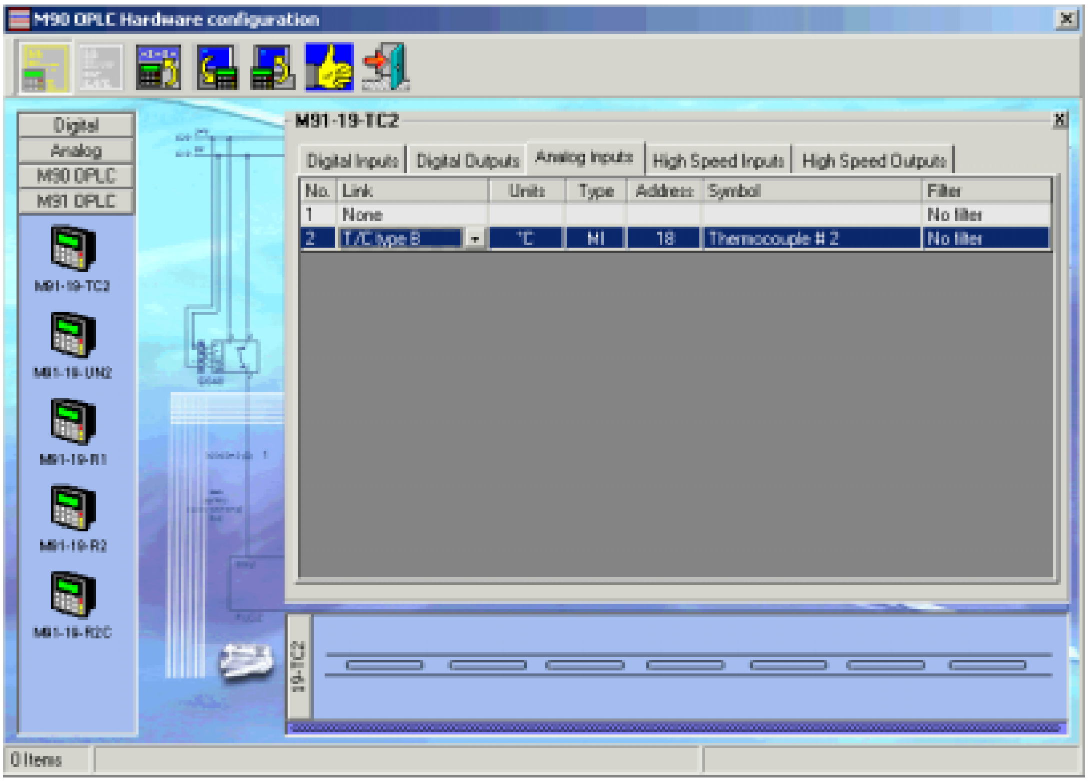
6. Enter the desired Address and Symbol of the MI Operand.



No.	Link	Units	Type	Address	Symbol	Filter
1	None					No filter
2	T/C type B					No filter

MI 18 Thermocouple # 2

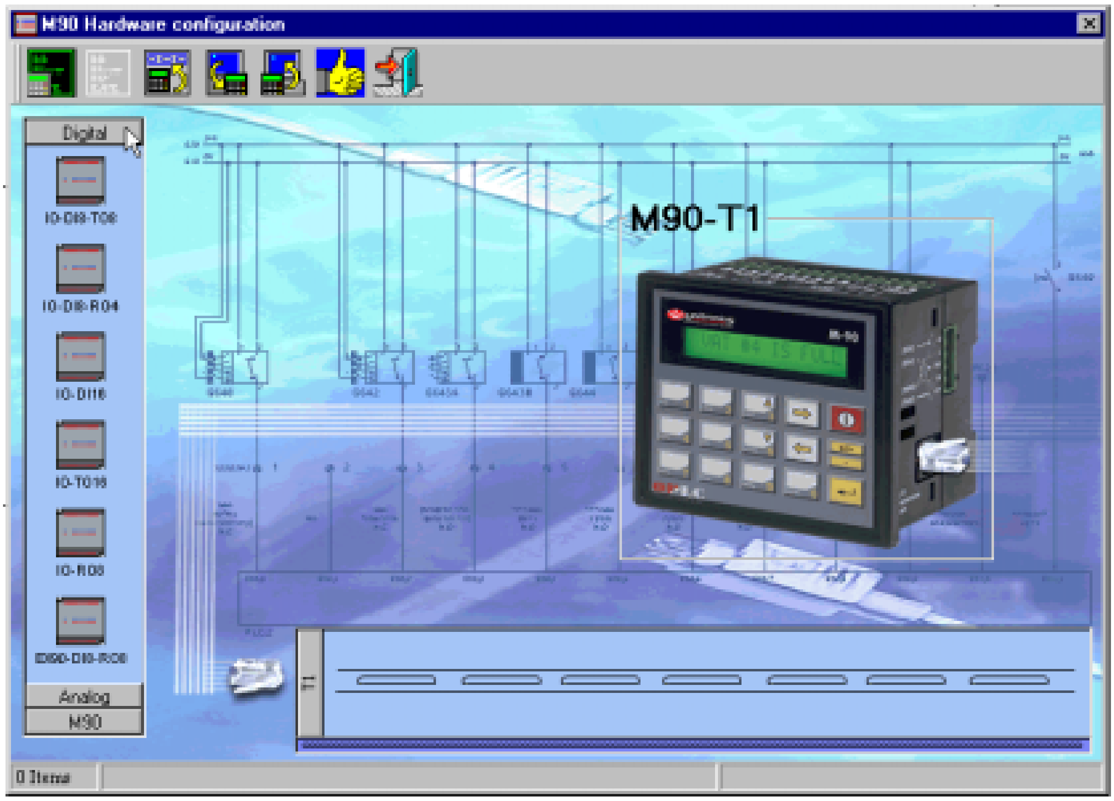
7. The Analog Input is now part of the configuration.



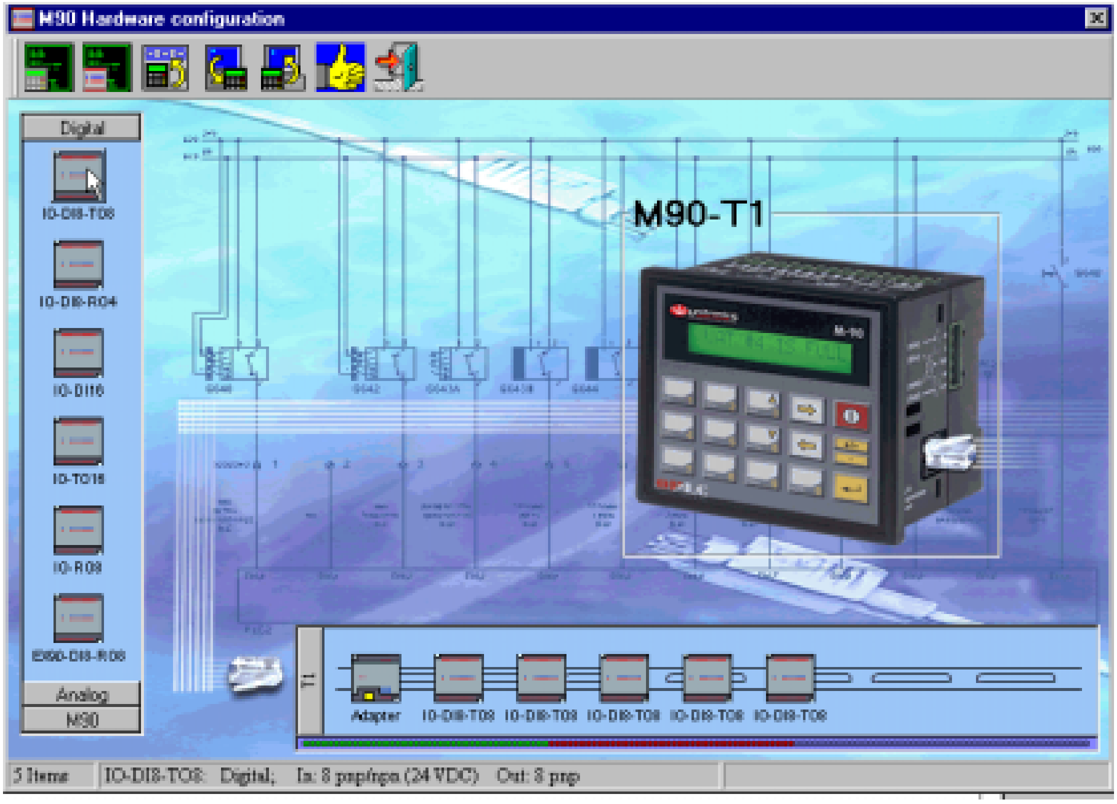
No.	Link	Units	Type	Address	Symbol	Filter
1	None					No filter
2	T/C type B	°C	MI	18	Thermocouple # 2	No filter

Configuring I/O Expansion Modules

1. Open the Digital or Analog menu according to the expansion you are connecting.

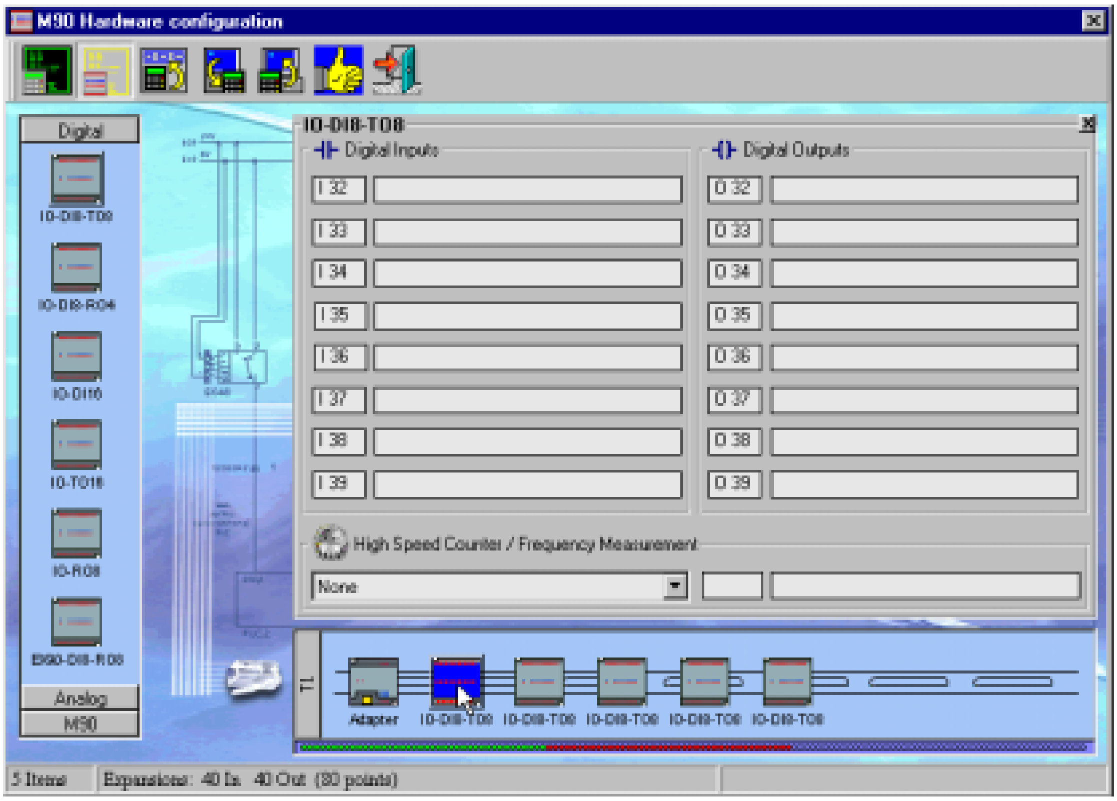


2. Double-click on the appropriate I/O module. The selected module(s) will appear on the Model Expansion bar.
3. Continue adding I/O expansion modules according to your expansion configuration.



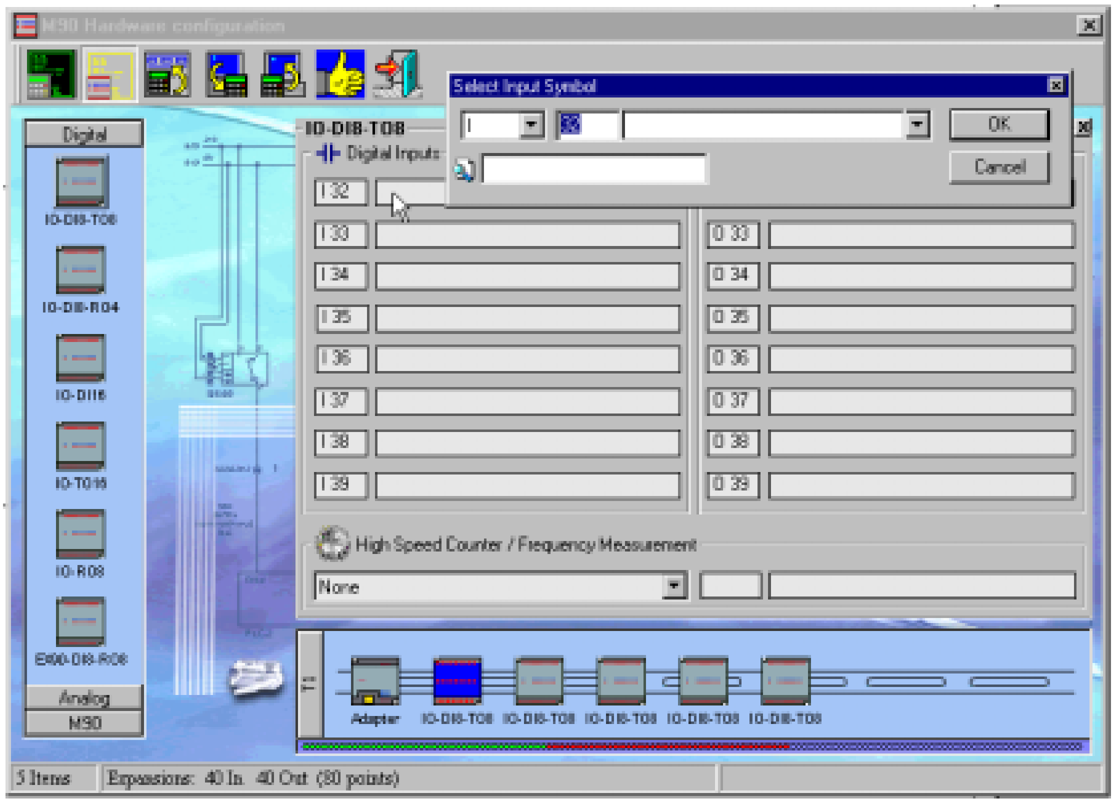
5 Items | IO-DI8-TO8: Digital; In: 8 pnp/npn (24 VDC) Out: 8 pnp

4. Double-click on an I/O expansion icon in the Model Expansion bar. The I/O Details window opens.

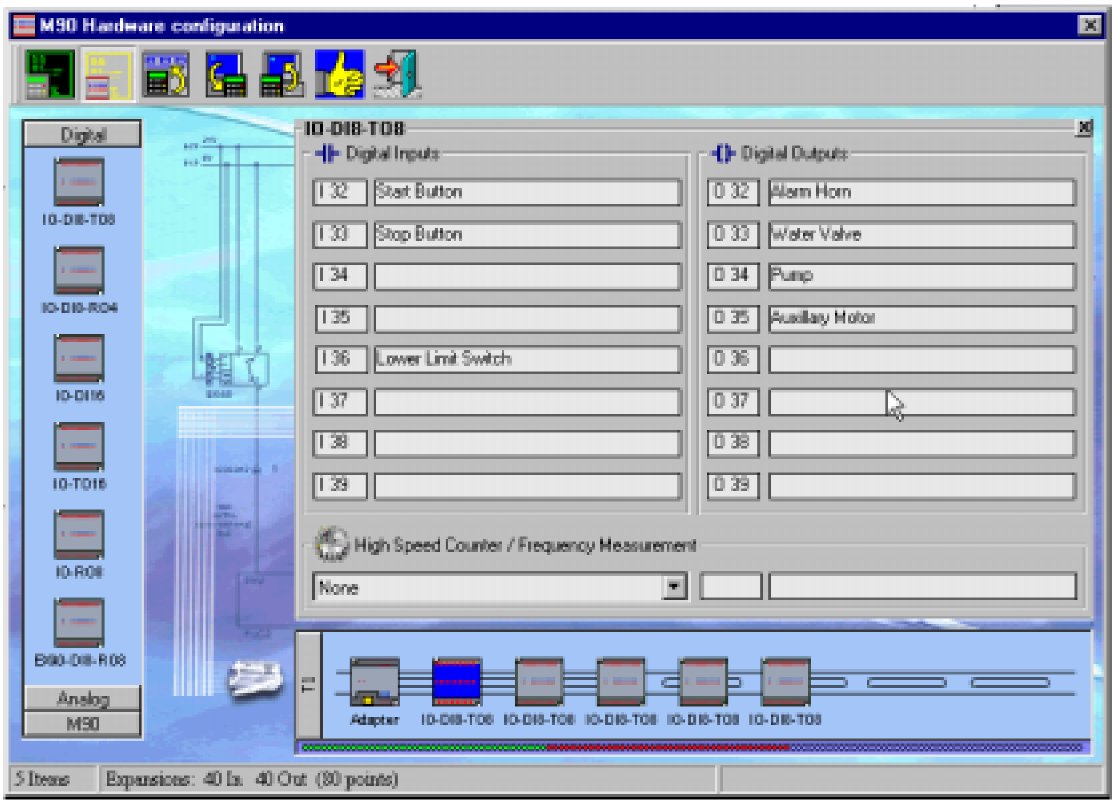


5 Items | Expansions: 40 In 40 Out (80 points)

5. Click on the appropriate Inputs / Outputs to enter the desired Addresses and Symbols.

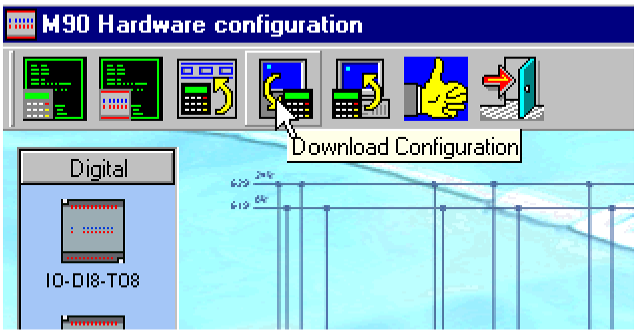


6. The Addresses and Symbols appear in the I/O Details window.

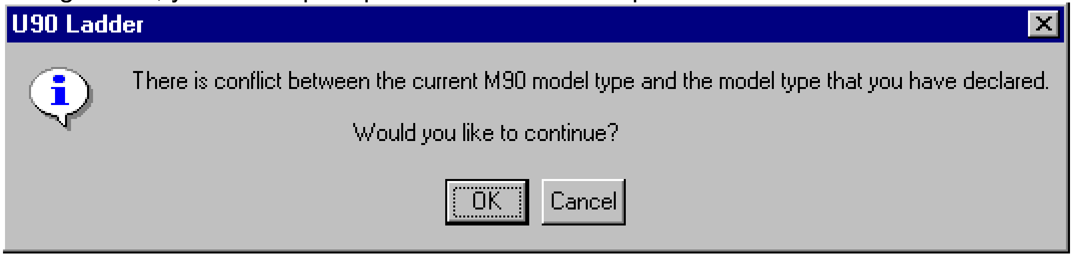


I 32	Start Button	O 32	Alarm Horn
I 33	Stop Button	O 33	Water Valve
I 34		O 34	Pump
I 35		O 35	Auxiliary Motor
I 36	Lower Limit Switch	O 36	
I 37		O 37	
I 38		O 38	
I 39		O 39	

7. Click the Download Configuration icon.



If there is a conflict between the current M90 hardware information and the project configuration, you will be prompted to choose how to proceed.

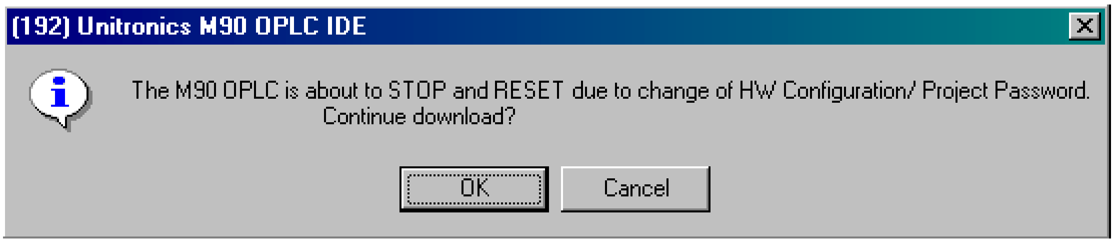

U90 Ladder

There is conflict between the current M90 model type and the model type that you have declared.

Would you like to continue?

If you decide to continue with the Download, the M90 OPLC will be stopped and reset during the Download procedure.

8. Click OK. The Download process is activated.


(192) Unitronics M90 OPLC IDE

The M90 OPLC is about to STOP and RESET due to change of HW Configuration/ Project Password. Continue download?

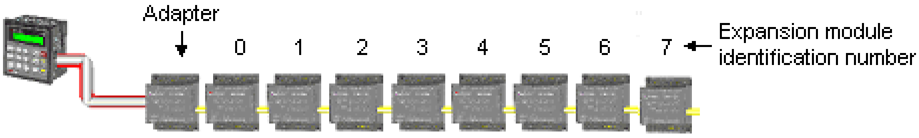
The Hardware configuration is now updated.

Note: If your application does not require that you use all of the I/Os on a particular I/O expansion module, do not select the unused I/Os when you configure the module. Selecting unused I/Os may add to the M90's scan time

Addressing: I/O Expansion Modules

Inputs and outputs located on I/O expansion modules that are connected into an M90 OPLC are assigned addresses that comprise a letter and a number. The letter indicates whether the I/O is an input (I) or an output (O). The number indicates the I/O's location in the system. This number relates to both the expansion module's position in the system, and to the position of the I/O on that module.

Expansion modules are numbered from 0-7 as shown in the figure below.



The formula below is used to assign addresses for I/O modules used in conjunction with the M90 OPLC.

X is the number representing a specific module's location (0-7). Y is the number of the input or output on that specific module (0-15).

The number that represents the I/O's location is equal to: **32 + x • 16 + y**

Example

- Input #3, located on expansion module #2 in the system, will be addressed as I 67, 67 = 32 + 2 • 16 + 3
- Output #4, located on expansion module #3 in the system, will be addressed as O 84, 84 = 32 + 3 • 16 + 4.

EX90-DI8-RO8 is a stand-alone I/O module. Even if it is the only module in the configuration, the EX90-DI8-RO8 is always assigned the number 7. Its I/Os are addressed accordingly.

Example

- Input #5, located on an EX90-DI8-RO8 connected to an M90 OPLC will be addressed as I 149, 149 = 32 + 7 • 16 + 5

Hardware Configuration is featured in several sample applications, such as the applications ' HSC x 1000', 'HSC saved', 'High-speed Output', 'Motor Speed', and 'Expansion HSC Reset'. These applications may be found by selecting Sample U90 Projects from the Help Menu.

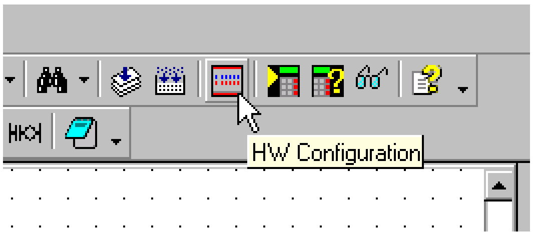
Configuring I/O Expansion Modules

Certain M90 models can be hooked up to I/O Expansion Modules.

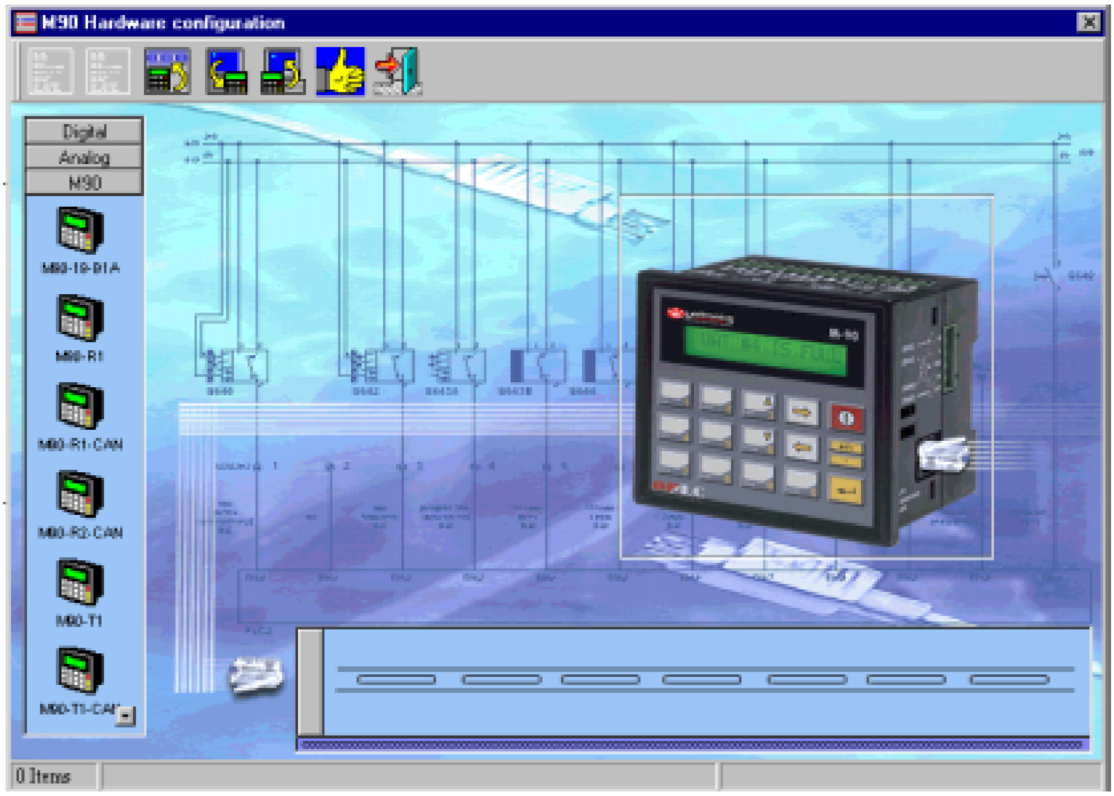
You must configure the M90 according to the I/O Expansion Modules you are connecting.

Adding I/O Expansion Modules to your Hardware Configuration

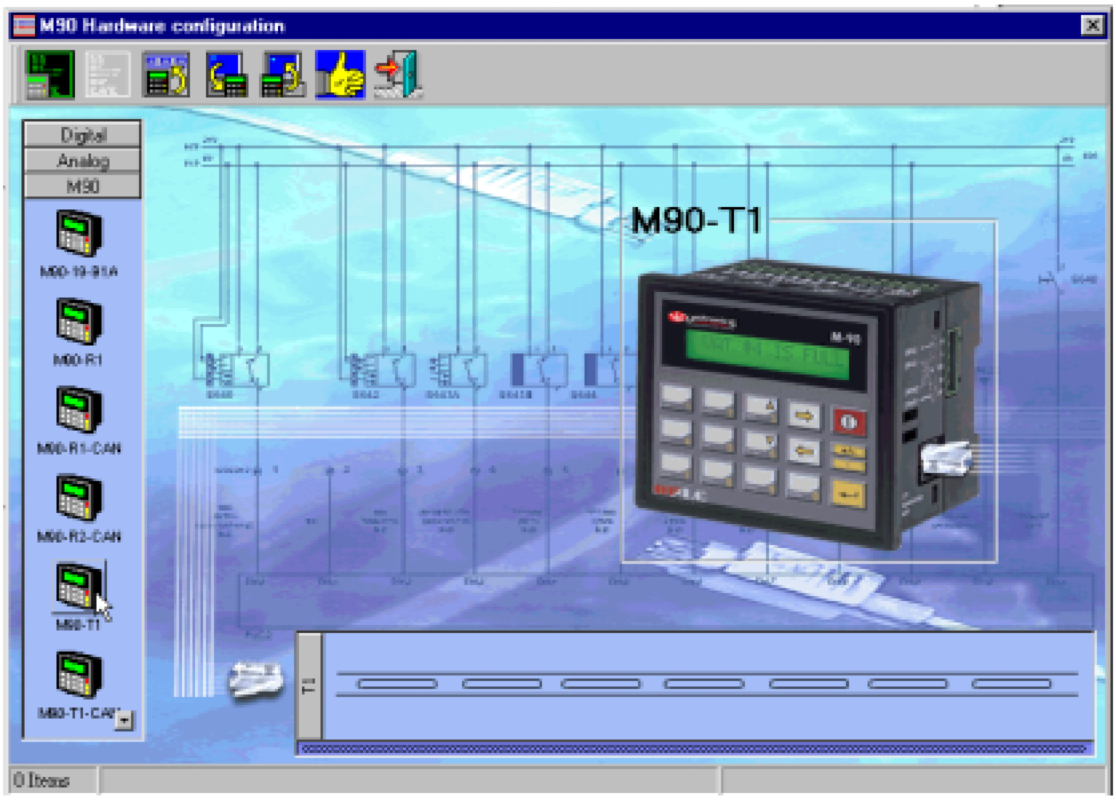
1. Click on the **Hardware Configuration** icon on the Standard toolbar.



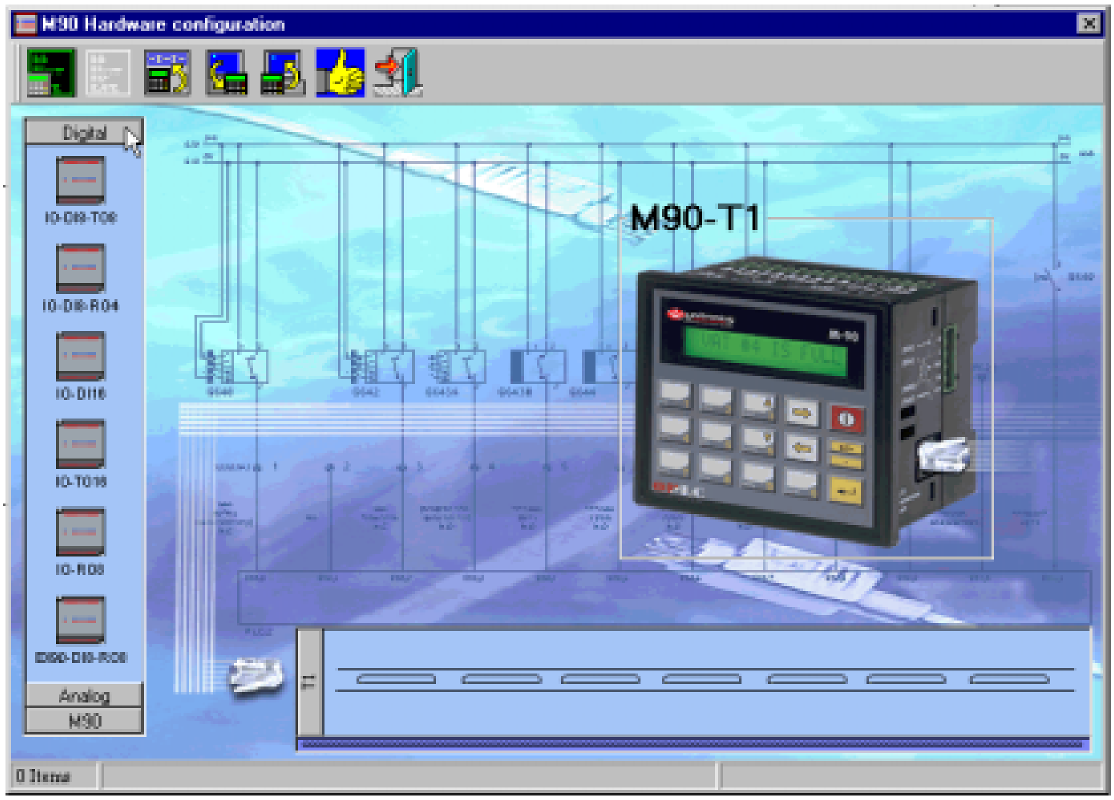
2. The **M90 Hardware configuration** window opens.



3. Select the M90 model for your project application from the M90 icon menu.



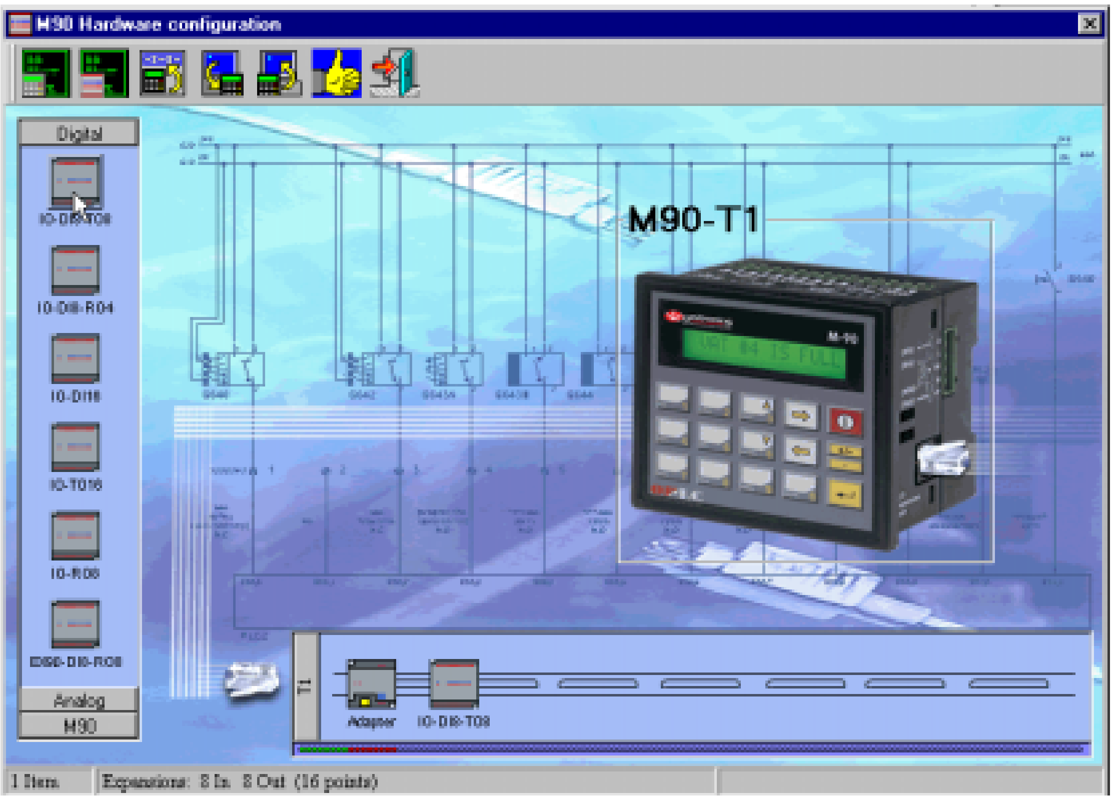
4. The selected model name appears above the M90 controller. Open the Digital or Analog menu according to the module you are connecting.



M90 Hardware configuration

0 Items

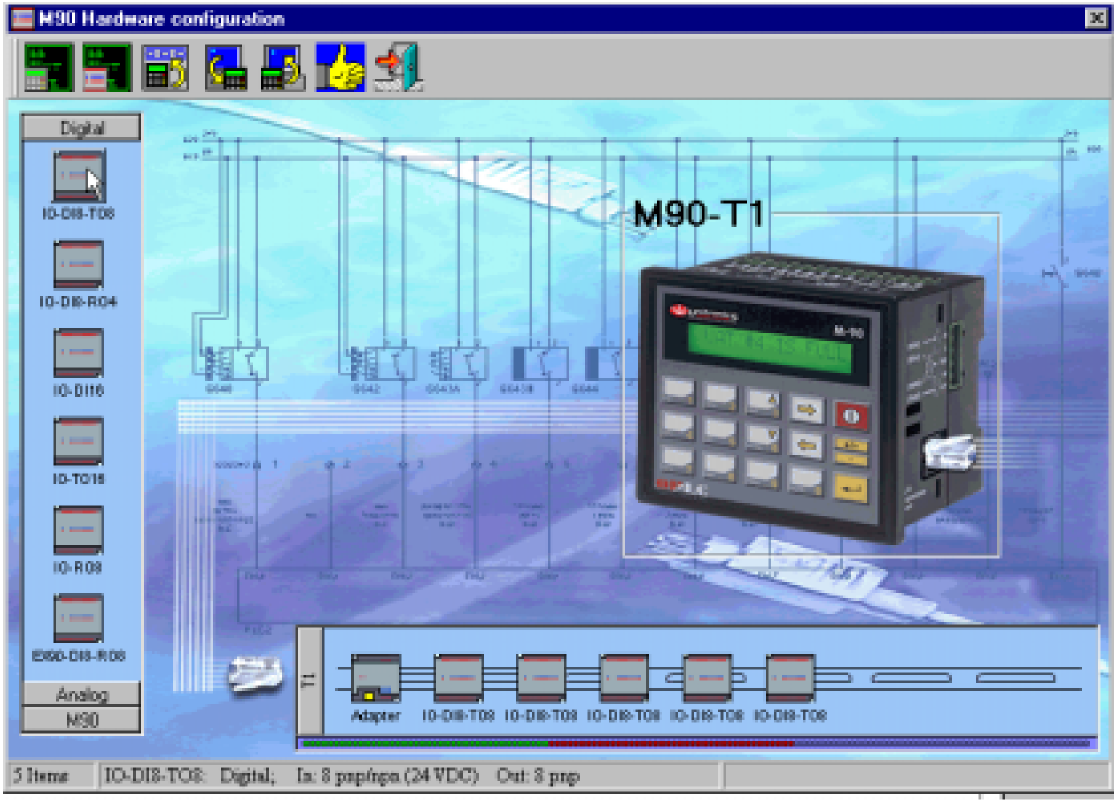
5. Double-click on the appropriate I/O module. The selected module(s) will appear on the Module Expansion bar.



M90 Hardware configuration

1 Item   Expansions: 8 In 8 Out (16 points)

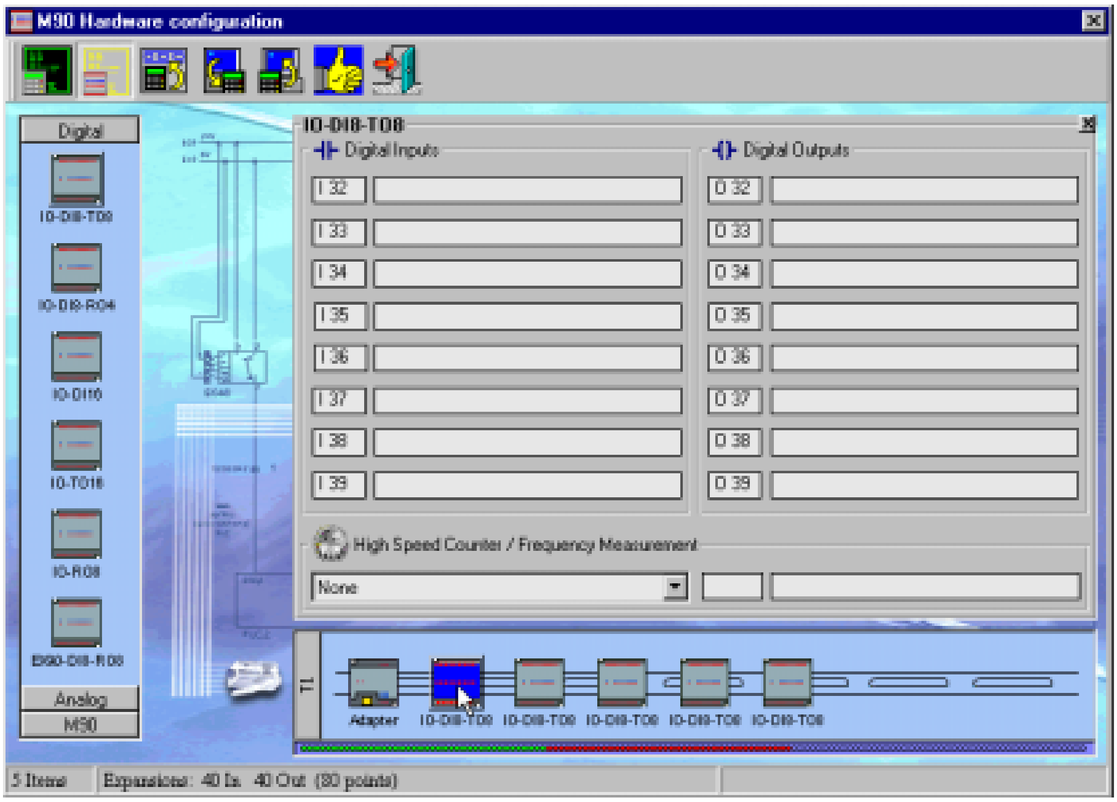
6. Continue adding I/O Expansion Modules according to your expansion configuration.



5 Items | IO-DI8-TO8: Digital; In: 8 pnp/npn (24 VDC) Out: 8 pnp

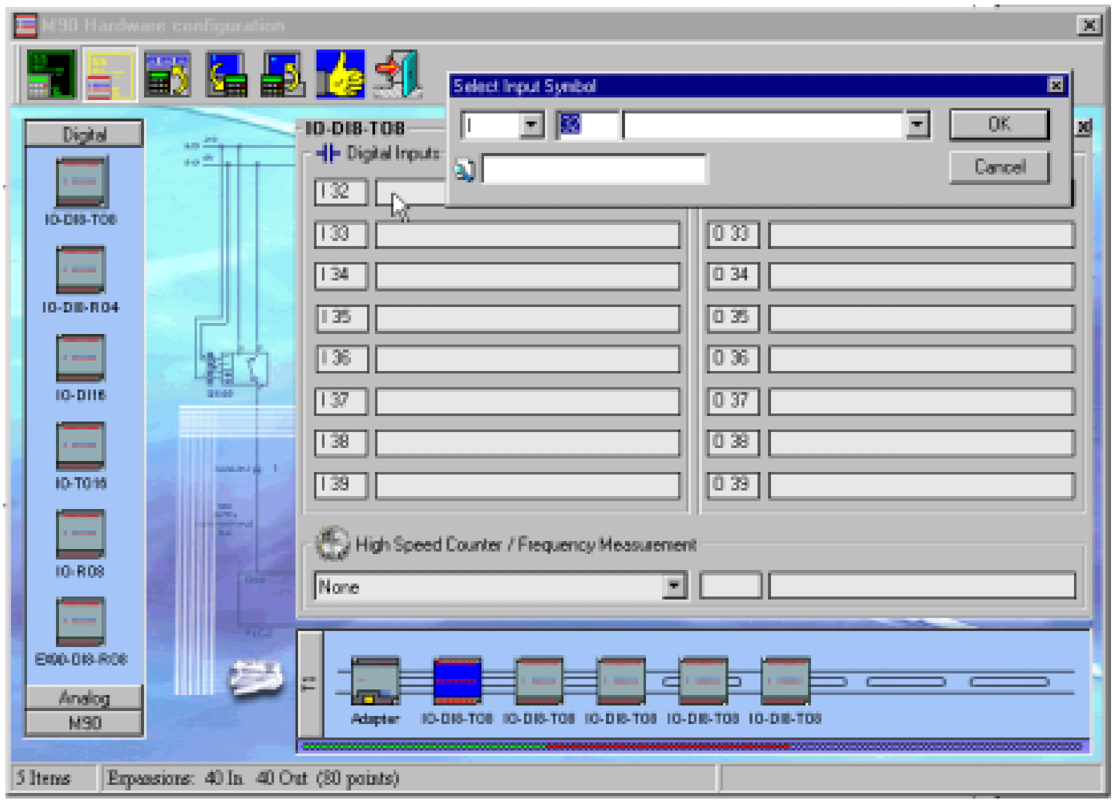
Configuring I/Os: Linking Operands

1. Double-click on an I/O expansion icon in the Model Expansion bar. An I/O Details window opens.

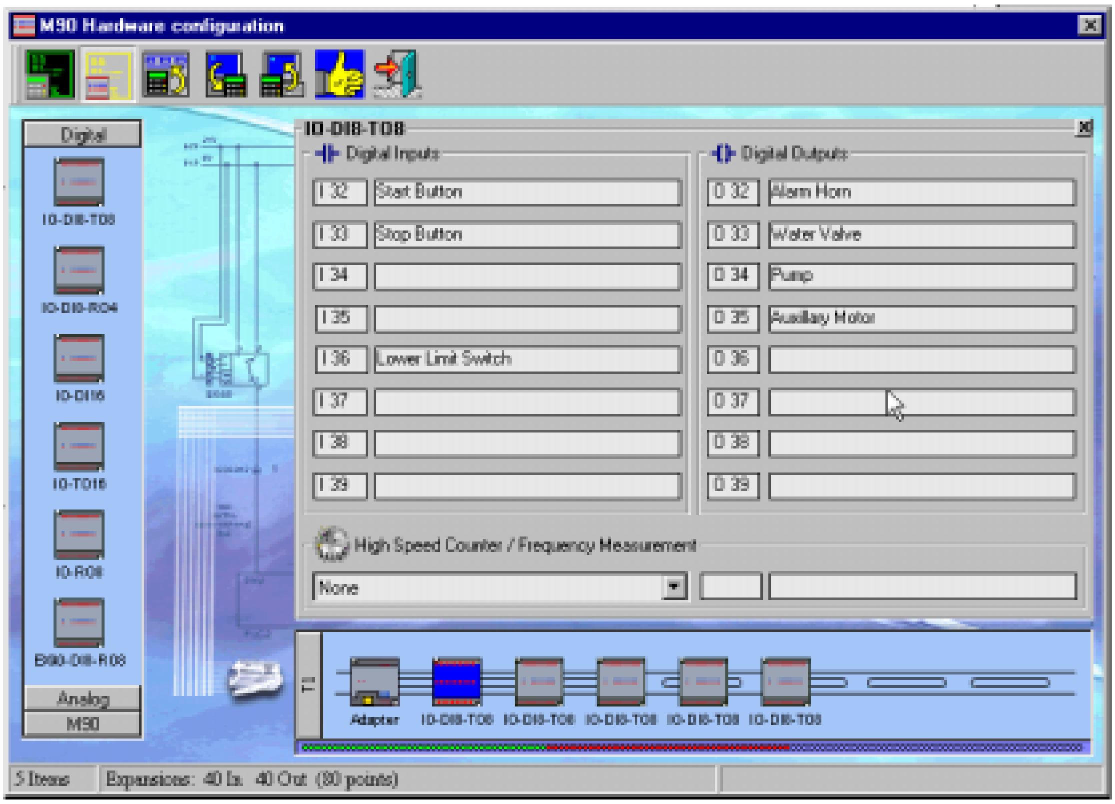


5 Items | Expansions: 40 In 40 Out (80 points)

2. Click on the appropriate Inputs / Outputs to enter the desired Addresses and Symbols.



3. The Addresses and Symbols appear in the I/O Details window.

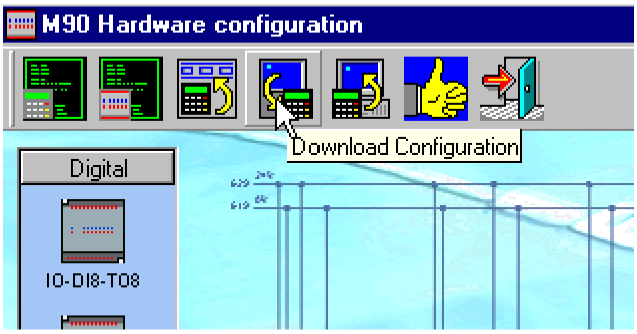


Digital Inputs		Digital Outputs	
I 32	Start Button	O 32	Alarm Horn
I 33	Stop Button	O 33	Water Valve
I 34		O 34	Pump
I 35		O 35	Auxiliary Motor
I 36	Lower Limit Switch	O 36	
I 37		O 37	
I 38		O 38	
I 39		O 39	

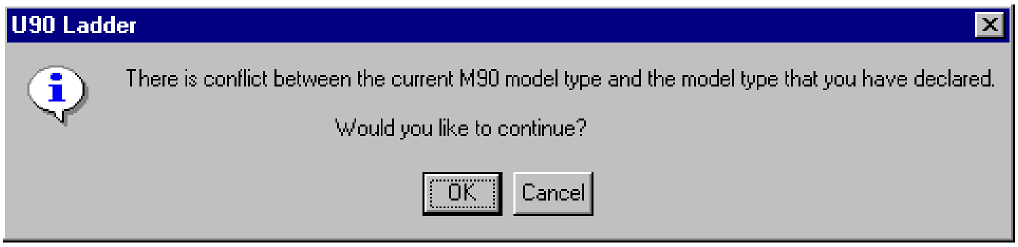
High Speed Counter / Frequency Measurement: None

Downloading Hardware Configuration properties

1. Click the Download Configuration icon.



2. If there is a conflict between the current M90 hardware information and the project configuration, you will be prompted to choose how to proceed.

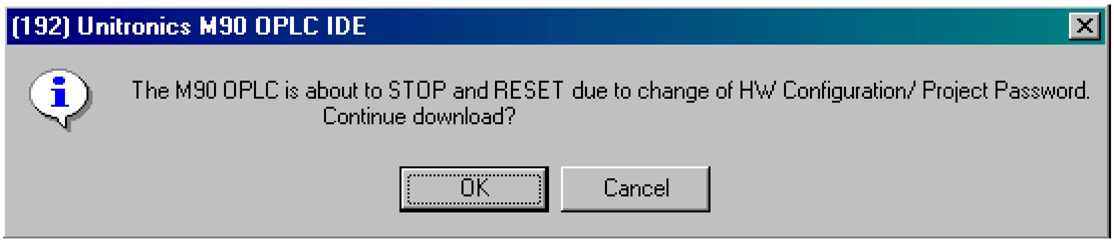


U90 Ladder

There is conflict between the current M90 model type and the model type that you have declared.

Would you like to continue?

3. If you decide to continue with the Download, the M90 OPLC will be stopped and reset during the Download procedure. Click OK. The Download process is activated.



(192) Unitronics M90 OPLC IDE

The M90 OPLC is about to STOP and RESET due to change of HW Configuration/ Project Password. Continue download?

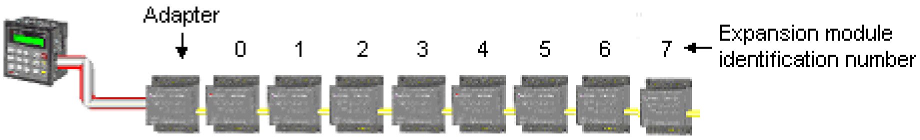
The Hardware configuration is now updated.

Note: If your application does not require that you use all of the I/Os on a particular I/O Expansion Module, do not select the unused I/Os when you configure the module. Selecting unused I/Os may add to the M90's scan time.

Addressing: I/O Expansion Modules

Inputs and outputs located on I/O expansion modules that are connected into an M90 OPLC are assigned addresses that comprise a letter and a number. The letter indicates whether the I/O is an input (I) or an output (O). The number indicates the I/O's location in the system. This number relates to both the expansion module's position in the system, and to the position of the I/O on that module.

Expansion modules are numbered from 0-7 as shown in the figure below.



The formula below is used to assign addresses for I/O modules used in conjunction with the M90 OPLC.

X is the number representing a specific module's location (0-7). Y is the number of the input or output on that specific module (0-15).

The number that represents the I/O's location is equal to: **32 + x • 16 + y**

Example

- Input #3, located on expansion module #2 in the system, will be addressed as I 67, 67 = 32 + 2 • 16 + 3
- Output #4, located on expansion module #3 in the system, will be addressed as O 84, 84 = 32 + 3 • 16 + 4.

EX90-DI8-RO8 is a stand-alone I/O module. Even if it is the only module in the configuration, the EX90-DI8-RO8 is always assigned the number 7. Its I/Os are addressed accordingly.

Example

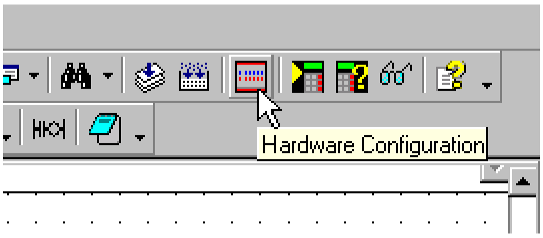
- Input #5, located on an EX90-DI8-RO8 connected to an M90 OPLC will be addressed as I 149, 149 = 32 + 7 • 16 + 5

Configuring an Analog Input

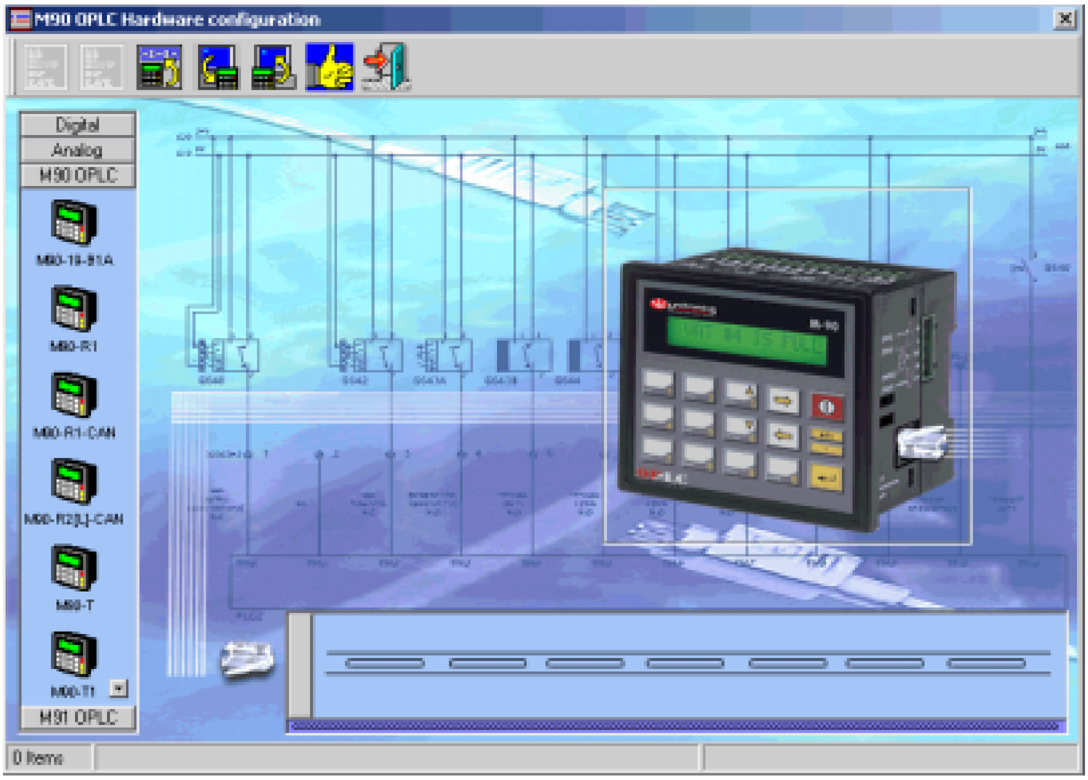
M90

To attach an Analog Input to an MI:

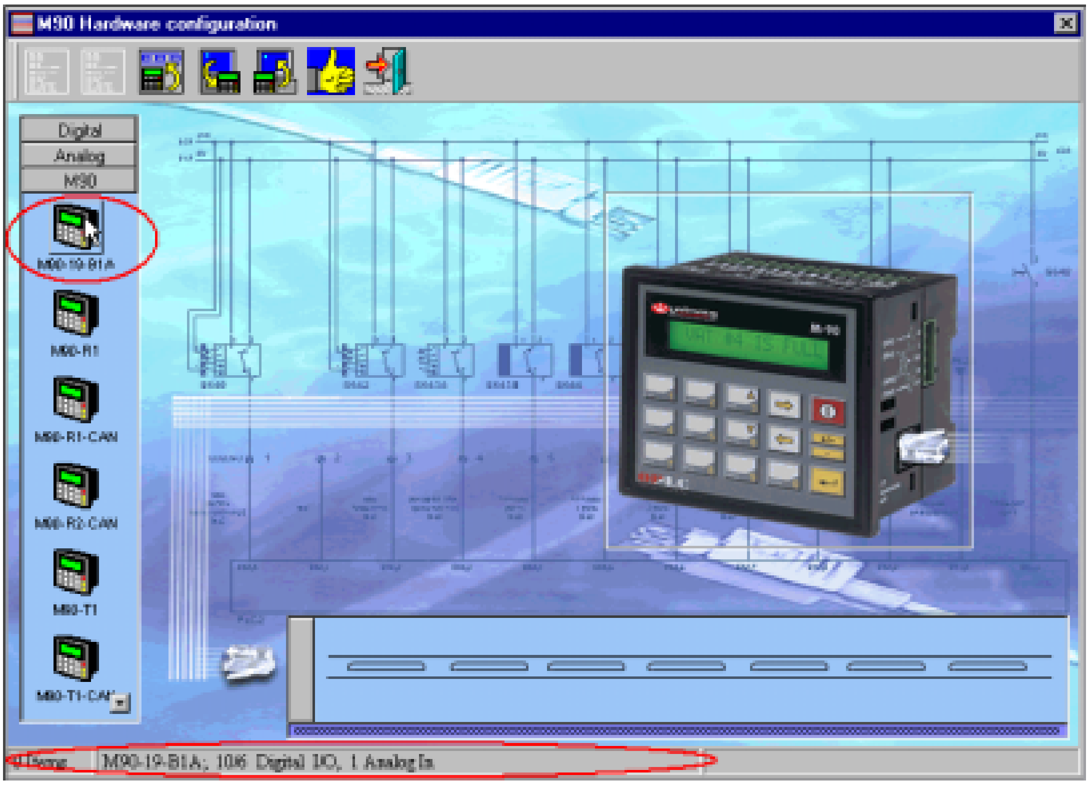
1. Click **Hardware Configuration** on the Standard toolbar.



2. The **M90 Hardware Configuration** window opens.

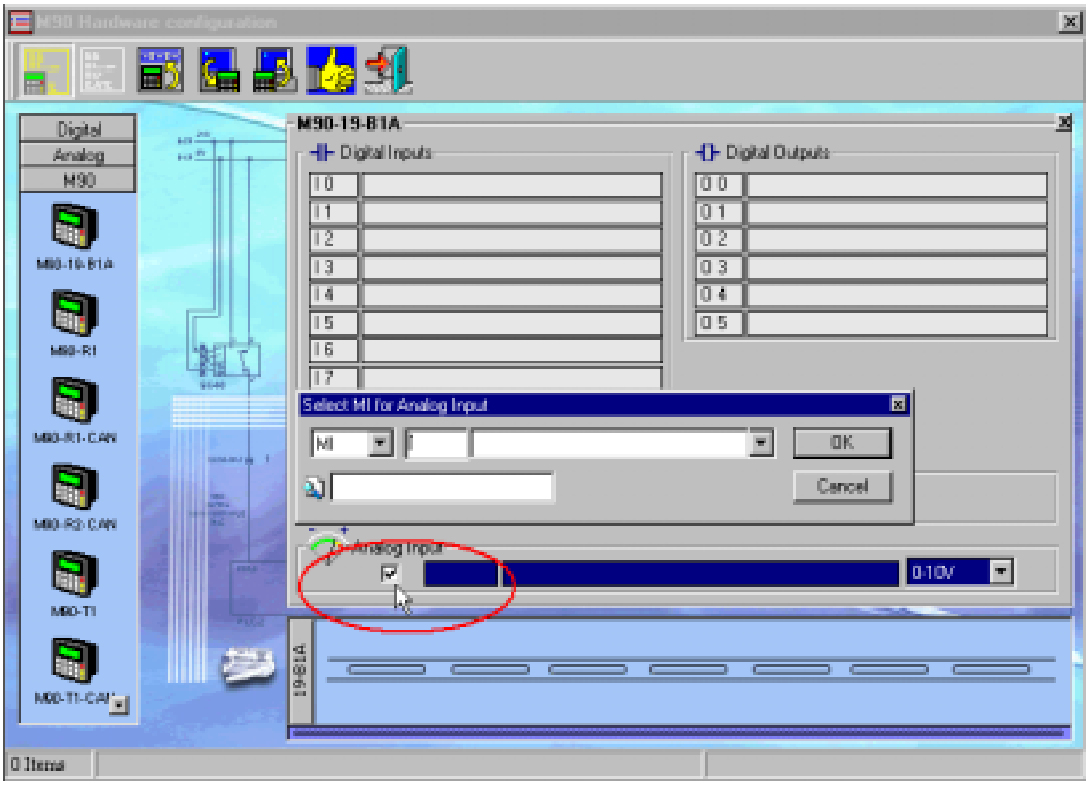


3. Click the appropriate M90 model.

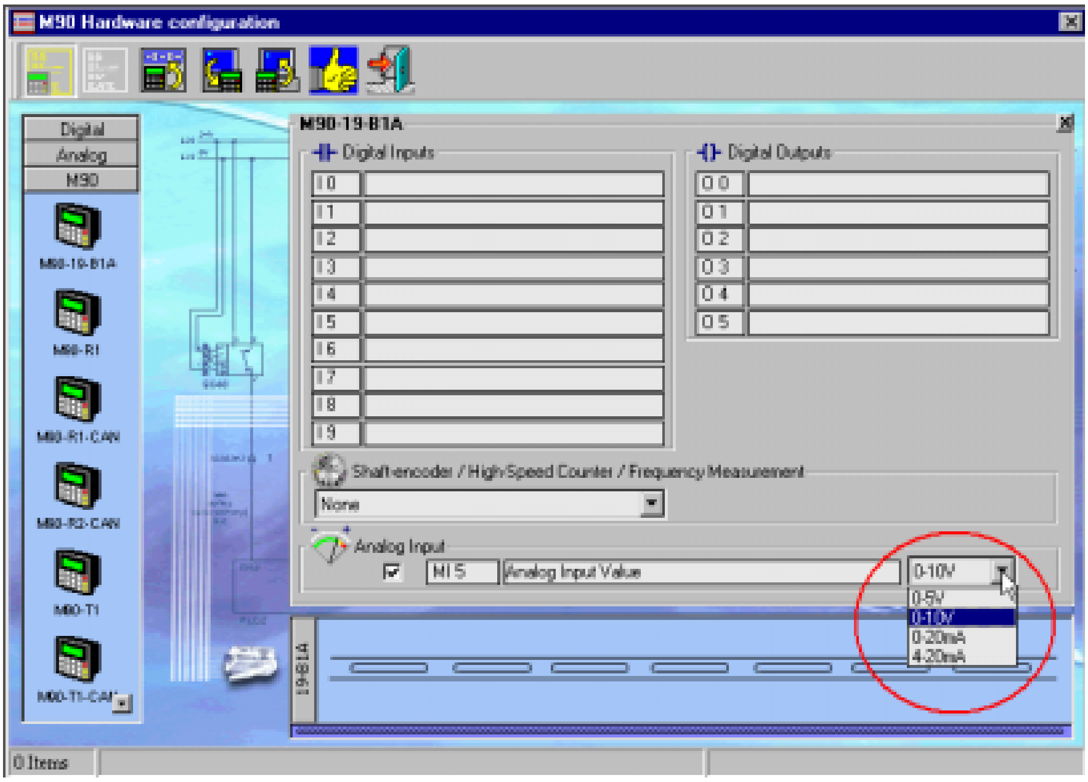


4. The I/O options for that model are displayed.

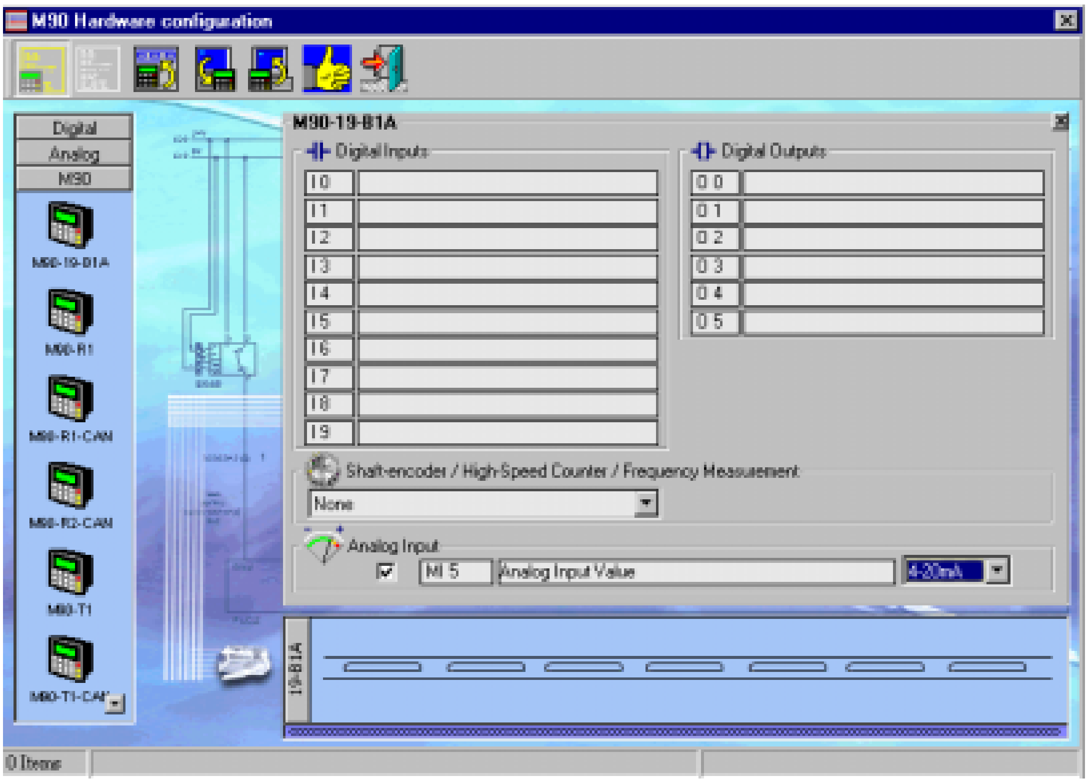
5. Check the **Analog Input** check box. The **Select MI for Analog Input** window opens.



6. Enter the desired Address and Symbol of the MI Operand. Select the Analog Input type from the drop-down menu.



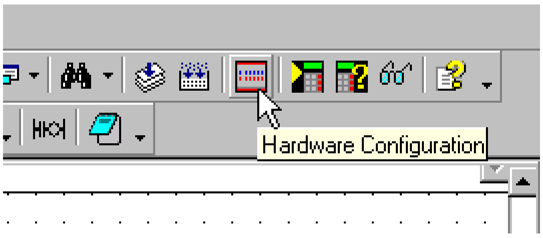
7. The M90 Hardware Configuration window now appears with the new Analog Input configuration.



M91

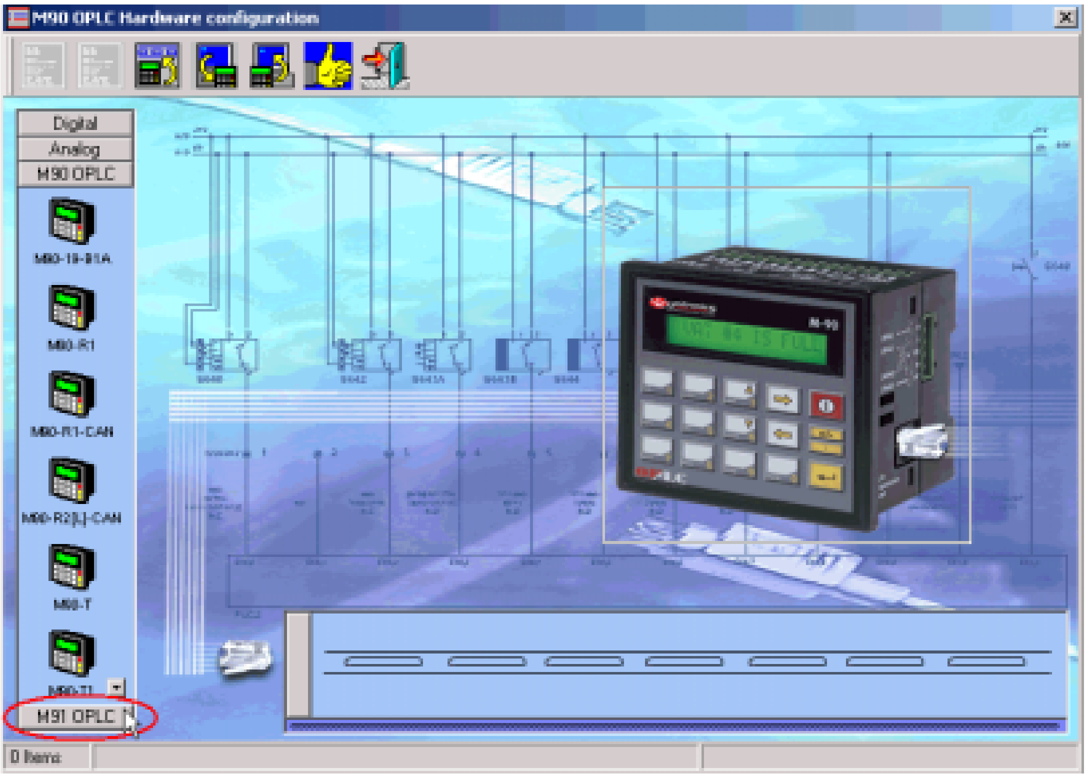
To attach an Analog Input to an MI:

1. Click **Hardware Configuration** on the Standard toolbar.



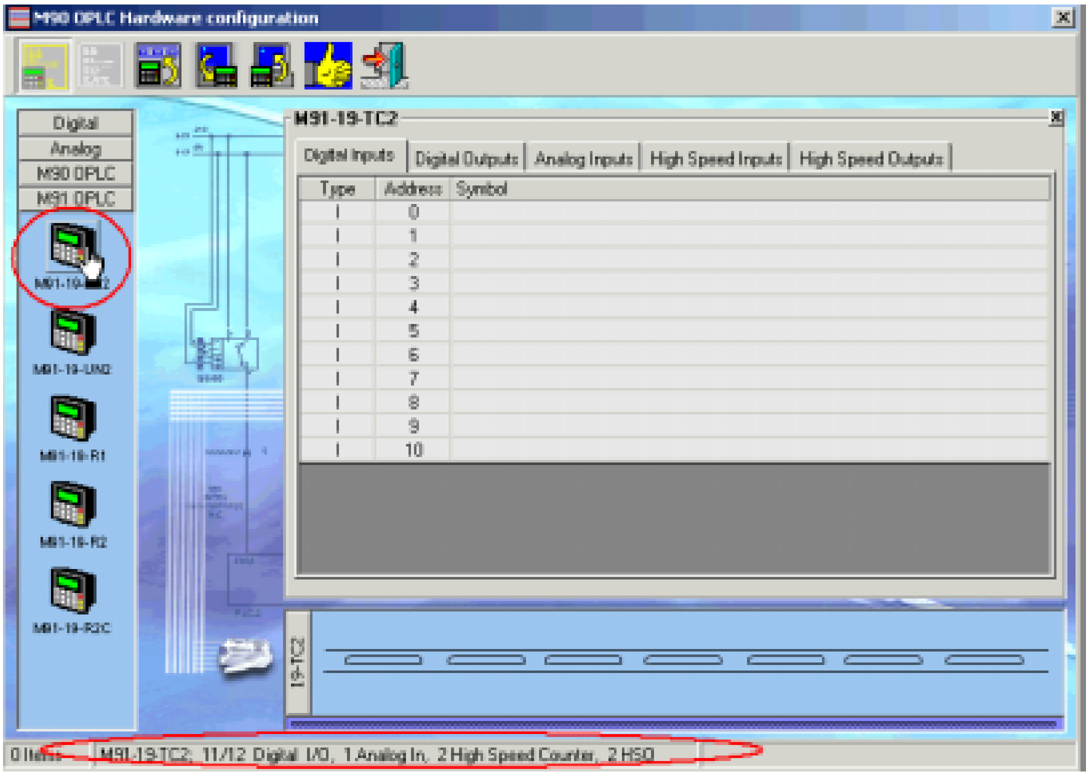
2. The **M90 Hardware Configuration** window opens.

3. Click on the M91 bar.



3. Select the appropriate M91 model; the model's I/O options are displayed.

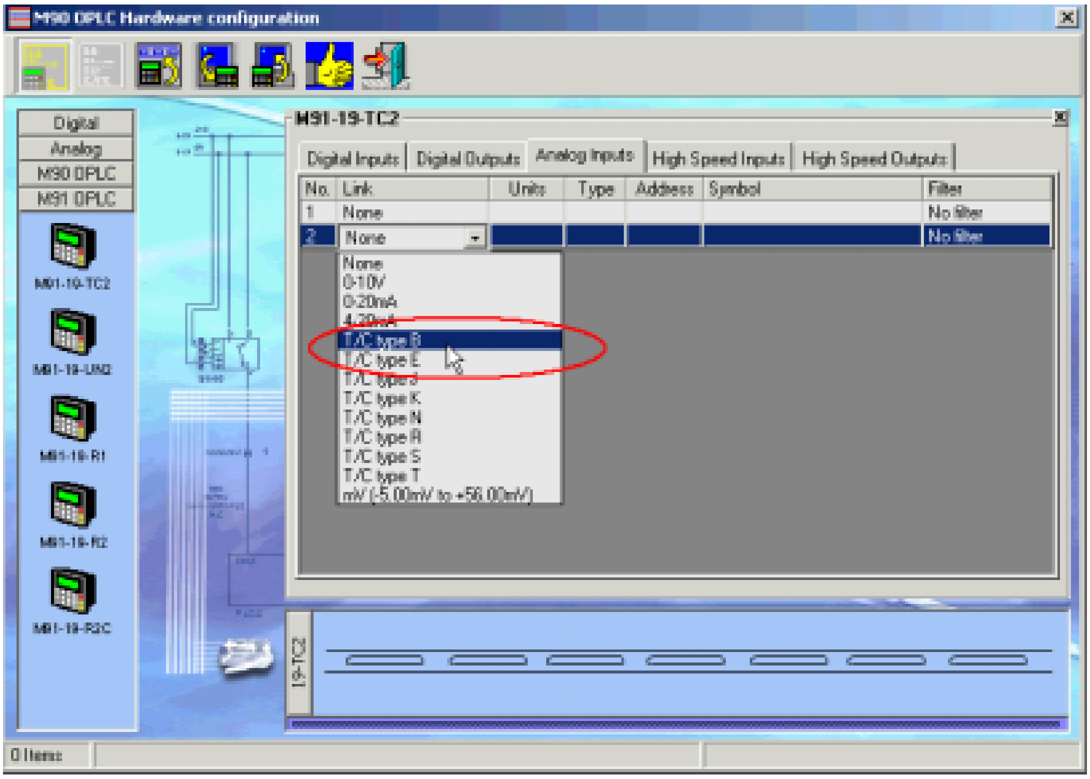
4. Click on the Analog Inputs tab.



Type	Address	Symbol
I	0	
I	1	
I	2	
I	3	
I	4	
I	5	
I	6	
I	7	
I	8	
I	9	
I	10	

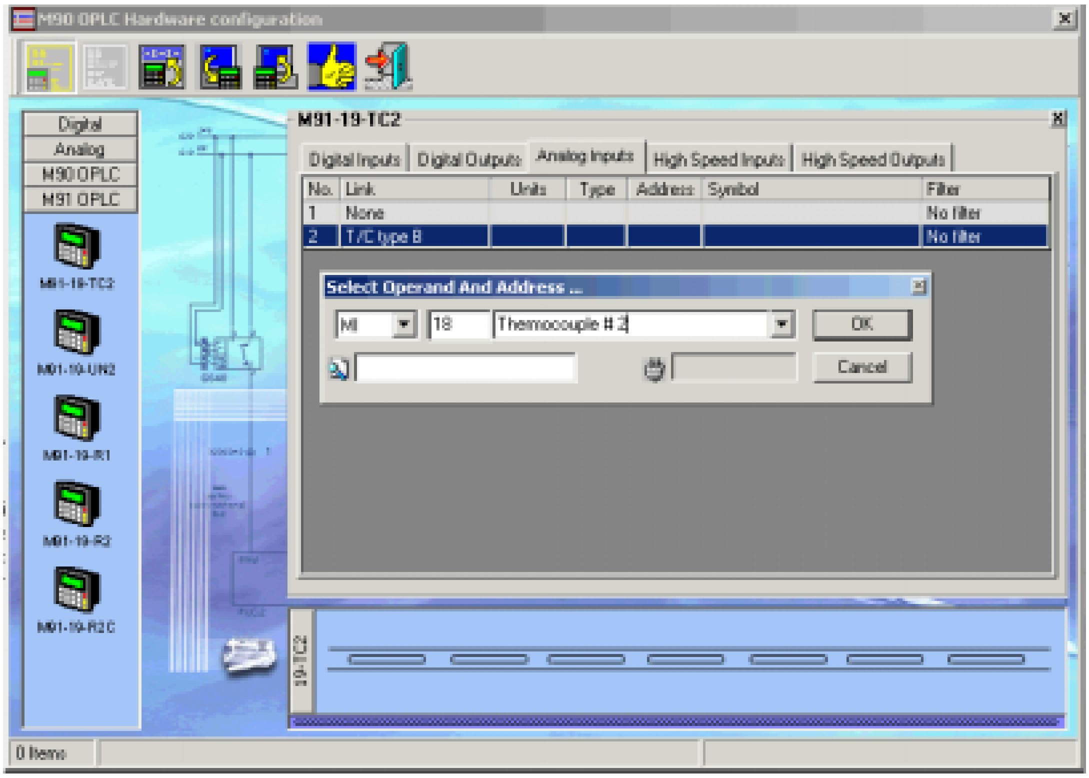
0 Items | M91-19-TC2; 11/12 Digital I/O, 1 Analog In, 2 High Speed Counter, 2 HSO

4. Click the Link field, then select the desired type of input. The **Select MI for Analog Input** window opens.



No.	Link	Units	Type	Address	Symbol	Filter
1	None					No filter
2	None					No filter

6. Enter the desired Address and Symbol of the MI Operand.



M90 OPLC Hardware configuration

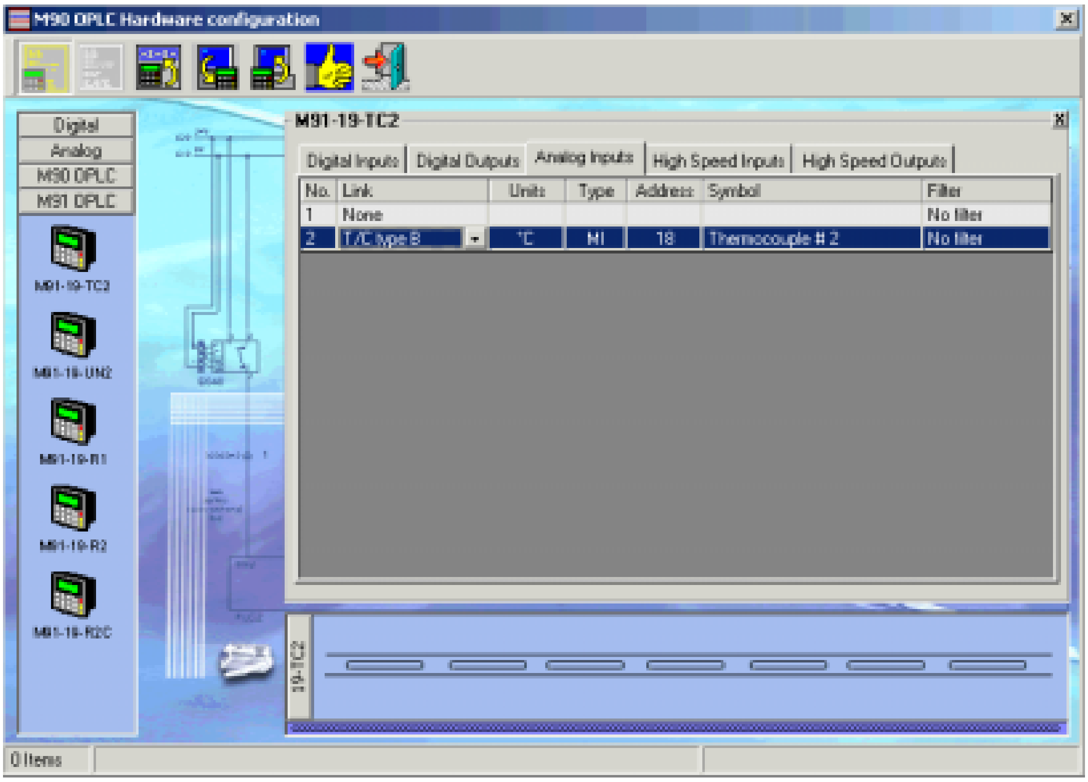
M91-19-TC2

No.	Link	Units	Type	Address	Symbol	Filter
1	None					No filter
2	T/C type B					No filter

Select Operand And Address ...

MI | 18 | Thermocouple # 2

7. The Analog Input is now part of the configuration.



M90 OPLC Hardware configuration

M91-19-TC2

No.	Link	Units	Type	Address	Symbol	Filter
1	None					No filter
2	T/C type B	°C	MI	18	Thermocouple # 2	No filter

Analog I/O Ranges

Note that devices used in conjunction with Unitronics controllers must be calibrated according to the available range. Below, Range refers to the value contained by the register that is linked to the I/O in Hardware Configuration.

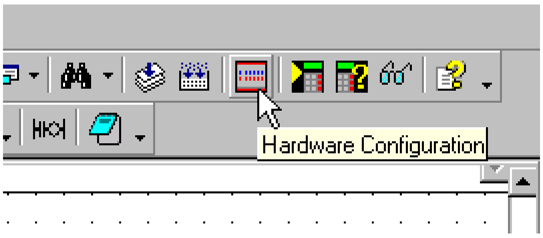
Analog output values are contained in the register that you link to the output in Hardware Configuration.

Model number	Resolution	Range
V200-18-E1 (Snap-in I/O module) V120-12-R1, V120-12-R2C M90 controllers (analog input) M91-19-R1, M91-19-R2, R2C	10 bit (0-10V, 0-20mA, 4-20mA)	0-1023, 1024 units (except at 4-20mA) 204 to 1024, 820 units (at 4-20mA)
V120-12-UN2 M90-19-UN2 M91-19-TC2	14 bit (0-10V, 4-20mA) Temperature ranges appear in the following table	0-16383, 16384 units (except at 4-20mA) 3277-16383, 13107 units (at 4-20mA)
IO-AI4-AO2 *Input*	12 bit (0-10V, 0-20mA, 4-20mA)	0-4095, 4096 units (except at4-20mA) 819 to 4095, 3277 units (at 4-20mA)
Output	12 bit +sign (±10V, 0-20mA, 4-20mA)	0-±4095(except at4-20mA) 819 to 4095, 3277 units (at 4-20mA)

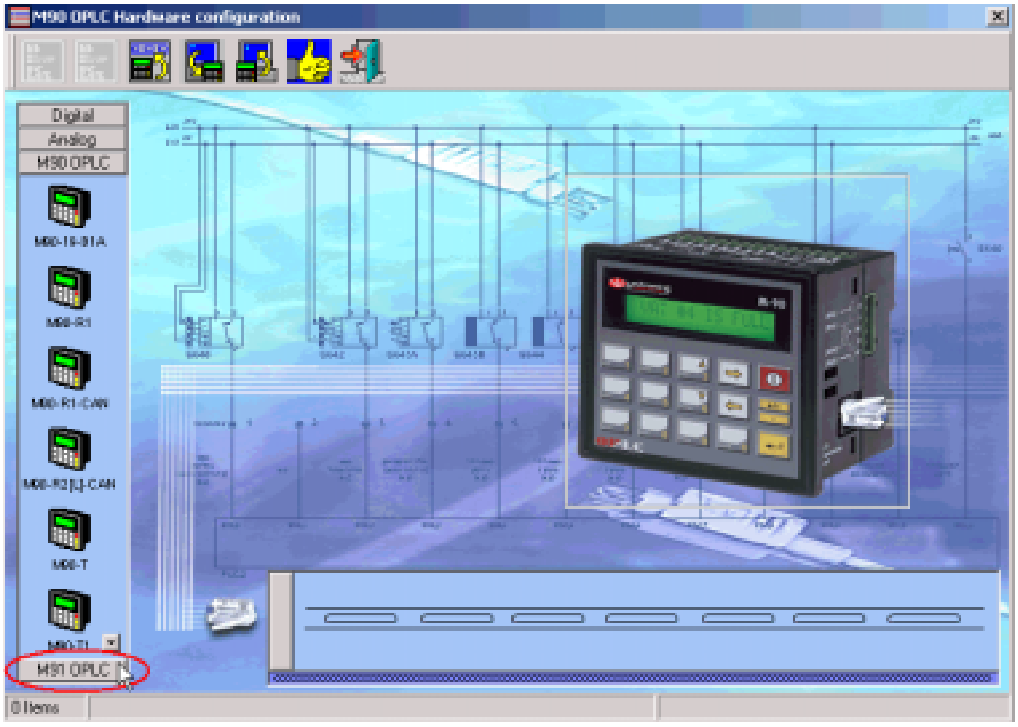
Model number	Type	Input ranges	Range
V120-12-UN2 M90-19-UN2 M91-19-TC2	mV	-5 to 56mV	-50 to 506°C
	B	200 to 1820°C (300 to 3276°F)	2000 to 18200°C (3000 to 32760°F)
	E	-200 to 750°C (-328 to 1382°F)	-2000 to 7500°C (-3280 to 13820°F)
	J	-200 to 760°C (-328 to 1400°F)	-2000 to 7600°C (-3280 to 14000°F)
	K	-200 to 1250°C (-328 to 2282°F)	-2000 to 12500°C (-3280 to 22820°F)
	N	-200 to 1300°C (-328 to 2372°F)	-2000 to 13000°C (-3280 to 23720°F)
	R	-0 to 1768°C (-32 to 3214°F)	-0 to 17680°C (-32 to 32140F)
	S	-0 to 1768°C (-32 to 3214°F)	-0 to 17680°C (-32 to 32140°F)
	T	-200 to 400°C (-328 to 752°F)	-200 to 4000°C (-3280 to 7520°F
IO-PT4		-50° to 460°C	-500° to 4600°C

Configuring a Thermocouple: M91 OPLC series

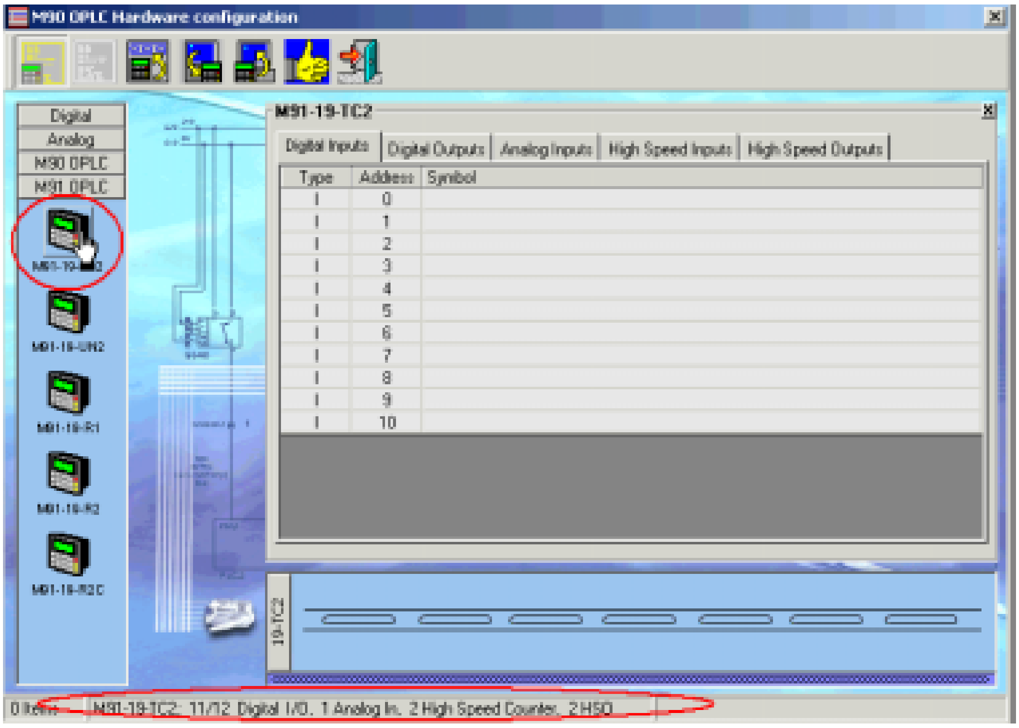
1. Click ***Hardware Configuration*** on the Standard toolbar.



2. The **M90 Hardware Configuration** window opens.
3. Click on the *M91 bar*.



4. Select the appropriate M91 model; the *model's I/O options* are displayed.

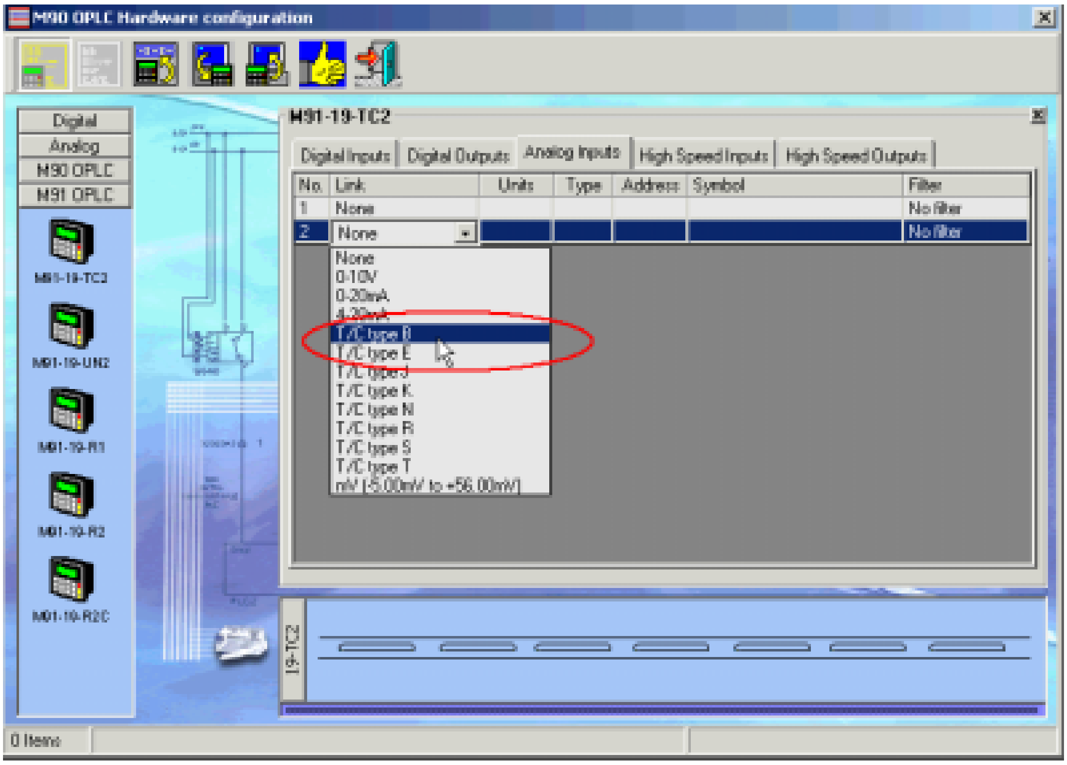


Type	Address	Symbol
I	0	
I	1	
I	2	
I	3	
I	4	
I	5	
I	6	
I	7	
I	8	
I	9	
I	10	

M91-19-TC2: 11/12 Digital I/O, 1 Analog In, 2 High Speed Counter, 2 HSO

5. Click on the Analog Inputs tab.

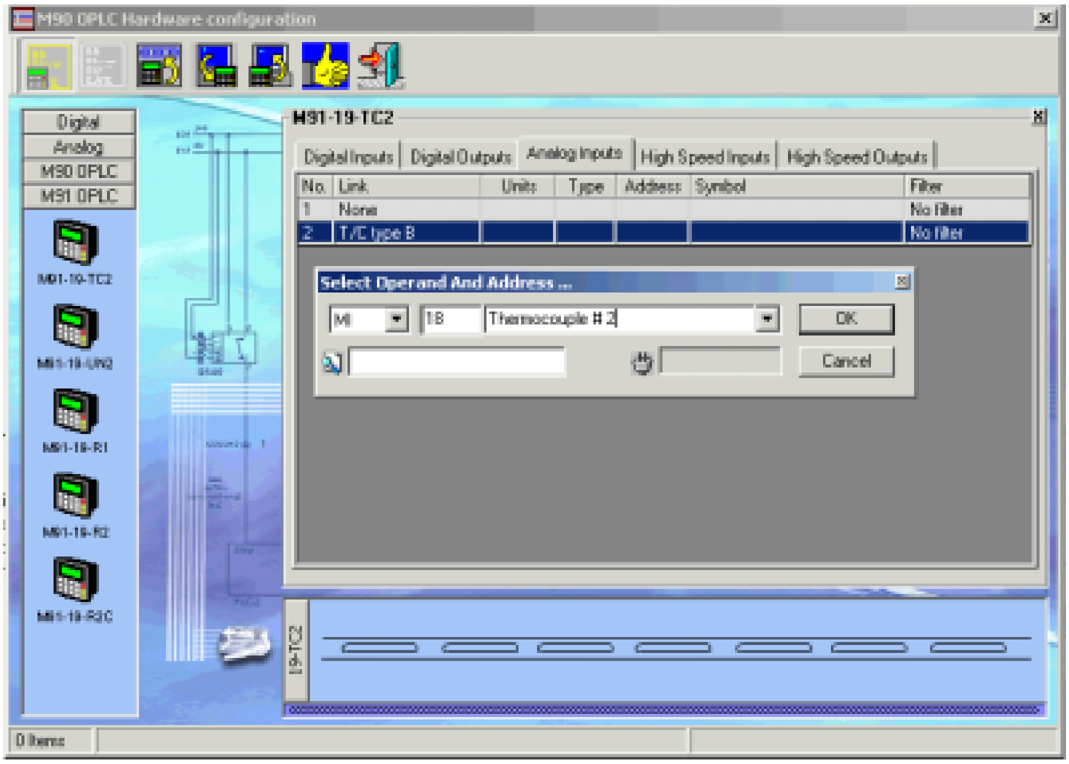
6. Click the Link field, then select the desired type of input. The **Select Operand and Address** box opens.



M91-19-TC2

No.	Link	Units	Type	Address	Symbol	Filter
1	None					No filter
2	None					No filter

7. Enter the desired Address and Symbol of the MI Operand.



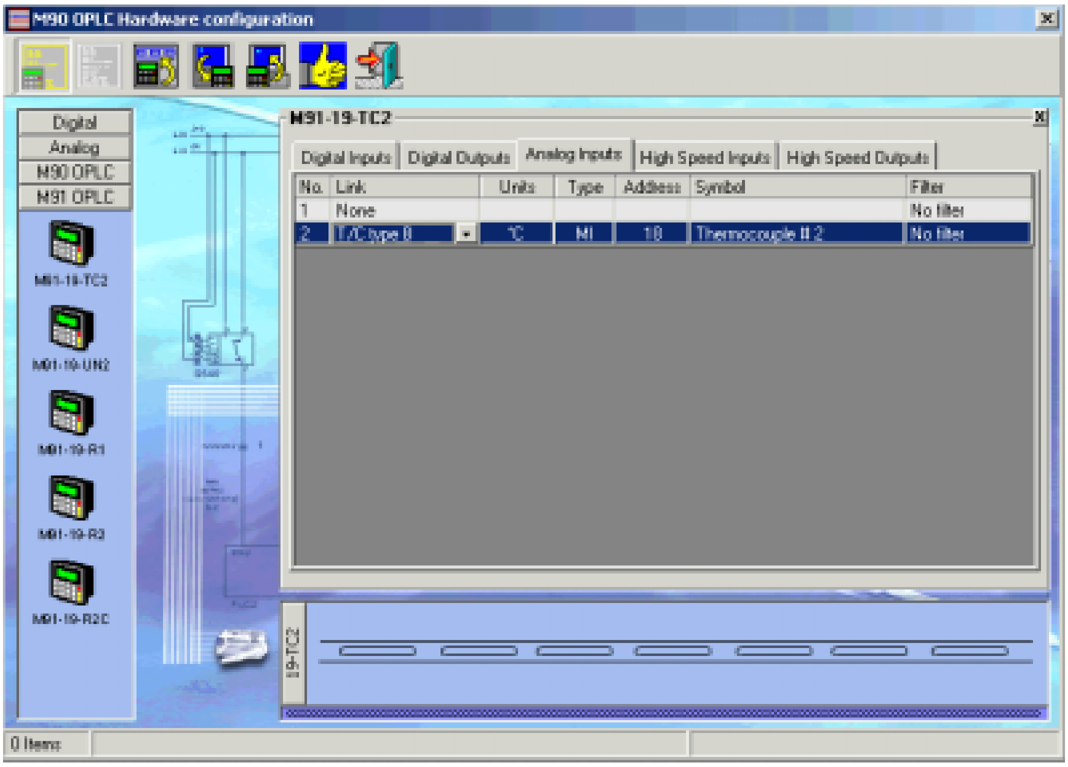
M91-19-TC2

No.	Link	Units	Type	Address	Symbol	Filter
1	None					No filter
2	T/C type B					No filter

Select Operand And Address ...

MI 18 Thermocouple # 2

8. The thermocouple is now part of the configuration.



No.	Link	Units	Type	Address	Symbol	Filter
1	None					No filter
2	T/C type 0	°C	MI	18	Thermocouple # 2	No filter

High-Speed Counters (HSC), Shaft Encoders, Frequency Measurer

The M90 series offers high-speed counter functions of the following types:

- Shaft encoder, at resolutions x2 and x4.
- High-speed counter.
- High-speed counter + reset,
- Frequency measurement, at 100, 500, and 1000 msec.

Some of the sample programs installed together with U90 Ladder include high-speed counters of different types.

HSC Types & Functions

High-speed counter functions are built into the M90 hardware. This is why you do not 'build' a high-speed counter within your Ladder program. Instead, you define it as part of the M90 OPLC's hardware configuration by:

1. Selecting the counter type as shown below
2. Linking it to an MI that contains the counter value.

Note that the counter value is an integer with a range of -32768 to +32767. After the counter reaches the maximum value of +3,2767 it will continue to count in the negative range.

The **last** on-board input on an M90 is the actual counter, and is capable of counting 5,000 pulses per second. Note that the M90 high-speed input is a pnp-type input, requiring a nominal voltage of 24V, a minimum of 15V.

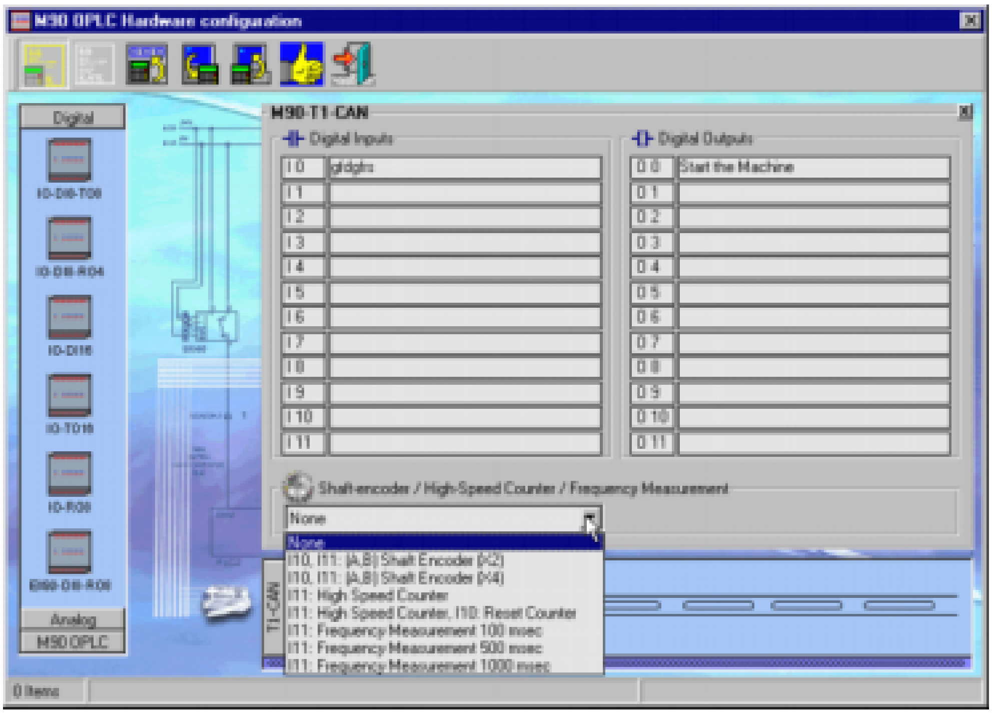
The next-to-last input also serves a purpose in certain high-speed counter functions:

- Shaft encoder function: the next-to-last input serves to indicate the direction of the encoder.
- High-speed counter + reset function: the next-to-last input serves to reset the counter.

When the next-to-last input is used in a high-speed counter function, it is normally OFF. It remains OFF until it receives a signal; the input then turns ON, stopping and resetting the high-speed counter. The high-speed counter begins counting pulses only after the counter reset turns OFF. Note that SB 10 High Speed Counter Reset Enable must be ON; it is ON by default.

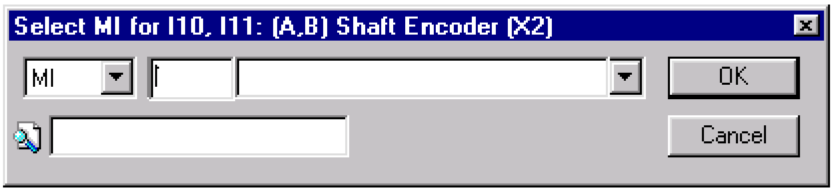
Configuring a High-speed counter

1. Select Hardware Configuration from the Controller menu. The *Hardware Configuration window opens*.


M90 OPLC Hardware configuration
Digital
IO-DI8-TO8
IO-DI8-RO4
IO-DI16
IO-TO16
IO-RO8
EX90-DI8-RO8
Analog
M90 OPLC
M90-T1-CAN
Digital Inputs
Digital Outputs
Shaft-encoder / High-Speed Counter / Frequency Measurement
None
None
I10, I11: (A,B) Shaft Encoder (X2)
I10, I11: (A,B) Shaft Encoder (X4)
I11: High Speed Counter
I11: High Speed Counter, I10: Reset Counter
I11: Frequency Measurement 100 msec
I11: Frequency Measurement 500 msec
I11: Frequency Measurement 1000 msec
T1-CAN
0 Items

I 0	gfdgfrs	O 0	Start the Machine
I 1		O 1	
I 2		O 2	
I 3		O 3	
I 4		O 4	
I 5		O 5	
I 6		O 6	
I 7		O 7	
I 8		O 8	
I 9		O 9	
I 10		O 10	
I 11		O 11	

2. Click on the icon representing your controller model. The appropriate hardware model window opens.
3. Select a high-speed counter type by clicking the drop-down arrow to display the options, then clicking one.
4. The *Select Operand Address box opens*. Select an MI to contain the counter value, and then click OK.



This MI contains the counter value which is current at the last program scan. Use this MI in your program like any other MI. You can reset the counter by placing a 0 value into this MI via the Store function. Note that in order to reset the counter, SB 10 High Speed Counter Reset Enable must be turned ON; SB 10 is ON by default.

Shaft Encoder

Selecting the shaft encoder function enables the counter to count both up (-3, -2, -1, 0, 1, 2, 3, …) and down (3, 2, 1, 0, -1, -2,-3 …). Note that the input requires you to use pnp-type shaft encoders.

High-speed Counter

If you select the high-speed counter function that does not include Reset, note that you must reset it within your Ladder program. This type of counter only counts up.

If you select the high-speed counter function with reset, the counter is capable of counting up within the positive range, 0-32767. This function uses the next-to-last input as a counter reset. Since the reset is done via the hardware, the reset is immediate and independent of the program scan.

Frequency Measurement

This counts the number of pulses over the selected period of time (sample rate): 100 msec, 500 msec, or 1000 msec (1 second), expressing the result in Hertz. For example, 155 pulses counted over 100 msec is equal to 1550Hz; 155 pulses counted over 500 msec is equal to 310Hz.

Compare Functions and Counter Values

It is probable that a counter value will **not** be read at the exact moment that a Compare function in your program is being carried out. This can cause an Equal (=) function to miss the desired counter value; if the counter does not reach the value required by the Equal function at the moment the function is carried out, the Equal function cannot register that the value has been reached. To avoid this problem, use functions Greater Than Or Equal To (≥) and Lesser Than Or Equal To (≤).

High-Speed Output: PWM

M90

M90 OS versions 2.00 (B01) and later enable you to use the last on-board output of M90 models T1 and T1-CAN in either:

- High Speed Output (HSO) mode
- Normal output mode.

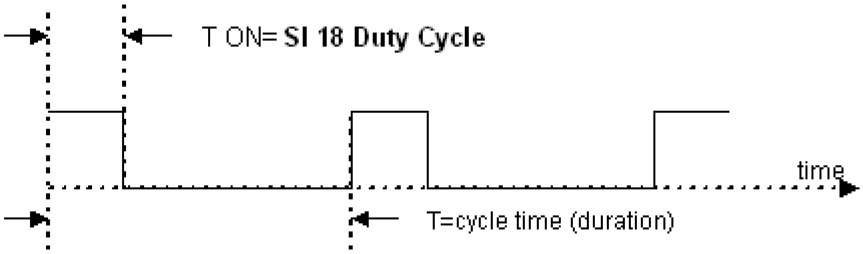
Using HSO mode gives you the ability to use an output as a PWM (Pulse Width Modulation) output. You can also use an output in HSO mode together with stepper motor controllers.

To use HSO mode:

1. Use System Integer SI 16 HSO Mode to change the operating mode of Output 11 from Normal mode to HSO mode: 0=Normal Mode, 1: HSO Mode.
   This should be part of your program's Power-up tasks.
2. Set the output frequency (F) by storing a value into SI 17 HSO Frequency.
   Note that F=1/T, where T is the duration time of a complete cycle.
   You can store a value of 0, or a value from 3-1500Hz; other frequency values are not supported.
3. Set the duty cycle—the ratio of the "on" period of a cycle to the total cycle period—by storing a value into SI 18 Duty Cycle. This value may be from 0-1000, and is expressed as a percentage.

   If, for example, the constant 750 is stored into SI 18, the duty cycle is equal to 75.0%. This means that the pulse will hold a positive state during 75.0% of the total cycle.
4. Use SB 16 HSO RUN to control the output; when SB 16 is ON, Output 11 operates.

In the figure below, SI 18 is equal to 250. This results in the duty cycle being 25% of the total cycle time.

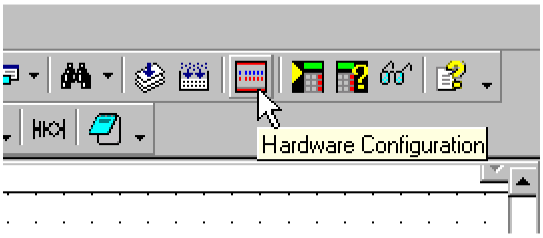


Note that:

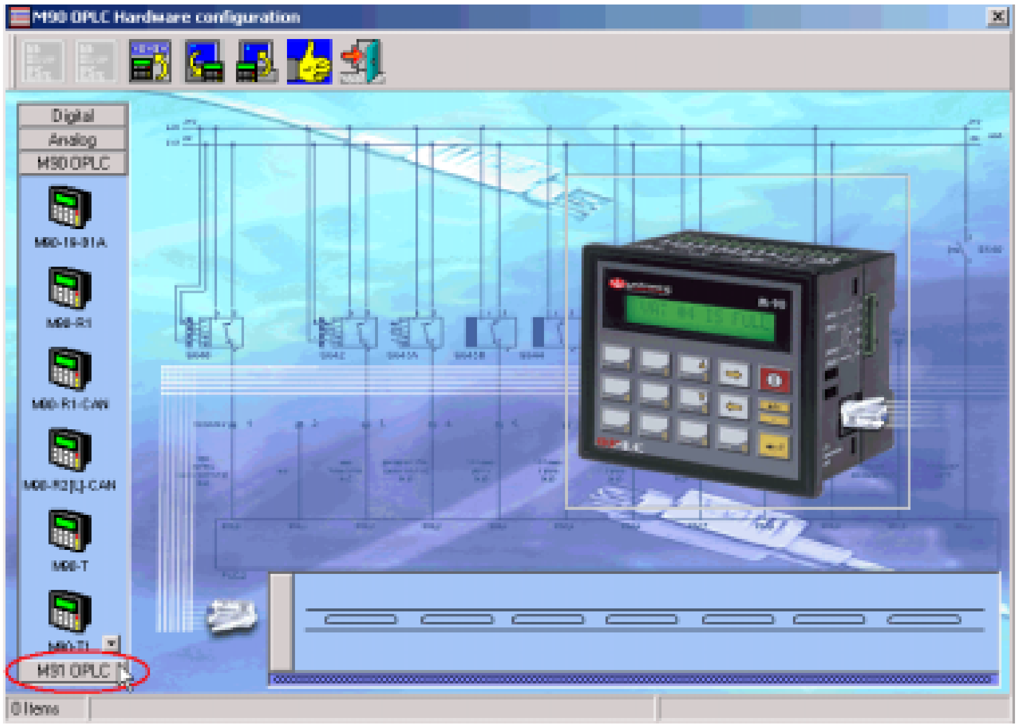
- If you store out-of-range values into SI 17 and SI 18, their values remain unchanged—they retain the last legal values stored.
- Note 2. All parameters except SI 16 may be changed during run-time.

M91

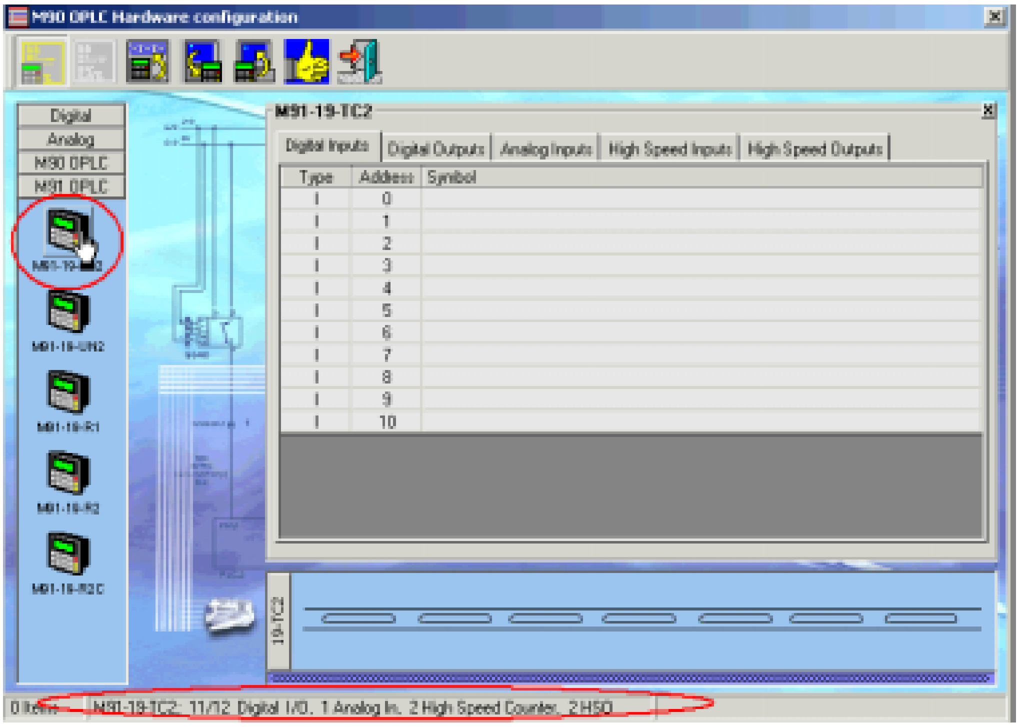
1. Click ***Hardware Configuration*** on the Standard toolbar.



2. The **M90 Hardware Configuration** window opens.
3. Click on the *M91 bar*.



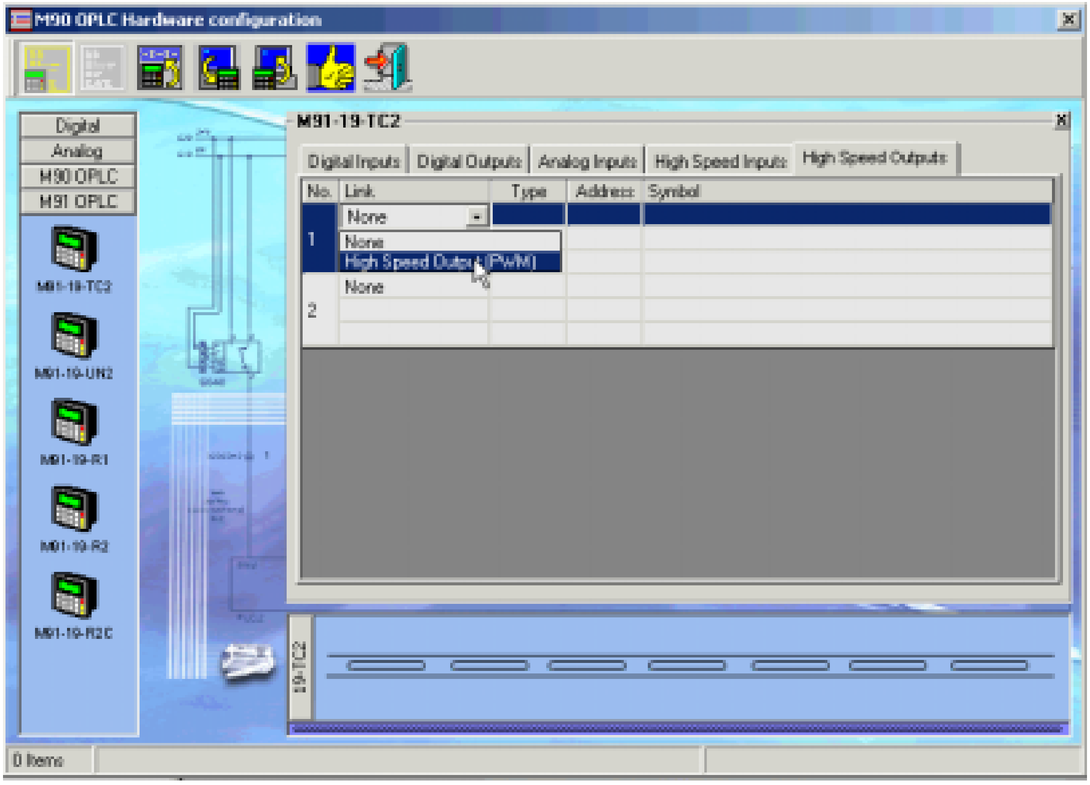
4. Select the appropriate M91 model; *the model's I/O options* are displayed.



Type	Address	Symbol
I	0	
I	1	
I	2	
I	3	
I	4	
I	5	
I	6	
I	7	
I	8	
I	9	
I	10	

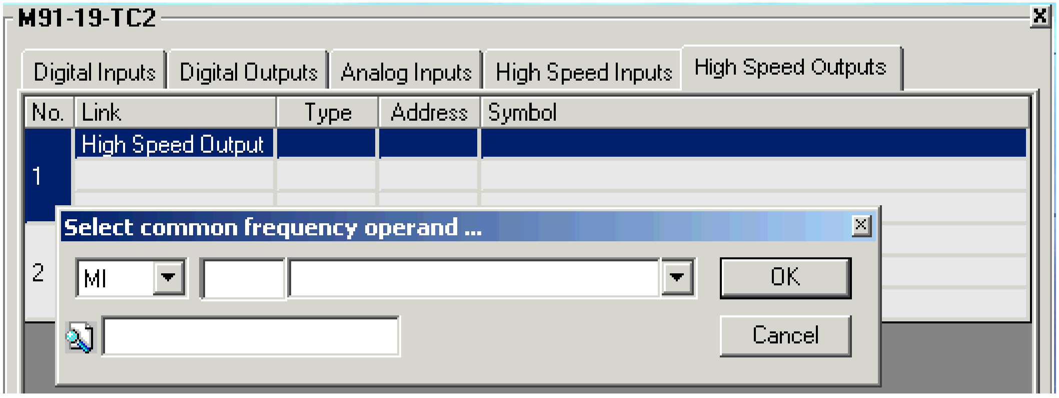
M91-19-TC2: 11/12 Digital I/O, 1 Analog In, 2 High Speed Counter, 2 HSO

3. Click on the High Speed Outputs tab, then select High Speed Output (PWM).



No.	Link	Type	Address	Symbol
1	None			
	None			
2	None			

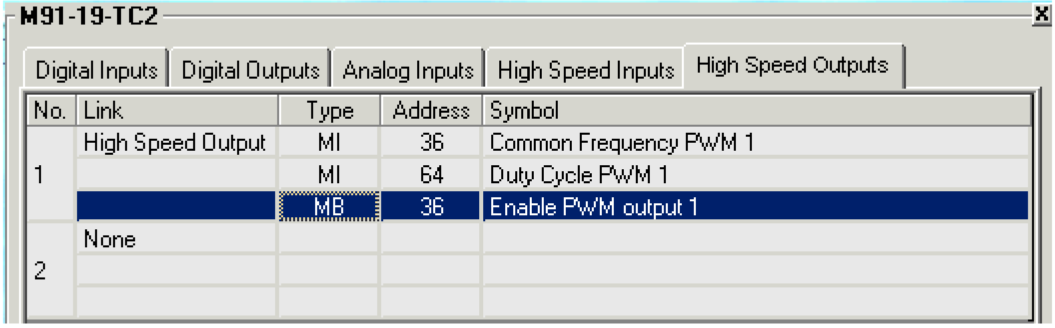
4. The Select Operand and Address box will open 3 times, enabling you to link MIs for Common Frequency & Duty Cycle, and MB for Enable Output.



No.	Link	Type	Address	Symbol
1	High Speed Output			
2				

MI

7. The PWM output is now part of the configuration.



No.	Link	Type	Address	Symbol
1	High Speed Output	MI	36	Common Frequency PWM 1
		MI	64	Duty Cycle PWM 1
		MB	36	Enable PWM output 1
2	None			

Analog Input value--Out Of Range

Expansion modules

If an expansion module's analog input is receiving current or voltage in excess of the absolute maximum rating, the corresponding Out Of Range indicator lights up.

IO-AI4-AO2

Analog value: from 0 to 4095 (12 bit). If the analog input is:

- below 0V/0mA, then the analog value will be 0.
- above 10V/20mA (about 2% above the full scale), then the analog value will be 4096.

IO-ATC8

Analog value: from 0 to 16383 (14 bit). If the analog input is:

- slightly below 0V/0mA (about 0.5% below 0V/0mA), then the analog value will be -1.
- slightly above 10V/20mA (about 0.5% above the full scale), then the analog value will be 16384.
- If the analog input is greatly below or above of the analog input range ,but still within the range of the absolute maximum rating, then the analog value will be 32767.

M90 models

M90-19-B1A, M90-R1, and M90-R2-CAN

Analog value: from 0 to 1023 (10 bit). If the analog input is:

- Below 0V/0mA, the analog value will be 0.
- Above 10V/20mA, the analog value will be 1023.

M91 models

M91-19-R1, M91-19-R2, and M91-19-R2C

Analog value: from 0 to 1023 (10 bit). If the analog input is:

- Below 0V/0mA, then the analog value will be 0.
- Above 10V/20mA (about 2% above the full scale), then the analog value will be 1024.

M91-19-TC2, M91-19-UN2, and M91-19-4UA2
Analog value: from 0 to 16383 (14 bit). If the analog input is:

- Slightly below 0V/0mA (about 0.5% below 0V/0mA), then the analog value will be -1.
- Slightly above 10V/20mA (about 0.5% above the full scale), then the analog value will be 16384.
- Greatly below or above of the analog input range, but still in the range of the absolute maximum rating, then the analog value will be 32767.

Note that the absolute maximum rating of the analog inputs for all the units is +/- 15V.

HMI

Display

What is an HMI?

HMI stands for Human Machine Interface. This is the interface between the operator and the controller.

The M90 HMI is the controller operating panel. The panel comprises a 15 key numeric keypad and a 16 character LCD Display screen.

The keypad is used to input data into the application, such as Timer values.

The M90's Display screen can show operator messages, variable information from the program and system information.

HMI messages are created in the Display Editor.

Variable information fields are created in the Variable Editor.

HMI applications are featured in several sample applications, such as the applications ' Display Jumps from Ladder', 'Names from List Var', 'Password', 'Special characters on List', 'Display of Events', and '5 Vars on Display'. These applications may be found by selecting Sample U90 Projects from the Help Menu.

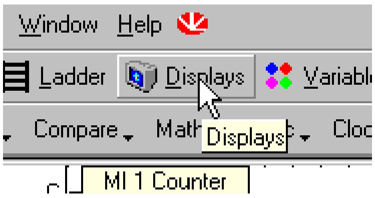
What are Displays?

Displays are shown on the controller's LCD screen according to the program conditions you set in your HMI program. Use the Display editor to create the HMI text, define the variable fields & parameters and assign jump conditions.

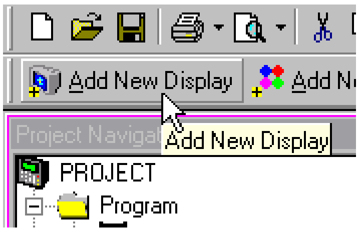
Creating and Naming a Display

To create a Display:

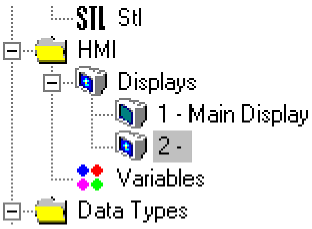
1. Click the Display icon on the Standard toolbar. The Display Editor will open.



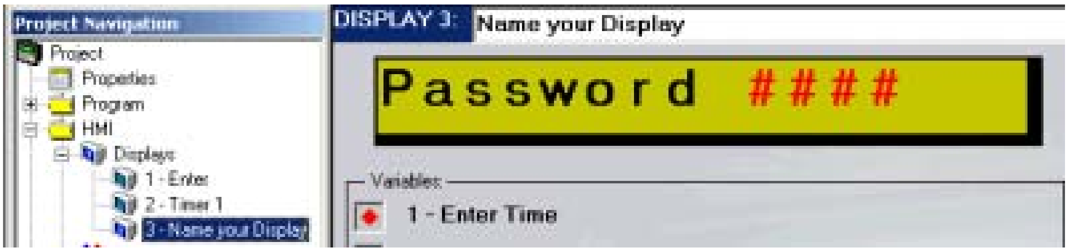
2. Click the Add New Display icon in the HMI toolbar.



3. A new Display is created.



4. Place the cursor in the name field.
5. Type in a name. Press enter.



DISPLAY 3: Name your Display

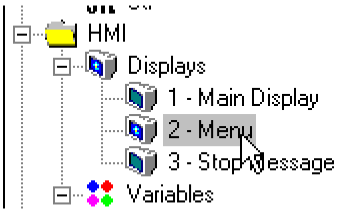
Password ####

The Display name also appears with the Display number in the Project Navigation Tree.

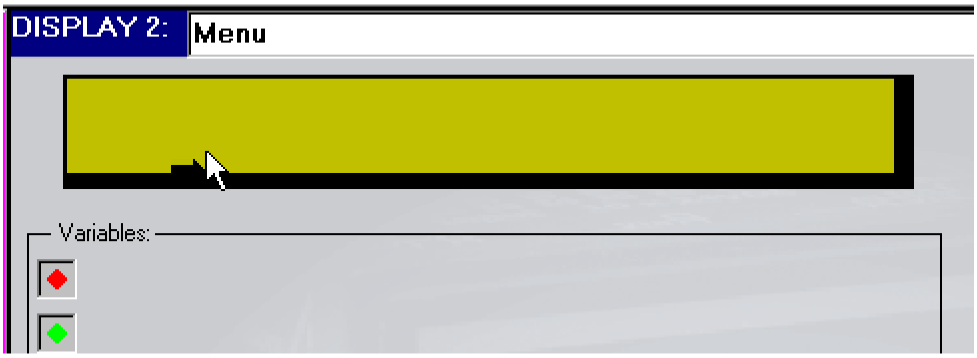
Creating a fixed text Display

To create a fixed text Display:

1. Select the desired Display from the Navigator Window.

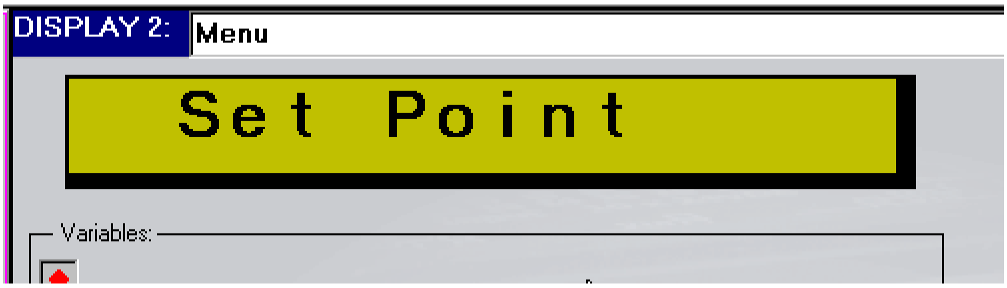


2. The Display opens in the Display Editor.



DISPLAY 2: Menu

3. Type in the fixed text to be displayed.



DISPLAY 2: Menu

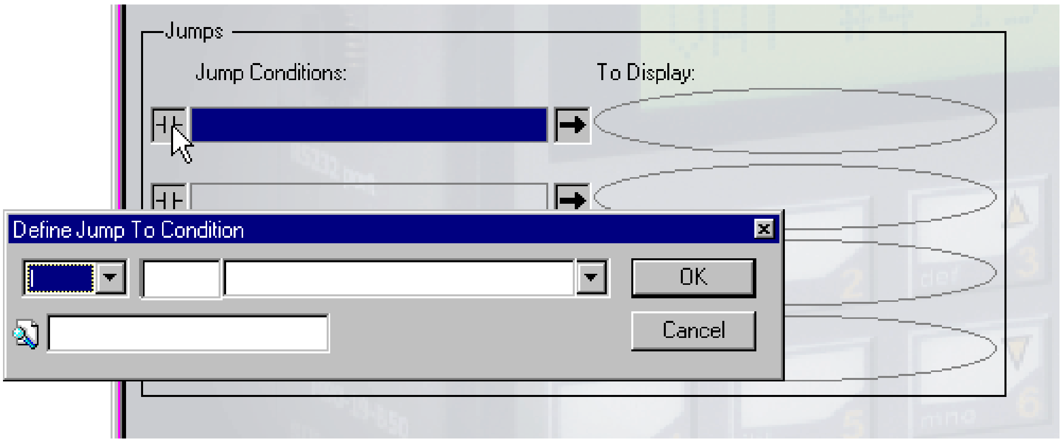
Set Point

Jump to Display: scrolling between Displays

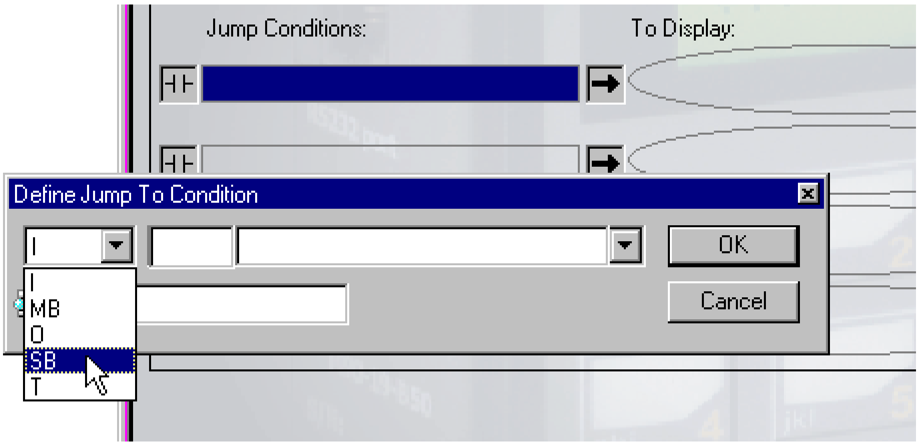
Display Jumps allow you to move between Displays via the M90 keypad or any bit positive transition. You can create up to 4 Jumps for each Display in the Display Editor. If you want to create more than 4 Jumps for a Display, you must create the logic conditions in the Ladder Editor.

To create a jump:

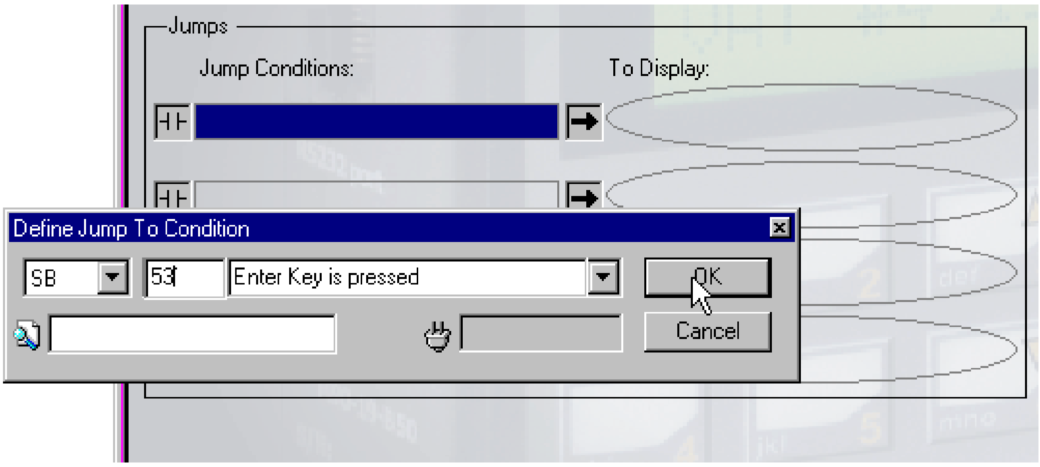
1. Click on a Jump Condition and the Define Jump to Condition dialog box opens.



2. Select a Jump Operand from the drop-down menu.

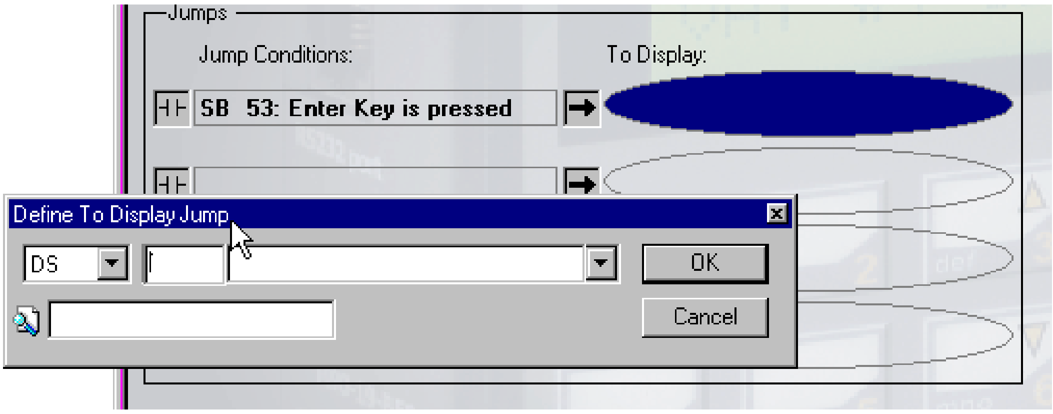


3. Enter the desired Address and symbol for the Jump Operand. Click OK.

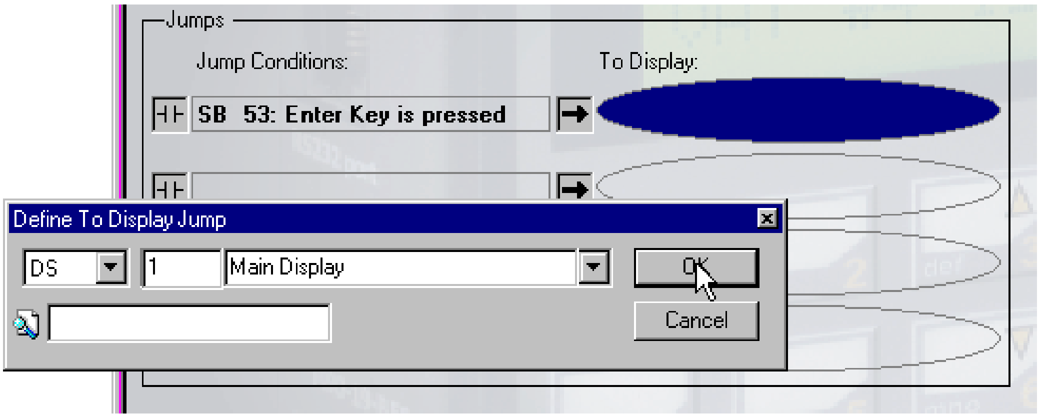


SB 53 Enter Key is pressed

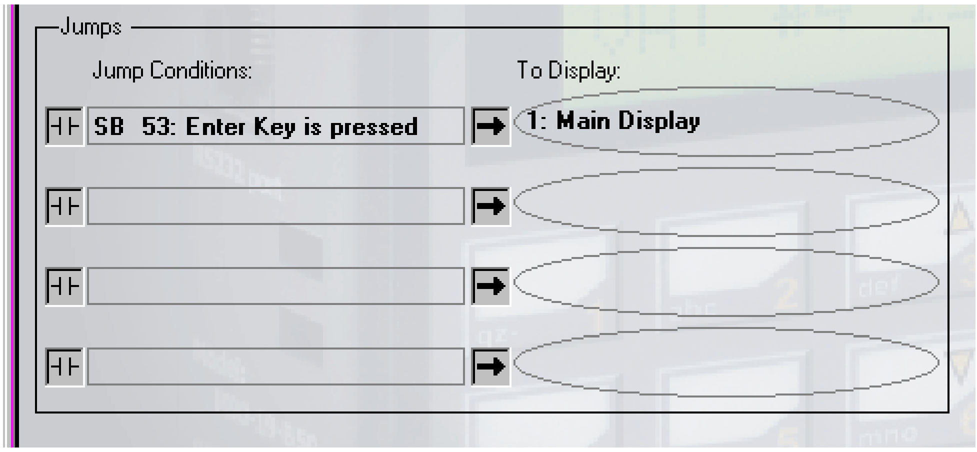
4. The Define To Display Jump dialog box opens.



5. Enter the Display number to which you want to jump. Click OK.



6. The result will be:



Note that Display Jump conditions based on MBs can **only** be linked to MB 0-127; jumps may not be linked to MB 128 -255.

Note ♦ When an HMI keypad entry variable is active, and the Enter key is pressed on the controller keypad, SB 30 HMI Keypad Entries Complete turns ON. This can be used as a Jump condition.

In addition , note that a Display may contain a total of 4 variables. Each one has an SB:

- SB 31 HMI Var 1 Keypad entry completed
- SB 32 HMI Var 2 Keypad entry completed
- SB 33 HMI Var 3 Keypad entry completed

- SB 34 HMI Var 4 Keypad entry completed

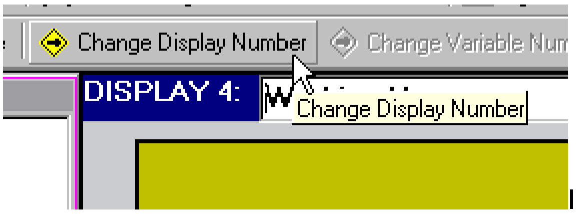
The condition of these SBs may be used as Jump Conditions, or to drive calculations in your program.

Changing a Display number

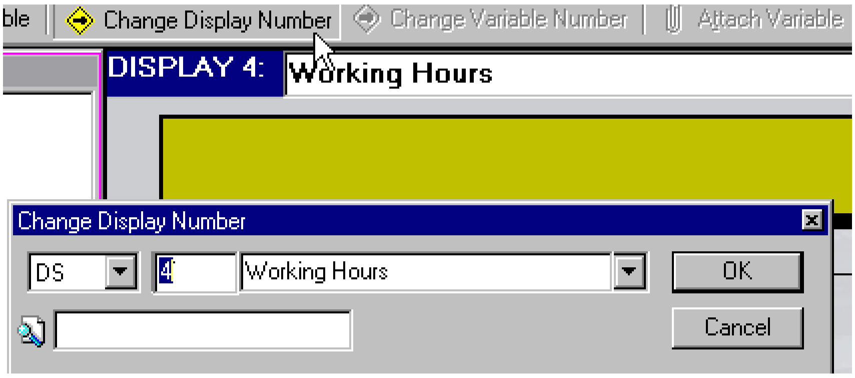
To change a Display number:

In the Display Editor:

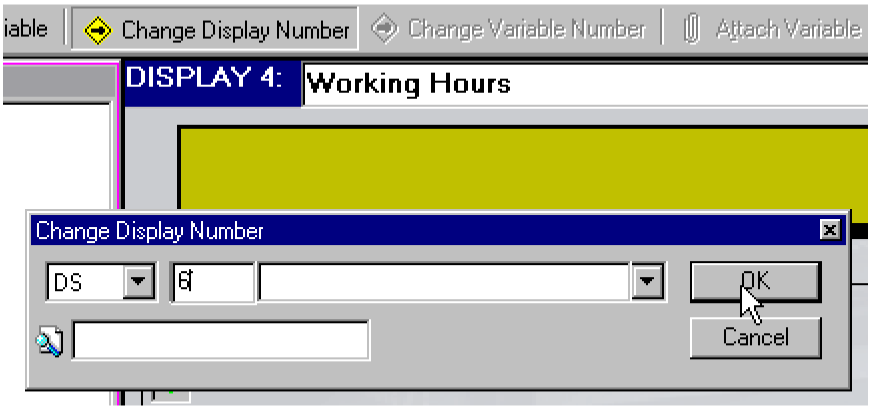
1. On the HMI toolbar, click the Change Display icon.



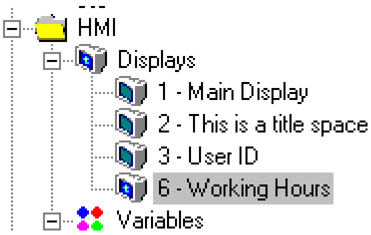
2. The Change Display Number dialog box opens.



3. Enter the new Display number in the Address field. Click OK.



4. The Display number changes. The Display title is unchanged.

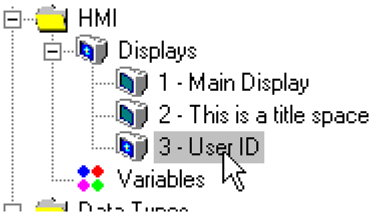


Deleting a Display

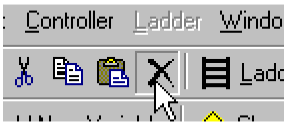
To delete a Display:

In the Display Editor:

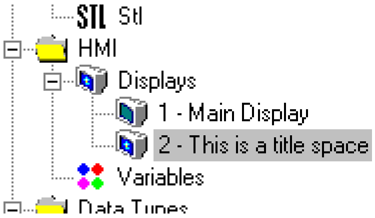
1. In the Navigation Window, click on the Display number you want to delete. The Display will open in the Display Editor.



2. On the Standard toolbar, click **Delete**.



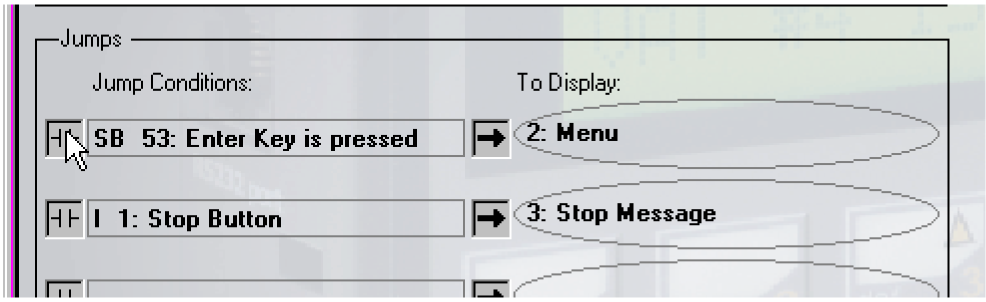
3. The Display is deleted. You see that the Display number disappears from the Navigation Window.



Changing a Jump condition

To change a Jump condition:

1. Click on the Jump Condition in the desired Display.

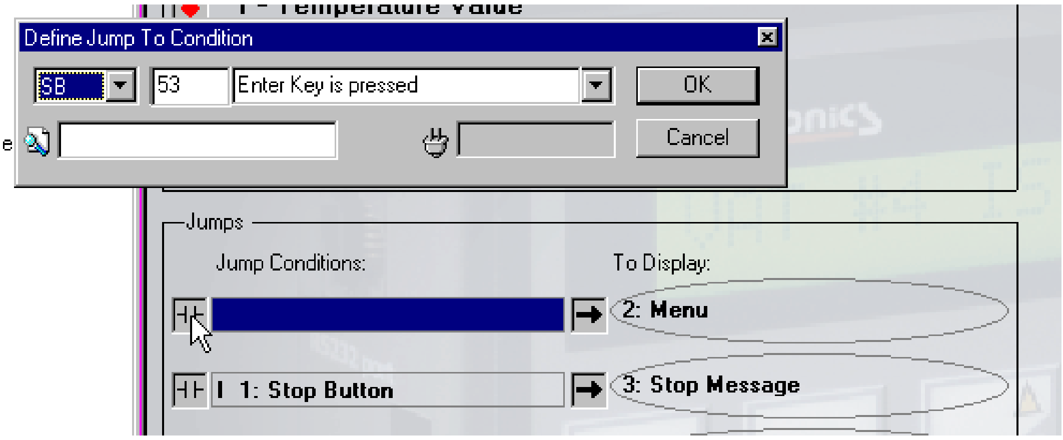


Jump Conditions: To Display:

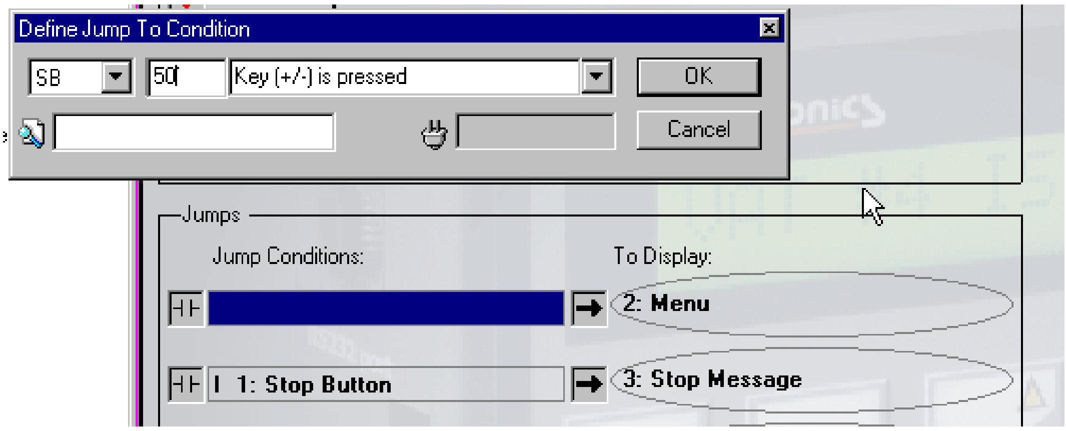
SB 53: Enter Key is pressed → 2: Menu

I 1: Stop Button → 3: Stop Message

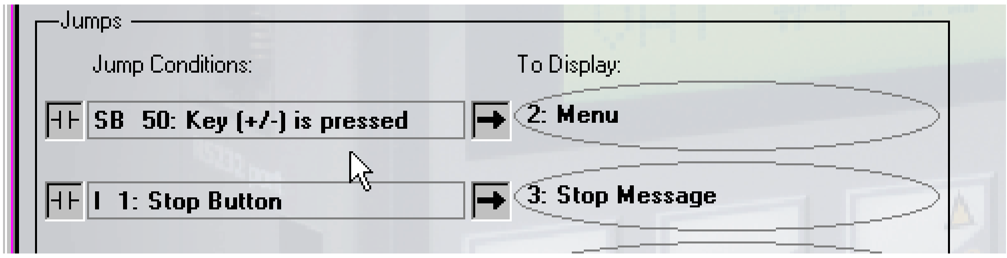
2. The Define Jump to Condition dialog box opens.



3. Make the appropriate changes.



4. The new Jump Condition now appears in the Display Editor.

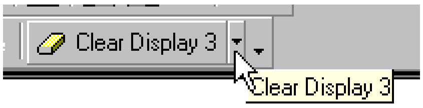


Clearing a Display

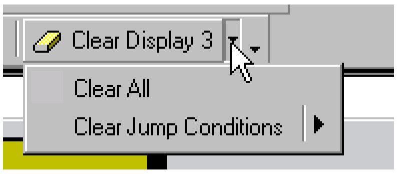
To clear the contents of a Display:

In the Display Editor:

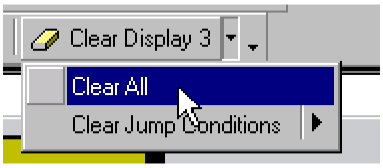
1. On the HMI toolbar, click the Clear Display icon.



2. Open the Clear Display menu. You can clear all Display parameters - or - only Jump conditions



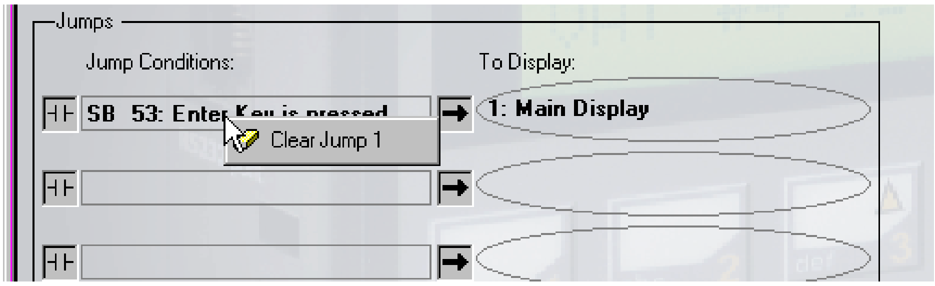
3. Select the parameters you wish to clear.



Clearing Jump conditions

To clear an existing Jump condition:

1. Right click on the Jump.
2. The Clear Jump icon appears.



3. Click the icon to clear the Jump.

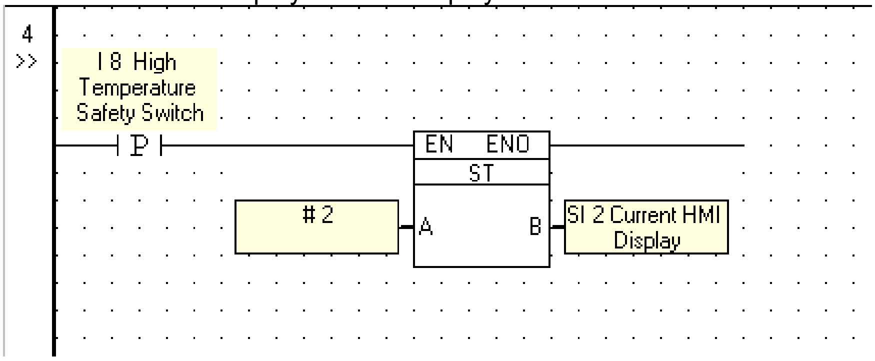
Creating more than four Jumps for a Display

You can create up to 4 Jumps for each Display in the Display Editor. If you want to create more than 4 Jumps for a Display, you must create the logic conditions in the Ladder Editor.

SI 2 contains the Current HMI Display number. You can jump to a specific Display by writing the Display number into SI 2.

Example:

- Writing #5 into SI 2 will cause Display #5 to be displayed on the controller.
- Writing #8 into SI 2 will cause Display #8 to be displayed on the controller.

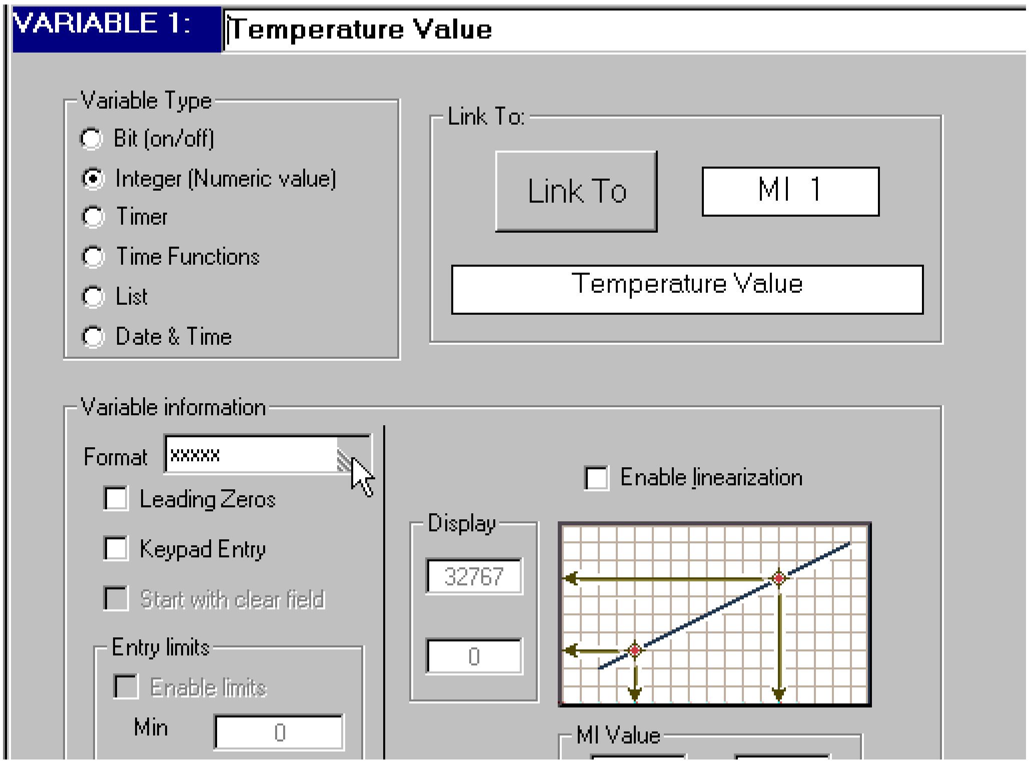


Take care to create the Displays **as well as** the logic conditions.

Display formats for MI and SI values

To set the M90 Display format for a MI or SI value:

1. Open the Format menu in the Variable information box in the Variable Editor.



VARIABLE 1: Temperature Value

Variable Type
- ○ Bit (on/off)
- ● Integer (Numeric value)
- ○ Timer
- ○ Time Functions
- ○ List
- ○ Date & Time

Link To:
Link To | MI 1
Temperature Value

Variable information
Format: xxxxx
- □ Leading Zeros
- □ Keypad Entry
- □ Start with clear field

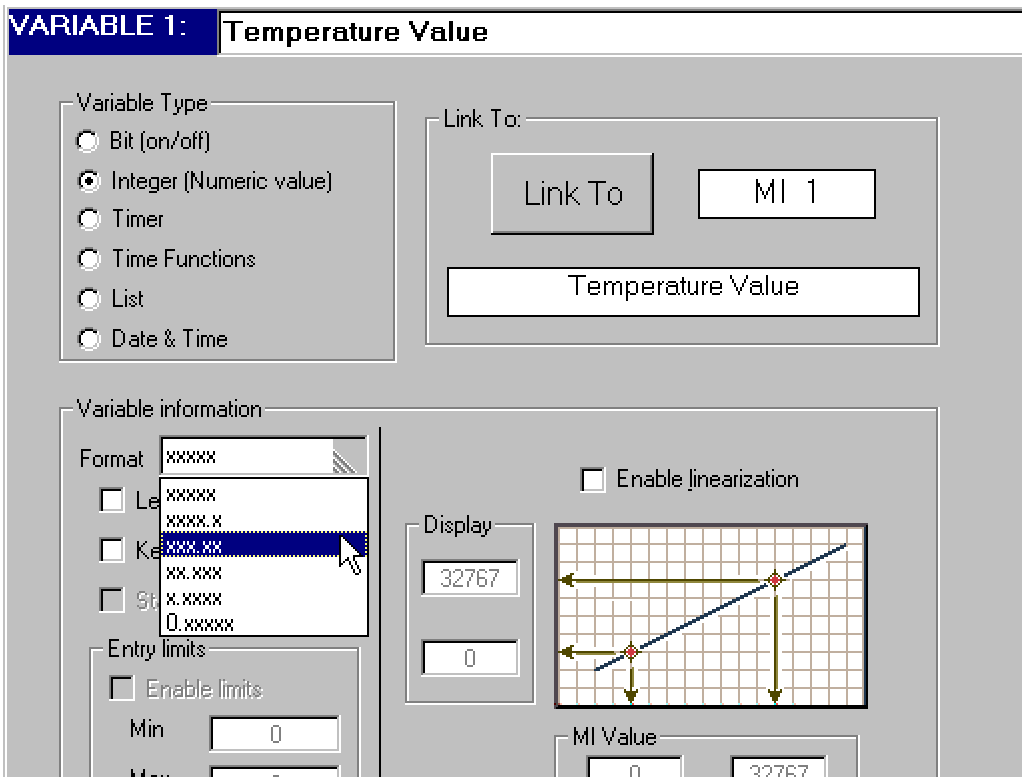
Entry limits
- □ Enable limits
- Min: 0

□ Enable linearization

Display: 32767, 0

MI Value

2. Select a Variable Format.



VARIABLE 1: Temperature Value

Variable Type
- ○ Bit (on/off)
- ● Integer (Numeric value)
- ○ Timer
- ○ Time Functions
- ○ List
- ○ Date & Time

Link To:
Link To | MI 1
Temperature Value

Variable information
Format: xxxxx

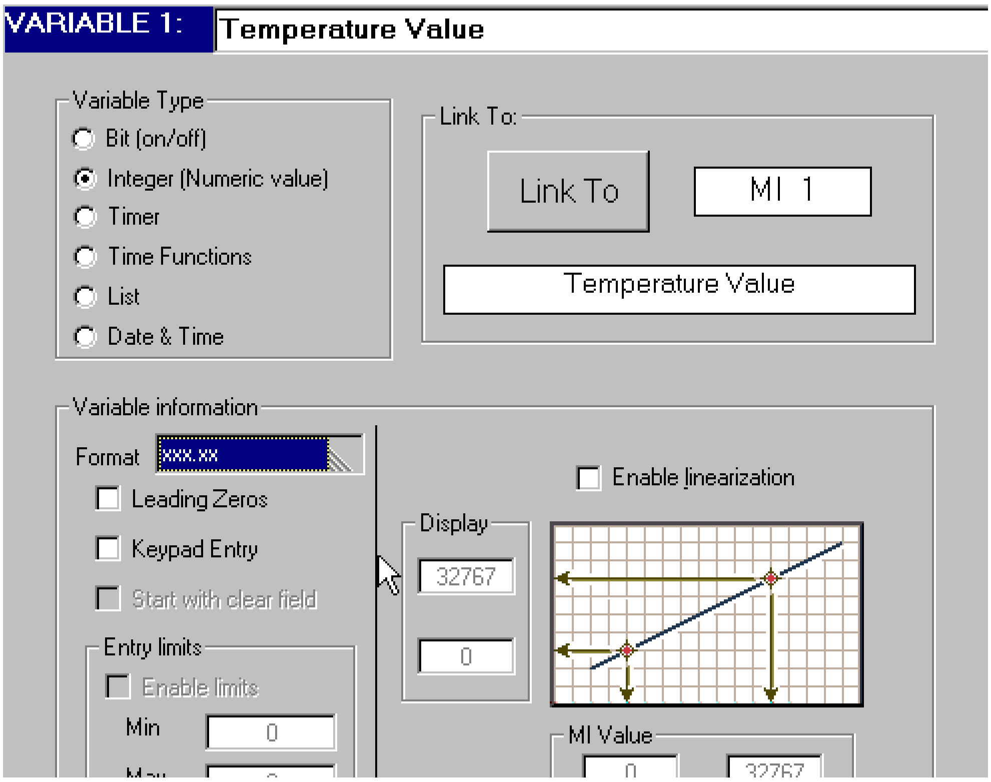
Entry limits
- □ Enable limits
- Min: 0

□ Enable linearization

Display: 32767, 0

MI Value: 0, 32767

3. The selected format appears in the Format window.



VARIABLE 1: Temperature Value

Variable Type

- ○ Bit (on/off)
- ● Integer (Numeric value)
- ○ Timer
- ○ Time Functions
- ○ List
- ○ Date & Time

Link To:

Link To | MI 1

Temperature Value

Variable information

Format xxx.xx

- □ Leading Zeros
- □ Keypad Entry
- □ Start with clear field

Entry limits

- □ Enable limits

Min 0

Max 0

□ Enable linearization

Display

32767

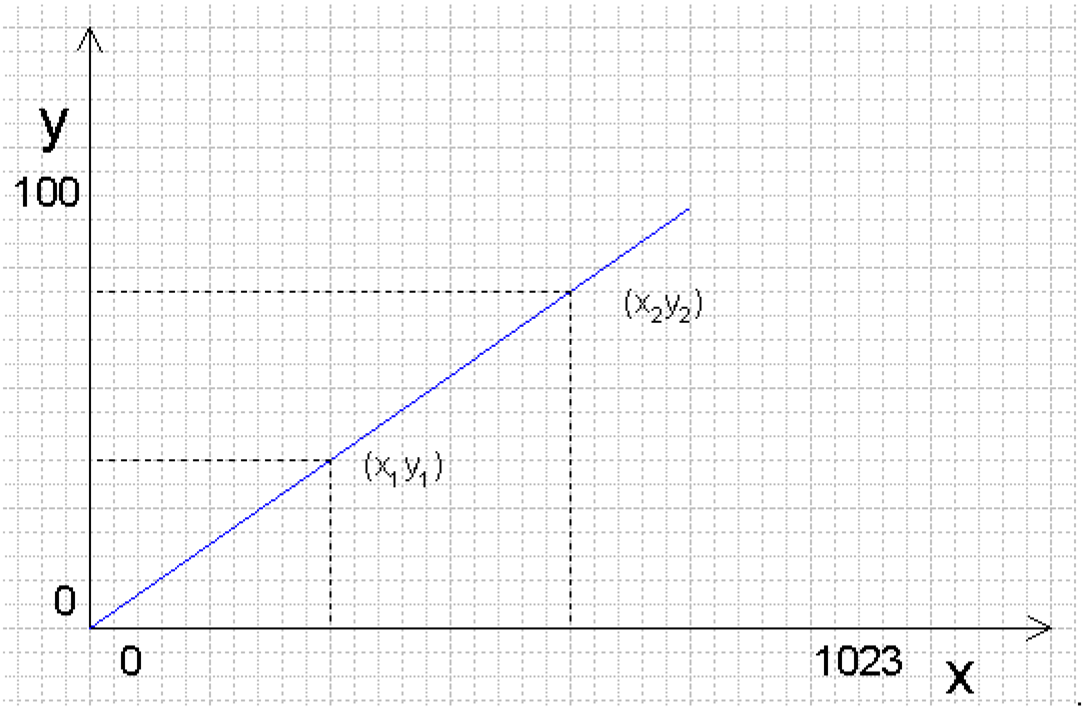
0

MI Value

0 32767

Linearization

Linearization can be used to convert analog values from I/Os into decimal or other integer values. An analog value from a temperature probe, for example can be converted to degrees Celsius and displayed on the controller's display screen.



Linearize values for Display

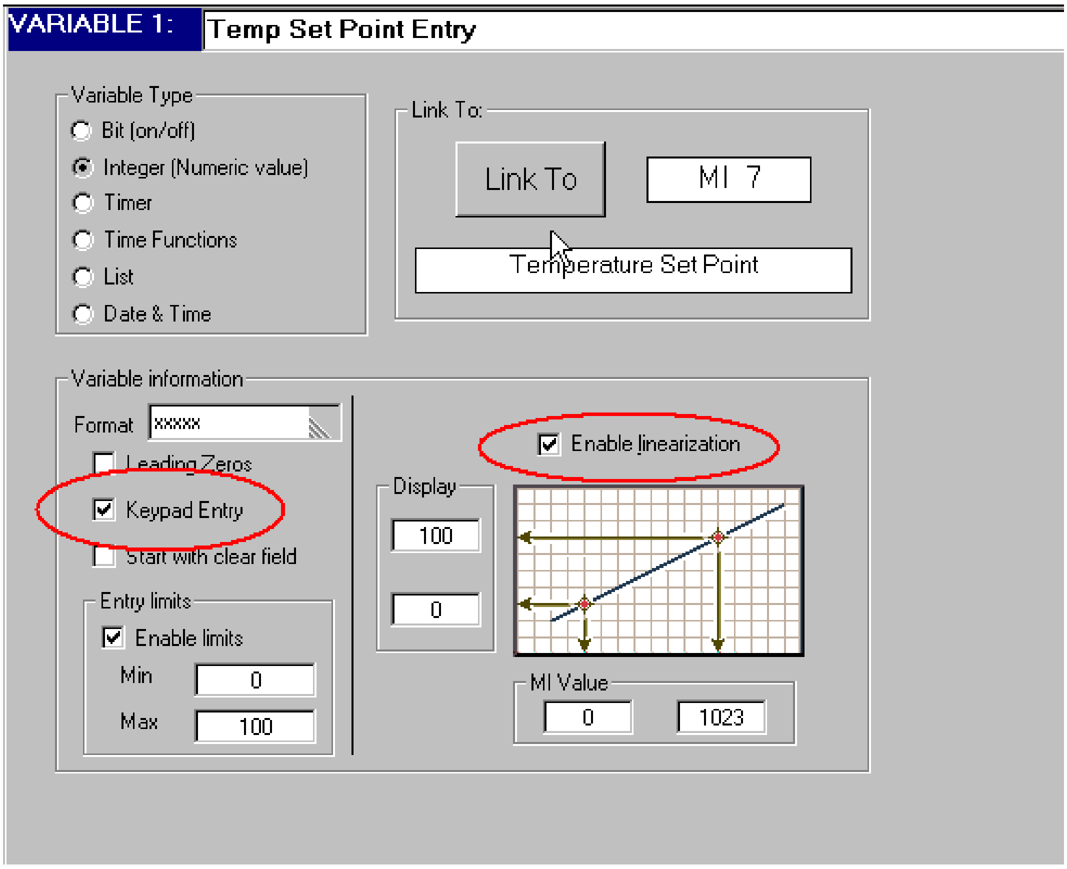
Note that the linearized value created in this way may be displayed-- **but** the value **cannot** be used anywhere else within the project for further calculations or operations.

You can enter an Analog value, such as temperature, via the M90 keypad, then convert that value into a Digital value for comparison with a digital value from a temperature probe by selecting **Enable Linearization** in the linked Variable.

This conversion process is Reverse Linearization.

To enable Analog to Digital conversion:

1. Create a Display for entering the analog value.
2. Create an Integer Variable.
3. Select **keypad entry** and **enable linearization**.
4. Enter the linearization values for the x and y axes.



VARIABLE 1: Temp Set Point Entry

Variable Type
○ Bit (on/off)
● Integer (Numeric value)
○ Timer
○ Time Functions
○ List
○ Date & Time

Link To:
Link To
MI 7
Temperature Set Point

Variable information
Format xxxxx
□ Leading Zeros
☑ Keypad Entry
□ Start with clear field

Entry limits
☑ Enable limits
Min 0
Max 100

☑ Enable linearization

Display
100
0

MI Value
0
1023

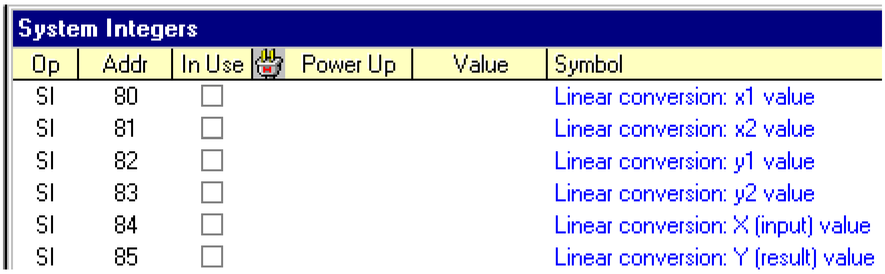
According to the above example:

- A temperature entry of 100^0 C will be converted to 1023 Digital value.
- A temperature entry of 50^0 C will be converted to 512 Digital value.

Linearize values in the Ladder

You can also linearize values in your Ladder and display them on the M90's LCD.

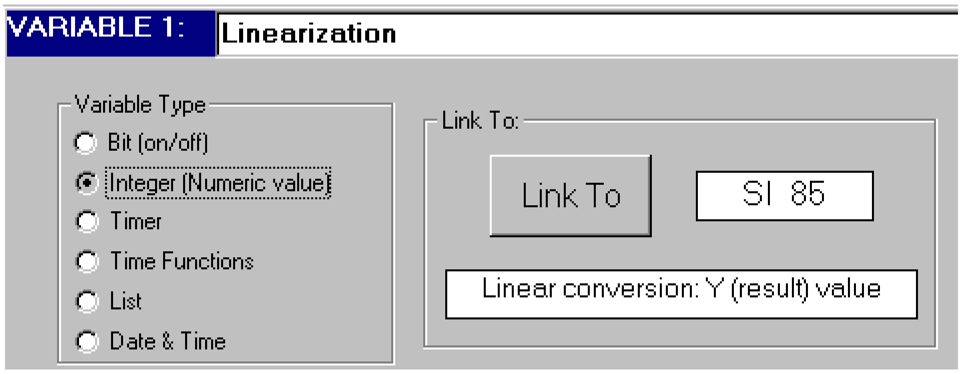
1. In your Ladder project, use SI 80 - 85 to set the (x,y) variable ranges. Use SB 80 to activate the **Linearization** function.



System Integers

Op	Addr	In Use	Power Up	Value	Symbol
SI	80	□			Linear conversion: x1 value
SI	81	□			Linear conversion: x2 value
SI	82	□			Linear conversion: y1 value
SI	83	□			Linear conversion: y2 value
SI	84	□			Linear conversion: X (input) value
SI	85	□			Linear conversion: Y (result) value

The linearization values created here can be displayed by linking SI 85 to a Display;the value **can** be used elsewhere within the project for further calculations or operations.



VARIABLE 1: Linearization

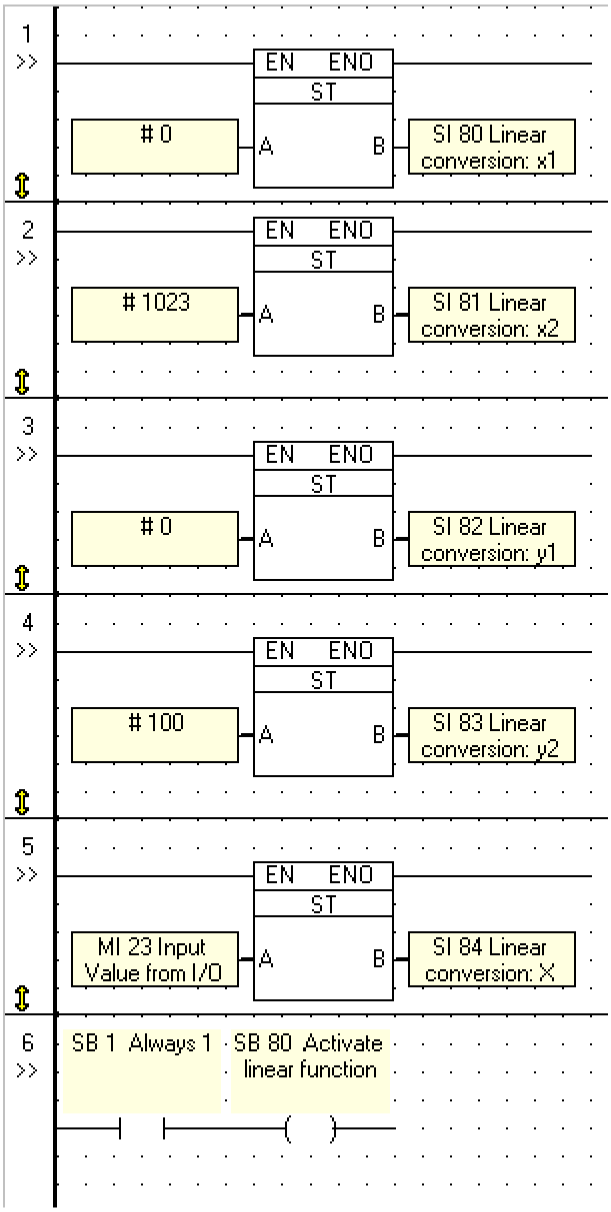
Variable Type

○ Bit (on/off)
● Integer (Numeric value)
○ Timer
○ Time Functions
○ List
○ Date & Time

Link To: SI 85

Linear conversion: Y (result) value

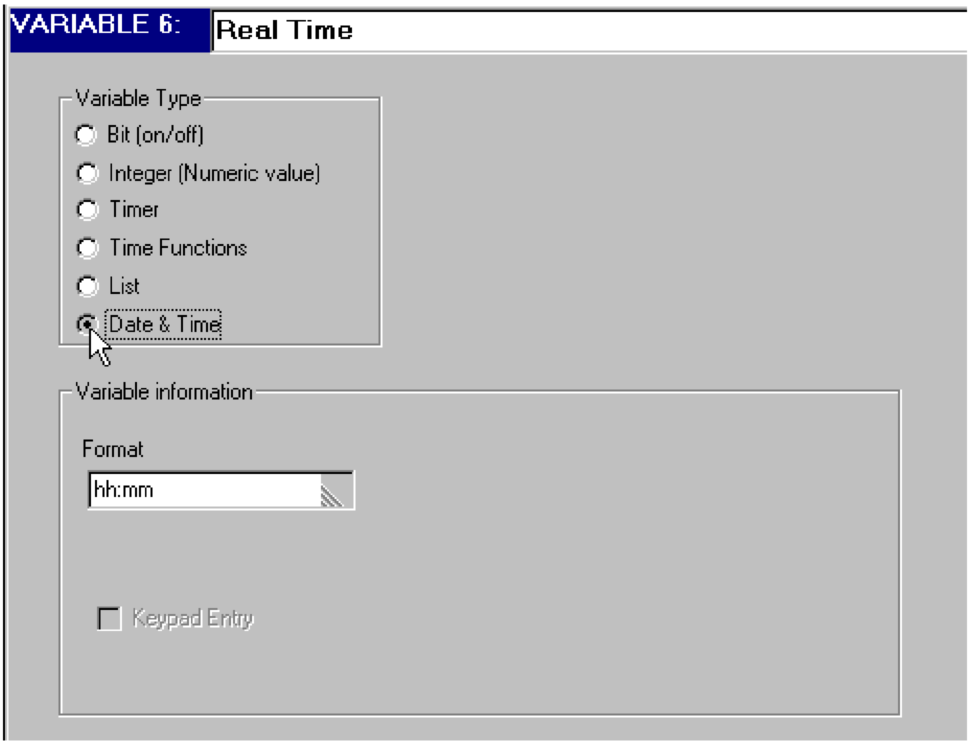
Example: write the variable ranges into SI 80 - 83, then writing an analog input into SI 84:


1
EN ENO
ST
# 0
A
B
SI 80 Linear conversion: x1
2
# 1023
SI 81 Linear conversion: x2
3
# 0
SI 82 Linear conversion: y1
4
# 100
SI 83 Linear conversion: y2
5
MI 23 Input Value from I/O
SI 84 Linear conversion: X
6
SB 1 Always 1
SB 80 Activate linear function

Display the Time and Date on the LCD

To display the time and date on the M90:

1. Select **Date & Time** from the Variable Type check box in the Variable Editor.



VARIABLE 6: Real Time

Variable Type

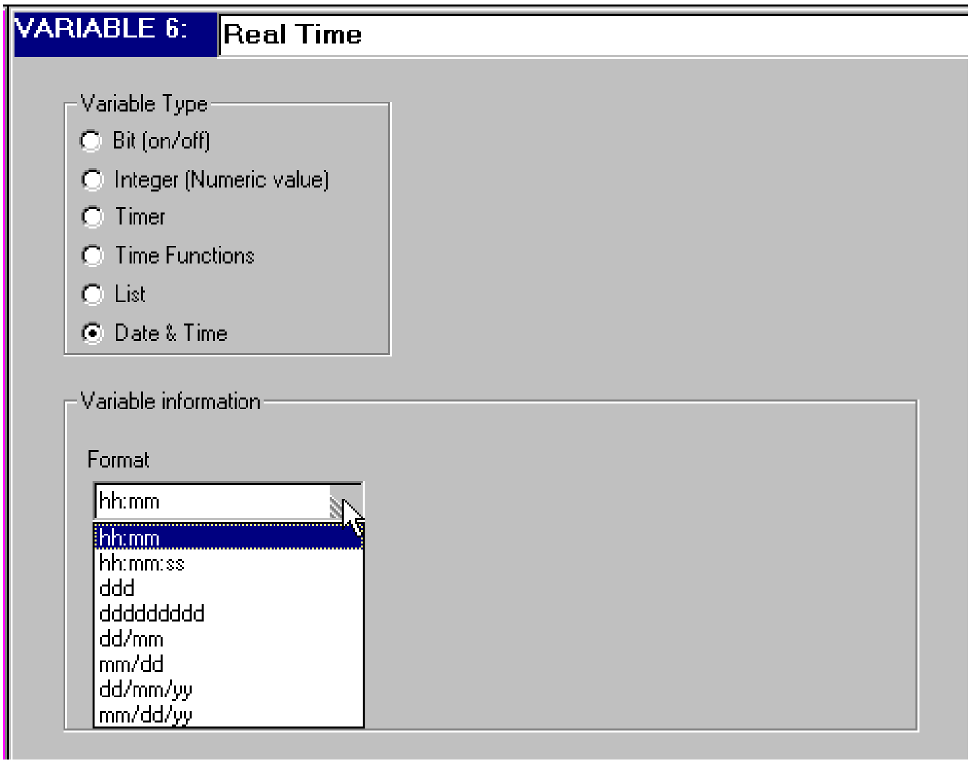
○ Bit (on/off)
○ Integer (Numeric value)
○ Timer
○ Time Functions
○ List
● Date & Time

Variable information

Format: hh:mm

□ Keypad Entry

2. Select the Time & Date Format in the Variable information box.



VARIABLE 6: Real Time

Variable Type

○ Bit (on/off)
○ Integer (Numeric value)
○ Timer
○ Time Functions
○ List
● Date & Time

Variable information

Format: hh:mm

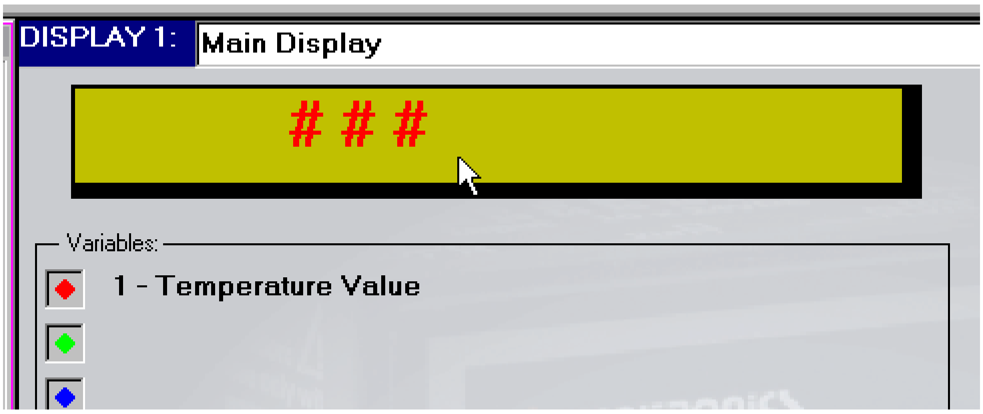
Make sure to define a Display field **large enough** for the selected Date & Time format.

Displaying Special Symbols on the LCD

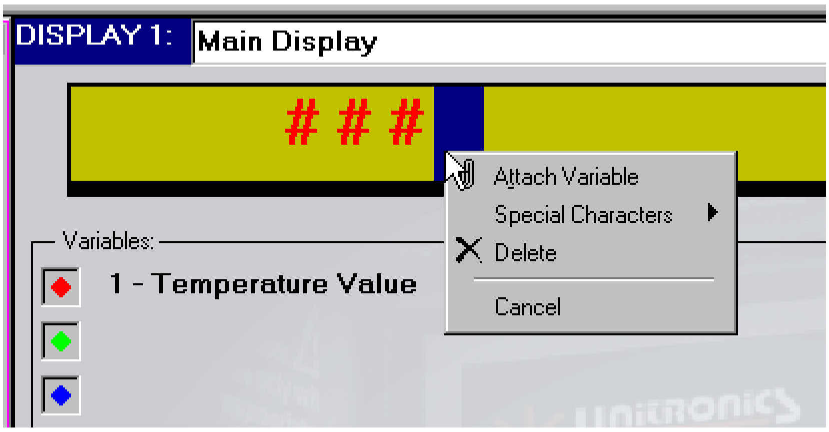
There are a number of Special Symbols such as arrows and degree signs, that may be displayed on the M90' LCD.

To enter a Special Symbol into a Display:

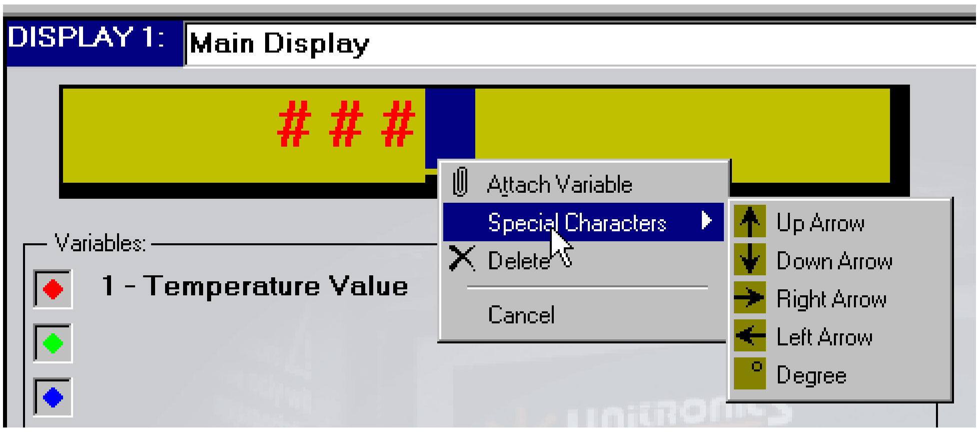
1. Choose the position in the Display field .



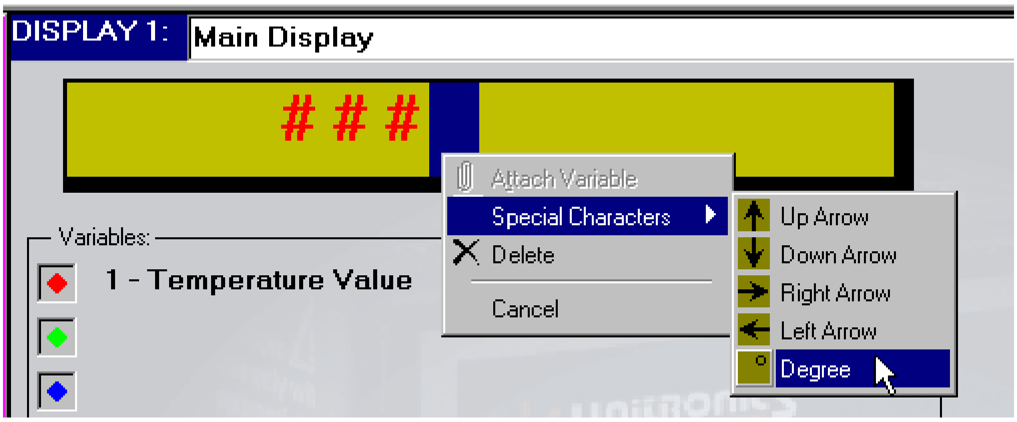
2. Right click to open the Variable modification menu.



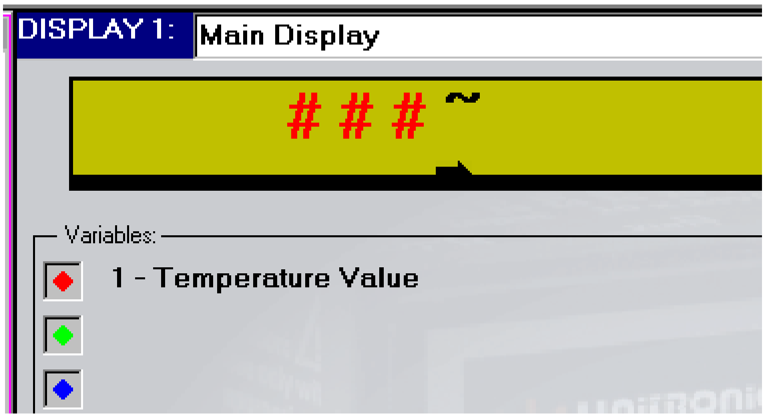
3. Select **Special Characters** from the menu. The Special Characters menu opens.



4. Select the Special Character you wish to add.



5. A ~ symbol will appear in the Display screen to show you that a Special Symbol was inserted. The selected symbol will appear on the controller.



Display Integer values as ASCII or Hexadecimal

You can:

- Display the values in an MI vector as ASCII characters.
- Display a register value in hexadecimal format.

To do this, attach a numeric Variable to a Display. The variable uses linearization to display the value(s) in the desired format.

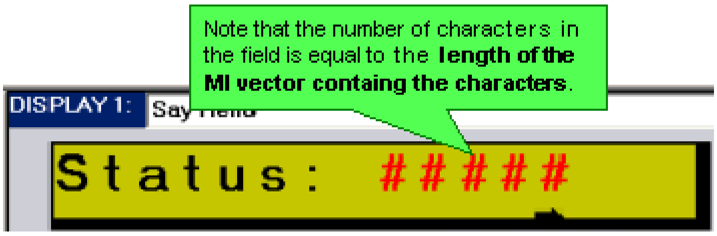
Note that non-supported ASCII characters will be shown as <space> characters.

ASCII -Hexadecimal character table

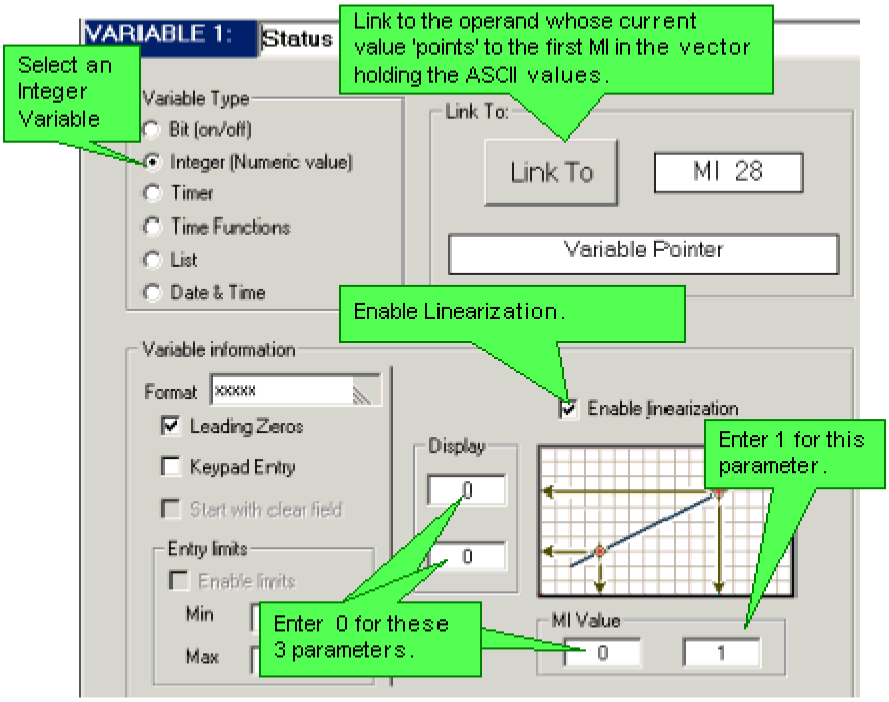
Vector as ASCII

When the application shown in the example below is downloaded, the ASCII characters 'Hello' will be displayed on the M90 screen when Key #3 is pressed.

1. Create a Variable Field in a Display, then attach a Variable.


Note that the number of characters in the field is equal to the **length of the MI vector containg the characters**.

2. Define the Variable as shown below.



Select an Integer Variable

Link to the operand whose current value 'points' to the first MI in the vector holding the ASCII values.

Enable Linearization.

Enter 1 for this parameter.

Enter 0 for these 3 parameters.

VARIABLE 1: Status

Variable Type
○ Bit (on/off)
● Integer (Numeric value)
○ Timer
○ Time Functions
○ List
○ Date & Time

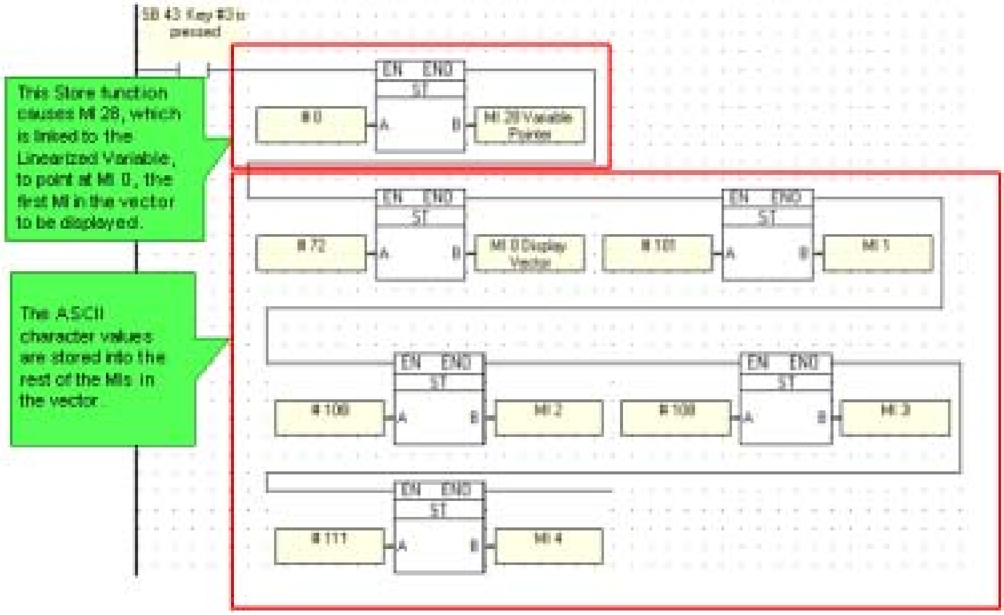
Link To:
Link To: MI 28
Variable Pointer

Variable information
Format: xxxxx
☑ Leading Zeros
☐ Keypad Entry
☐ Start with clear field

Entry limits
☐ Enable limits
Min
Max

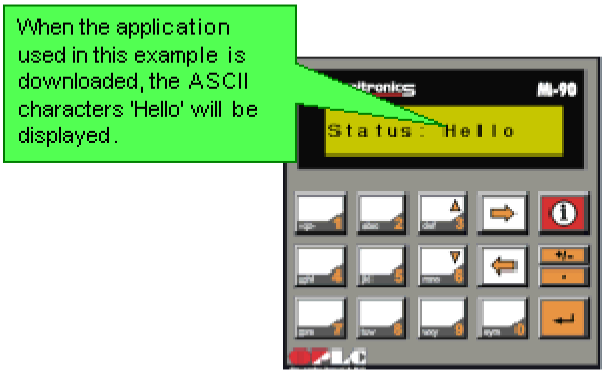
☑ Enable linearization
Display: 0, 0
MI Value: 0, 1

3. The Ladder net below sets the Variable pointer and stores ASCII values into the MI vector.



This Store function causes MI 28, which is linked to the Linearized Variable, to point at MI 0, the first MI in the vector to be displayed.

The ASCII character values are stored into the rest of the MIs in the vector.

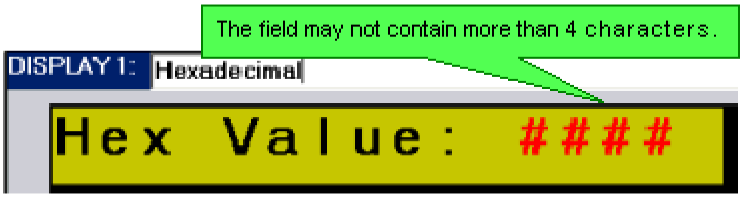


When the application used in this example is downloaded, the ASCII characters 'Hello' will be displayed.

Register Value in Hexadecimal

When the application shown in the example below is downloaded, the hexadecimal value of 63 will be displayed on the M90 screen.

1. Create a Variable Field in a Display, then attach a Variable. Note that if the field is too short, only the right-most characters are displayed. For example, the hex value 63(3F) cannot be shown in a field one character long.

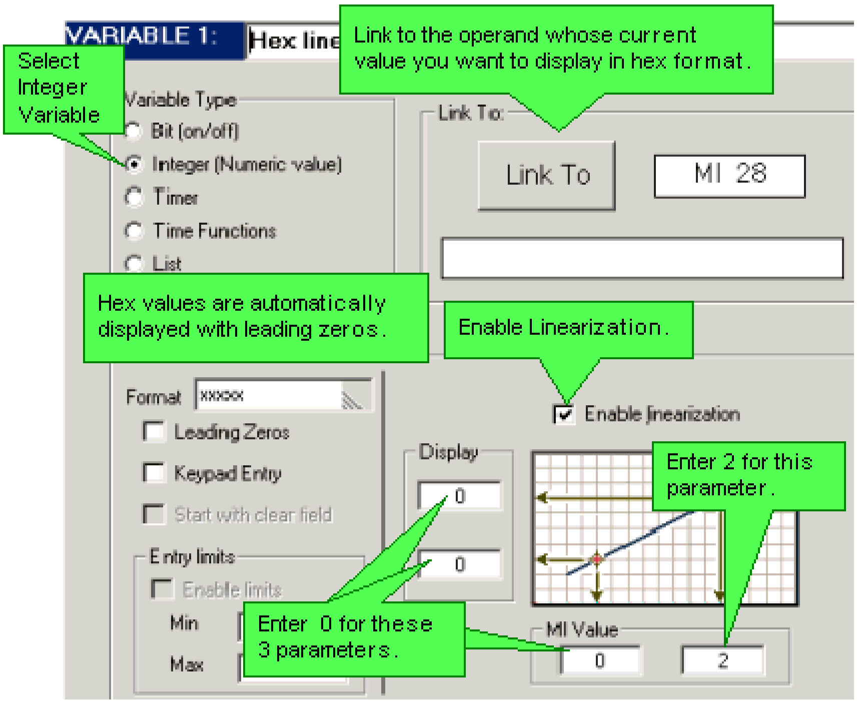


The field may not contain more than 4 characters.

DISPLAY 1: Hexadecimal

Hex Value: ####

2. Define the Variable as shown below.



VARIABLE 1: Hex line

Select Integer Variable

Link to the operand whose current value you want to display in hex format.

Variable Type
○ Bit (on/off)
● Integer (Numeric value)
○ Timer
○ Time Functions
○ List

Link To: MI 28

Hex values are automatically displayed with leading zeros.

Enable Linearization.

Format xxxxx
□ Leading Zeros
□ Keypad Entry
□ Start with clear field

Entry limits
□ Enable limits
Min
Max

☑ Enable linearization

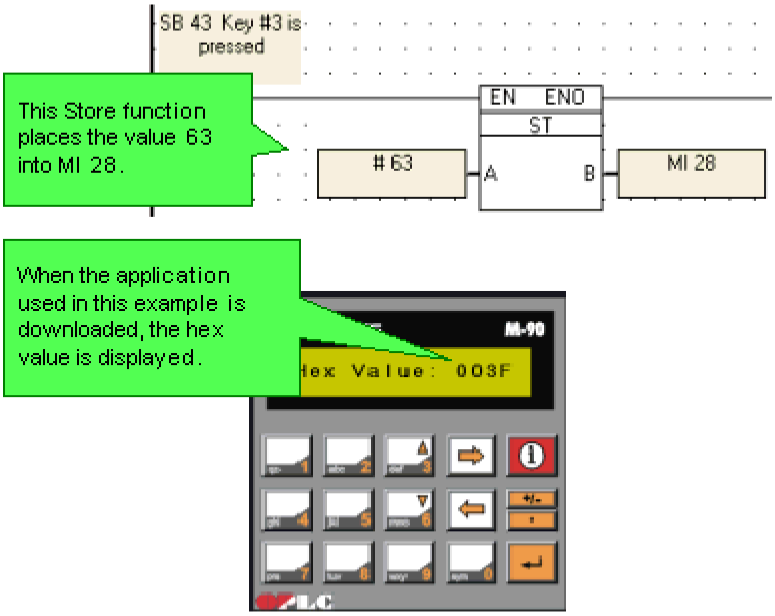
Display 0 0

MI Value 0 2

Enter 2 for this parameter.

Enter 0 for these 3 parameters.

3. The Ladder net below stores the value into the MI.



This Store function places the value 63 into MI 28.

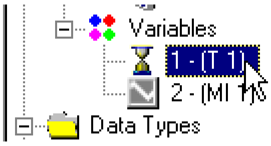
When the application used in this example is downloaded, the hex value is displayed.

Scrolling between Displays via the M90 keypad

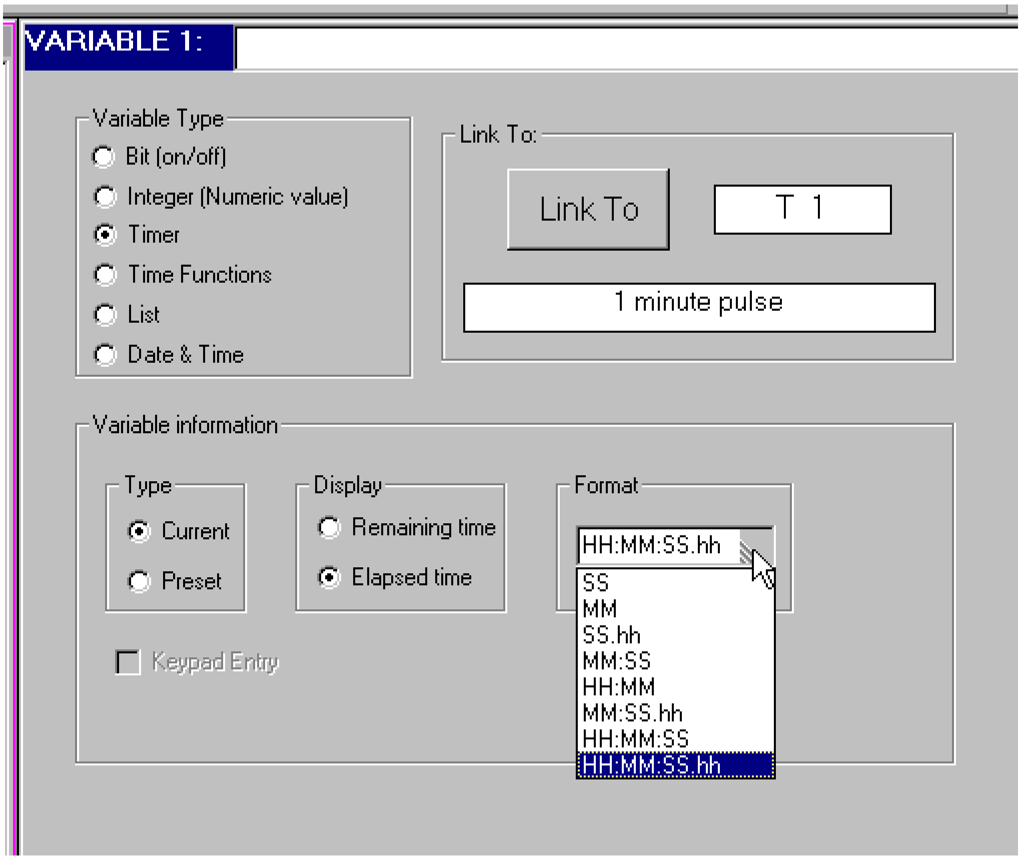
Use Jump conditions to scroll between Display screens using the M90 keypad.

Selecting a Timer Display format

1. From the Navigator Window, create or choose an existing Timer Variable.



2. Open the Timer format drop-down menu in the Variable Editor.



VARIABLE 1:

Variable Type

○ Bit (on/off)
○ Integer (Numeric value)
● Timer
○ Time Functions
○ List
○ Date & Time

Link To:

Link To: T 1

1 minute pulse

Variable information

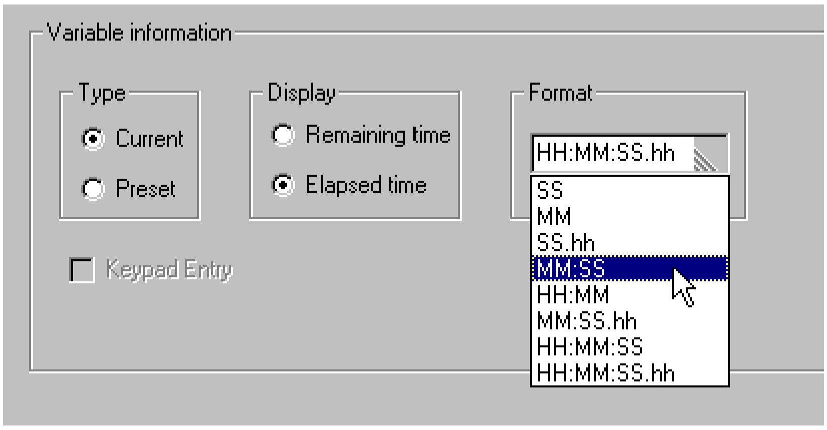
Type: ● Current ○ Preset

Display: ○ Remaining time ● Elapsed time

Format: HH:MM:SS.hh

□ Keypad Entry

3. Select the Timer format from the drop-down menu in the Variable Editor.



Variable information

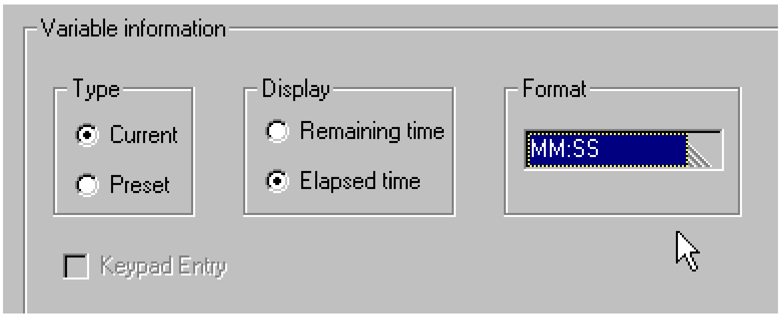
Type: ● Current ○ Preset

Display: ○ Remaining time ● Elapsed time

Format: HH:MM:SS.hh

□ Keypad Entry

4. The selected format is displayed in the Format window.



Variable information

Type: ● Current ○ Preset

Display: ○ Remaining time ● Elapsed time

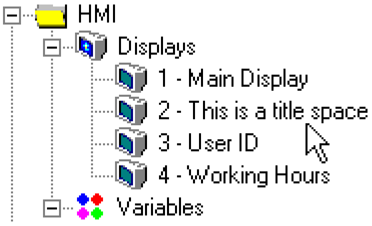
Format: MM:SS

□ Keypad Entry

<u>Toggling between Displays</u>

To move quickly between Displays:

1. Click the Display number in the Navigation Window that you want to view.



HMI
Displays
1 - Main Display
2 - This is a title space
3 - User ID
4 - Working Hours
Variables

2. The Display immediately appears in the Display Editor.

How many displays can I create?

Yes, there is a limit of 80 text displays. Do remember that, in addition to the text displays, there are 120 List Variables that can be displayed on a controller.

Variable

Variables

You insert Variables into a Display to:

- Show varying values and text on the controller screen.
- Enter values into the controller.

Use the Variable Editor to link variables to the operands that contain the data you want to use in your program. You can use variables in your HMI program to display text that varies according to current conditions or events. Variable integers also can receive data input from the M90's keypad keys, such as an employee ID number, or a set point for process control.

Displaying Variable Values in a Display

To display data from an HMI variable within an M90 display, you must:

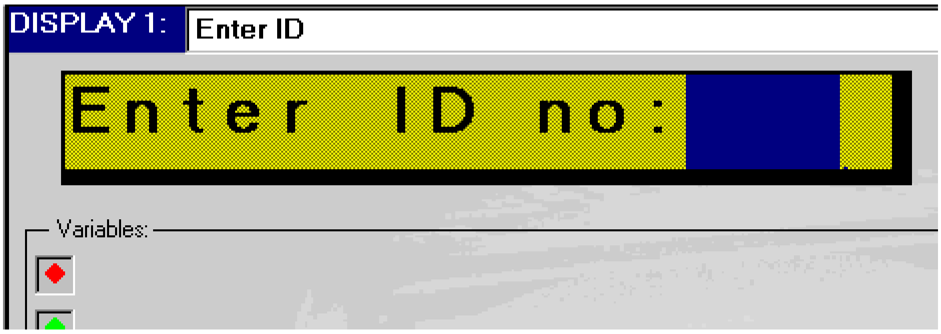
- Create a field within the display that is long enough to hold the variable data.
- Attach a variable to the field.

To Create a Field

1. Click your cursor in the display. This is the starting point of the field.
2. To create the field, either:
3. Drag the cursor across the display. The field you create is automatically highlighted in blue.

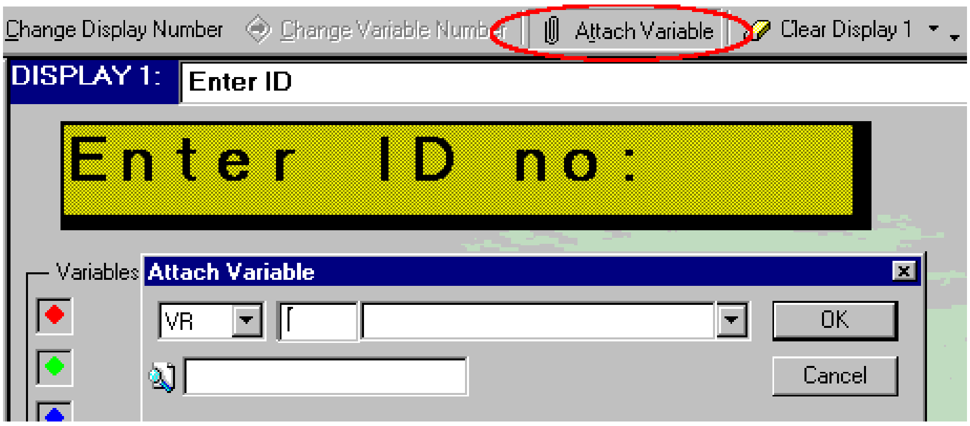
   OR
4. Hold the SHIFT key down, and press the right-pointing arrow key. Each time you press the arrow key, a space is automatically highlighted in blue.

In the figure below, the display contains a field two spaces long.

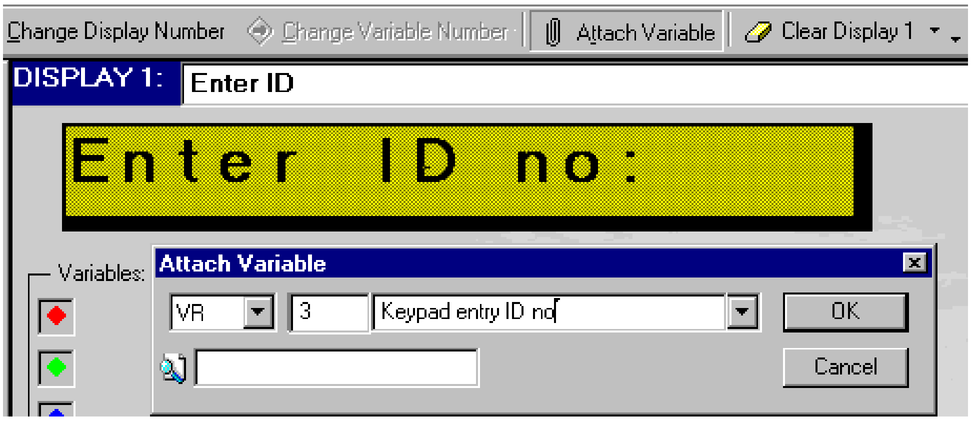


To Attach a Variable

1. Click Attach Variable on the HMI toolbar. T he Attach Variable dialog box opens as shown below.

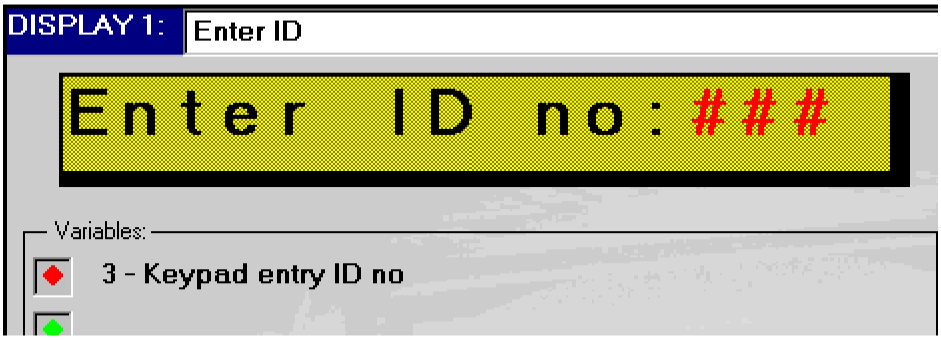


2. Enter the number of the desired variable as shown below and press OK. If you do not enter a variable number, the program assigns a default variable.



VR 3 Keypad entry ID no

3. The variable-linked spaces now appear as red pound signs, and the variable itself appears in the Variable pane of this Display as shown below.



DISPLAY 1: Enter ID

Enter ID no: ###

Variables:

3 - Keypad entry ID no

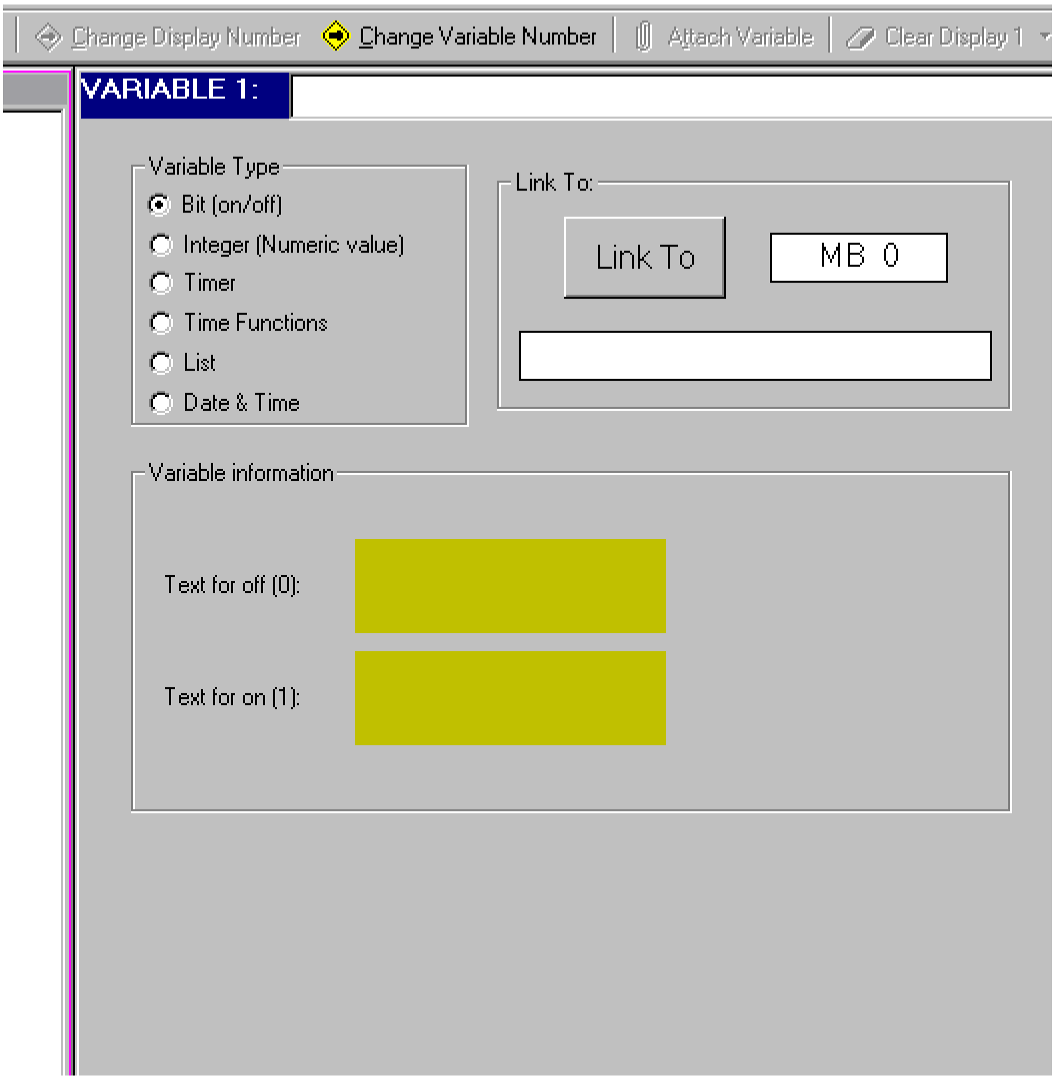
Use the Variable Editor to:

- Set variable types and properties.
- Create up to 120 list variables to display fixed text messages.
- Enable data entry via the M90 keypad.

Up to fifty variables may be included in your application. The different types of variables are listed below.

Variable Type	Linked to	Display Options:
Bit	MB	Create a text display for ON and OFF.
Integer	MI	Choose integer display format; enable linearization and keypad entry.
Timer	T	Display either elapsed time or remaining time and allow timer modification via the M90 keypad.
Time Functions	MI	Display and modify Time function from hour up to year.
List	MI	Create up to 120 additional fixed text messages for different values of an MI / SI.
Date & Time	RTC	Set the display format (from Hours/Minutes to Month/Day/Year) and enable keypad entry.

Variable Editor view:



VARIABLE 1:

Variable Type
● Bit (on/off)
○ Integer (Numeric value)
○ Timer
○ Time Functions
○ List
○ Date & Time

Link To: MB 0

Variable information

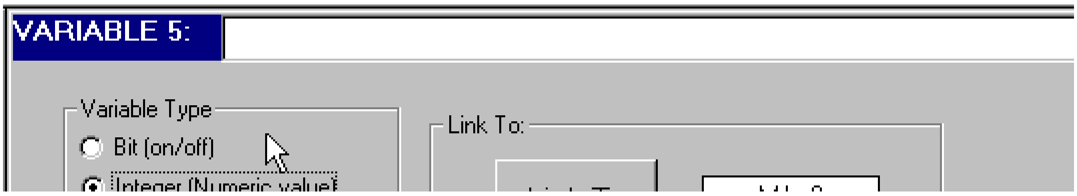
Text for off (0):

Text for on (1):

Naming a Variable

To assign a title to a Variable:

1. Open a Variable in the Variable Editor.

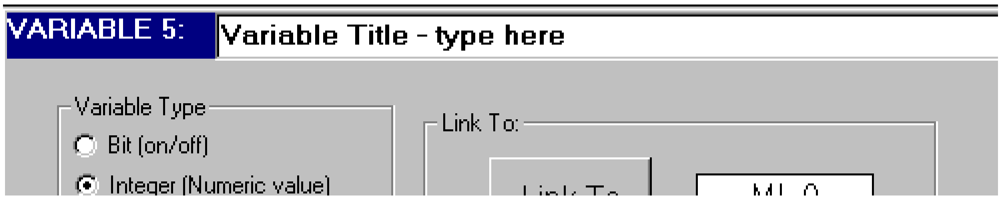


VARIABLE 5:

Variable Type
○ Bit (on/off)
● Integer (Numeric value)

Link To: MI 0

2. Type the Variable name in the title field.

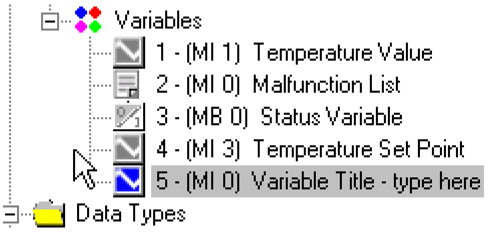


VARIABLE 5: Variable Title - type here

Variable Type
○ Bit (on/off)
● Integer (Numeric value)

Link To: MI 0

The Variable name appears with the Variable number in the Navigator window.



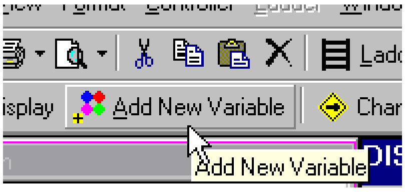
Variables
- 1 - (MI 1) Temperature Value
- 2 - (MI 0) Malfunction List
- 3 - (MB 0) Status Variable
- 4 - (MI 3) Temperature Set Point
- 5 - (MI 0) Variable Title - type here

Data Types

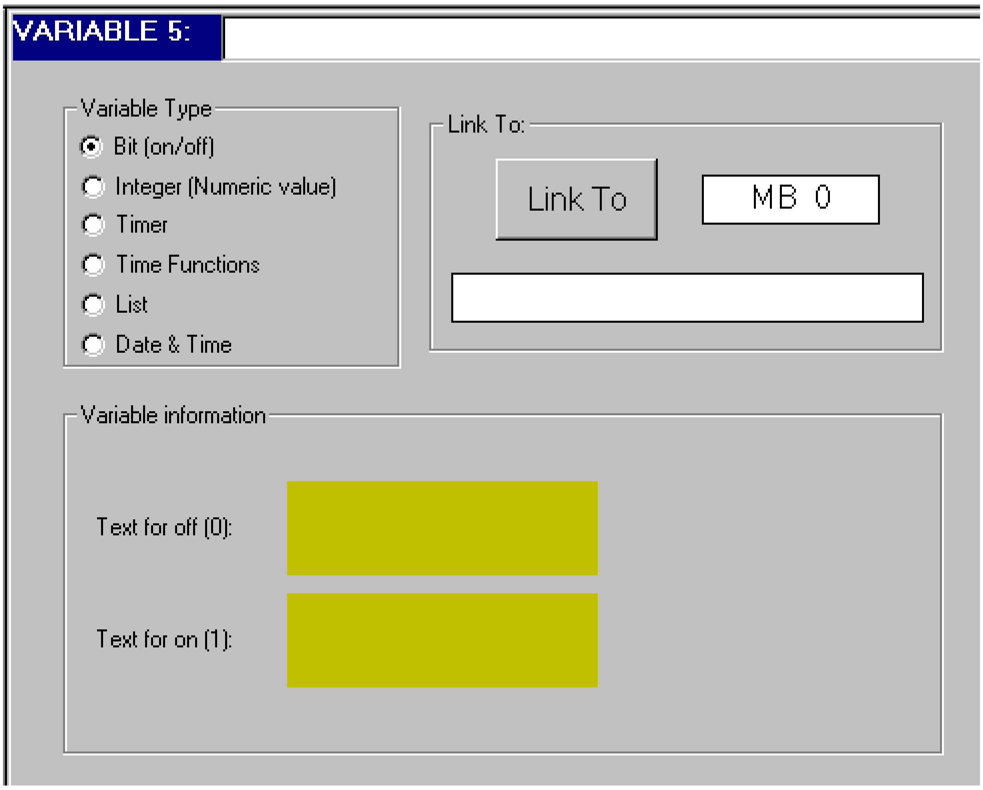
Creating Variables

To create a new Variable:

1. Click the Add New Variable icon on the HMI toolbar.



2. A new Variable opens in the Variable Editor.



VARIABLE 5:

Variable Type
- ● Bit (on/off)
- ○ Integer (Numeric value)
- ○ Timer
- ○ Time Functions
- ○ List
- ○ Date & Time

Link To:

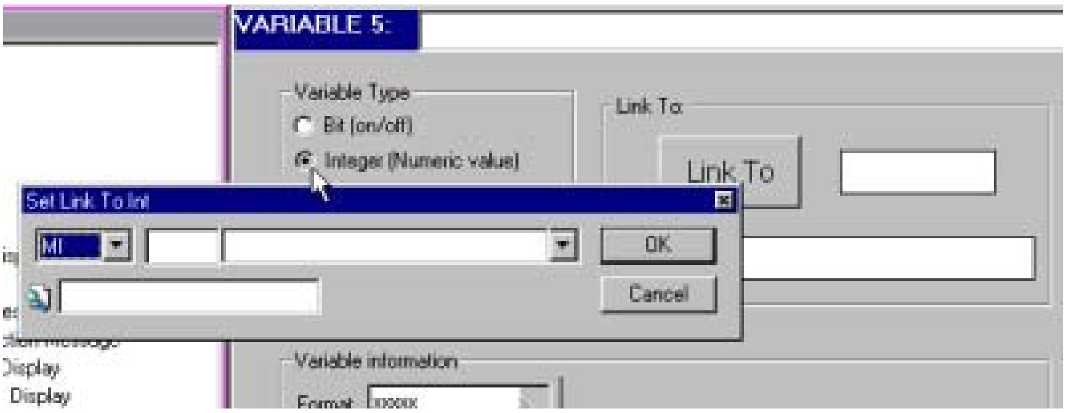
Link To | MB 0

Variable information

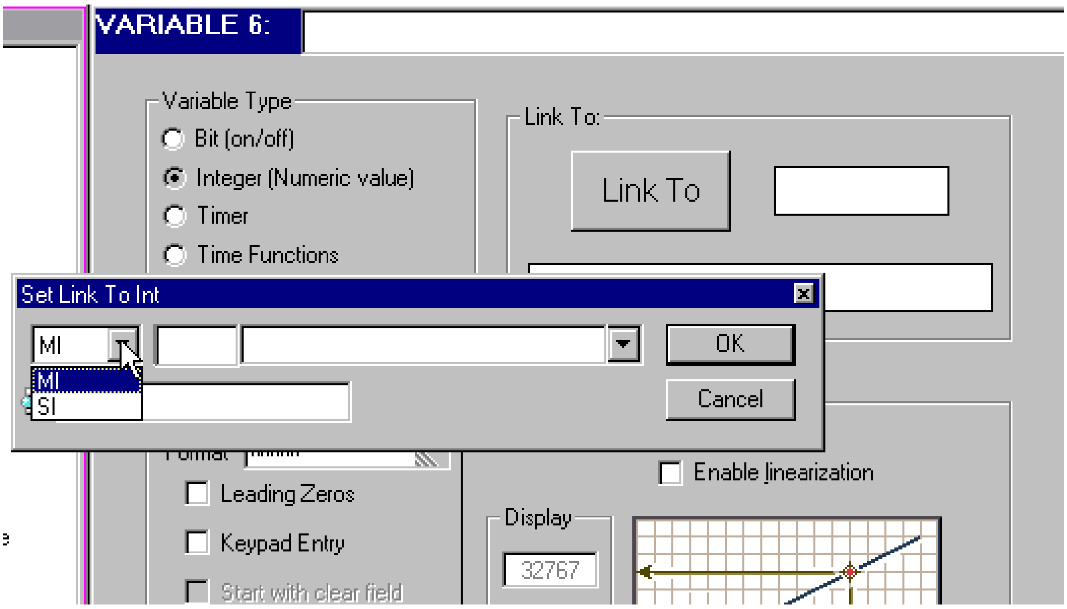
Text for off (0):

Text for on (1):

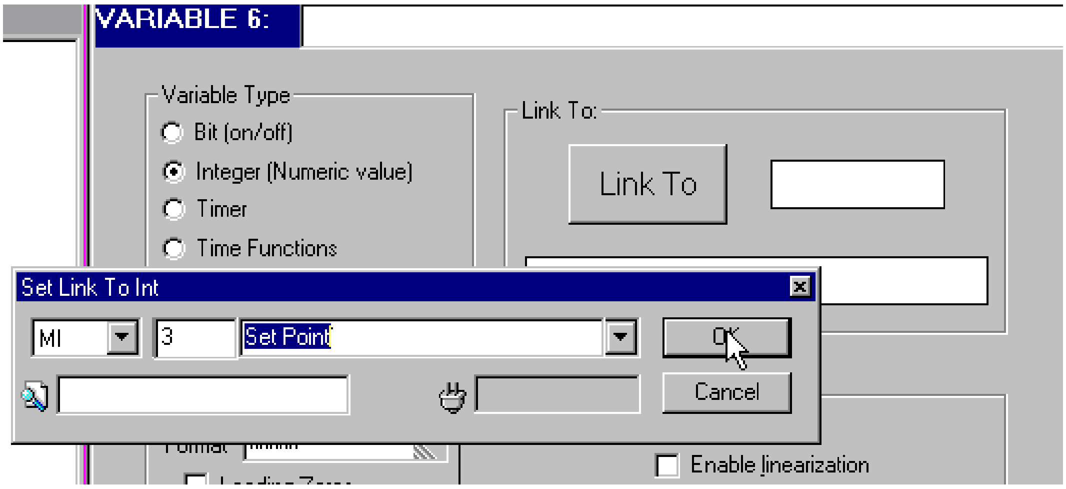
3. Select the desired Variable Type.



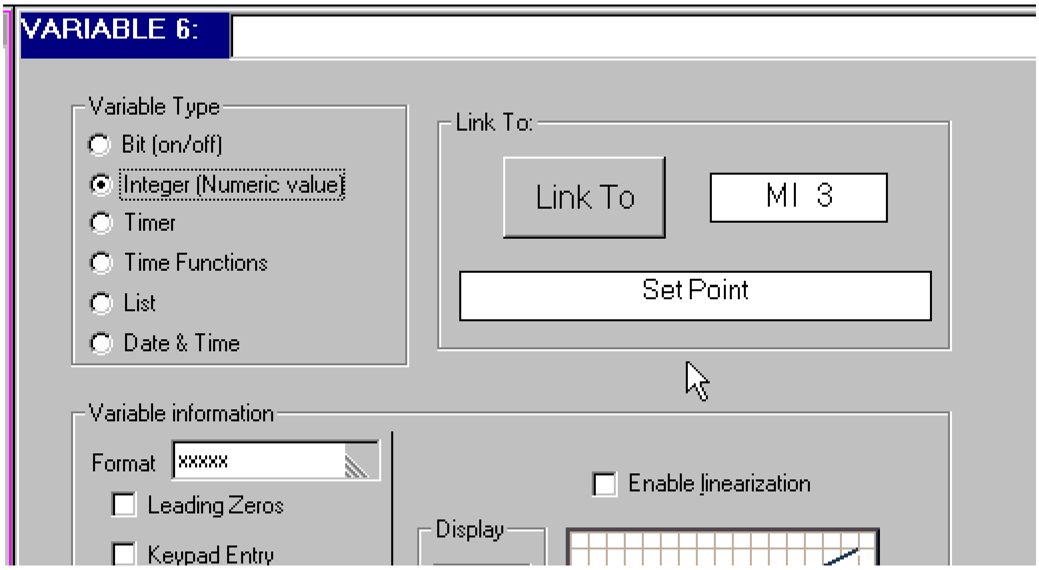
4. Select the Operand type.



5. Enter the Operand Address and Symbol.



6. The new Variable appears with the appropriate link in the Variable Editor.



VARIABLE 6:

Variable Type

○ Bit (on/off)

● Integer (Numeric value)

○ Timer

○ Time Functions

○ List

○ Date & Time

Link To:

Link To MI 3

Set Point

Variable information

Format xxxxx

□ Leading Zeros

□ Keypad Entry

□ Enable linearization

Display

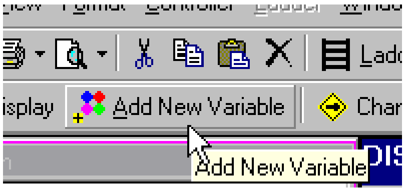
Showing an MI value on the controller's LCD

To display an MI value on the controller display:

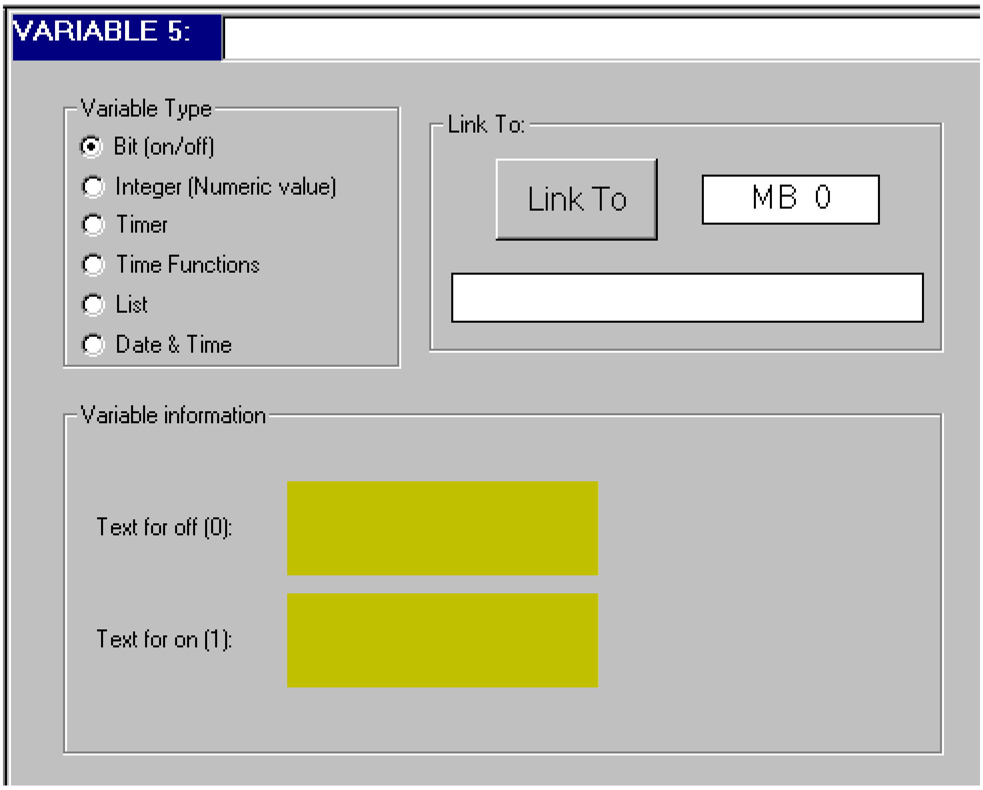
1. *Create a Variable*

   To create a new Variable:

   1. Click the Add New Variable icon on the HMI toolbar.

   2. A new Variable opens in the Variable Editor.



VARIABLE 5:

Variable Type

● Bit (on/off)
○ Integer (Numeric value)
○ Timer
○ Time Functions
○ List
○ Date & Time

Link To:

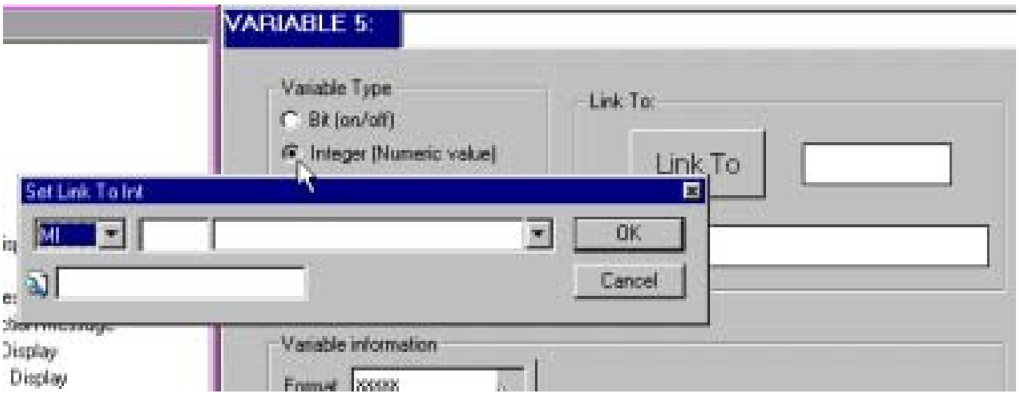
Link To: MB 0

Variable information

Text for off (0):

Text for on (1):

3. Select the desired Variable Type.



VARIABLE 5:

Variable Type

○ Bit (on/off)
● Integer (Numeric value)

Link To:

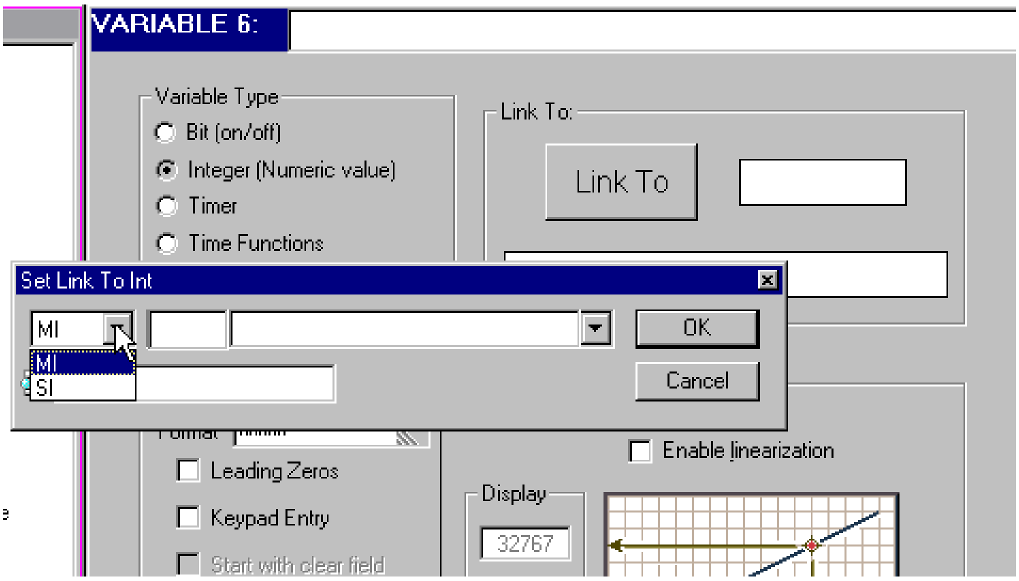
Set Link To Int

MI

Variable information

Format: xxxxx

4. Select the Operand type.



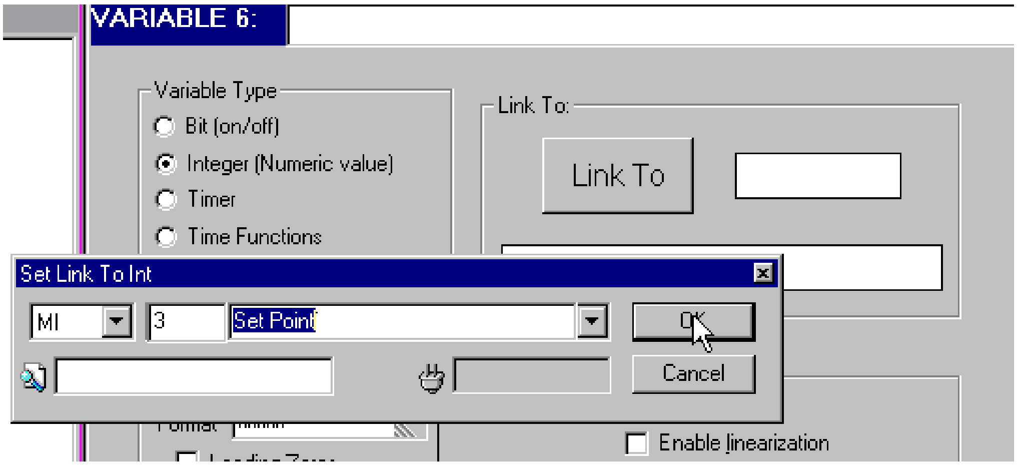
VARIABLE 6:

Variable Type
○ Bit (on/off)
● Integer (Numeric value)
○ Timer
○ Time Functions

Set Link To Int
MI
MI
SI

Link To:
☐ Leading Zeros
☐ Keypad Entry
☐ Start with clear field
☐ Enable linearization
Display: 32767

5. Enter the Operand Address and Symbol.



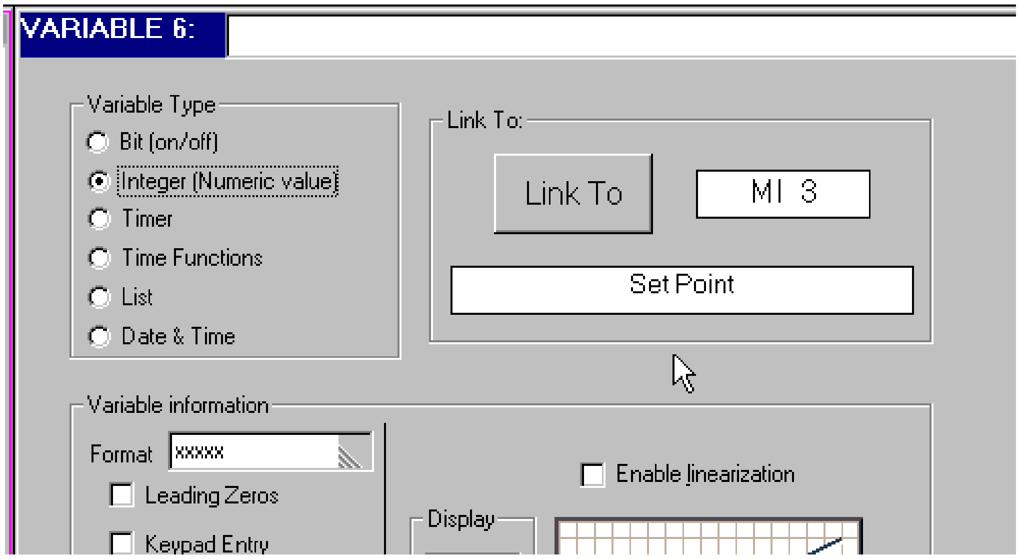
VARIABLE 6:

Variable Type
○ Bit (on/off)
● Integer (Numeric value)
○ Timer
○ Time Functions

Set Link To Int
MI 3 Set Point

Link To:
☐ Enable linearization

6. The new Variable appears with the appropriate link in the Variable Editor.



VARIABLE 6:

Variable Type
○ Bit (on/off)
● Integer (Numeric value)
○ Timer
○ Time Functions
○ List
○ Date & Time

Link To: MI 3
Set Point

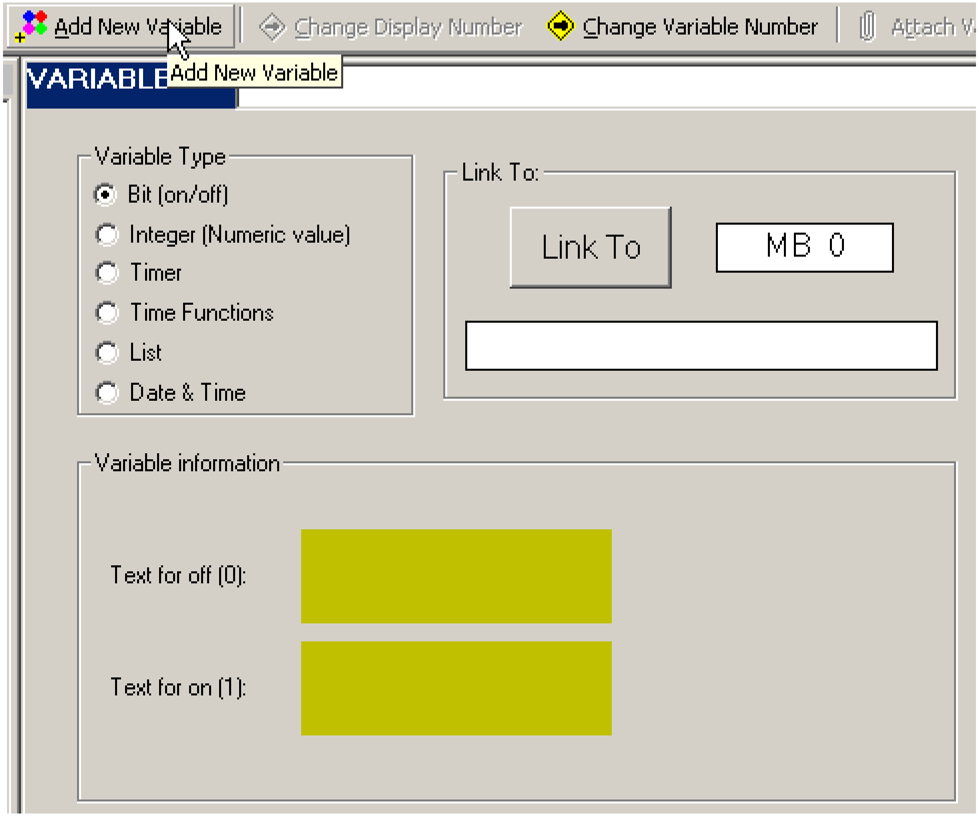
Variable information
Format xxxxx
☐ Leading Zeros
☐ Keypad Entry
☐ Enable linearization
Display

7. *Create a Variable Field in a Display* and attach it to the Variable.

List Variable: Display text according to a changing MI value

To display different texts for different values of the same MI:

1. Create a new Variable.



VARIABLE

Variable Type
- ● Bit (on/off)
- ○ Integer (Numeric value)
- ○ Timer
- ○ Time Functions
- ○ List
- ○ Date & Time

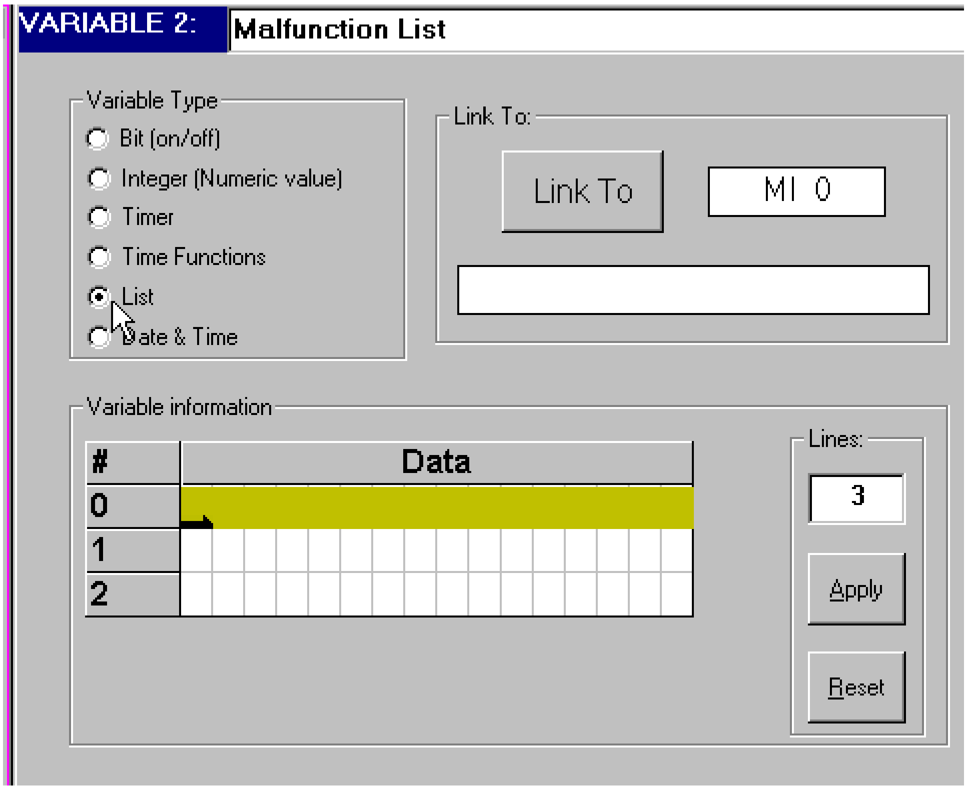
Link To: MB 0

Variable information

Text for off (0):

Text for on (1):

2. Select **List** Variable type.



VARIABLE 2: Malfunction List

Variable Type
- ○ Bit (on/off)
- ○ Integer (Numeric value)
- ○ Timer
- ○ Time Functions
- ● List
- ○ Date & Time

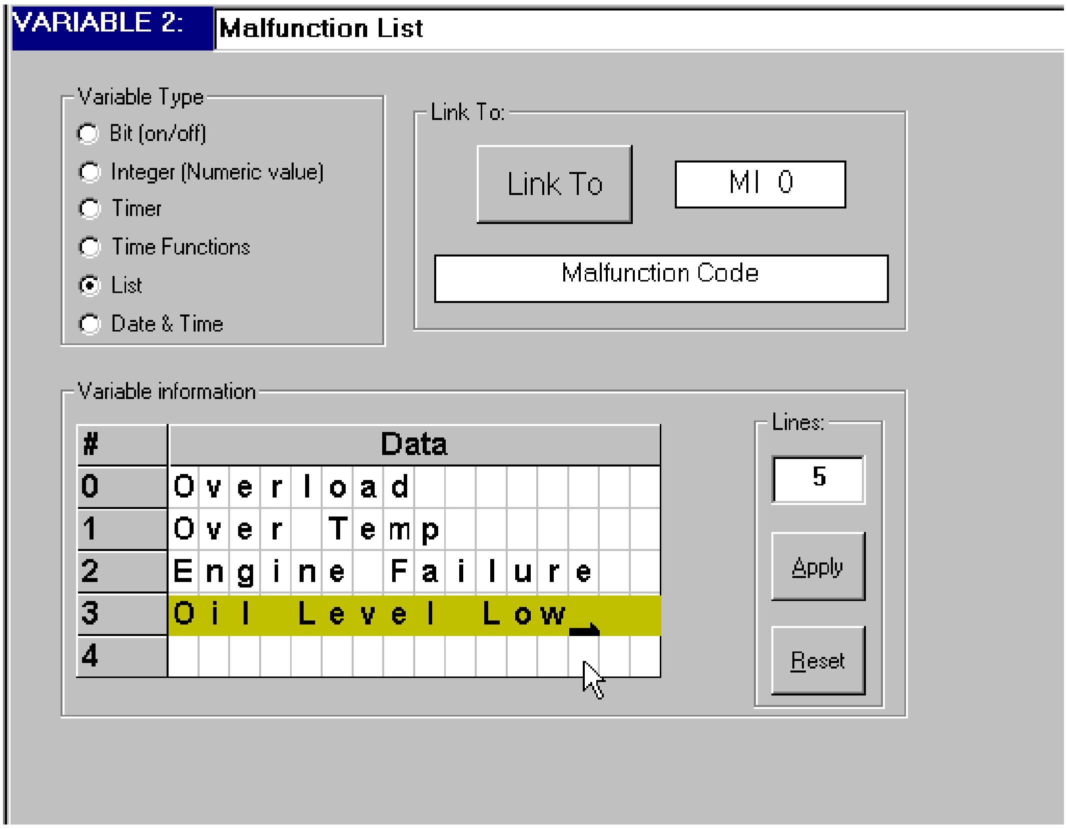
Link To: MI 0

Variable information

#	Data
0	
1	
2	

Lines: 3

3. Enter the desired text for each possible value of the linked MI.



VARIABLE 2: Malfunction List

Variable Type

○ Bit (on/off)
○ Integer (Numeric value)
○ Timer
○ Time Functions
● List
○ Date & Time

Link To:

MI 0

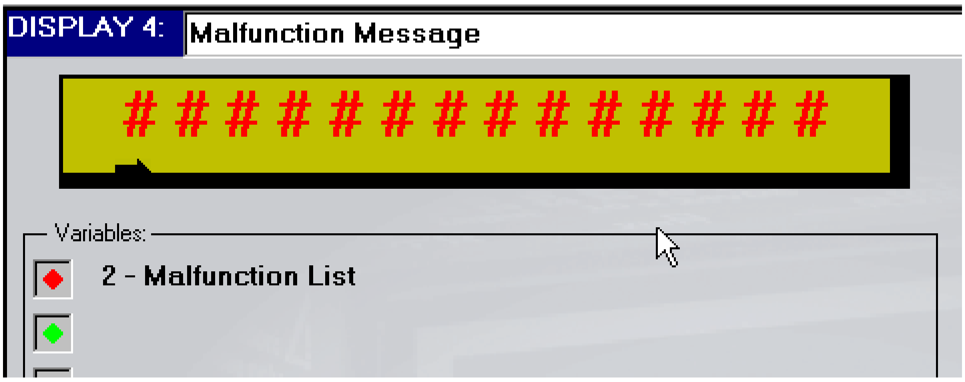
Malfunction Code

Variable information

#	Data
0	Overload
1	Over Temp
2	Engine Failure
3	Oil Level Low
4	

Lines: 5

4. Attach the Variable to a Display field.



DISPLAY 4: Malfunction Message

# # # # # # # # # # # # # #

Variables:

2 - Malfunction List

The text on the Display will be determined by the value written into MI 0 in the Ladder.

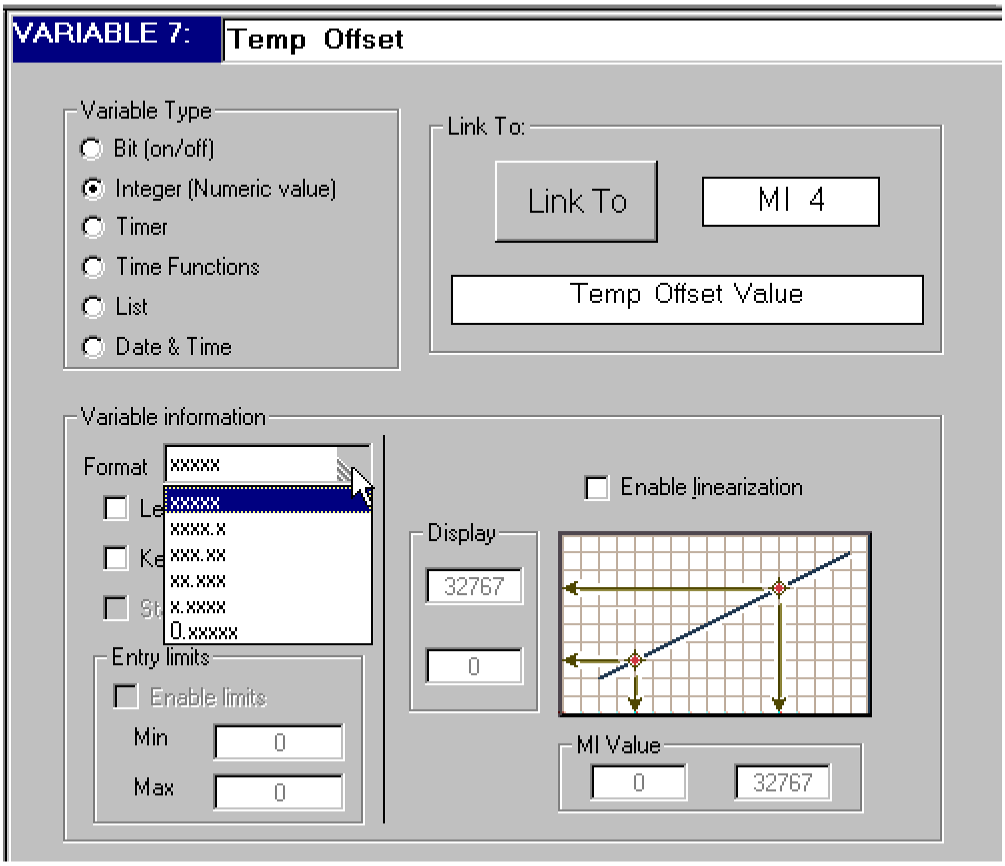
Example:

If MI 0 = 2, then the message will be **Engine Failure**.

Keypad Entry values

To enter a decimal number into a MI from the M90 keypad:

1. Select the desired decimal format from the Variable information box for the Integer Variable.



VARIABLE 7: Temp Offset

Variable Type
○ Bit (on/off)
● Integer (Numeric value)
○ Timer
○ Time Functions
○ List
○ Date & Time

Link To:
Link To MI 4
Temp Offset Value

Variable information
Format xxxxx
xxxxx
xxxx.x
xxx.xx
xx.xxx
x.xxxx
0.xxxxx

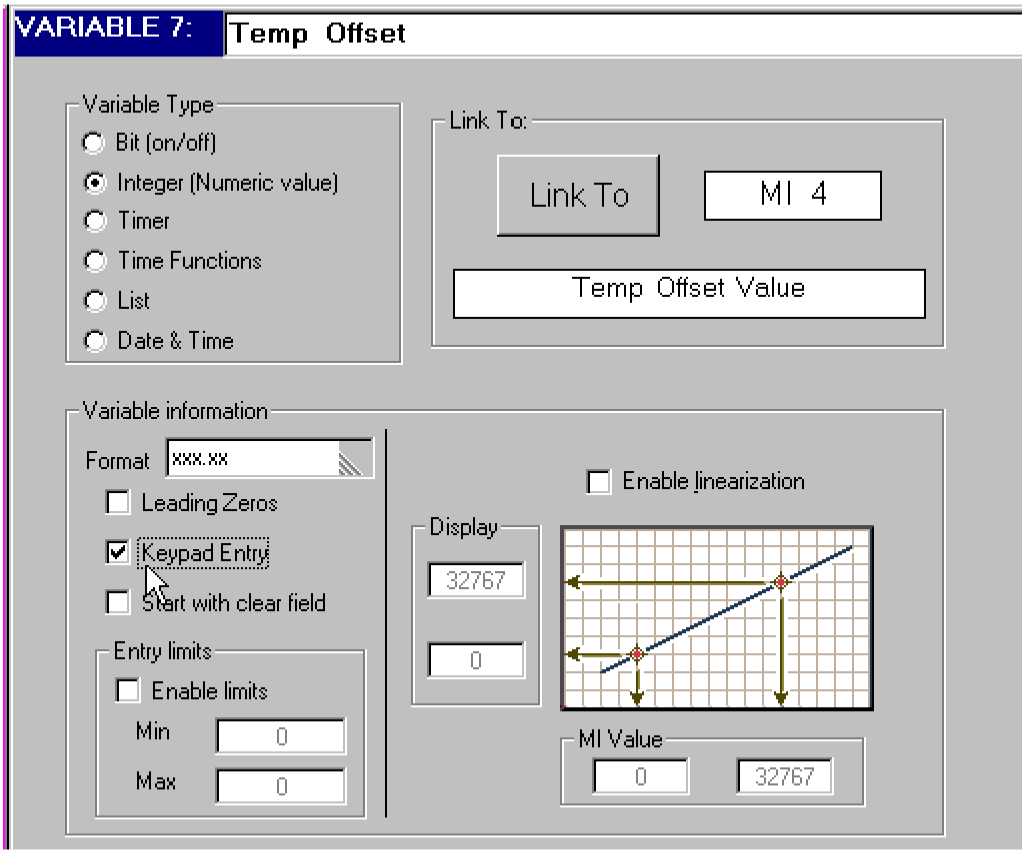
□ Enable linearization

Display 32767 0

Entry limits
□ Enable limits
Min 0
Max 0

MI Value 0 32767

2. Select **Keypad Entry** from the Variable information box.



VARIABLE 7: Temp Offset

Variable Type
○ Bit (on/off)
● Integer (Numeric value)
○ Timer
○ Time Functions
○ List
○ Date & Time

Link To:
Link To MI 4
Temp Offset Value

Variable information
Format xxx.xx
□ Leading Zeros
☑ Keypad Entry
□ Start with clear field

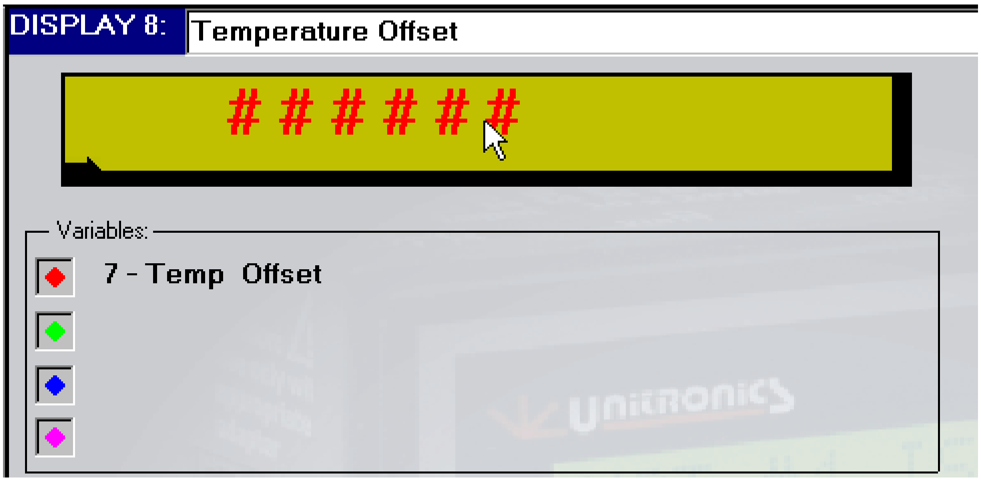
□ Enable linearization

Display 32767 0

Entry limits
□ Enable limits
Min 0
Max 0

MI Value 0 32767

3. Attach the Variable to a field in the desired Display.



Note ♦ When an HMI keypad entry variable is active, and the Enter key is pressed on the controller keypad, SB 30 HMI Keypad Entries Complete turns ON. This can be used as a Jump condition.

In addition , note that a Display may contain a total of 4 variables. Each one has an SB:

- SB 31 HMI Var 1 Keypad entry completed
- SB 32 HMI Var 2 Keypad entry completed
- SB 33 HMI Var 3 Keypad entry completed
- SB 34 HMI Var 4 Keypad entry completed

The condition of these SBs may be used as Jump Conditions, or to drive calculations in your program.

Force: HMI Keypad Entry Complete, SB 39

A flashing cursor on the M90 LCD screen indicates that the M90 is waiting for a keypad entry. You can turn off the flashing cursor by turning SB 39 ON.

This can enable you to use the same HMI screen to first enable keypad entry, and then to simply display the entered value.

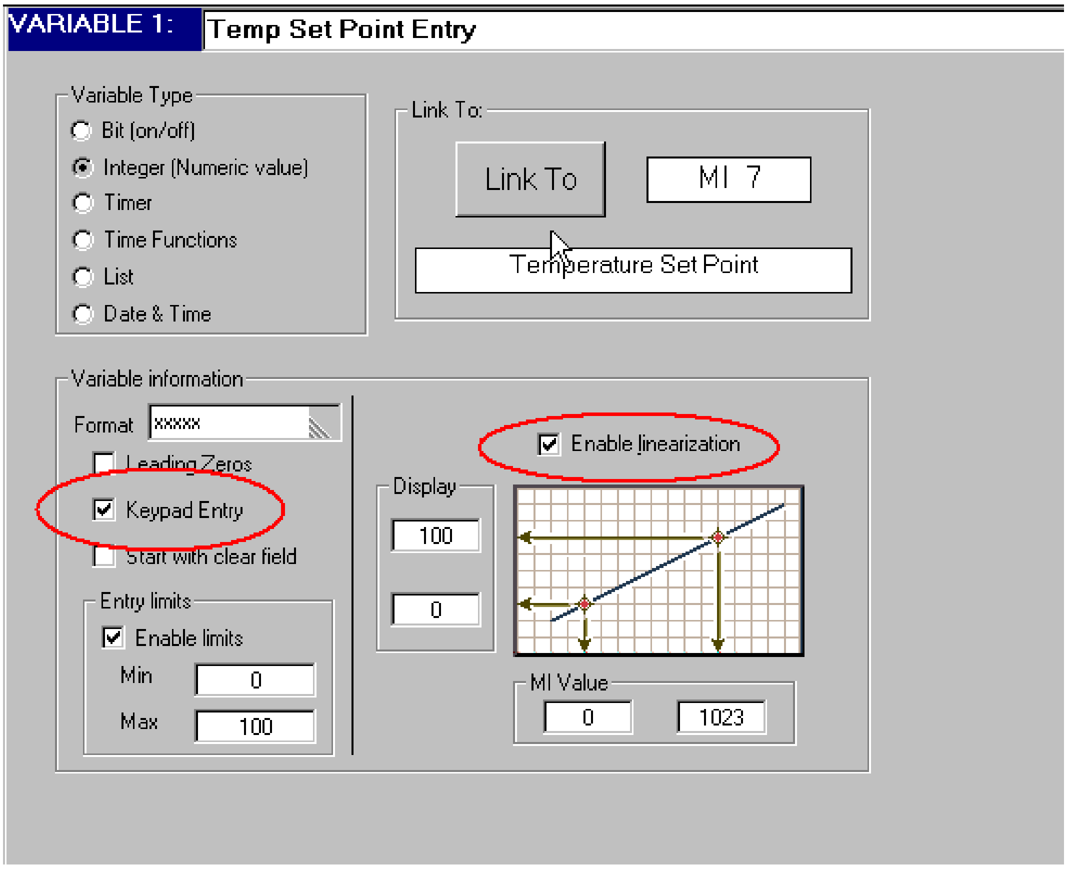
Converting Display values: Linearization

If you want to enter an Analog value, such as temperature, via the M90 keypad and convert that value into a Digital value for comparison with a digital value from a temperature probe, you use the **Enable Linearization** feature in the linked Variable.

This conversion process is Reverse Linearization.

To enable Analog to Digital conversion:

1. Create a Display for entering the analog value.
2. Create an Integer Variable.
3. Select **keypad entry** and **enable linearization**.
4. Enter the linearization values for the x and y axes.



VARIABLE 1: Temp Set Point Entry

Variable Type
○ Bit (on/off)
● Integer (Numeric value)
○ Timer
○ Time Functions
○ List
○ Date & Time

Link To:
Link To | MI 7
Temperature Set Point

Variable information
Format: xxxxx
□ Leading Zeros
☑ Keypad Entry
□ Start with clear field

Entry limits
☑ Enable limits
Min: 0
Max: 100

☑ Enable linearization

Display: 100 / 0

MI Value: 0 / 1023

According to the above example:

- A temperature entry of 100^0 C will be converted to 1023 Digital value.
- A temperature entry of 50^0 C will be converted to 512 Digital value.

Defining a Variable field and attaching a Variable

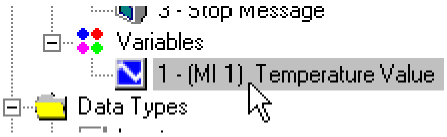
To display data from an HMI variable within an M90 display, you must:

- Create a field within the display that is long enough to hold the variable data.
- Attach a variable to the field.

Displaying an MI value with a leading zero

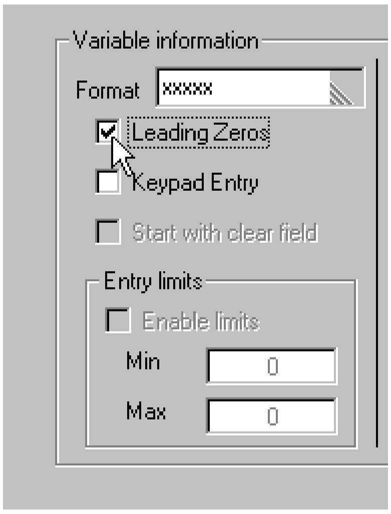
To display an MI with a Leading Zero:

1. Select the desired Variable from the Navigator Window.



3 - Stop Message
Variables
1 - (MI 1) Temperature Value
Data Types

2. Select Leading Zeros from the Variable Information check box.



Variable information

Format xxxxx

☑ Leading Zeros

□ Keypad Entry

□ Start with clear field

Entry limits

□ Enable limits

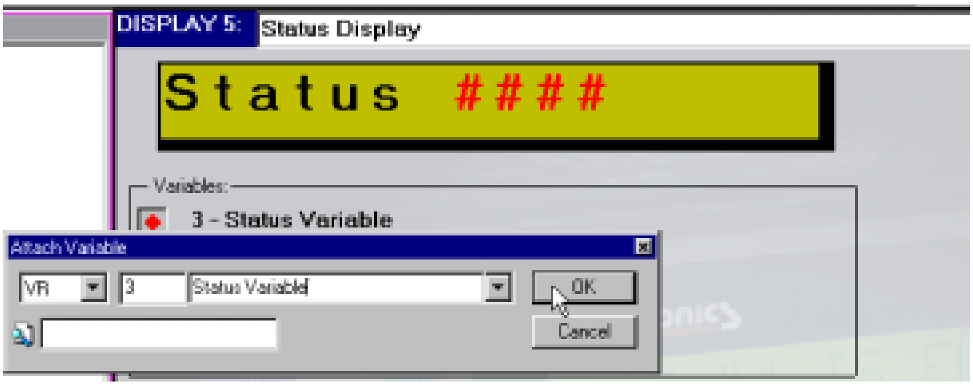
Min 0

Max 0

Displaying text according to the value of a MB or SB

To display a text according to the value of a MB or SB:

1. Create a Display and variable field.



DISPLAY 5: Status Display

Status ####

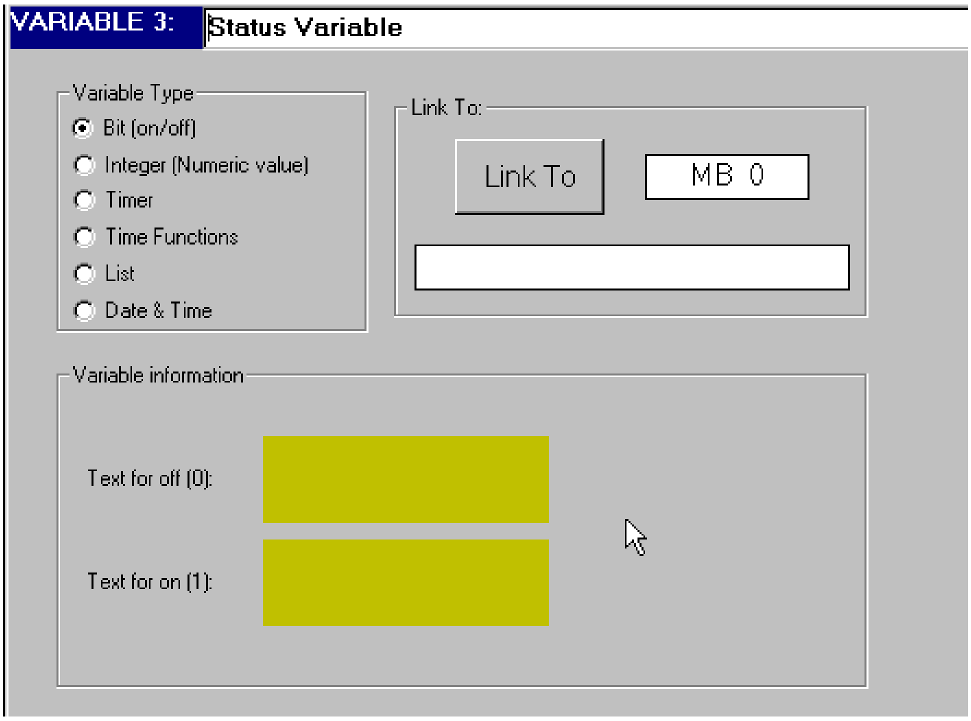
Variables:

3 - Status Variable

Attach Variable

VR 3 Status Variable

2. Create a **Bit** type variable attached to the field .



VARIABLE 3: Status Variable

Variable Type

● Bit (on/off)

○ Integer (Numeric value)

○ Timer

○ Time Functions

○ List

○ Date & Time

Link To:

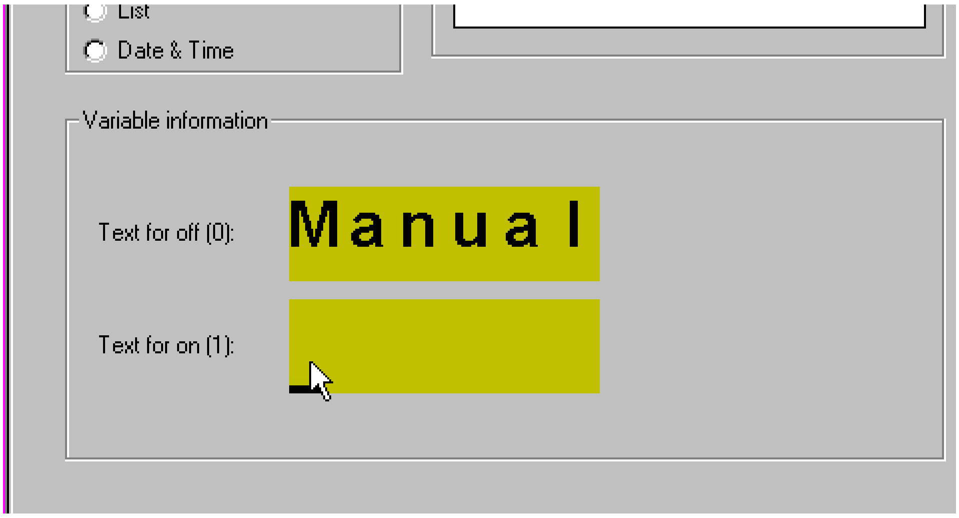
MB 0

Variable information

Text for off (0):

Text for on (1):

3. Enter a text Display for the "0" value of the MB / SB.



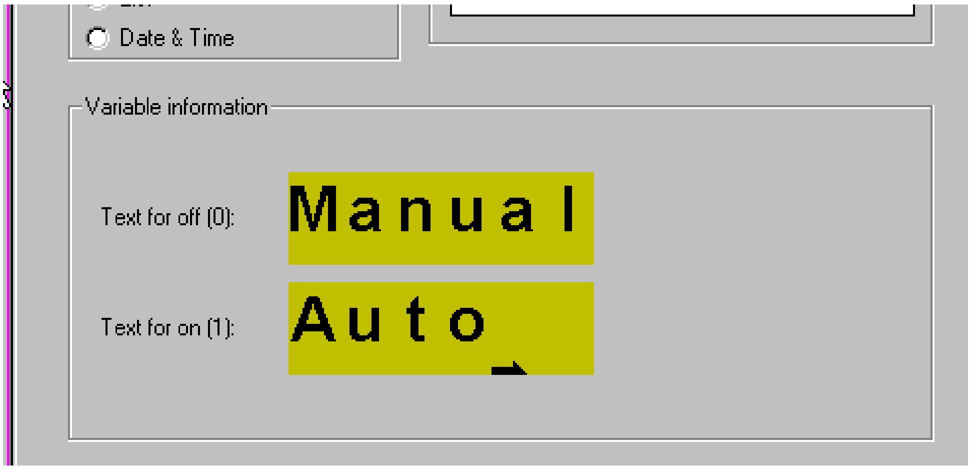
○ List
○ Date & Time

Variable information

Text for off (0): Manual

Text for on (1):

4. Enter a text Display for the "1" value of the MB / SB.



○ Date & Time

Variable information

Text for off (0): Manual

Text for on (1): Auto

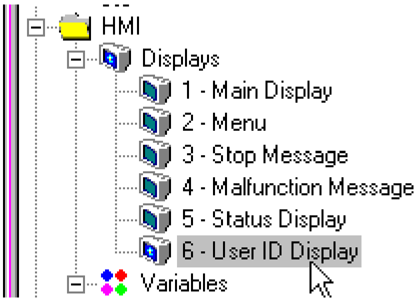
The text will be displayed according to the value of the MB / SB. Note that the Display field must be large enough for the defined text.

For the above example, the Display field must be 6 characters.

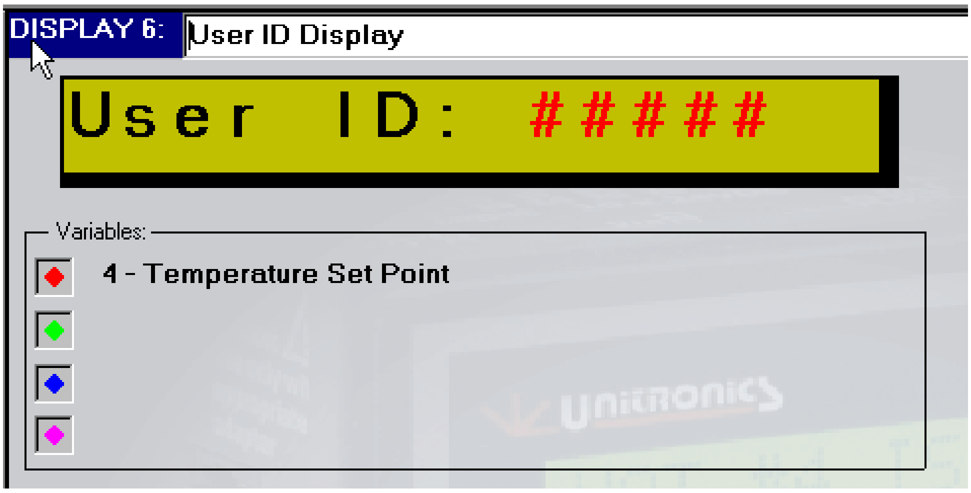
Opening a Variable from a Display

To move quickly from a Display to the Variable linked to the Display:

1. Select the desired Display from the Navigator window.



2. The Display opens in the Display Editor.



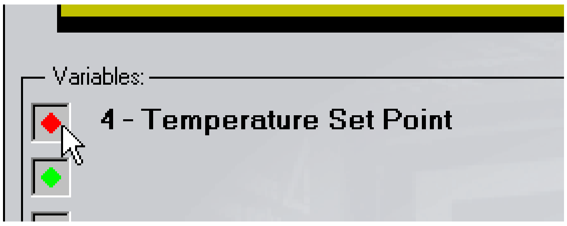
DISPLAY 6: User ID Display

User ID: #####

Variables:

4 - Temperature Set Point

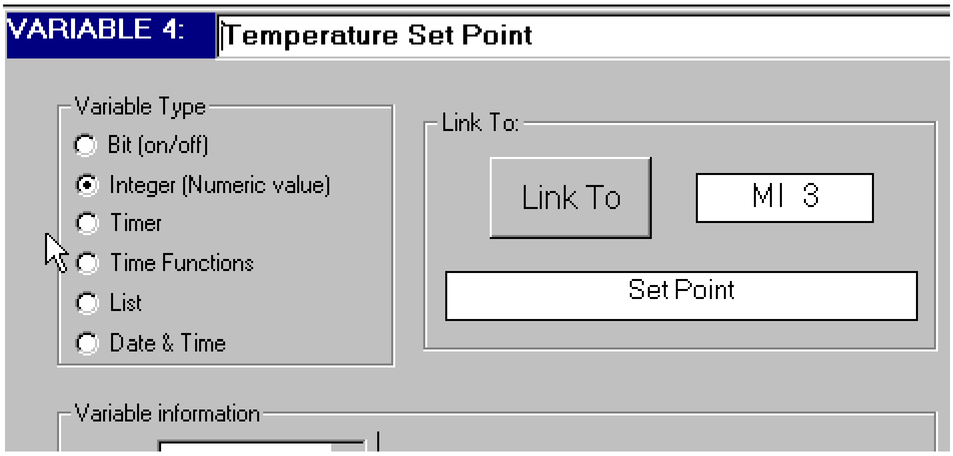
3. Select the Variable.



Variables:

4 - Temperature Set Point

4. The Variable opens in the Variable Editor.



VARIABLE 4: Temperature Set Point

Variable Type
- ○ Bit (on/off)
- ● Integer (Numeric value)
- ○ Timer
- ○ Time Functions
- ○ List
- ○ Date & Time

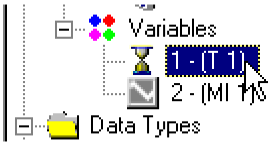
Link To: MI 3

Set Point

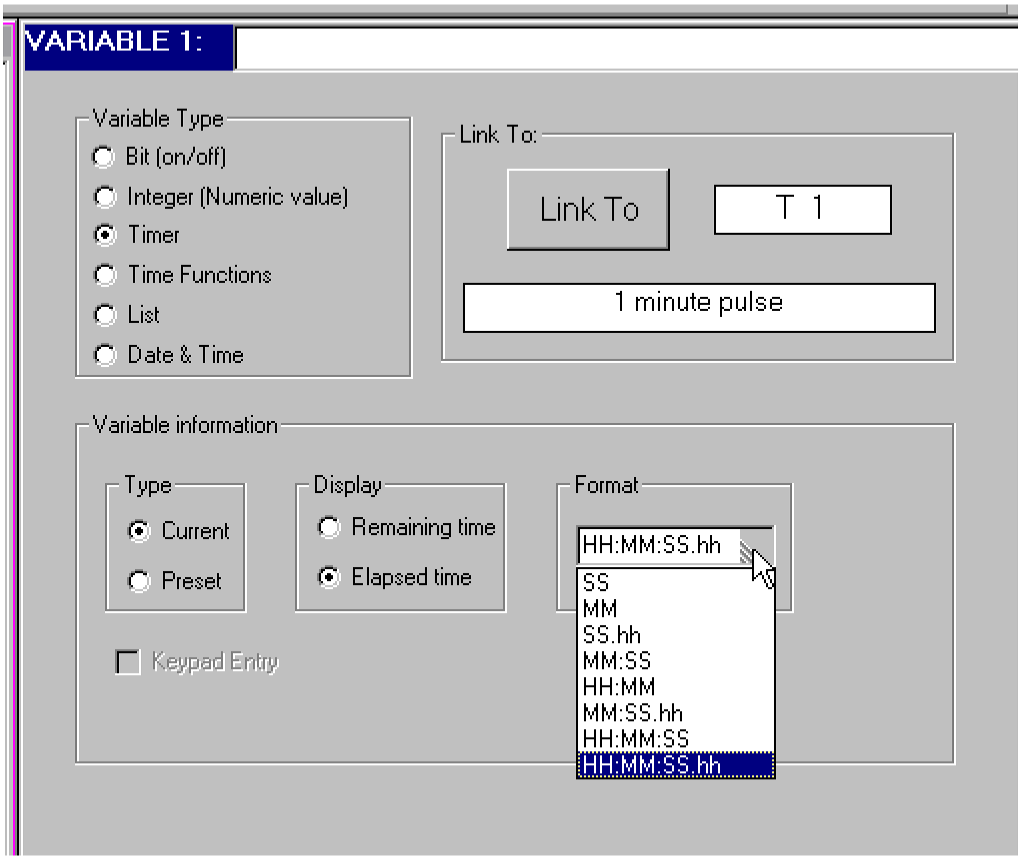
Variable information

Selecting a Timer Display format

1. From the Navigator Window, create or choose an existing Timer Variable.



2. Open the Timer format drop-down menu in the Variable Editor.



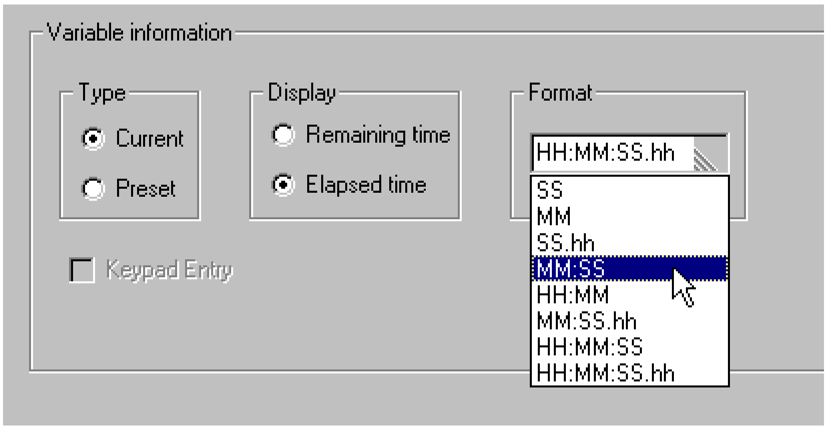
VARIABLE 1:

Variable Type
○ Bit (on/off)
○ Integer (Numeric value)
● Timer
○ Time Functions
○ List
○ Date & Time

Link To:
Link To: T 1
1 minute pulse

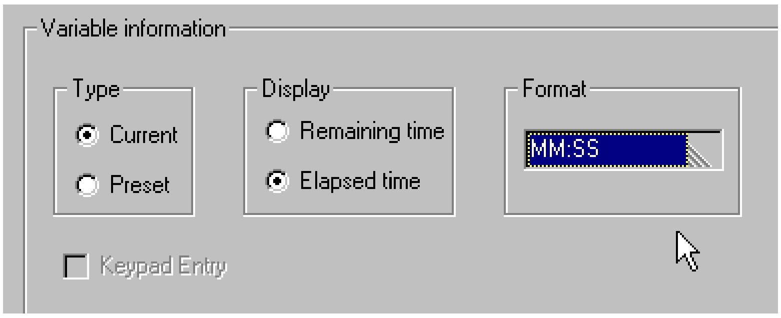
Variable information
Type: ● Current ○ Preset
Display: ○ Remaining time ● Elapsed time
Format: HH:MM:SS.hh
□ Keypad Entry

3. Select the Timer format from the drop-down menu in the Variable Editor.



Variable information
Type: ● Current ○ Preset
Display: ○ Remaining time ● Elapsed time
Format: HH:MM:SS.hh
□ Keypad Entry

4. The selected format is displayed in the Format window.



Variable information
Type: ● Current ○ Preset
Display: ○ Remaining time ● Elapsed time
Format: MM:SS
□ Keypad Entry

Communications

About Communications

You can use the RS232 port of your M90 for several purposes:

- Direct Communications: Your PC is connected to an M90 by the proprietary programming cable that is supplied together with your M90 OPLC.
- Modem Communications: Your PC connects to a remote M90 OPLC via modem.
- To communicate with devices that use the RS232 standard, such as GSM modems for SMS messaging.
- Network communications: You use your PC to access the RS232 port of an M90 that is integrated into an M90 CANbus network. This M90 can act as an RS232-to-CANbus bridge; via this bridge, you can access any M90 in the network.

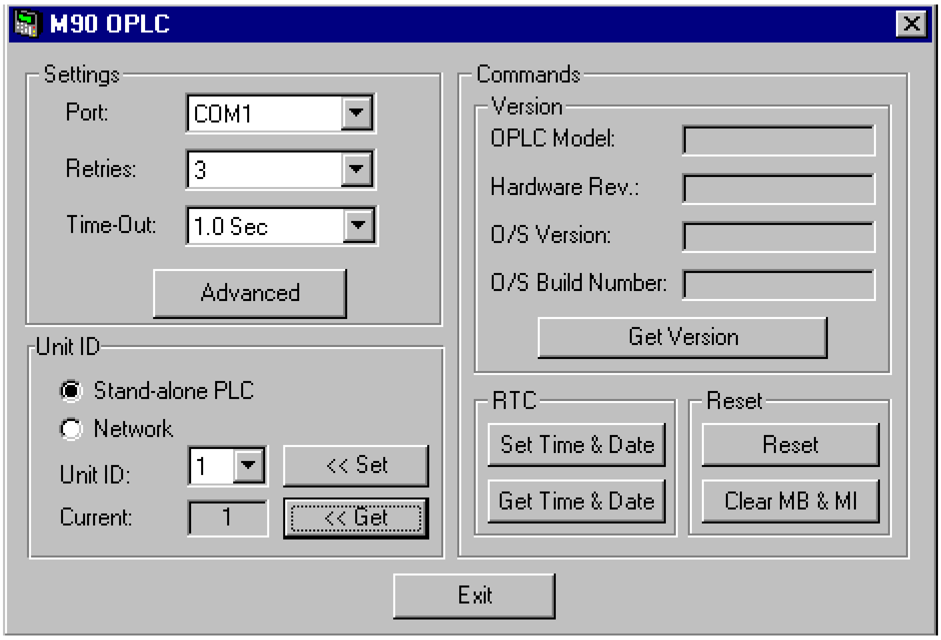
Note that an M90 cannot use both SMS messaging and modem communications.

In addition, you cannot use Direct Communications and Modem Communications simultaneously. If your PC is connected directly with an M90 and you dial a remote M90 via modem, all communications are automatically diverted to the remote unit. You will not be able to access the directly connected M90 until you 'hang up', terminating the call.

If you encounter problems, refer to the Troubleshooting Communications sections in this Help.

M90 Communication Settings

Display the M90's current communication settings by selecting M90 OPLC from the Controller menu. The M90's default communication settings are shown below.



M90 OPLC

Settings

Port: COM1

Retries: 3

Time-Out: 1.0 Sec

Unit ID

● Stand-alone PLC

○ Network

Unit ID: 1

Current: 1

M90 OPLC

Settings

Port, Retries and Time-Out are the communication settings between U90 Ladder and the M90. Click the Advanced button to view the advanced RS232 parameters.
The other settings in this box belong to your project, and relate to the M90.

Unit ID

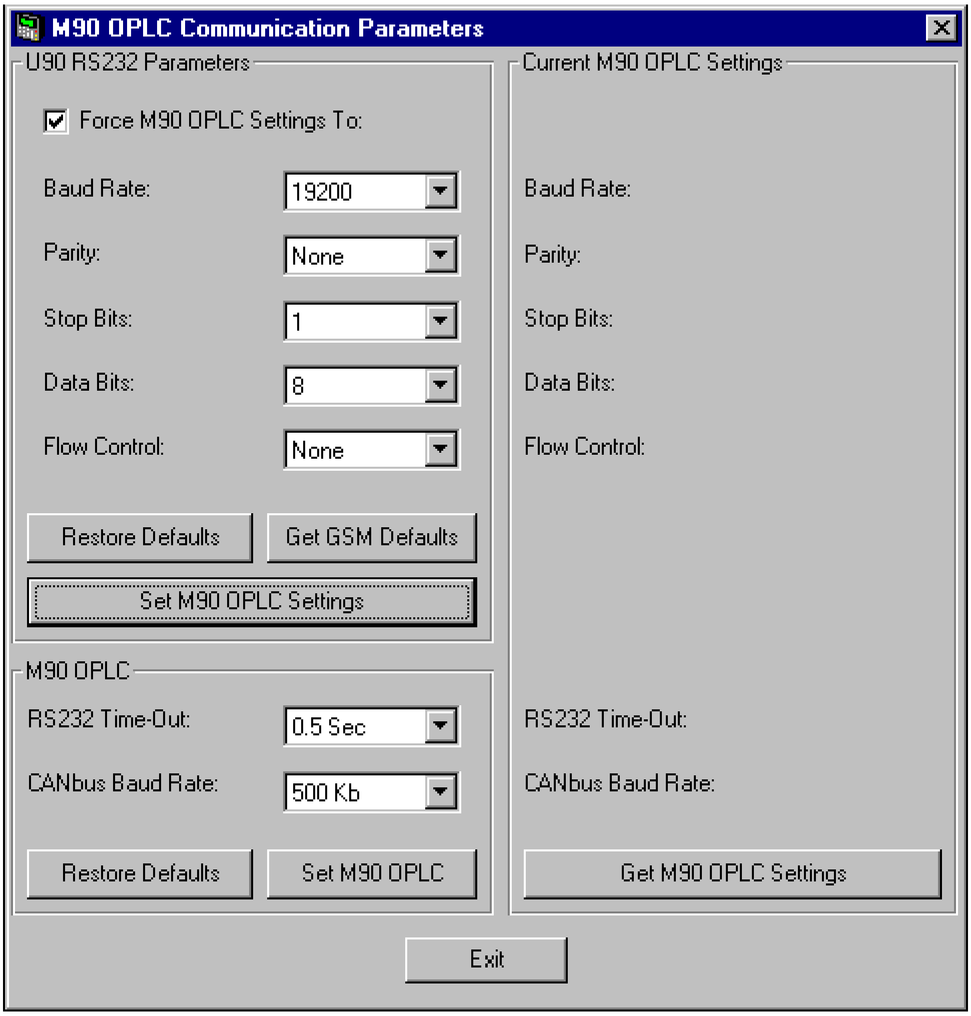
Note that by default, projects are defined as 'Stand-alone'. If you want to integrate your M90 into a network, you must define the M90 as a member of a network and assign it an ID number.
Click Get to retrieve the ID number of a directly connected M90.
Click Set to change the ID number.

Commands

To display information about the M90 unit connected to your PC, whether directly connected or within a network, click Get Version.
You can also view the current time and date settings within the M90 by clicking Get Time & Date, or import your PC's settings by clicking Set Time & Date.
You can also click on Reset to initialize the M90, and click on Clear MB & MI to initialize values.

Advanced Settings

Click on Advanced. The M90 OPLC Communication Parameters box opens as shown below.



M90 OPLC Communication Parameters

U90 RS232 Parameters

☑ Force M90 OPLC Settings To:

Baud Rate: 19200

Parity: None

Stop Bits: 1

Data Bits: 8

Flow Control: None

Current M90 OPLC Settings

Baud Rate:

Parity:

Stop Bits:

Data Bits:

Flow Control:

M90 OPLC

RS232 Time-Out: 0.5 Sec

CANbus Baud Rate: 500 Kb

RS232 Time-Out:

CANbus Baud Rate:

U90 RS232 Parameters

These settings are part of your U90 project. If you need to modify the default settings, click on the arrows to reveal the options.
If this project is defined to 'Use Modem', we recommend that you change these settings to match the settings of the modem. If this project is defined as 'Use SMS', we recommend that you enter the settings of the GSM modem.

Force M90 OPLC Settings To:

This is checked by default, making the settings that you have selected become part of your U90 Ladder project. These settings will be installed in the M90 whenever communications are activated, overwriting the previous settings.

Restore Defaults

Click this to restore defaults

Get GSM Defaults

Click this to enter the settings used to communicate with standard GSM modems.

Set M90 OPLC Settings

Click here to write your selected settings into the M90.

Advanced

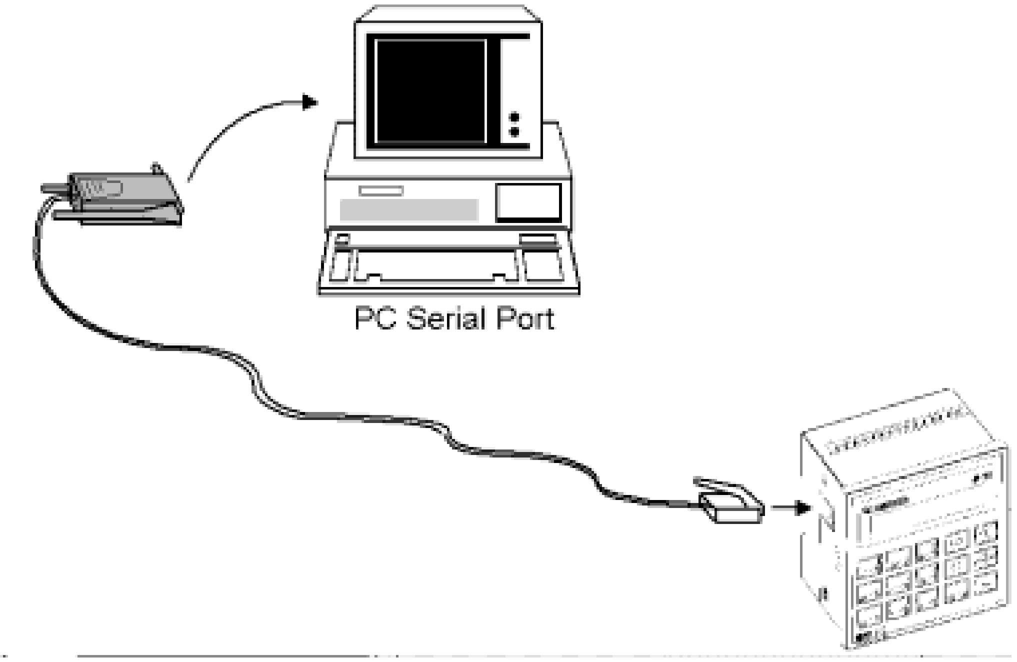
RS232 Time-Out settings may be edited.
Make sure that the CANbus baud rate is the same for all networked M90 units.

Current M90 OPLC Settings

Click Get M90 OPLC Settings to retrieve the settings of the M90 unit to which you are directly connected.
Note that this option does not work if you have defined the project as a network project.

Direct Communications- PC to M90

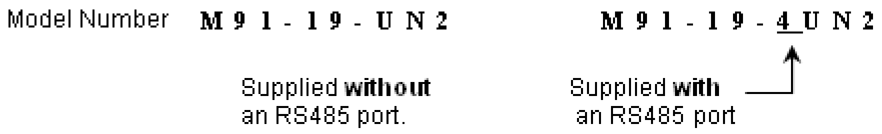
Direct Communications: when your PC is connected to an M90 by the proprietary programming cable that is supplied together with your M90 OPLC as shown below.



COM Port Mode: RS232/RS485 (M91 only)

Certain OPLCs can be ordered with an RS485 port. Within the controller, the jumper settings determine the COM port function according to RS232 or RS485; RS485 termination settings are also determined via jumper.

To check if your controller was supplied with an installed RS485 port, check the device's model number.


Model Number
M 9 1 - 1 9 - U N 2
M 9 1 - 1 9 - 4 U N 2
Supplied **without** an RS485 port.
Supplied **with** an RS485 port

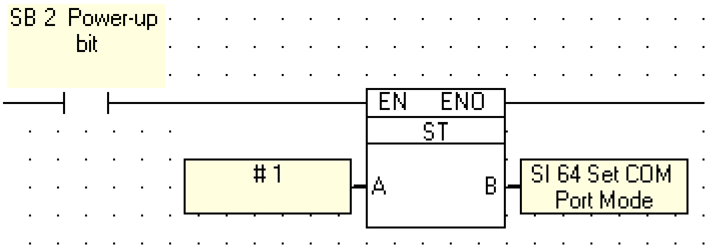
For more information regarding hardware COM settings, check the documentation M91 RS485 Port Settings.

Setting the COM Port Mode

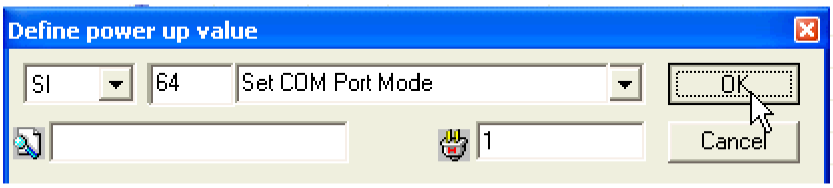
The value of SI 64, Set COM Port Mode, determines if the port will function according to RS232 or RS485. When SI 64 contains 0, the port is set to RS232, when SI 64 contains 1, the port is set to RS485.

The value in SI can only be changed via Power-up, whether via the Ladder application or by setting a Power-up value for SI 64.

Changing Mode via Ladder



Changing Mode via Power-up Value



SI 64 Set COM Port Mode

1

Note ♦ In order to change the mode, **Power-up must take place**; as for example if the power cable is temporarily disconnected.

♦ By factory default, SI 64 contains 0.

♦ When a port is set to RS485, both RS232 and RS485 can be used simultaneously if flow control signals DTR and DSR are not used.

Modems

About Modems

You can use the M90 with either a PSTN modem or a cellular modem. When you use a cellular GSM modem, you can also program the M90 to both send and receive SMS messages from a GSM cellular phone.

Modems may be used to communicate data; they may also be used to download and upload applications from remote M90 devices to a PC.

Note ♦ The PC-modem cable is not the same type of cable used to connect between the controller and the modem. Ensure that the cable used to connect the PC to the modem provides connection points for all of the modem's pins.
♦ If call are routed via a switchboard, note that the switchboard settings may interfere with communications. Consult with your switchboard provider.
♦ **PC/PLC modem communications**: Both PC and controller must use the same type of modem: either PSTN or GSM. Internal modems must be used in conjunction with the driver provided by the modem's manufacturer.

For advanced users, check: How the M90 works with a modem.

Configuring my PC's modem

You can configure your PC's modem to dial an M90's modem. Via a PC-modem-to-M90-modem connection, you can:

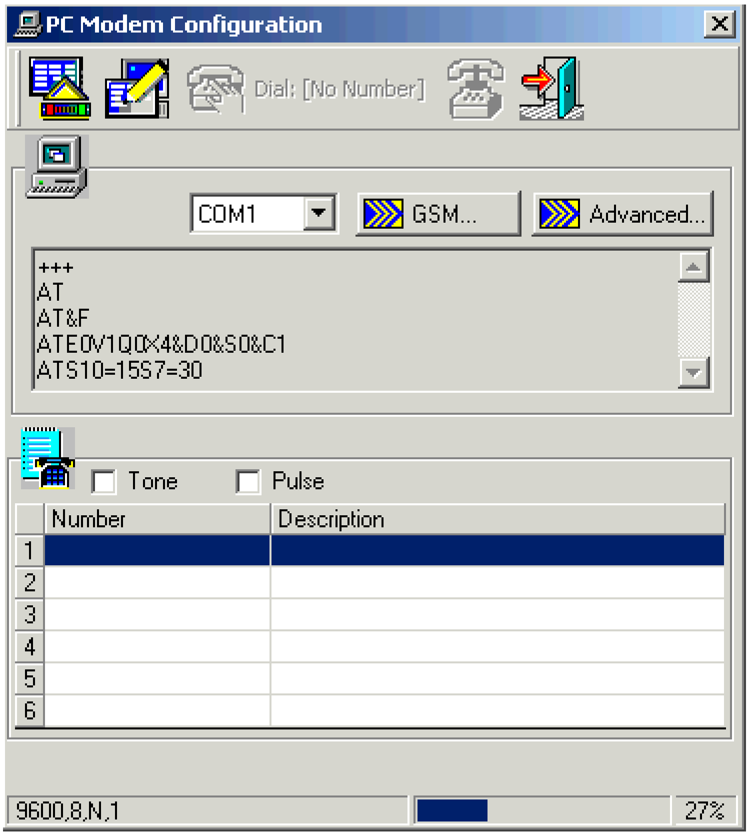
- Download and upload applications
- Test and troubleshoot problems in remote M90 units and applications.

Note ♦ PC-to-M90 communications are via Direct Com. This means that PC modem installation procedures are not necessary.

Configuring your PC's modem

1. Display the PC Modem Configuration box by selecting M90 OPLC Settings from the Controller menu, then clicking on the Modem Setup button.

   Note that the default port setting for internal PC modems is commonly COM 3 or COM 4. Most modems automatically match the parameters of incoming data: baud rate, data bits, parity & stop bits. The U90 Ladder fixed modem settings are: 9600, 8 data bits, no parity, 1 stop bit. You may need to manually change your modem's communication settings to match these.
   You can also select a GSM modem by clicking the GSM button and selecting a modem type.



```
+++
AT
AT&F
ATE0V1Q0X4&D0&S0&C1
ATS10=15S7=30
```

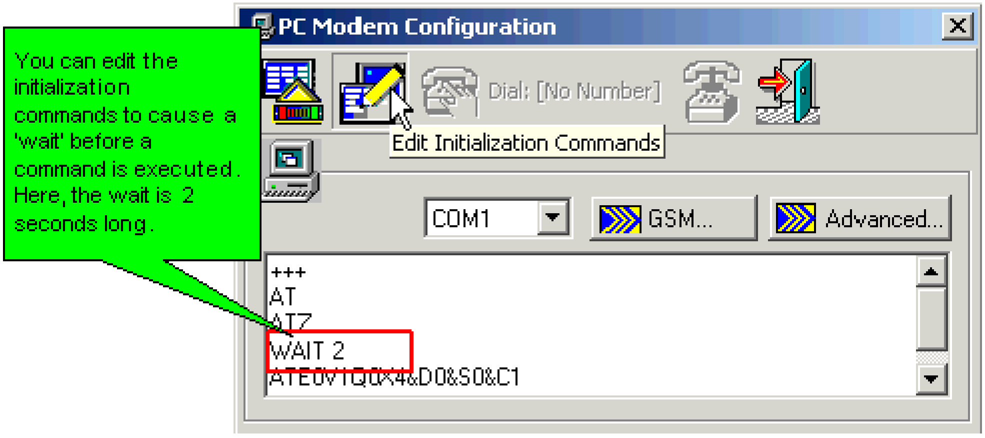
	Number	Description
1		
2		
3		
4		
5		
6		

9600,8,N,1 27%

The default modem initialization commands that appear here are standard for most modems. If your modem requires different commands, you can edit them.

2. To edit initialization commands, click on the Edit Initialization Commands button shown below. The window containing the commands turns white; you can now add, delete or edit commands.

   Note that you can restore the default commands by clicking the Default Initialization button.

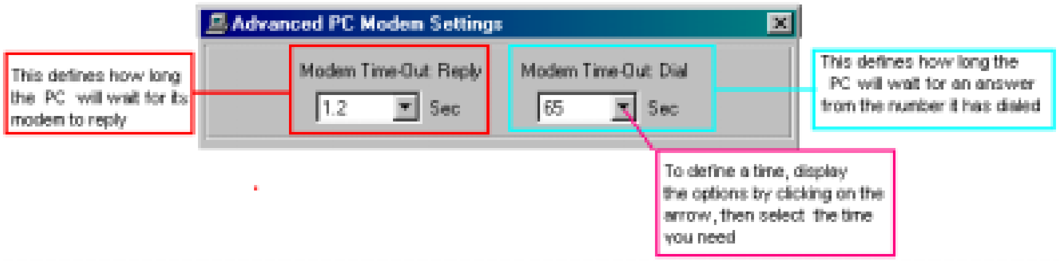


You can edit the initialization commands to cause a 'wait' before a command is executed. Here, the wait is 2 seconds long.

```
+++
AT
ATZ
WAIT 2
ATE0V1Q0X4&D0&S0&C1
```

3. Select whether to use pulse or tone dialing, as is required by the system, by clicking on the appropriate box. You can also leave both blank (default).

4. Click the Advanced button to edit Time-Out settings.



Advanced PC Modem Settings

Modem Time-Out Reply: 1.2 Sec

Modem Time-Out Dial: 65 Sec

This defines how long the PC will wait for its modem to reply

This defines how long the PC will wait for an answer from the number it has dialed

To define a time, display the options by clicking on the arrow, then select the time you need

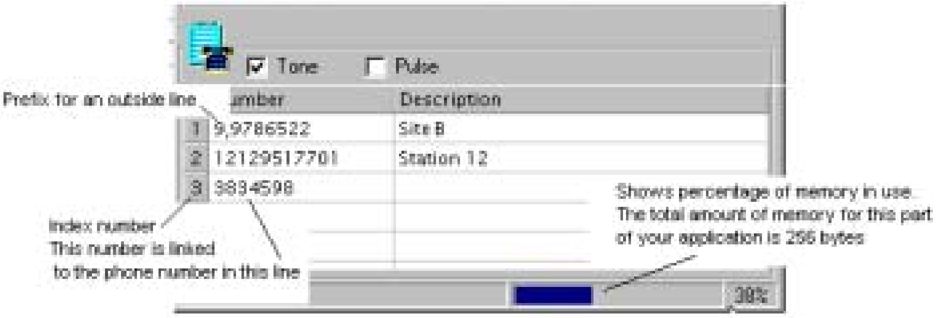
Phone Book

The Phone Book is where you define the list of numbers that the PC can dial. You can enter up to six numbers. Each phone number is automatically linked to an index number. Each phone number can be up to 18 characters long. You can also add a description to identify the location or other details of the number to be dialed.

Entering numbers in the Phone Book

1. Click on an empty line in the Phone Book, then type in the number, **exactly** as you would dial from a standard phone, including area codes. To dial an outside line, enter the prefix number required and follow it with a comma as shown below.

   This comma causes the short pause, or delay, that is required by many systems.



☑ Tone □ Pulse

	umber	Description
1	9,9786522	Site B
2	12129517701	Station 12
3	3834598	

38%

Prefix for an outside line

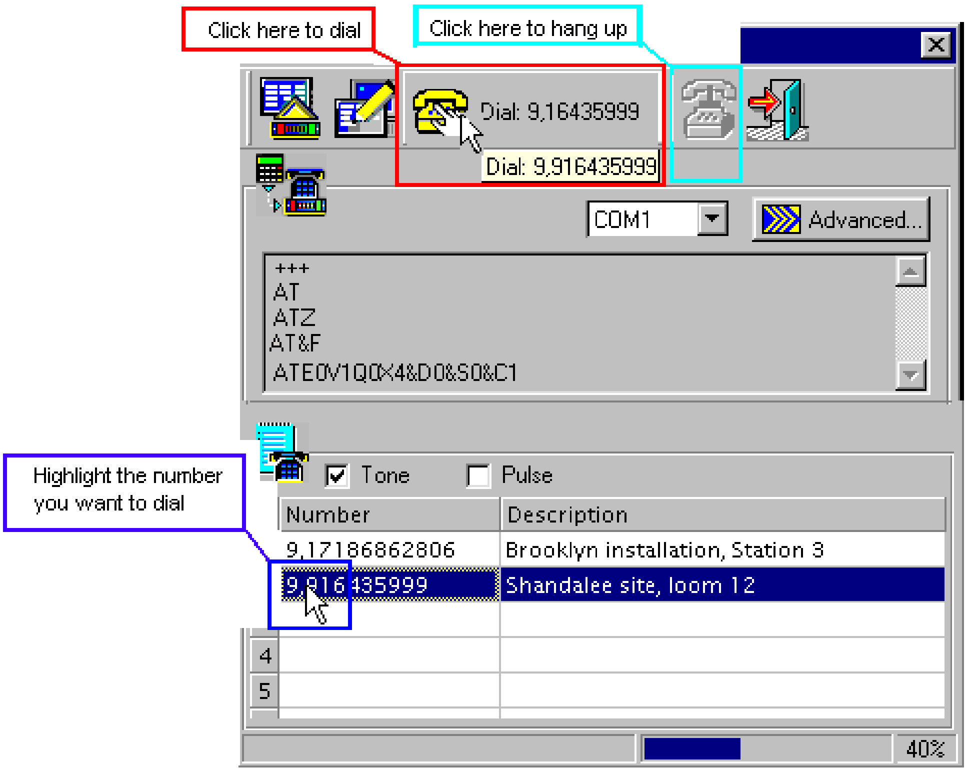
Index number
This number is linked
to the phone number in this line

Shows percentage of memory in use.
The total amount of memory for this part
of your application is 256 bytes

To edit the phone book, click in a number or description, then make your changes.

Dialing a remote M90

1. To dial, highlight the number you want to dial, then click on the Dial button as shown below.



Click here to dial

Click here to hang up

Highlight the number you want to dial

```
+++
AT
ATZ
AT&F
ATE0V1Q0X4&D0&S0&C1
```

☑ Tone ☐ Pulse

	Number	Description
	9,17186862806	Brooklyn installation, Station 3
	9,916435999	Shandalee site, loom 12
4		
5		

Note that this Phone Book is used only by the PC's modem, although it is similar in appearance to the M90's Phone Book.

Communication Log

When you dial a remote modem using U90 Ladder, a window opens up in the bottom of your screen. The log of events is quickly displayed in this window. This log is stored as a .txt file. You can view this log by navigating to the U90 folder and opening a file named U90ldxxx.txt.

This log is stored as a .txt file. You can view this log by navigating to Unitronics\U90_Ladder\U90Ldxxx and opening a file named ComLog.txt.

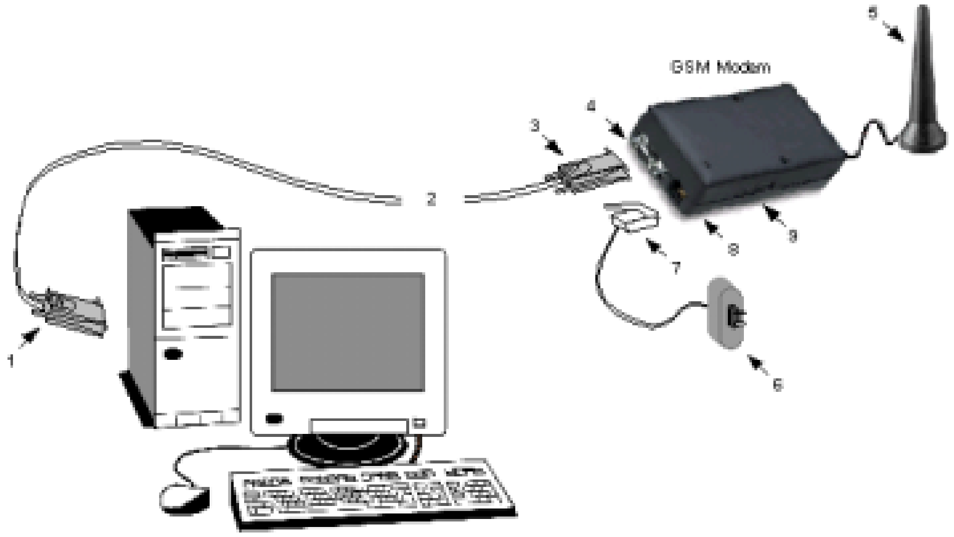
In this file, the most recent log of events appears **last**.

Note ♦ The PC-modem cable is not the same type of cable used to connect between the controller and the modem. Ensure that the cable used to connect the PC to the modem provides connection points for all of the modem's pins.
♦ If call are routed via a switchboard, note that the switchboard settings may interfere with communications. Consult with your switchboard provider.
♦ **PC/PLC modem communications**: Both PC and controller must use the same type of modem: either landline or GSM.
♦ Internal PC modems must be used in conjunction with the driver provided by the modem's manufacturer.

Using a PC to access an M90 via GSM modem

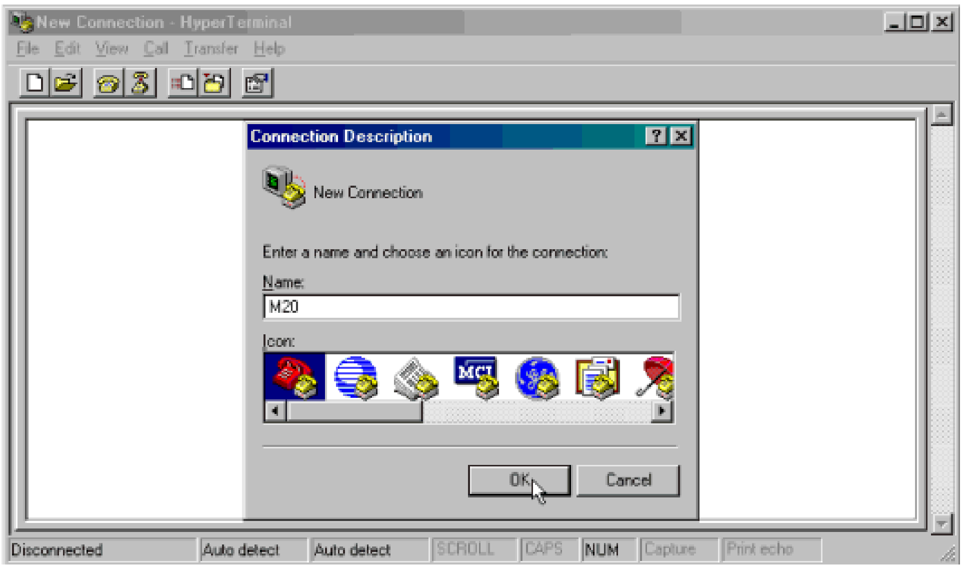
To use a PC running U90 Ladder to access a remote M90 OPLC for programming and maintenance via GSM networks:

1. Connect your M90 to the GSM modem according to the instructions supplied with the GSM Modem Kit.
2. Connect your PC to the GSM modem.



1	RS232 connector
2	RS232 cable MJ10-22-CS28 (available by separate order)
3	RS232 connector
4	GSM Modem serial port
5	GSM antenna
6	Power supply PS-GSM modem (available by separate order)
7	RJ11 connector
8	GSM modem power supply
9	SIM card drawer

3. U90 Ladder's modem communication rate is set at 9600 bps. To enable the modem to communicate with U90 Ladder, change the modem's default communication rate from 19200 bits per second (bps) to 9600 bps via Hyperterminal.
    1. Open Hyperterminal. The program can generally be located by clicking the Start button in the lower left corner of your screen, then selecting Programs>Accessories>Communications>Hyperterminal. The New Connection window opens as shown below.
       Note ♦ Hyperterminal generally starts by pointing to the internal modem, if one is installed on the PC.



New Connection - HyperTerminal

Connection Description

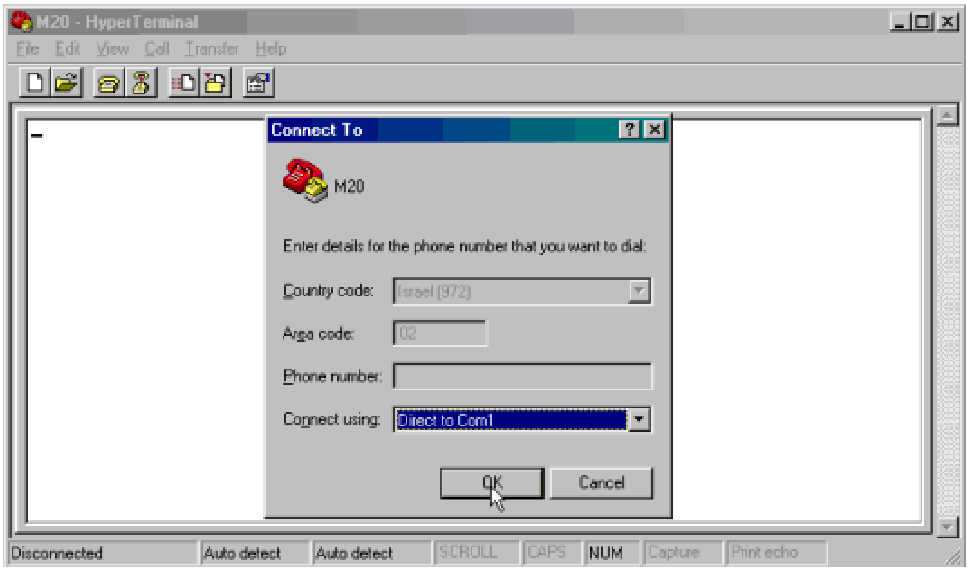
New Connection

Enter a name and choose an icon for the connection:

Name: M20

Icon:

2. Enter a name for the new connection and select an icon, and then click OK. The Connect To box opens.
3. Select a COM port for the modem, and then click OK.



M20 - HyperTerminal

Connect To

M20

Enter details for the phone number that you want to dial:

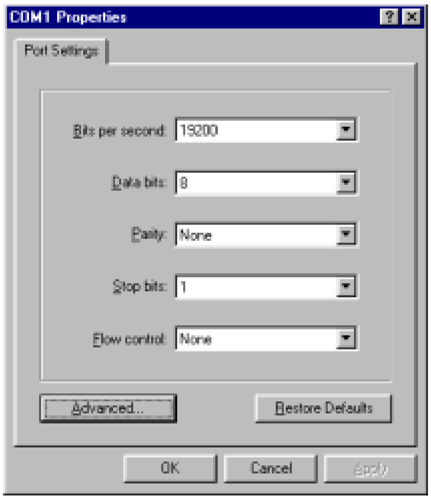
Country code: Israel (972)

Area code: 02

Phone number:

Connect using: Direct to Com1

4. The Port Settings box opens as shown below. To enable your PC to communicate with the modem, set the COM port parameters to a BPS of either 9600 or 19200, Data bits=8, Parity=N, Stop bits=1, Flow control=None, and then click OK.



COM1 Properties

Port Settings

Bits per second: 19200

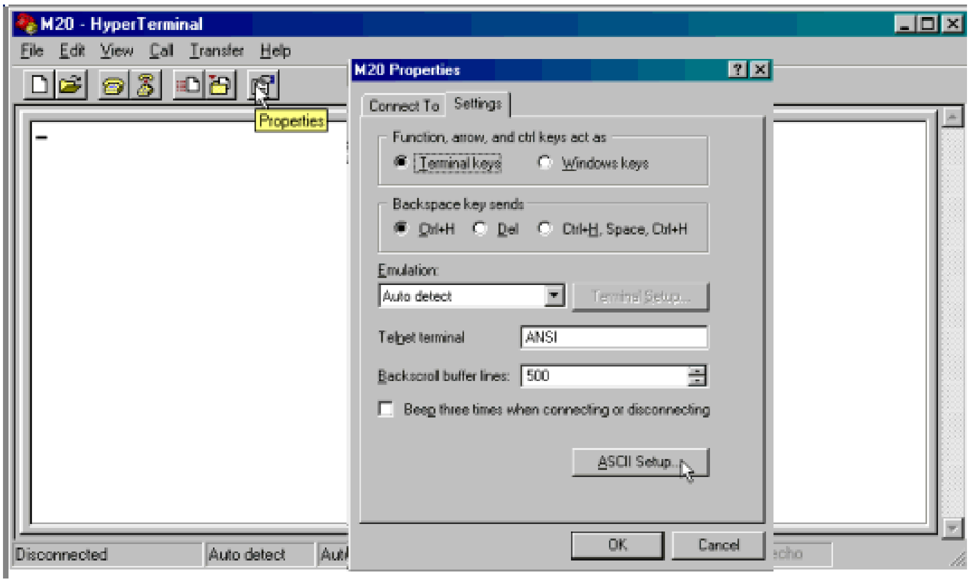
Data bits: 8

Parity: None

Stop bits: 1

Flow control: None

5. Open the modem’s Properties box by clicking on the Properties button, then open ASCII Setup.



M20 - HyperTerminal

M20 Properties

Function, arrow, and ctrl keys act as: ● Terminal keys ○ Windows keys

Backspace key sends: ● Ctrl+H ○ Del ○ Ctrl+H, Space, Ctrl+H

Emulation: Auto detect

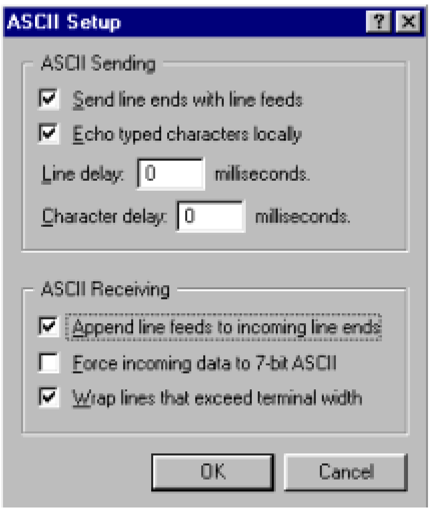
Telnet terminal: ANSI

Backscroll buffer lines: 500

□ Beep three times when connecting or disconnecting

Disconnected | Auto detect

6. Select the options shown below, and then click OK.



ASCII Setup

ASCII Sending

☑ Send line ends with line feeds

☑ Echo typed characters locally

Line delay: 0 milliseconds.

Character delay: 0 milliseconds.

ASCII Receiving

☑ Append line feeds to incoming line ends

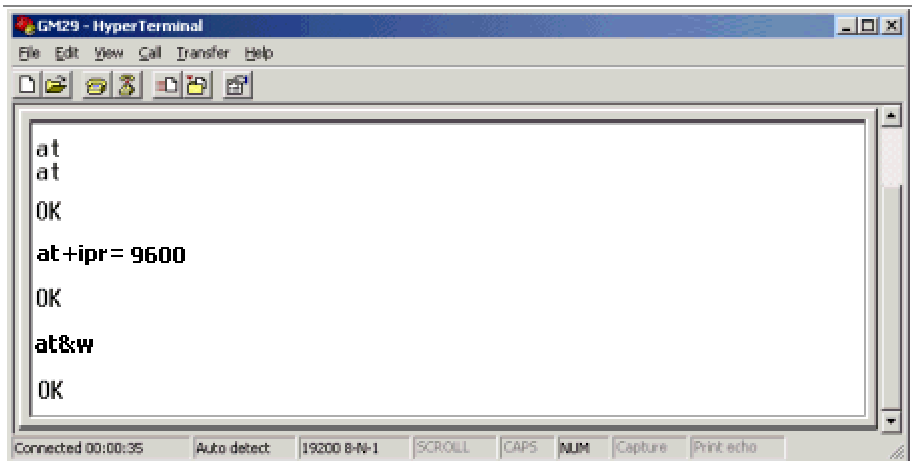
□ Force incoming data to 7-bit ASCII

☑ Wrap lines that exceed terminal width

Hyperterminal is now connected to your PC via Com 1; the ASCII settings now enable you to enter commands via the PC keyboard and see the replies from the modem within the Hyperterminal window.

To test the connection, type AT; if the connection is valid the modem will respond 'OK'.

To change the modem's baud rate, type AT+IPR=9600&W; the command '&W' burns the new baud rate into the modem's non-volatile memory.

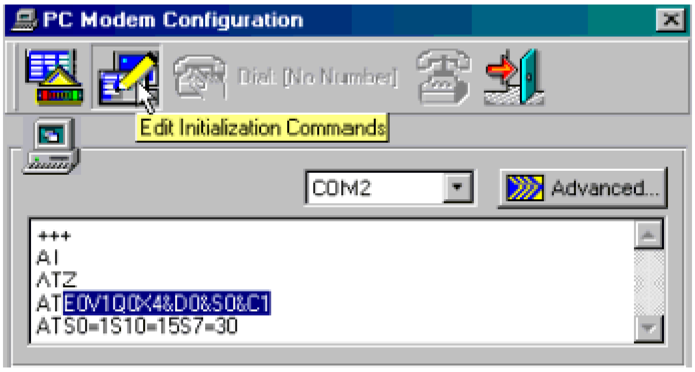


```
at
at
OK
at+ipr= 9600
OK
at&w
OK
```

You can reset the modem's communication rate by returning to this window and typing AT+IPR=19200&W.

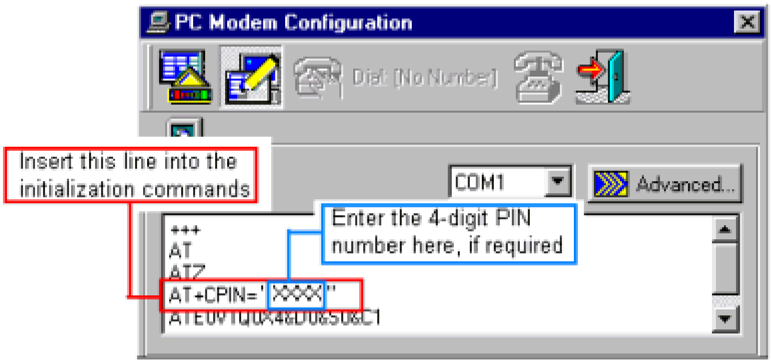
4. Configure U90 Ladder's modem initialization commands.

   1. Start U90 Ladder. Open the PC Modem Configuration box by selecting PC Modem Configuration from the Controller menu.
      To enable U90 Ladder to communicate with the GSM modem, you must edit the initialization commands.
   2. Access the initialization commands by clicking on the Edit Initialization Commands button shown below. The window containing the commands turns white; you can now edit commands.



```
+++
AI
ATZ
ATE0V1Q0X4&D0&S0&C1
ATS0=1S10=15S7=30
```

   3. If you are using a SIM card that has a PIN number, enter a new initialization command AT+CPIN="XXXX", where XXXX is the 4-digit PIN #.



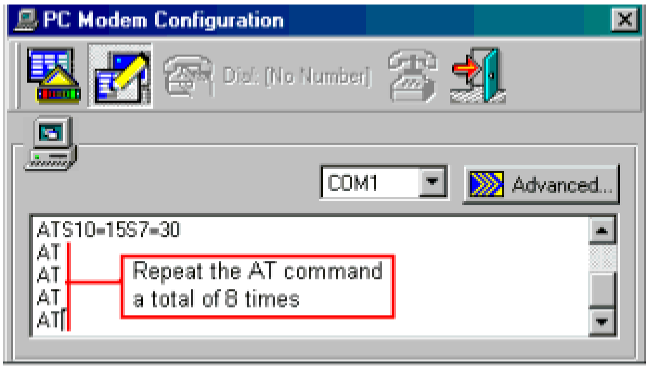
COM1

```
+++
AT
ATZ
AT+CPIN="XXXX"
ATE0V1Q0X4&D0&S0&C1
```

Insert this line into the initialization commands

Enter the 4-digit PIN number here, if required

4. End the list of commands by entering the AT command eight times as shown below.

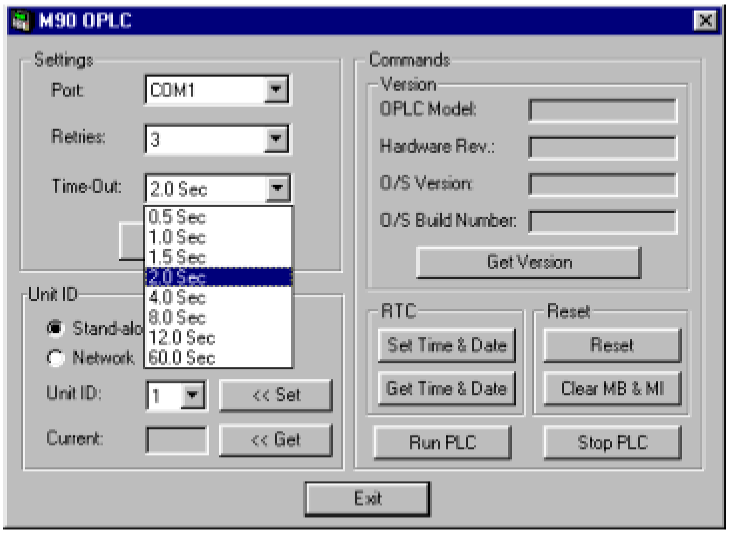


COM1

```
ATS10=15S7=30
AT
AT
AT
AT
```

Repeat the AT command a total of 8 times

5. After you have made these changes, close the PC Modem Configuration box.
6. Open the M90 OPLC box by selecting M90 OPLC from the Controller menu.
7. Set the M90 OPLC's Time-Out to 2 seconds as shown below. This should allow sufficient time for PC-to-M90 communications via the GSM modem.



Settings

Port: COM1

Retries: 3

Time-Out: 2.0 Sec

0.5 Sec
1.0 Sec
1.5 Sec
2.0 Sec
4.0 Sec
8.0 Sec
12.0 Sec
60.0 Sec

Unit ID

● Stand-alo

○ Network

Unit ID: 1

Current:

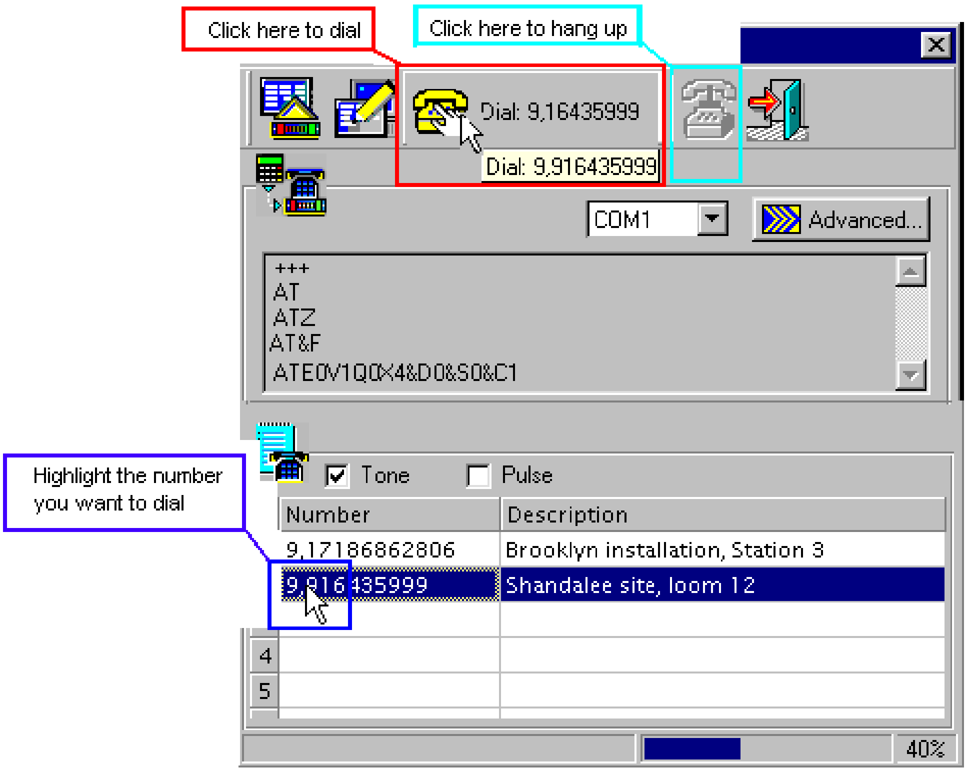
OPLC Model:

Hardware Rev.:

O/S Version:

O/S Build Number:

5. Dial the remote M90 modem from your PC.



Click here to dial

Click here to hang up

Highlight the number you want to dial

```
+++
AT
ATZ
AT&F
ATE0V1Q0X4&D0&S0&C1
```

☑ Tone □ Pulse

	Number	Description
	9,17186862806	Brooklyn installation, Station 3
	9,916435999	Shandalee site, loom 12
4		
5		

40%

Note ♦ Both GSM modems must contain SIM cards capable of data transfer. Check with your SIM card supplier to see if your SIM card is capable of data transfer.

♦ Note that only 3V SIM cards can be used with the GSM modem supplied with the Unitronics' GSM Modem Kits.

Configuring the M90 to use a modem

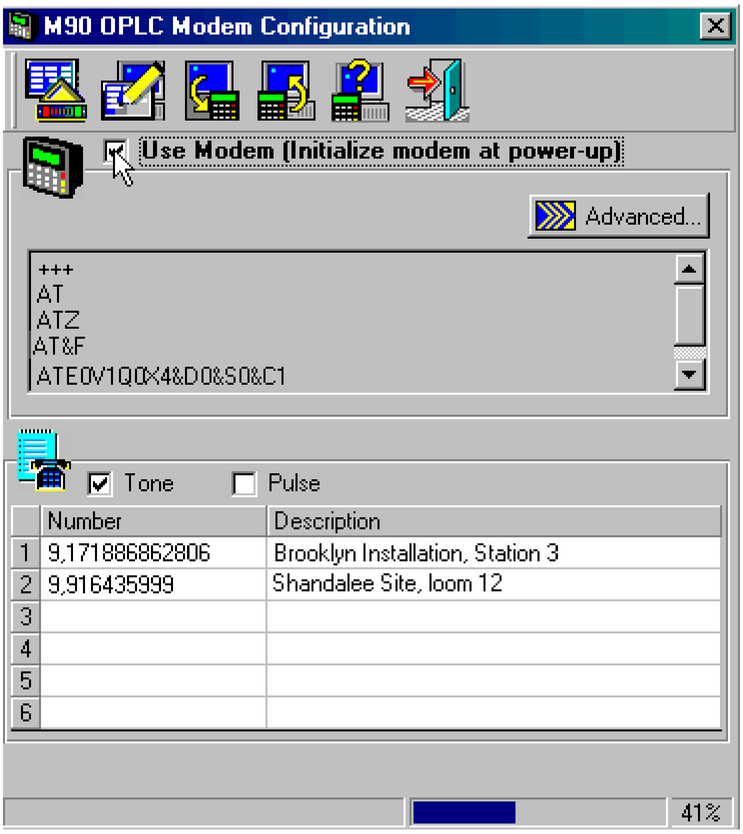
The M90 can use a modem to send and receive calls. A programmer can also use a PC's modem to communicate with a remote M90 that is connected up to a modem.

Most modems automatically match the parameters of incoming data: baud rate, data bits, parity & stop bits. The M90's embedded modem settings are: 9600, 8 data bits, no parity, 1 stop bit. You may need to manually change your modem's communication settings to match these.

Note ♦ If call are routed via a switchboard, note that the switchboard settings may interfere with communications. Consult with your switchboard provider.
♦ **PC/PLC modem communications**: Both PC and controller must use the same type of modem: either landline or GSM. Internal PC modems must be used in conjunction with the driver provided by the modem's manufacturer.

M90 modem configuration

1. Open the M90 Modem Configuration box by selecting Modem Configuration from the Controller menu.
2. To enable the M90 to use a modem, check the 'Use Modem' box shown below. This causes the M90 to automatically turn on SB 72, Initialize Modem, at power-up.



M90 OPLC Modem Configuration

☑ Use Modem (Initialize modem at power-up)

Advanced...

```
+++
AT
ATZ
AT&F
ATE0V1Q0X4&D0&S0&C1
```

☑ Tone □ Pulse

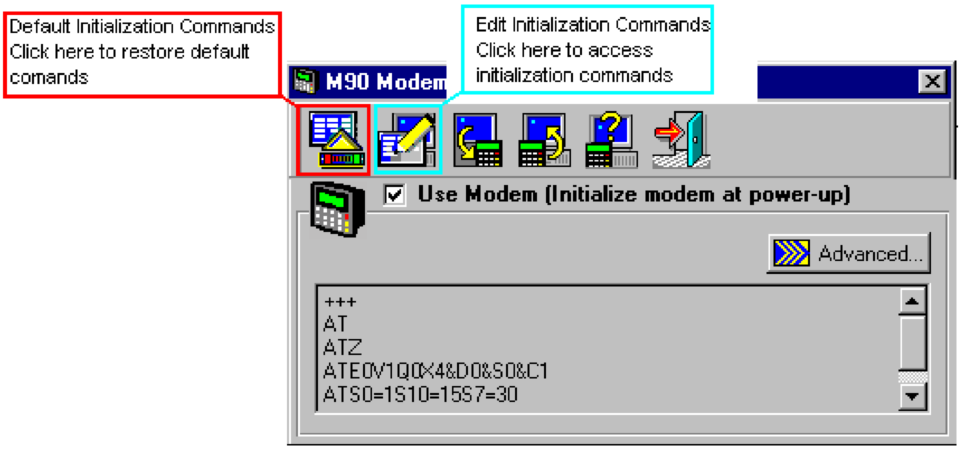
	Number	Description
1	9,171886862806	Brooklyn Installation, Station 3
2	9,916435999	Shandalee Site, loom 12
3		
4		
5		
6		

41%

The default modem initialization commands that appear above are standard for most modems. If your modem requires different commands, you can edit them.

3. To edit initialization commands, click on the Edit Initialization Commands button shown below.
4. The window containing the commands turns white; you can now add, delete or edit commands.

   Note that you can restore the default commands by clicking the Default Initialization button.



Default Initialization Commands
Click here to restore default comands

Edit Initialization Commands
Click here to access initialization commands

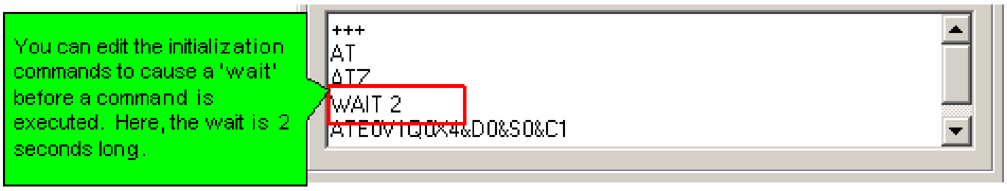
M90 Modem

☑ Use Modem (Initialize modem at power-up)

Advanced...

```
+++
AT
ATZ
ATE0V1Q0X4&D0&S0&C1
ATS0=1S10=15S7=30
```

You can also enter a wait command.



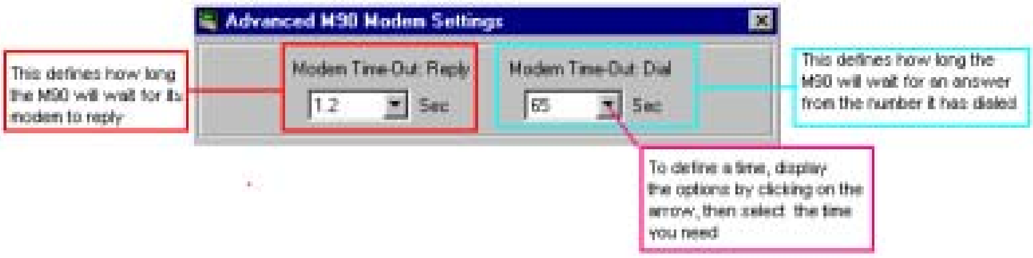
You can edit the initialization commands to cause a 'wait' before a command is executed. Here, the wait is 2 seconds long.

```
+++
AT
ATZ
WAIT 2
ATE0V1Q0X4&D0&S0&C1
```

5. Select whether to use pulse or tone dialing, as is required by your system, by clicking on the appropriate box. You can also leave both blank (default).

   You can also edit the modem's time-out settings:

6. Display the Modem Time-out settings by clicking the Advanced button. Set the appropriate times as shown below.



This defines how long the M90 will wait for its modem to reply

Modem Time-Out: Reply 1.2 Sec

Modem Time-Out: Dial 65 Sec

This defines how long the M90 will wait for an answer from the number it has dialed

To define a time, display the options by clicking on the arrow, then select the time you need

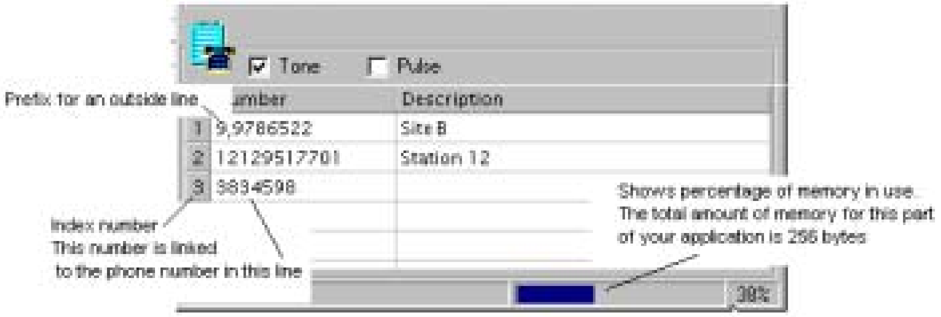
Phone Book

The Phone Book is where you define the list of numbers that the M90 can dial. You can enter up to six numbers. Each phone number is automatically linked to an index number. Each phone number can be up to 18 characters long. You can also add a description to identify the location or other details of the number to be dialed.

Entering numbers in the Phone Book

1. Click on an empty line in the Phone Book, then type in the number, **exactly** as you would dial from a standard phone, including area codes. To dial an outside line, enter the prefix number required and follow it with a comma as shown below.

   This comma causes the short pause, or delay, that is required by many systems.



☑ Tone ☐ Pulse

	umber	Description
1	9,9786522	Site B
2	12129517701	Station 12
3	3834598	

Prefix for an outside line

Index number
This number is linked
to the phone number in this line

Shows percentage of memory in use.
The total amount of memory for this part
of your application is 256 bytes

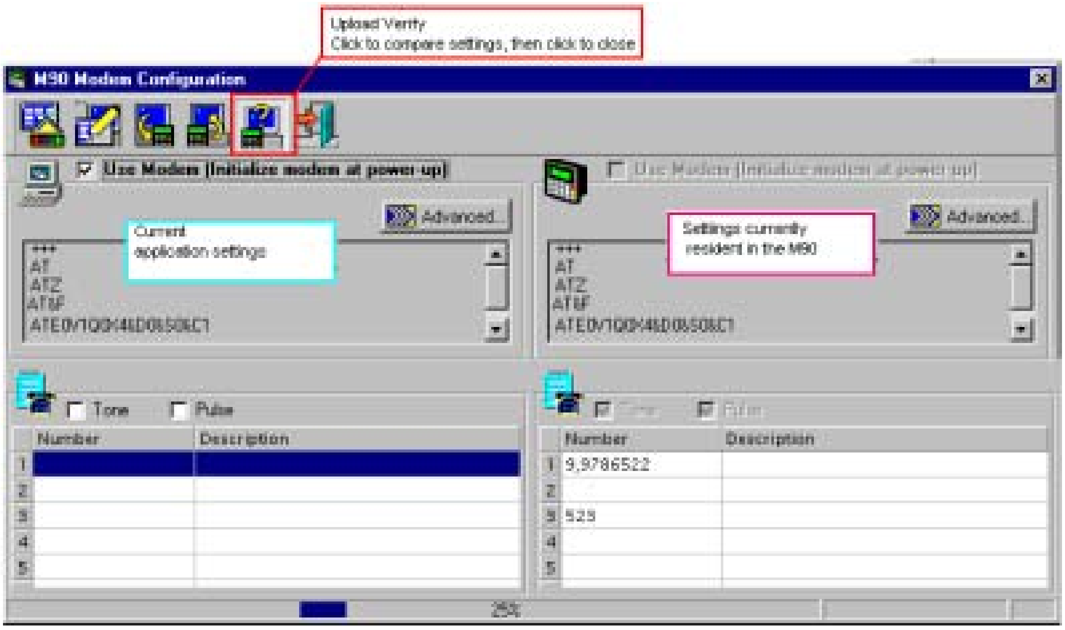
To edit the phone book, click in a number or description, then make your changes.

Downloading, uploading, and comparing settings

You download modem settings **to** the M90 by clicking the Download button on the tool bar. You can also upload settings **from** the M90 by clicking the Upload button. Note that downloading overwrites any settings that may already be in the M90; uploading settings overwrites any settings that you have made in your application.

You can **compare** your application's modem settings to the settings that are already within the M90--before downloading or uploading:

1. Display the settings for both your application and the M90 by clicking on the Upload Verify button.
2. Two windows open. The left window shows the settings you have set in your application; you can edit these settings. The right window shows the current settings within the M90; these are read-only.


Upload Verify
Click to compare settings, then click to close
M90 Modem Configuration
Current application settings
Settings currently resident in the M90
Advanced...
Advanced...
25%

☑ Use Modem (Initialize modem at power-up)

```
+++
AT
ATZ
AT&F
ATE0V1Q0X4&D0&S0&C1
```

□ Tone □ Pulse

	Number	Description
1		
2		
3		
4		
5		

□ Use Modem (Initialize modem at power-up)

```
+++
AT
ATZ
AT&F
ATE0V1Q0X4&D0&S0&C1
```

☑ Tone ☑ Pulse

	Number	Description
1	9,9786522	
2		
3	523	
4		
5		

Note that an M90 cannot be configured for **both** SMS messaging and modem communications. If this is done, SMS messaging will override modem communications--the M90 will not be able to use the modem.

For Advanced users check: How the M90 works with a modem.

Modem Communications-- System Bits and Integers

Relevant System Bits, System Integers, and Modem Error Messages are listed below.

Modems: General

System Bits		
SB	**Symbol**	**Description**
72	Initialize Modem	Causes modem initialization. Remains ON until initialization is complete, then turns off. Note that: • This SB turns ON at power-up. You can disable this SB at power-up to avoid initializing the modem. • You may use this SB to initialize the modem at any point during your application.
73	Modem Initialization: Succeeded	Signals that modem has been initialized. When SB 73 is ON, M90 is ready to both make and receive calls.
74	Modem Initialization: Failed	Signals that modem initialization failed. SI 70 contains the error code.
75	Modems Connected	Turns ON when connection is established
76	Disconnect Modem	Ends call (hang-up)

77	Dial Remote Modem	Dials the phone number represented by the index number stored in SI 71

System Integers

SI	Symbol	Description
70	Modem: Error Code	Contains an error code resulting from a modem error. The list is shown below.
71	Modem: Phone Number	Contains the phone number to be dialed. You create a phone book when you configure the modem. Each phone number in the phone book is linked to an index number. Use the Store Direct function to place the index number of desired phone number in SI 71, then activate SB 77 to dial it.

Error Messages (SI 70)

Number	Error Message	Description
0	No Error	No error.
1	No Carrier	No carrier signal found--reason unknown. Check your communication cables.
2	Modem Did Not Reply	The modem referred to is the one on the M90 side.
3	No Dial Tone	No dial tone.
4	Line is Busy	The number dialed is engaged.
5	No Carrier While Dialing	Carrier signal was lost during dialing.
6	Modem Report Error	May be due to an incorrect number or unknown initialization commands.
7	Modem Report Unknown Message	An unrecognized message.
8	No Phone Number	SI 71 contains a number that is not linked to any phone number stored in the phone book.
9	RS232 Port Busy	The RS232 port is already in use.

SMS System Bits and Integers

Listed below are the System Bits, System Integers, and Error Messages that are used by the M90 in SMS messaging.

System Bits

SB	Symbol	Description
180	Initialize GSM Modem for SMS	This is necessary to enable use of the SMS feature. Note that the modem must first be initialized using SB 70.
181	SMS: Initialization Succeeded	Signals that GSM modem has been initialized. The modem is now ready to send and receive SMS messages.
182	SMS: Initialization Failed	Signals that GSM modem has failed. SI 180 contains the error code.
183	Send SMS	Send the string that is represented by the index number stored in SI 182, to the phone number represented by the index number stored in SI 181.
184	SMS: Transmission succeeded	Signals that SMS has been successfully transmitted
185	SMS: Transmission Failed	Signals that SMS has failed. SI 180 contains the error code
186	SMS Received	Signals that a defined SMS has been received. SI 183 contains the index number identifying the origin of the SMS, if this number has been stored in the SMS phone book. If the number is not found, SI

		183 equals 0. SI 184 contains the index number of the SMS string that has been received. Only messages that have been defined in the SMS messages list can be received by the M90.
187	Error in Received SMS	This bit signals one of the errors listed below. SI 180 contains the error code.
188	Ignore Received SMS	Allows the user to block reception of SMS messages
189	Print SMS message	This prints a message with CR (Carriage Return) & LF (Line Feed)
190	Print SMS message	This prints a message with LF (Line Feed)
191	Print SMS message	This prints a message without CR (Carriage Return) or LF (Line Feed)
192	Get GSM antenna signal quality	Get GSM antennae signal quality. The signal quality is contained in SI 185 GSM Signal Quality.
193	Delete SMS messages from SIM	Deletes all of the SMS messages from the SIM card
194	Print SMS message	This prints a message including STX and ETX.

System Integers

SI	Symbol	Description
180	SMS Error Code	Contains an error code resulting from a SMS error. The list is shown below
181	SMS: Send to Phone Number	Contains the index number of a phone number within the GSM phone book. Use the Store Direct function to place the index number of the desired phone number in SI 181. Storing the value '0' into SI 181 causes a message to be sent to the **last** number to which an SMS message was sent. When auto-acknowledge is selected, the number 7 will be automatically placed into S1 181 when the SMS is acknowledged.
182	SMS: String Number to Send	Contains the index number that represents the SMS string to be sent. Use the Store Direct function to place the index number of the desired SMS string in SI 182.
183	Origin of Received SMS	Contains the index number that represents the phone number from which the SMS was sent. If this number is not defined in the GSM phone book, SI 183 will contain 0.
184	Received SMS String	Contains the index number that represents the SMS that has been received. If this number is not defined in the SMS message list, SI 184 will contain 0.
185	GSM Signal Quality	GSM antenna signal quality. If this is less than 11, reposition the antenna. You can use SB 192, Get GSM antennae signal quality,together with this SI.

Error Messages (SI 180)

Number	Error Message	Description
0	No error	No error found
1	GSM Modem Not Initialized	The GSM modem was not initialized. Before using the SMS feature the modem must be initialized. Refer to relevant help sections.
2	GSM Modem Did Not Reply	The GSM modem referred to is the one on the M90 side.
3	Modem Reports Unknown Message	Modem returns an unrecognized reply
5	Wrong PIN number	The Personal Identification Number that was given does not match that of the SIM card installed in the M90's GSM modem.

6	Failed Registration	GSM modem did not register successfully, for example if no network was found, or if the modem antenna is not functioning.
7	No Phone Number	SI 181 contains a number that is not linked to any phone number stored in the GSM phone book.
8	Transmit: Undefined String number	SI 182 contains a string number that is not linked to any string number stored in the SMS Messages List.
9	Unauthorized Origin	This SMS string has been transmitted from an unauthorized phone number.
11	Illegal String Received	The string received is not linked to any string stored in the SMS Messages List. SI 184 will contain 0.
14	RS232 Port Busy	The RS232 port is already in use; for example, the modem is currently connected.
16	SMS not successfully sent to all numbers	The SMS message was not successfully sent to all the phone numbers for which it was configured.
17	PUK number needed	The SIM card is locked due to too many attempts to enter an incorrect PIN number.

Networks

About M90 networks

You can create a decentralized control network of up to 63 controllers using CANbus-enabled M90 models. This is sometimes called a multi-master network. In an M90 network, CANbus enables inter-PLC data exchange. Technical specifications and wiring diagrams are given in the M90 User Guide.

Using your PC, you can access a networked M90 unit via its RS232 port. You can then view, read, and write data into any unit. This feature can also allow you to view your network via a SCADA program.

Assigning a Unit ID number

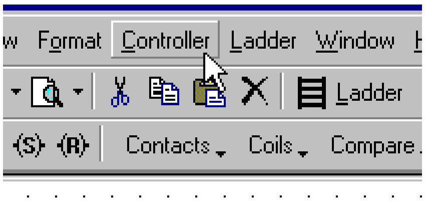
When you create an M90 network, you must assign a Unit ID number to each controller. A Unit ID number is unique. It **must** be used **only** once within a network.

You use this number for two purposes:

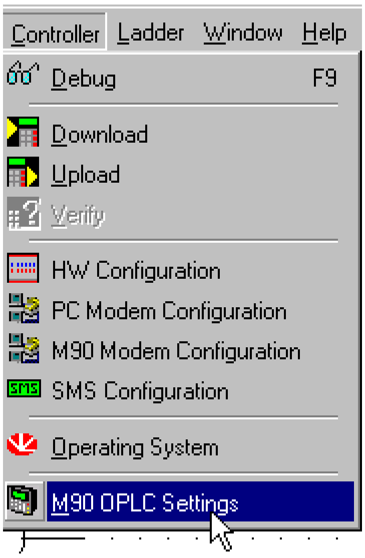
- To enable the M90 controllers to exchange data.
- To access a networked M90 via your PC.

To set a Unit ID number:

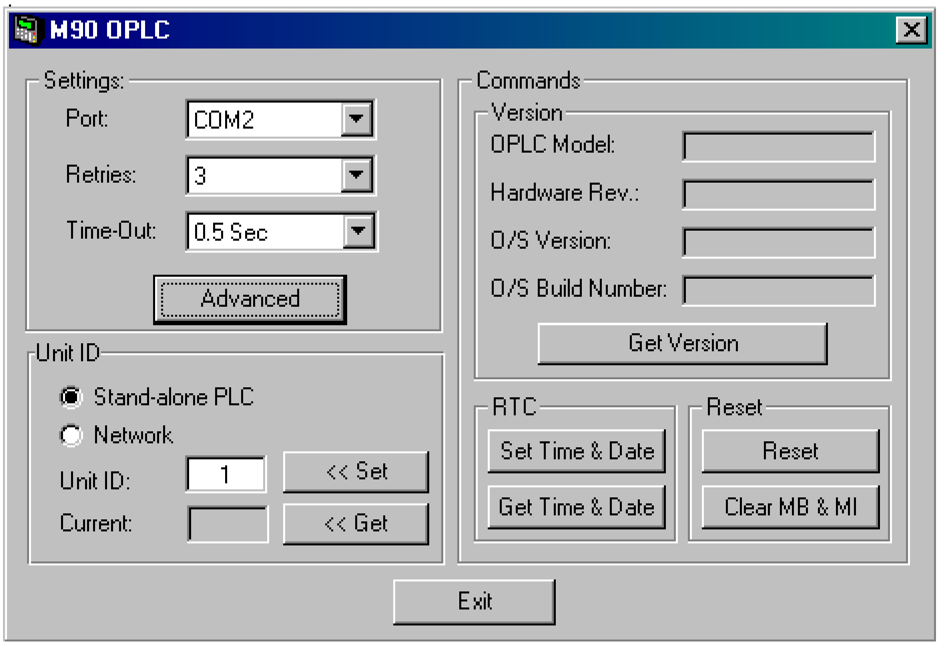
1. Click Controller on the Standard menu bar.



2. Select **M90 OPLC Settings** from the Controller menu.



3. The **M90 OPLC Settings** window opens.

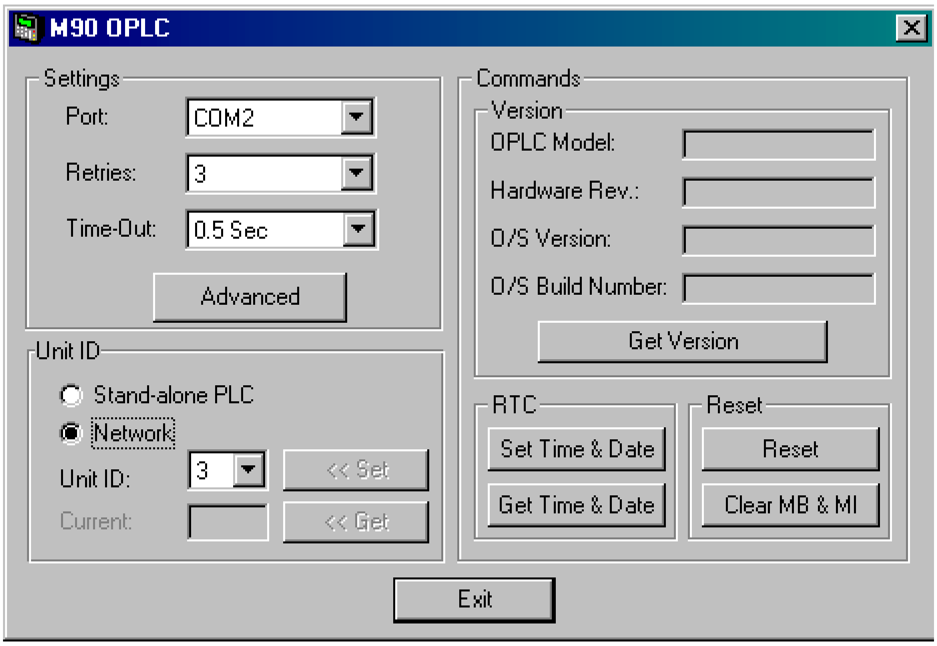

M90 OPLC

Settings:
Port: COM2
Retries: 3
Time-Out: 0.5 Sec

Unit ID
● Stand-alone PLC
○ Network
Unit ID: 1
Current:

Commands
Version
OPLC Model:
Hardware Rev.:
O/S Version:
O/S Build Number:

4. Enter the new ID number in the Unit ID window.



M90 OPLC

Settings

Port: COM2

Retries: 3

Time-Out: 0.5 Sec

Unit ID

○ Stand-alone PLC

● Network

Unit ID: 3

Current:

Commands

Version

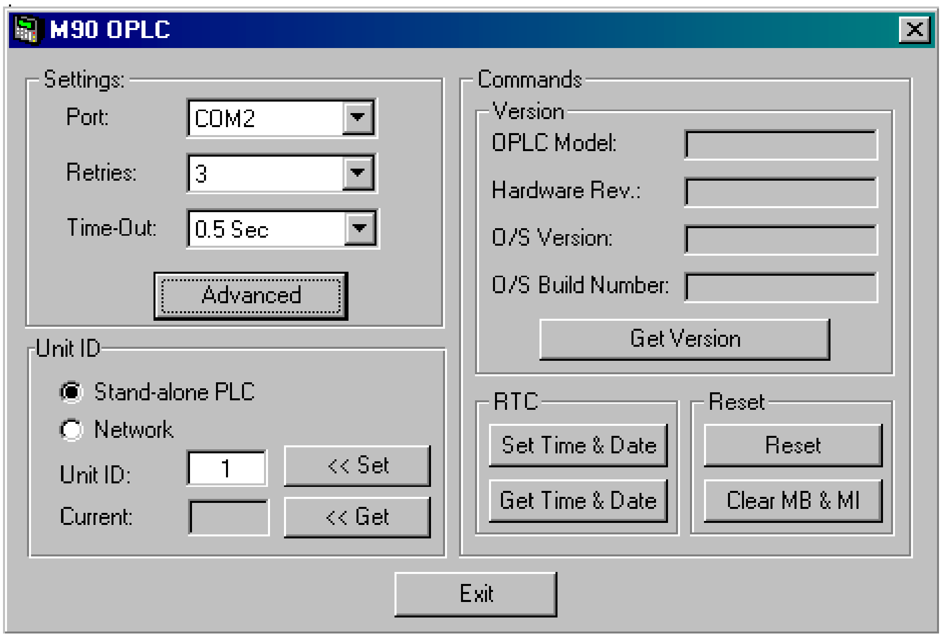
OPLC Model:

Hardware Rev.:

O/S Version:

O/S Build Number:

5. Click **<< Set** to enter the new IN number.



M90 OPLC

Settings:

Port: COM2

Retries: 3

Time-Out: 0.5 Sec

Unit ID

● Stand-alone PLC

○ Network

Unit ID: 1

Current:

Commands

Version

OPLC Model:

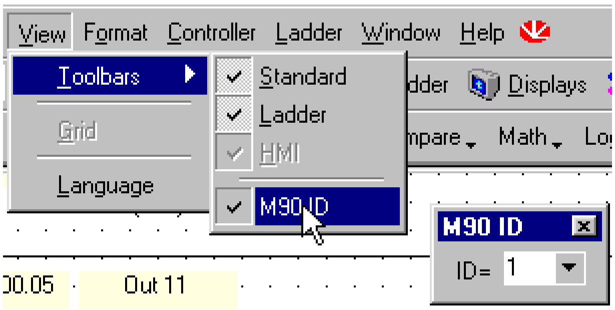
Hardware Rev.:

O/S Version:

O/S Build Number:

Displaying the Unit ID Tool Bar

1. Display the Unit ID by selecting M90 ID from the controller.
2. The Unit ID tool bar opens as shown below.



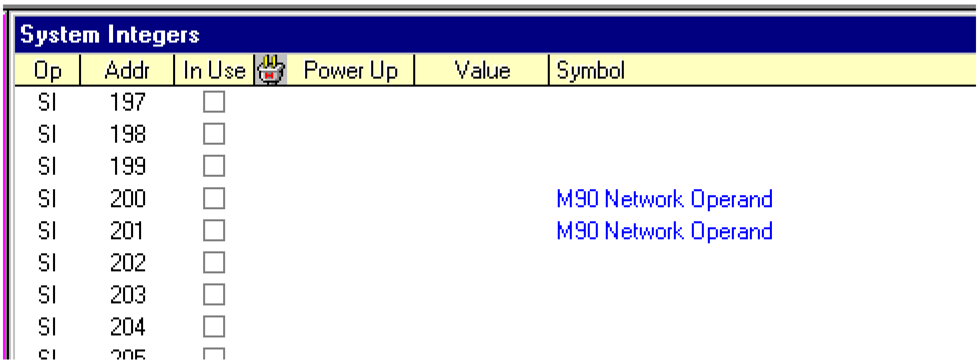
To download via an M90 bridge to a networked M90, you must select the unique ID of the networked M90. When you enter '0' as the Unit ID number, you communicate directly with the M90 that you are using as a bridge to the network.

Enabling M90 to M90 data exchange within a CANbus network

When you create a CANbus network, you assign each controller a unique Unit ID number, 1 through 63.

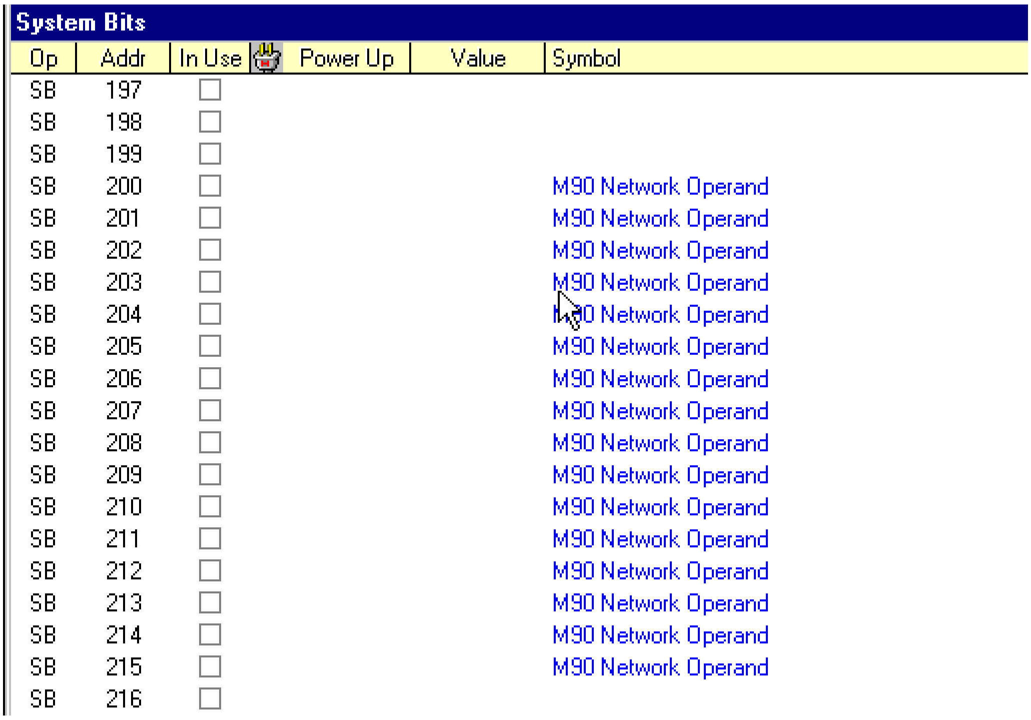
You use these unique ID numbers when you write your network control program. You address operands using the unique ID number. This allows the M90 units to access data from other controllers, using special SIs and SBs in combination with the Unit ID number.

Each controller can read the information contained in SI 200 & SI 201 and SB 200- SB 207 and 16 first Inputs I 0 - I 15 in other units.



System Integers

Op	Addr	In Use	Power Up	Value	Symbol
SI	197	□			
SI	198	□			
SI	199	□			
SI	200	□			M90 Network Operand
SI	201	□			M90 Network Operand
SI	202	□			
SI	203	□			
SI	204	□			
SI	205	□			



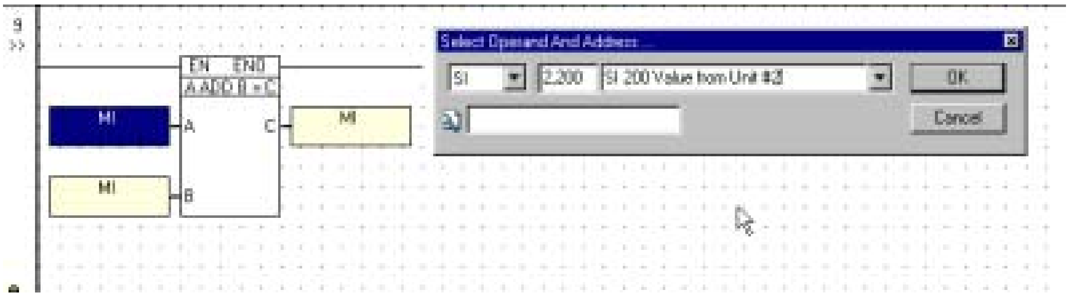
System Bits

Op	Addr	In Use	Power Up	Value	Symbol
SB	197	□			
SB	198	□			
SB	199	□			
SB	200	□			M90 Network Operand
SB	201	□			M90 Network Operand
SB	202	□			M90 Network Operand
SB	203	□			M90 Network Operand
SB	204	□			M90 Network Operand
SB	205	□			M90 Network Operand
SB	206	□			M90 Network Operand
SB	207	□			M90 Network Operand
SB	208	□			M90 Network Operand
SB	209	□			M90 Network Operand
SB	210	□			M90 Network Operand
SB	211	□			M90 Network Operand
SB	212	□			M90 Network Operand
SB	213	□			M90 Network Operand
SB	214	□			M90 Network Operand
SB	215	□			M90 Network Operand
SB	216	□			

To read the information from a controller, the addressing to an SI or SB must be combined with the Unit ID number of the controller being read from.

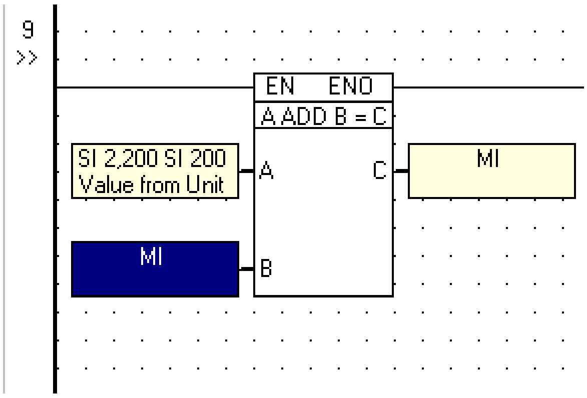
Example:

We want to add the value in SI 200 in unit #2 with another MI.



Select Operand And Address ...

SI | 2,200 | SI 200 Value from Unit #2



SI 2,200 SI 200 Value from Unit

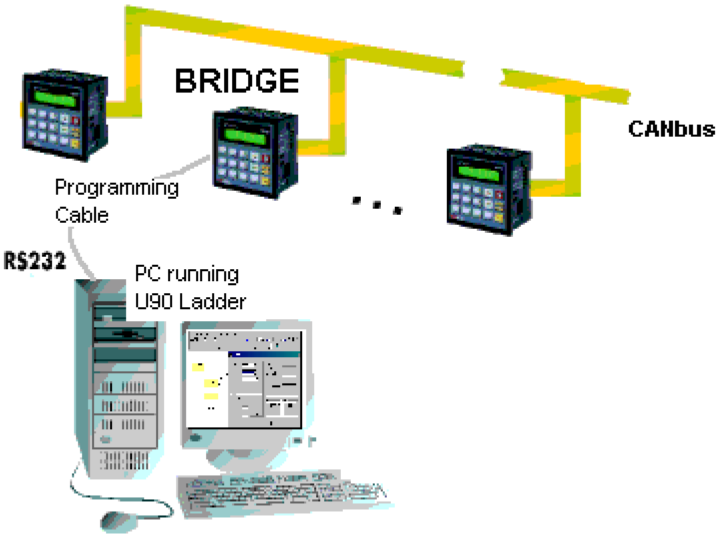
Note: You can connect up to **63** units in a CANbus network. Each controller can read information from up to **8** other controllers in the network.

CANbus networking is featured in several sample applications, check the application ' 8 PLCs + Alarm' . These applications may be found by selecting Sample U90 Projects from the Help Menu.

Using your PC to access a network

You can use your PC to access any M90 unit within a network. To do this, you connect your PC to any M90 in the network using the programming cable supplied with the M90 controller as shown below. This M90 is your 'bridge' to the rest of the network.

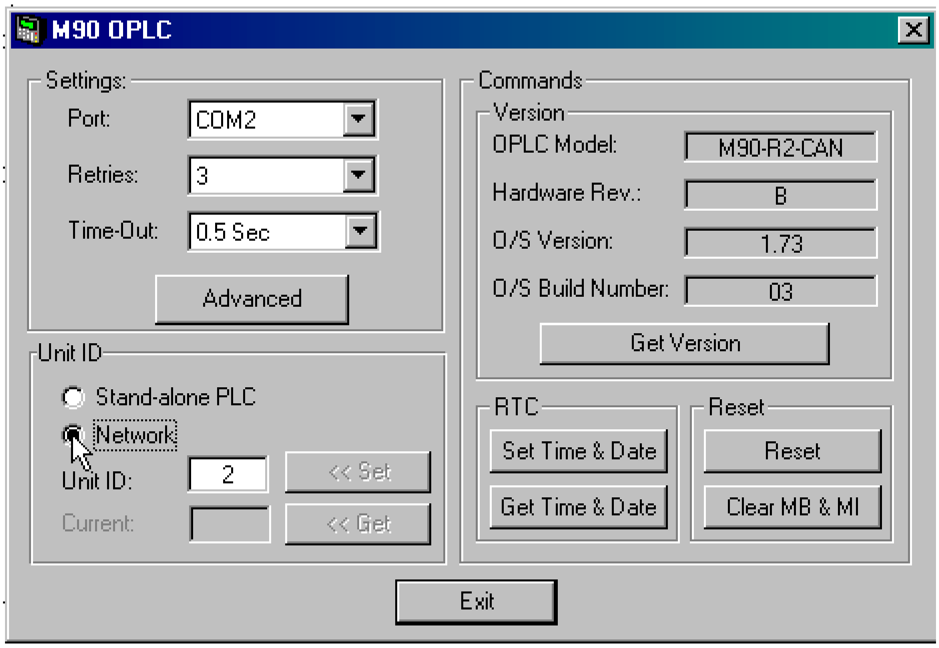
Via the bridge, you can download, upload, and edit programs--you can perform any action that can be performed via direct communications. You can also view runtime data. This does not affect the running of the control program.



Note that different PCs can access a network at the same time, using different M90 units as bridges. However, 2 different PCs cannot simultaneously access the same M90 unit.

To communicate with different M90 units via the bridge, you:

1. Select Network as shown below.
2. Select the M90 you wish to communicate with by entering its unique ID number.



M90 OPLC

Settings:
Port: COM2
Retries: 3
Time-Out: 0.5 Sec

Unit ID
○ Stand-alone PLC
● Network
Unit ID: 2
Current:

Commands
Version
OPLC Model: M90-R2-CAN
Hardware Rev.: B
O/S Version: 1.73
O/S Build Number: 03

According to the figure above, the PC would communicate with M90 number 2.

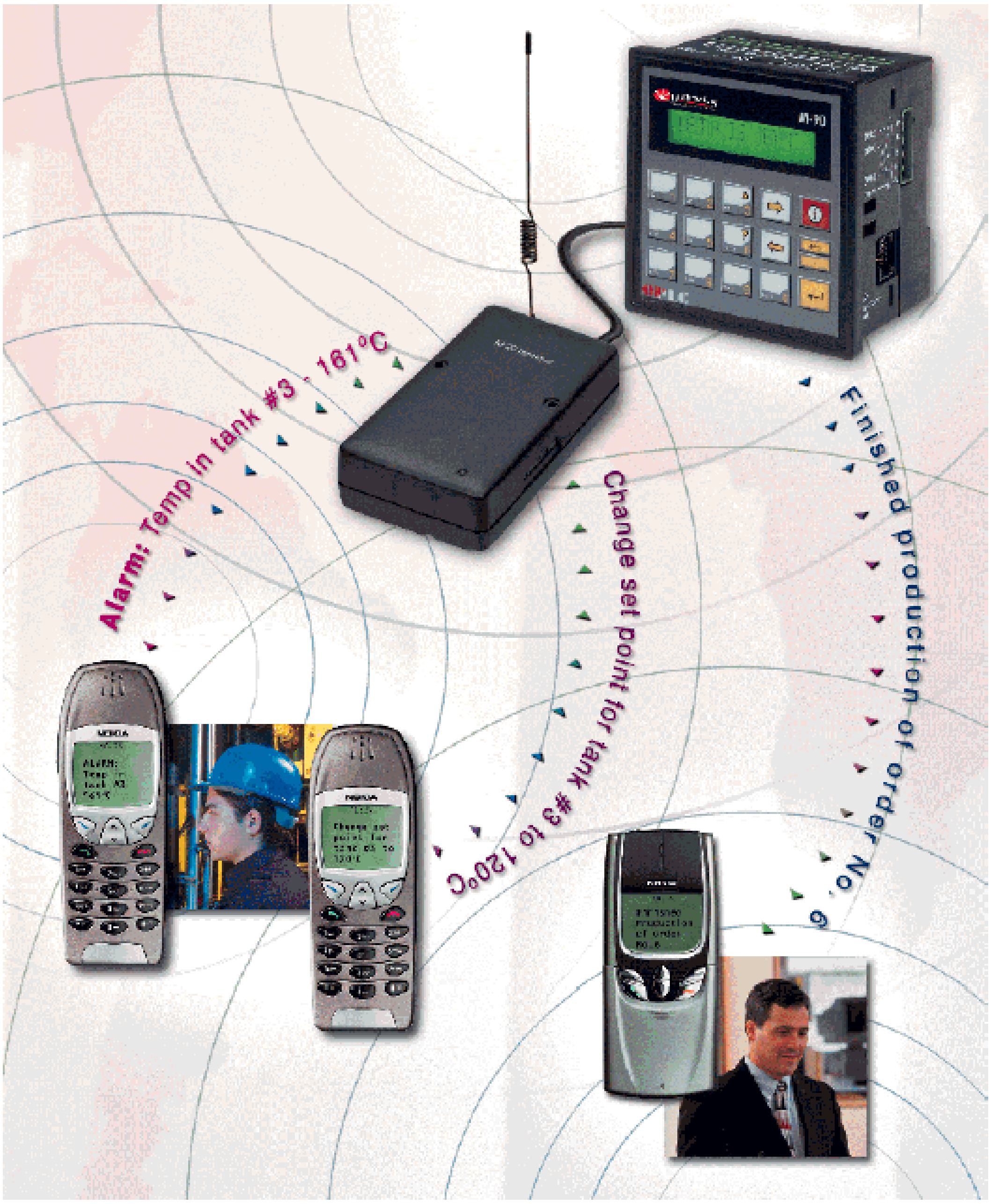
However, note that once your project is defined as a **Network** project, U90 Ladder **cannot** automatically detect the bridges' communication settings. In order to communicate via the bridge, your current communication settings must be **identical** with those of the bridge. Note too, that the bridge's RS232 baud rate cannot be set below 9600.

SMS

About SMS messaging

SMS messaging is a feature of GSM-based cellular telephone services. SMS-enabled M90 controllers can use SMS messaging to send and receive data to and from a cell phone. Both fixed text and variable data can be communicated. This feature can be used to transmit data and for remote diagnostics.

SMS messaging is featured in several sample applications; these may be found by selecting Sample U90 Projects from the Help Menu.


Alarm: Temp in tank #3 - 161°C
Change set point for tank #3 to 120°C
Finished production of order No. 6

In order to use this feature, you must connect an SMS-enabled M90 model to a GSM modem, which is sometimes called a cellular IP modem. Other modems do not support connection to a cellular network.

SMS messaging is subject to all of the limitations of normal cellular network use, as for example network availability.

Note that SMS messages are limited to the English character set.

Overview of M90 SMS messaging

To enable the M90 to use SMS messaging, you must:

1. Create the SMS phone book, which determines where the M90 can send SMS messages.
2. Create SMS messages.
3. Configure the SMS Message Properties for each SMS message.

4. Configure your SMS messaging features.
5. Download the project to the M90.
6. Connect the M90 to a GSM modem

After you have performed the above procedures, you can use SMS messages in your application.

Once SMS messages have been created, configured, and downloaded to the M90, the M90 can receive these messages from a GSM cell phone.

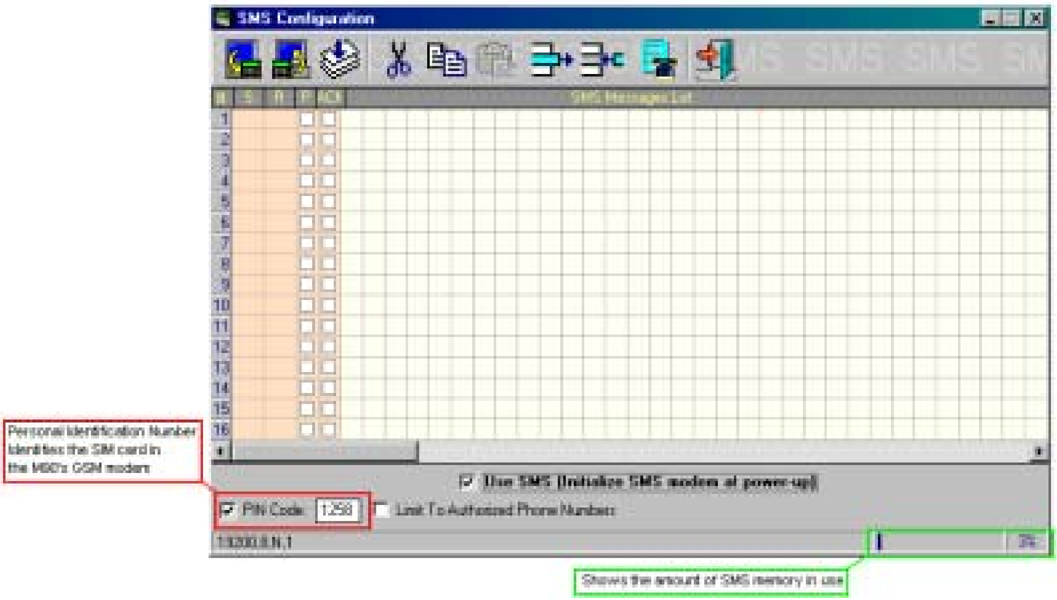
Note that you must use the English character set to write SMS messages.

Configuring SMS messaging features

In order to use the SMS feature, your M90 must be connected to a GSM modem. To enable the M90 to use the GSM modem, select the 'Use SMS Messaging' option shown below. This causes the M90 to turn on SB 180, Initialize GSM Modem, at power-up.

The M90's embedded GSM modem settings are: 19200, 8 data bits, no parity, 1 stop bit. You may need to manually change your modem's communication settings to match these parameters.

If your GSM modem requires a PIN code to connect to a GSM network, enter it as shown below.



Personal Identification Number Identifies the SIM card in the M90's GSM modem

☑ Use SMS (Initialize SMS modem at power-up)

☑ PIN Code 1258 ☐ Limit To Authorized Phone Numbers

19200,8,N,1

3%

Shows the amount of SMS memory in use

Limit to Authorized Phone Numbers

Select this option to prevent the M90 from receiving SMS messages from any number not listed in the SMS phone book.

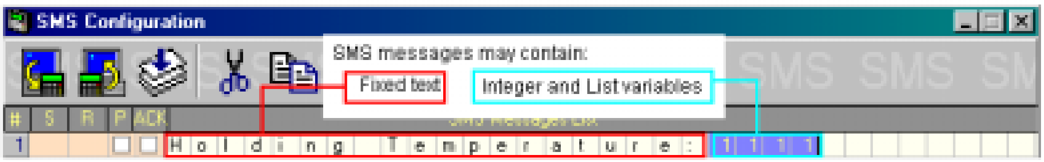
Creating SMS messages

You can create up to 99 SMS messages, or up to a total of 1k, whichever comes first. Each SMS message can contain up to 140 characters. SMS messages can include both fixed text and variable data.

Creating SMS text messages

Note that you must use the English character set to write SMS messages.

1. Open the SMS editor by selecting SMS Configuration from the Controller menu.
2. Enter fixed text by placing your cursor within a line and typing normally. You may use any keyboard symbols except for number symbols (#). These have a specific purpose which is described below.


SMS Configuration
SMS messages may contain:
Fixed text
Integer and List variables
# S R P ACK
SMS Messages List
1 Holding Temperature: 1111

1. Cut and copy messages by clicking on the Cut button. This removes all of the text and variables from a message, but does not delete the line.
2. Copy messages by clicking on the Copy button. This copies all of the text and variables.
3. Paste by clicking on the Paste button. You can paste over an existing message. This action erases any information in the line.
4. Use the Insert button to add a line **below** the line containing the cursor.
5. Use the Delete button to remove a line **below** the line containing the cursor.

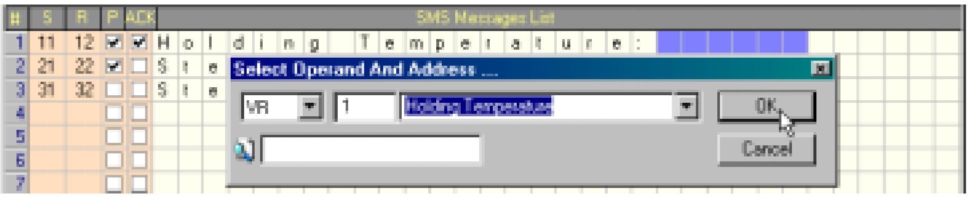
Attaching variables

You can attach up to 9 Integer or List Variables to an SMS message. Each variable can include up to 16 characters. Attaching variables to an SMS message is similar to attaching variables to an HMI display. However, the variable must already be in the variable list--you cannot link a variable before it has been created.

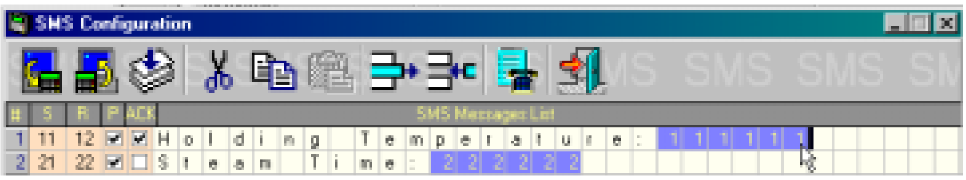
Integer variables can be sent and received with SMS messages. List variables can only be sent to a cell phone.

As with HMI variables, you must create a Display Field for the display of the variable's value.

1. Click your cursor where you want to locate the variable text.
2. Hold down the **Shift** key on your PC keyboard, while you press the right-pointing arrow key. A square is highlighted each time you press the arrow key. The first square displays the number of highlighted squares.
3. Release the **Shift** key. The Select Operand and Address box opens.
4. Enter the variable number and description, then click OK as shown below.



5. The SMS message now appears together with the variable field.


SMS Configuration
# S R P ACK
SMS Messages List
1 11 12 Holding Temperature: 111111
2 21 22 Steam Time: 222222

Deleting variables

1. Place your cursor in the highlighted Variable field.
2. Press the Backspace or Delete key until the entire field is erased.

Testing messages

1. To test your messages, click on the Compile button. If, for example, you have attached 'illegal' variables--not integer or list variables--the first illegal variable will be displayed.

Sending SMS messages from a GSM cell phone

To send SMS messages from your cell phone you must:

- Write and download SMS messages to the M90 as described in Creating SMS messages.
- Write an SMS message in your cell phone.
- Send the message to the M90's GSM modem

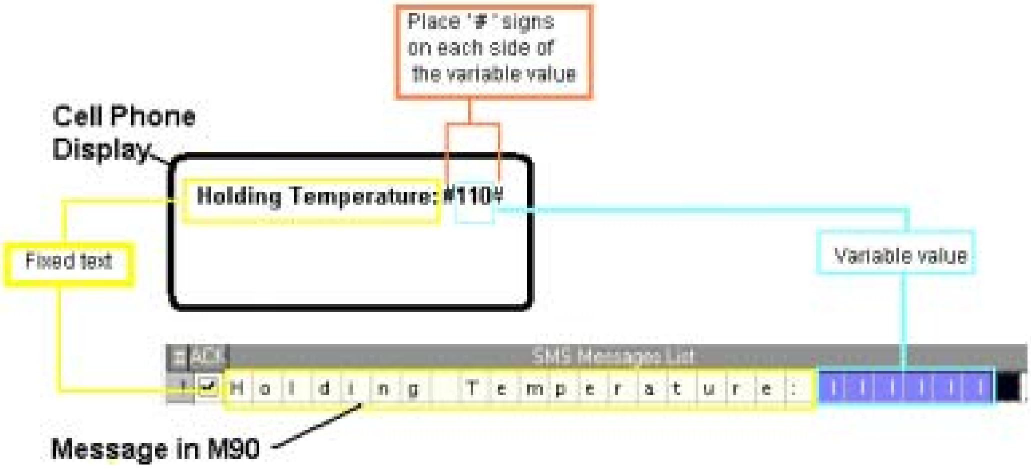
Note that you can only send messages that have already been set in the M90. In addition, if an M90 is configured with the Limited to Authorized Phone Numbers option, you will not be able to send it SMS messages if your number is not in the list .

Writing SMS messages in your cell phone

You write an SMS message using your cell phone keypad. Make sure that:

- The fixed text in your cell phone is identical to the M90's SMS message **in every detail**: spaces, characters--and note that characters are case-sensitive.
- You bracket variable values with number signs (#) as shown below. These signs '#' do not count as spaces.
- The variable field in the M90 is big enough to hold the value.

The figure below shows the same SMS message: as it appears on a cell phone display, and as it appears in the M90's SMS Messages List.



When you send this message from your cell phone, the value 110 will be written into Variable 1 in the M90.

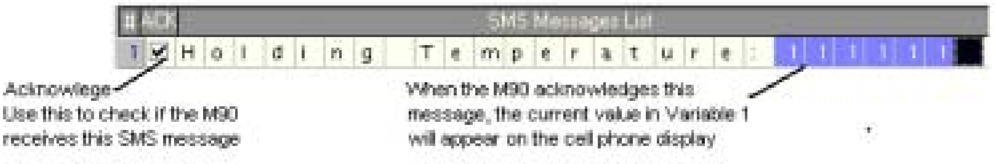
Sending the message to the M90

1. Enter the number of the M90's GSM modem exactly as you would enter any GSM cell phone number, then send the message.

Checking that the M90 has received the SMS message

You can check if the M90 received your message by using the Acknowledge feature:

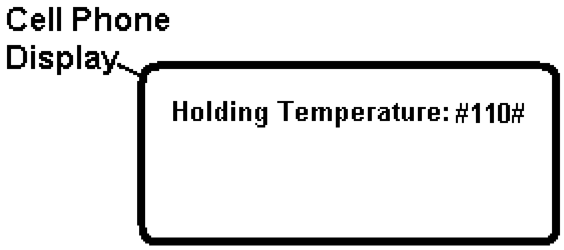
1. Select the **ACK** box as shown below.



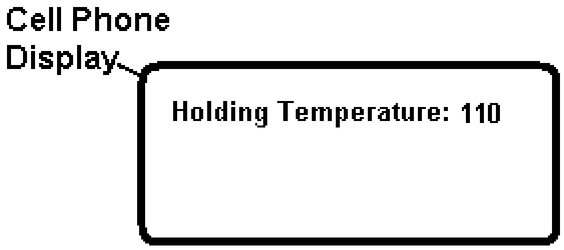
Acknowlege
Use this to check if the M90 receives this SMS message

When the M90 acknowledges this message, the current value in Variable 1 will appear on the cell phone display

2. Use your cell phone to send the message **"Holding Temperature:#110#"** to the M90.



3. The M90 receives this SMS message.
4. The M90 immediately returns the message to your cell phone, together with the current variable value.
5. You can now view this SMS message on your cell phone display, together with changes in the variable value.



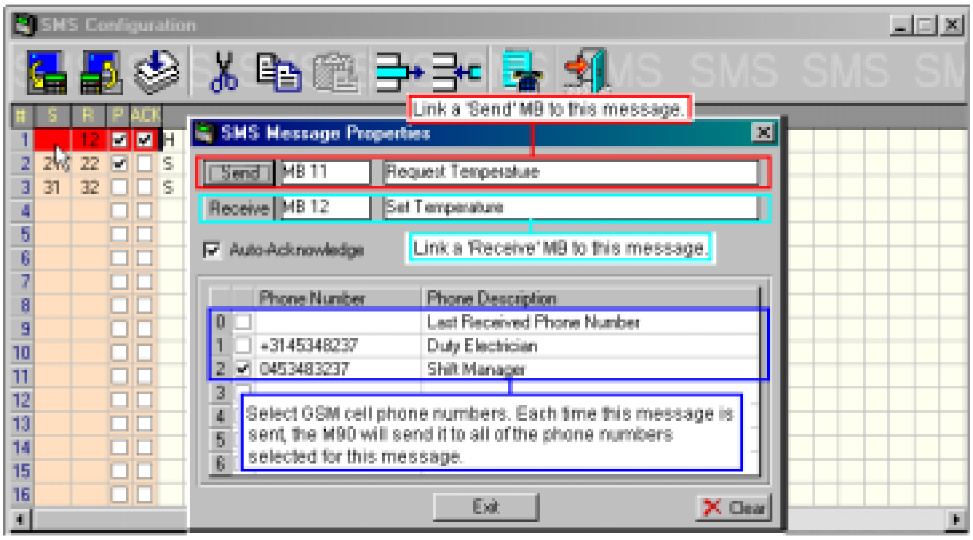
Variable Types

Although SMS messaging supports Integer and List variables, note that you cannot send List variables via cell phone.

SMS Message Properties

Before you can use an SMS message in your application, you must configure its properties.

1. Open the SMS Messages Properties box by clicking in the fields at the beginning of a message as shown below.



#	S	R	P	ACK	
1		12	☑	☑	H
2	21	22	☑	□	S
3	31	32	□	□	S
4			□	□	
5			□	□	
6			□	□	
7			□	□	
8			□	□	
9			□	□	
10			□	□	
11			□	□	
12			□	□	
13			□	□	
14			□	□	
15			□	□	
16			□	□	

SMS Message Properties

Link a 'Send' MB to this message.

Send | MB 11 | Request Temperature

Receive | MB 12 | Set Temperature

Link a 'Receive' MB to this message.

☑ Auto-Acknowledge

		Phone Number	Phone Description
0	□		Last Received Phone Number
1	□	+3145348237	Duty Electrician
2	☑	0453483237	Shift Manager
3			
4			
5			
6			

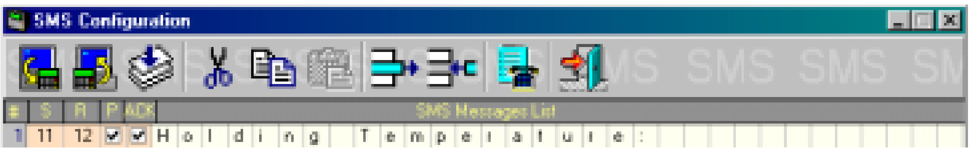
Select GSM cell phone numbers. Each time this message is sent, the M90 will send it to all of the phone numbers selected for this message.

2. Link a Send MB to this message by clicking on the Send button. The Select Operand and Address box opens.
3. Select an MB, then press OK. The MB's number and description appear in the Send fields.
4. Repeat Steps 2 & 3 to link a Receive MB.

   Note that a message does not need to be linked to both a Send and Receive MB.
5. Link the GSM cell phone numbers to this message by checking the boxes of the desired numbers. You can also select Last Received Phone Number. This will cause this SMS message to be sent to the origin of the **last** SMS message received by the M90.

   Note that you cannot edit the SMS phone book while you are configuring SMS Message
6. When you have finished, click Exit.

In the message below, the Send MB is 11, the Receive MB is 12, and the checked box under P means that phone numbers have been linked to this message. ACK has also been selected.



#	S	R	P	ACK	SMS Messages List
1	11	12	☑	☑	Holding Temperature:

ACK-Acknowledge message

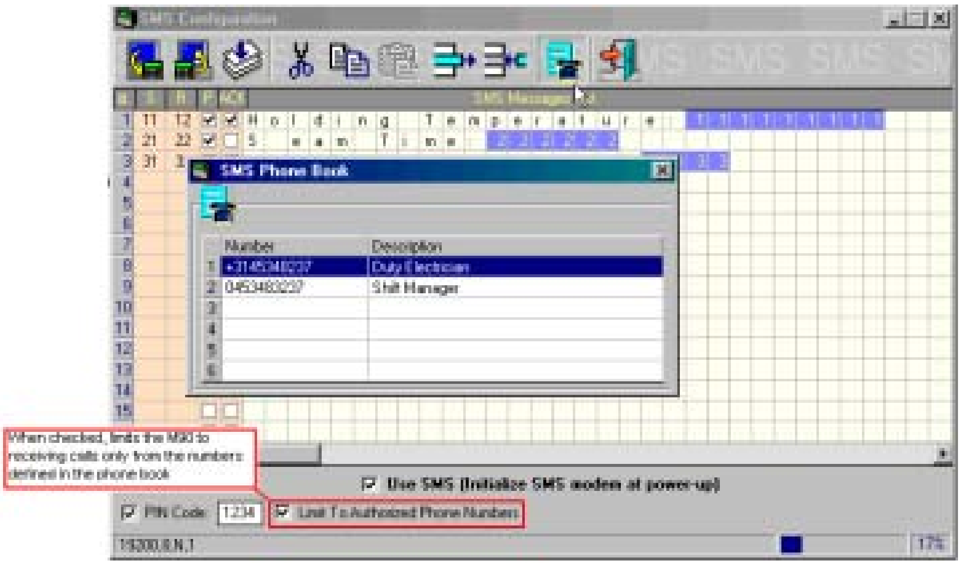
This feature allows a cell phone user to check if the M90 has received a particular message.

SMS phone book

The SMS phone book is where you define the list of GSM cell phone numbers that the M90 can use for SMS messaging. The phone book holds 6 numbers; however, you can dial more numbers by using an MI pointer. Each phone number can be up to 18 characters long. You can also add a description to identify who is being called.

Entering numbers in the Phone Book

1. Open the Phone Book by clicking the button on the toolbar.
2. Click on an empty line in the Phone Book shown below, then type in the number.



	Number	Description
1	+3145348237	Duty Electrician
2	0453483237	Shift Manager
3		
4		
5		
6		

When checked, limits the M90 to receiving calls only from the numbers defined in the phone book

☑ Use SMS (Initialize SMS modem at power-up)

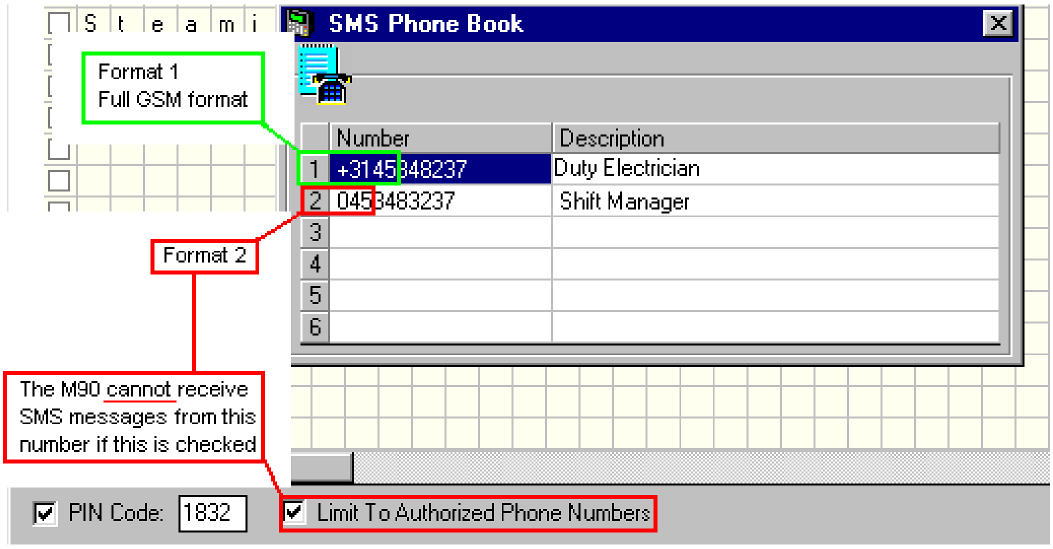
☑ PIN Code: 1234 ☑ Limit To Authorized Phone Numbers

19200,8,N,1 17%

Note that there are two formats for entering phone numbers shown below. If **Limit to Authorized Phone Numbers** is *not* selected, the M90 can send and receive SMS messages to/from any number in the Phone Book.

If **Limit to Authorized Phone Numbers** *is* selected:

- Format 1: The M90 **can** receive messages from this number. This is because the number is in full GSM format, including the '+' in front of the country code.
- Format 2: The M90 **cannot** receive messages from this number.



	Number	Description
1	+3145348237	Duty Electrician
2	0453483237	Shift Manager
3		
4		
5		
6		

The M90 cannot receive SMS messages from this number if this is checked

☑ PIN Code: 1832 ☑ Limit To Authorized Phone Numbers

To edit the phone book, click in a number or description, then make your changes.

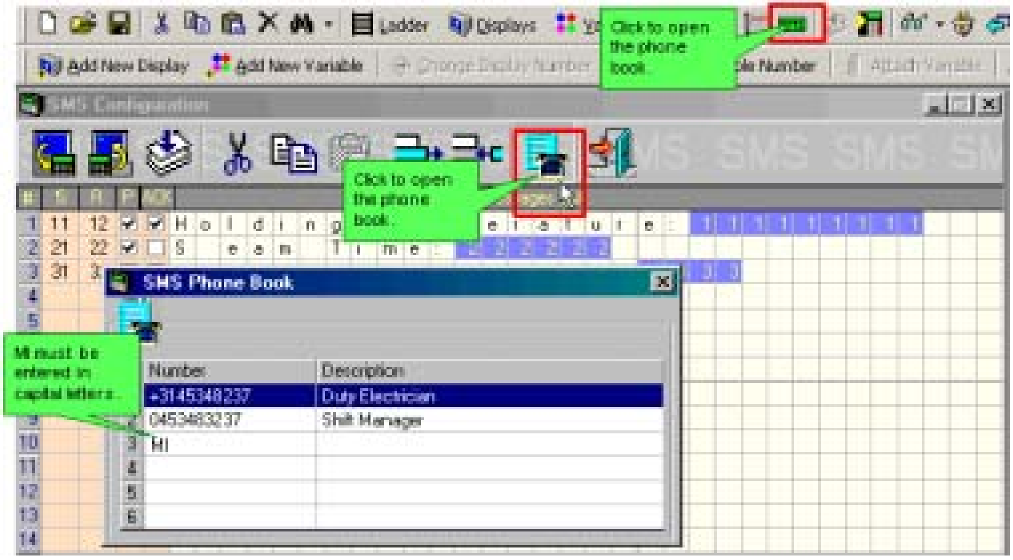
SMS Phone Number: via MI Pointer

Use this utility to use an MI vector as one of the phone numbers in the SMS phone book. This allows you to:

- Enable a number to be dialed via the M90's keypad.
- Exceed the 6 number limit of the SMS phone book.

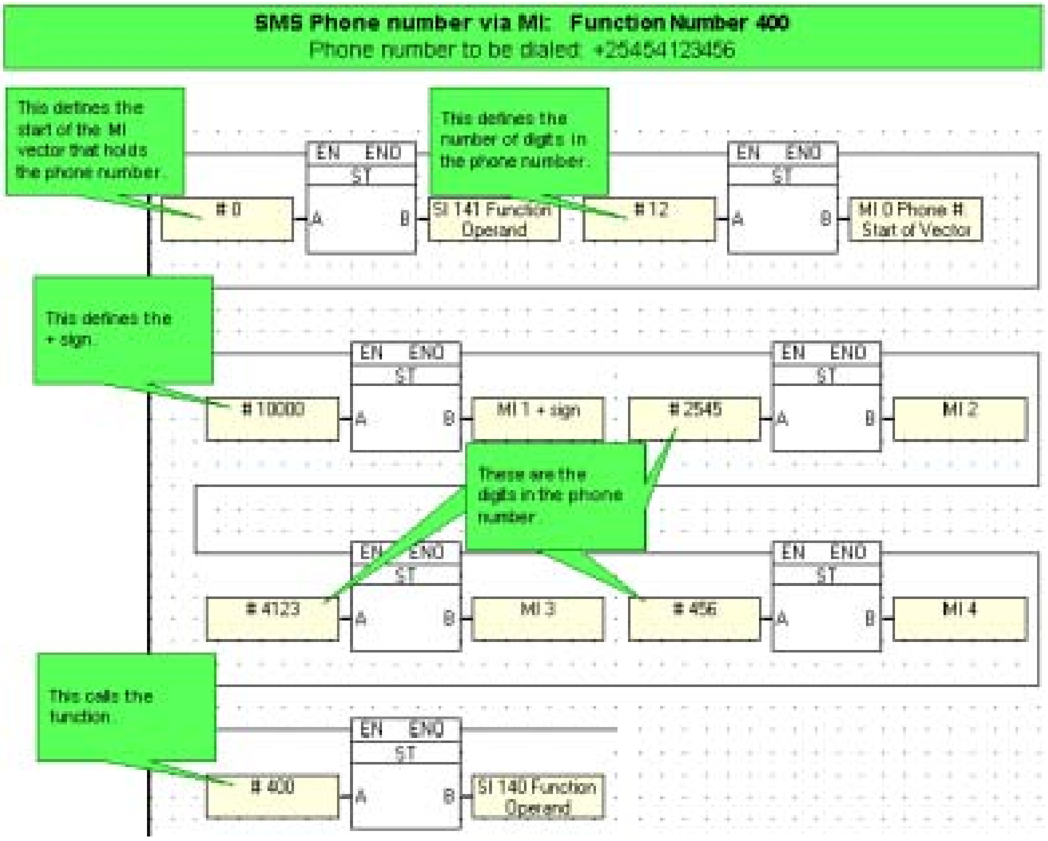
Note that since there is no Ladder element for this function; you perform it by:

- Storing the start address of the MI vector needed to contain the phone number into SI 141,
- Entering the characters MI, in capital letters, in the *SMS phone book*,


Click to open the phone book.
Click to open the phone book.
SMS Configuration
SMS Phone Book
MI must be entered in capital letters.

	Number	Description
1	+3145348237	Duty Electrician
2	0453483237	Shift Manager
3	HI	
4		
5		
6		

- Using the index number of that line to call the number, which enables the number in the MI vector to be called,
- Storing 400 into SI 140 to select the function. Storing the function number calls the function. In your application, call the function **after** you have entered all of the other parameters. Note that when you run Test (Debug) Mode, the current value in SI 140 will **not** be displayed.



SMS Phone number via MI: Function Number 400

Phone number to be dialed: +25454123456

This defines the start of the MI vector that holds the phone number.

This defines the number of digits in the phone number.

This defines the + sign.

These are the digits in the phone number.

This calls the function.

SMS System Bits and Integers

Listed below are the System Bits, System Integers, and Error Messages that are used by the M90 in SMS messaging.

System Bits

SB	**Symbol**	**Description**
180	Initialize GSM Modem for SMS	This is necessary to enable use of the SMS feature. Note that the modem must first be initialized using SB 70.
181	SMS: Initialization Succeeded	Signals that GSM modem has been initialized. The modem is now ready to send and receive SMS messages.
182	SMS: Initialization Failed	Signals that GSM modem has failed. SI 180 contains the error code.
183	Send SMS	Send the string that is represented by the index number stored in SI 182, to the phone number represented by the index number stored in SI 181.
184	SMS: Transmission succeeded	Signals that SMS has been successfully transmitted
185	SMS: Transmission Failed	Signals that SMS has failed. SI 180 contains the error code
186	SMS Received	Signals that a defined SMS has been received. SI 183 contains the index number identifying the origin of the SMS, if this number has been stored in the SMS phone book. If the number is not found, SI 183 equals 0. SI 184 contains the index number of the SMS string that has been received. Only messages that have been defined in the SMS messages list can be received by the M90.

187	Error in Received SMS	This bit signals one of the errors listed below. SI 180 contains the error code.
188	Ignore Received SMS	Allows the user to block reception of SMS messages
189	Print SMS message	This prints a message with CR (Carriage Return) & LF (Line Feed)
190	Print SMS message	This prints a message with LF (Line Feed)
191	Print SMS message	This prints a message without CR (Carriage Return) or LF (Line Feed)
192	Get GSM antenna signal quality	Get GSM antennae signal quality. The signal quality is contained in SI 185 GSM Signal Quality.
193	Delete SMS messages from SIM	Deletes all of the SMS messages from the SIM card
194	Print SMS message	This prints a message including STX and ETX.

System Integers

SI	Symbol	Description
180	SMS Error Code	Contains an error code resulting from a SMS error. The list is shown below
181	SMS: Send to Phone Number	Contains the index number of a phone number within the GSM phone book. Use the Store Direct function to place the index number of the desired phone number in SI 181. Storing the value '0' into SI 181 causes a message to be sent to the **last** number to which an SMS message was sent. When auto-acknowledge is selected, the number 7 will be automatically placed into S1 181 when the SMS is acknowledged.
182	SMS: String Number to Send	Contains the index number that represents the SMS string to be sent. Use the Store Direct function to place the index number of the desired SMS string in SI 182.
183	Origin of Received SMS	Contains the index number that represents the phone number from which the SMS was sent. If this number is not defined in the GSM phone book, SI 183 will contain 0.
184	Received SMS String	Contains the index number that represents the SMS that has been received. If this number is not defined in the SMS message list, SI 184 will contain 0.
185	GSM Signal Quality	GSM antenna signal quality. If this is less than 11, reposition the antenna. You can use SB 192, Get GSM antennae signal quality,together with this SI.

Error Messages (SI 180)

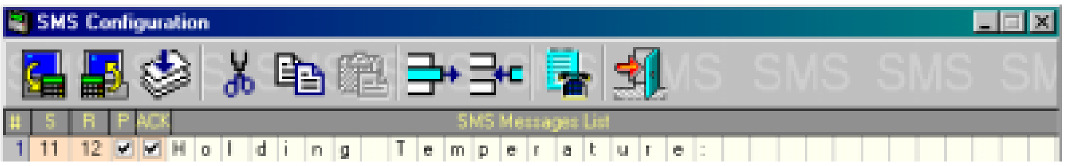
Number	Error Message	Description
0	No error	No error found
1	GSM Modem Not Initialized	The GSM modem was not initialized. Before using the SMS feature the modem must be initialized. Refer to relevant help sections.
2	GSM Modem Did Not Reply	The GSM modem referred to is the one on the M90 side.
3	Modem Reports Unknown Message	Modem returns an unrecognized reply
5	Wrong PIN number	The Personal Identification Number that was given does not match that of the SIM card installed in the M90's GSM modem.
6	Failed Registration	GSM modem did not register successfully, for example if no network was found, or if the modem antenna is not functioning.
7	No Phone	SI 181 contains a number that is not linked to any phone number

	Number	stored in the GSM phone book.
8	Transmit: Undefined String number	SI 182 contains a string number that is not linked to any string number stored in the SMS Messages List.
9	Unauthorized Origin	This SMS string has been transmitted from an unauthorized phone number.
11	Illegal String Received	The string received is not linked to any string stored in the SMS Messages List. SI 184 will contain 0.
14	RS232 Port Busy	The RS232 port is already in use; for example, the modem is currently connected.
16	SMS not successfully sent to all numbers	The SMS message was not successfully sent to all the phone numbers for which it was configured.
17	PUK number needed	The SIM card is locked due to too many attempts to enter an incorrect PIN number.

Using SMS messages in your application

To cause the M90 to send an SMS message, you use the Send MB which is linked to that message. In the figures below, the Send MB is 11. When MB 11 is turned ON in your application, this message will be sent. The Send MB is turned OFF automatically after the message has been sent.

The Receive MB is 12. When this message is received by the M90, MB 12 will turn ON. You must turn the Receive MB OFF in your application in order to register the next time this message is received.



#	S	R	P	ACK	SMS Messages List
1	11	12	☑	☑	Holding Temperature:

M90 Receives the SMS message

When the M90 receives this SMS message, MB 12 will turn ON .

M90 Sends the SMS message

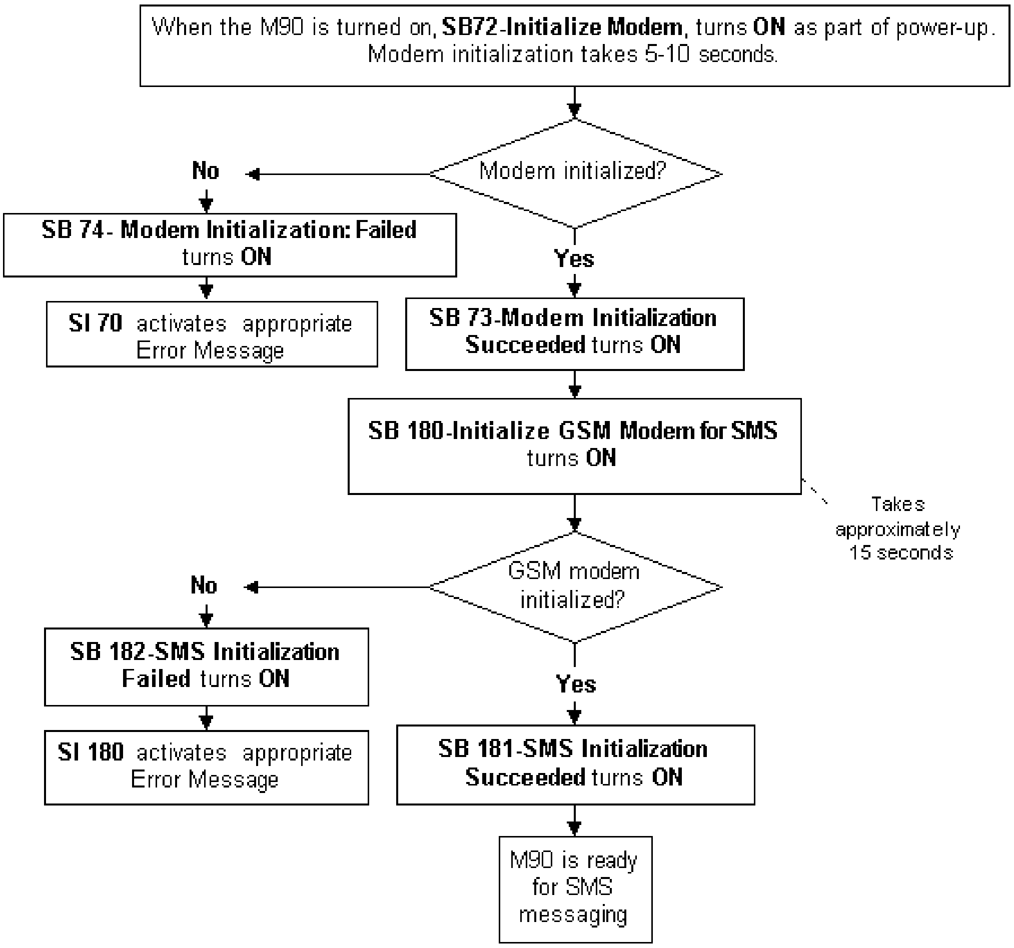
When you turn MB 11 on in your application, the M90 will send this message to all of the selected phone numbers.

How the M90 works with SMS messaging

To allow the M90 to use SMS messaging, you select 'Use SMS messaging' in the M90 OPLC SMS Configuration box.

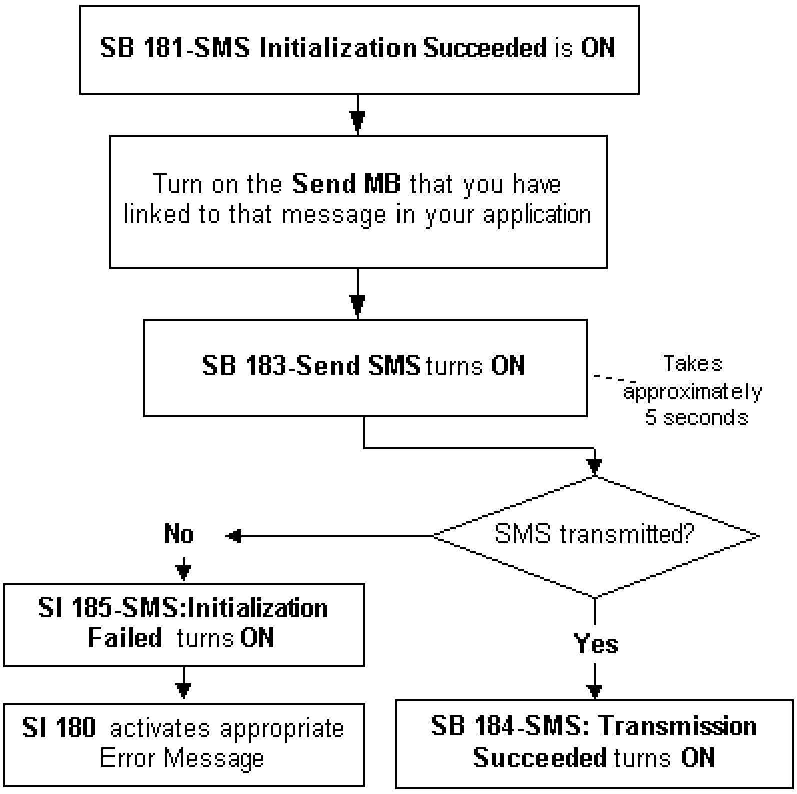
The charts below show the actual process--exactly how the M90 initializes and works with a GSM modem. This information is provided for advanced users who may require it for a specific application, or for troubleshooting.

Initialization


When the M90 is turned on, SB72-Initialize Modem, turns ON as part of power-up. Modem initialization takes 5-10 seconds.
Modem initialized?
No
SB 74- Modem Initialization: Failed turns ON
SI 70 activates appropriate Error Message
Yes
SB 73-Modem Initialization Succeeded turns ON
SB 180-Initialize GSM Modem for SMS turns ON
Takes approximately 15 seconds
GSM modem initialized?
No
SB 182-SMS Initialization Failed turns ON
SI 180 activates appropriate Error Message
Yes
SB 181-SMS Initialization Succeeded turns ON
M90 is ready for SMS messaging

After initialization, the M90 can send and receive SMS messages.

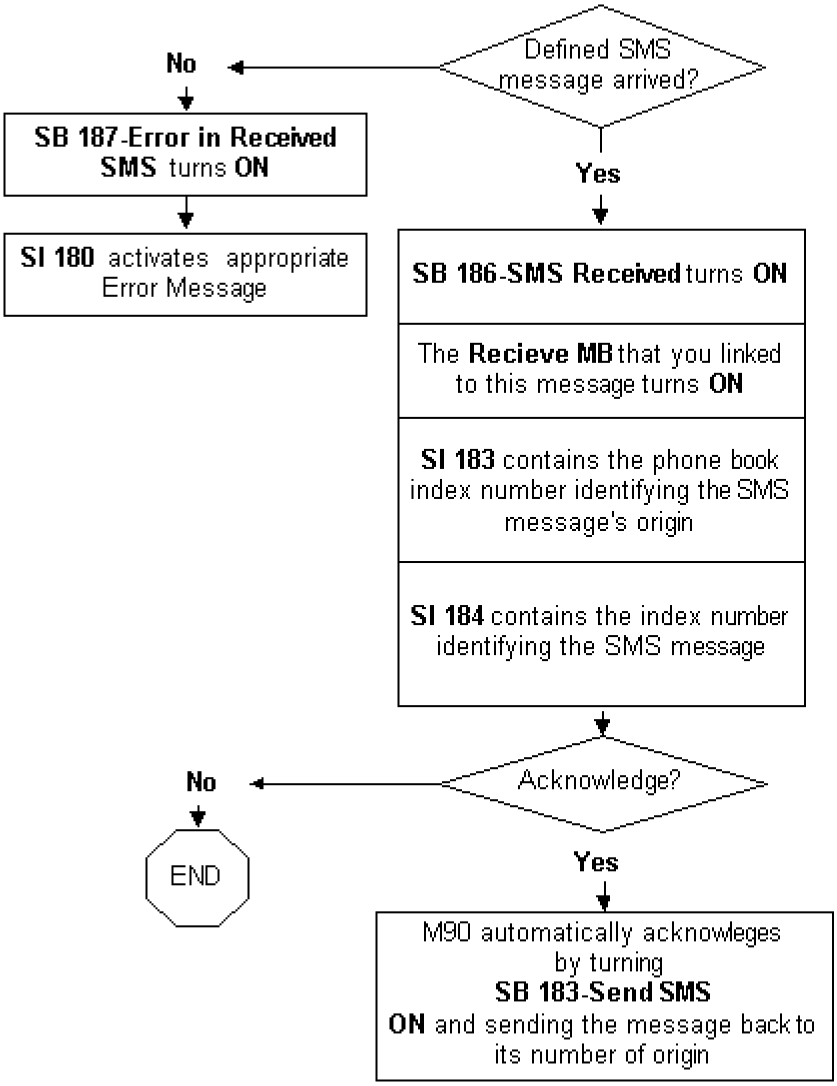
Sending Messages


SB 181-SMS Initialization Succeeded is ON
Turn on the Send MB that you have linked to that message in your application
SB 183-Send SMS turns ON
Takes approximately 5 seconds
SMS transmitted?
No
Yes
SI 185-SMS:Initialization Failed turns ON
SI 180 activates appropriate Error Message
SB 184-SMS: Transmission Succeeded turns ON

Note that a cell phone will not be able to receive a message if its SIM card is full.

Receiving SMS messages

The chart below shows how the M90 receives SMS messages. It also shows what happens if the M90 receives an SMS message marked 'Acknowledge'.


Defined SMS message arrived?
No
SB 187-Error in Received SMS turns ON
SI 180 activates appropriate Error Message
Yes
SB 186-SMS Received turns ON
The Recieve MB that you linked to this message turns ON
SI 183 contains the phone book index number identifying the SMS message's origin
SI 184 contains the index number identifying the SMS message
Acknowledge?
No
END
Yes
M90 automatically acknowleges by turning SB 183-Send SMS ON and sending the message back to its number of origin

Listed below are the System Bits, System Integers, and Error Messages that are used by the M90 in SMS messaging.

System Bits

SB	Symbol	Description
180	Initialize GSM Modem for SMS	This is necessary to enable use of the SMS feature. Note that the modem must first be initialized using SB 72.
181	SMS: Initialization Succeeded	Signals that GSM modem has been initialized. The modem is now ready to send and receive SMS messages.
182	SMS: Initialization Failed	Signals that GSM modem has failed. SI 180 contains the error code.
183	Send SMS	Send the string that is represented by the index number stored in SI 182, to the phone number represented by the index number stored in SI 181.
184	SMS: Transmission succeeded	Signals that SMS has been successfully transmitted
185	SMS: Transmission Failed	Signals that SMS has failed. SI 180 contains the error code

186	SMS Received	Signals that a defined SMS has been received. SI 183 contains the index number identifying the origin of the SMS, if this number has been stored in the SMS phone book. If the number is not found, SI 183 equals 0. SI 184 contains the index number of the SMS string that has been received. Only messages that have been defined in the SMS messages list can be received by the M90.
187	Error in Received SMS	This bit signals one of the errors listed below. SI 180 contains the error code.
188	Ignore Received SMS	Allows the user to block reception of SMS messages

System Integers

SI	Symbol	Description
180	SMS Error Code	Contains an error code resulting from a SMS error. The list is shown below
181	SMS: Send to Phone Number	Contains the index number of a phone number within the GSM phone book. Use the Store Direct function to place the index number of the desired phone number in SI 181. Storing the value '0' into SI 181 causes a message to be sent to the last number to which an SMS message was sent. When auto-acknowledge is selected, the number 7 will be automatically placed into S1 181 when the SMS is acknowledged.
182	SMS: String Number to Send	Contains the index number that represents the SMS string to be sent. Use the Store Direct function to place the index number of the desired SMS string in SI 182.
183	Origin of Received SMS	Contains the index number that represents the phone number from which the SMS was sent. If this number is not defined in the GSM phone book, SI 183 will contain 0.
184	Received SMS String	Contains the index number that represents the SMS that has been received. If this number is not defined in the SMS message list, SI 184 will contain 0.

Error Messages (SI 180)

Number	Error Message	Description
0	No error	No error found
1	GSM Modem Not Initialized	The GSM modem was not initialized. Before using the SMS feature the modem must be initialized. Refer to relevant help sections.
2	GSM Modem Did Not Reply	The GSM modem referred to is the one on the M90 side.
3	Modem Reports Unknown Message	Modem returns an unrecognized reply
5	Wrong PIN number	The Personal Identification Number that was given does not match that of the SIM card installed in the M90's GSM modem.
6	Failed Registration	GSM modem did not register successfully, for example if no network was found, or if the modem antenna is not functioning.
7	No Phone Number	SI 181 contains a number that is not linked to any phone number stored in the GSM phone book.
8	Transmit: Undefined String number	SI 182 contains a string number that is not linked to any string number stored in the SMS Messages List.
9	Unauthorized Origin	This SMS string has been transmitted from an unauthorized phone number.
11	Illegal String Received	The string received is not linked to any string stored in the SMS Messages List. SI 184 will contain 0.

14	RS232 Port Busy	The RS232 port is already in use; for example, the modem is currently connected.
16	SMS not successfully sent to all numbers	The SMS message was not successfully sent to all the phone numbers for which it was configured.
17	PUK number needed	The SIM card is locked due to too many attempts to enter an incorrect PIN number.

SMS messaging problems

You can begin troubleshooting by entering Information Mode. You can then check the status of relevant System Bits and Integers to help diagnose the problem.

To begin diagnosing the problem, check the error codes contained in SI 70 and SI 180. Refer to the error code tables Modem Communications-- System Bits and Integers and in SMS System Bits and Integers

The tables below show the more common causes of SMS communication problems.

Problem	**SI 70 value**	**Possible Cause & Recommended Action**
Modem fails to initialize	2: Modem Did Not Reply	M90-to-GSM modem cable: Make sure that the cable is securely connected. Check the pin-out. of the M90-to-modem adapter cables. Note that if you use cables comprising this pin-out, you must set the RS232 parameter Flow Control to N (none). Incompatible communication settings. Most modems automatically match the parameters of incoming data: baud rate, data bits, parity & stop bits. The M90's embedded GSM modem settings are: 19200, 8 data bits, no parity, 1 stop bit. You may need to manually change your modem's communication settings to match these.
	0: No Error	SB 72 / SB180: OFF In order to work with a GSM modem , you must select 'Use GSM modem' in the SMS configuration box . This causes SB 72 Initialize Modem and SB 180 Initialize GSM modem to turn ON when the M90 powers up. Check that these SBs are not disabled in your program.
	6: Modem Report Error	Check the modem initialization commands. Refer to Configuring the M90 to use a modem.

Other Common Problems:

Problem	**SI 180 value**	**Possible Cause & Recommended Action**
GSM modem not Initialized	1	Refer to table above
Wrong PIN number	5	Check the PIN number contained in the SMS Configuration box, leave it empty if your SIM card has no PIN number.
Failed Registration	6	GSM modem did not register successfully, for example if no network was found, or if the modem antenna is not functioning.
PUK number needed	17	The SIM card is locked due to too many attempts to enter an incorrect PIN number.
Cell phone does not receive	No value	Check the cell phone's SIM card; it may be full. You can clear the SIM card using SB 193 Delete all SMS

message	messages from SIM card.

Ladder

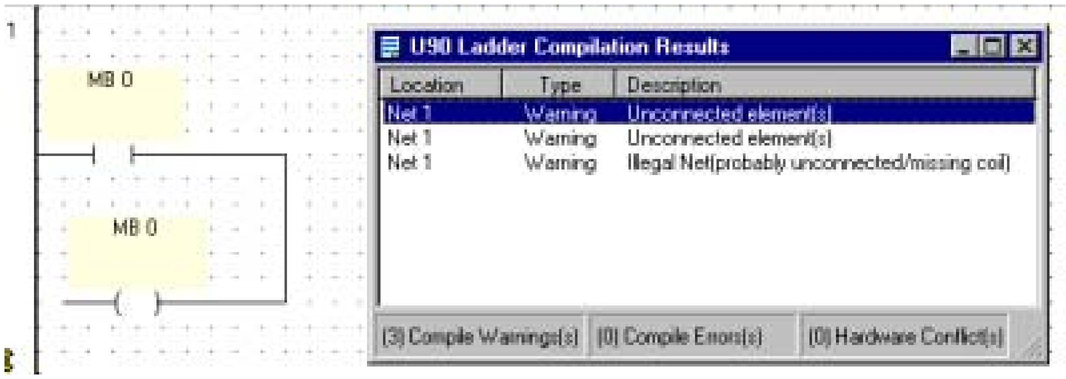
Ladder Net

A U90 Ladder net is the smallest division of a ladder diagram in Unitronics' U90 Ladder software.

Your first ladder element on the left must be connected to the left side of the ladder in each net. You do **not** need to connect the last element on the right to the right side of the ladder in each net.

You should place only one ladder rung on a Ladder net.

Power flows through the ladder elements in a net from left to right. If you build a net that would result in reverse power flow (right to left) the following error message occurs:



U90 Ladder Compilation Results

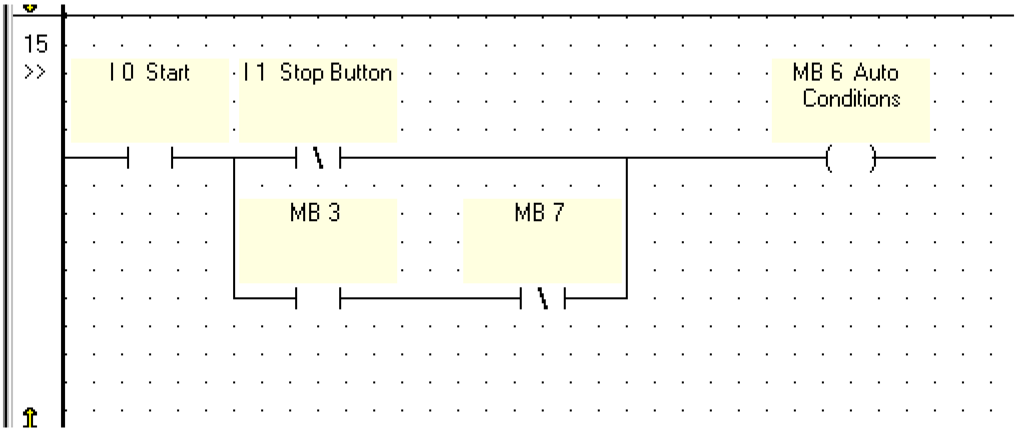
Location	Type	Description
Net 1	Warning	Unconnected element(s)
Net 1	Warning	Unconnected element(s)
Net 1	Warning	Illegal Net(probably unconnected/missing coil)

(3) Compile Warnings(s) (0) Compile Errors(s) (0) Hardware Conflict(s)

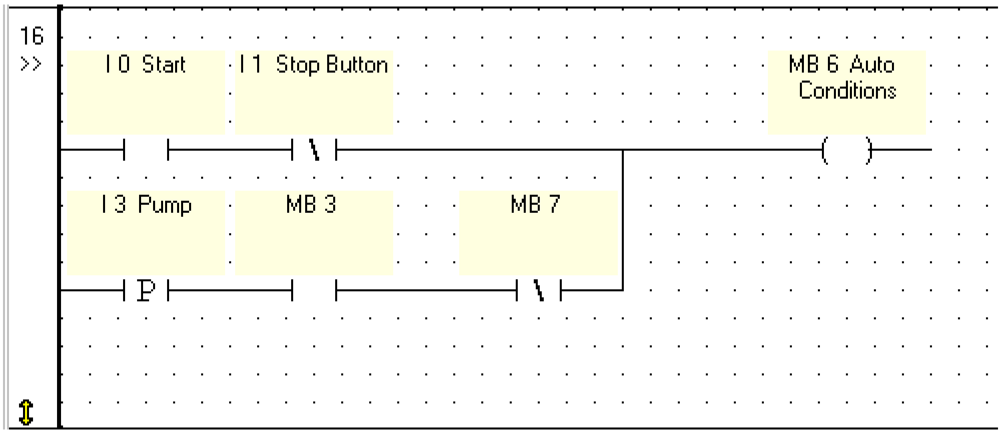
Placing more than one rung in a net may cause compiler problems in your project.

Examples:

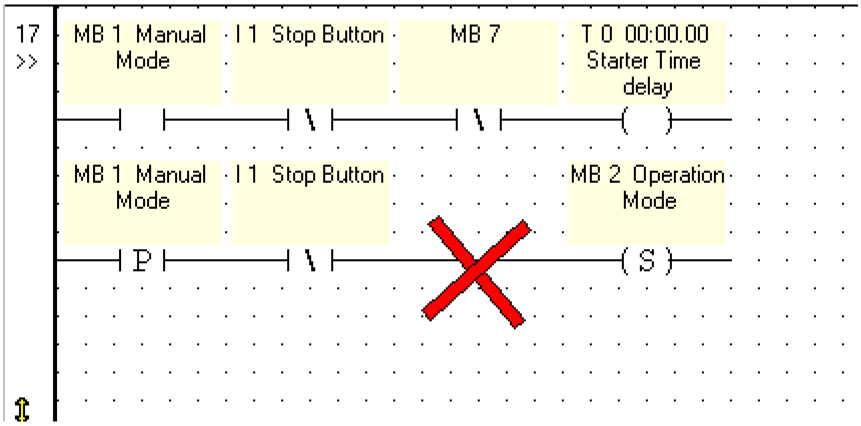
This net is constructed properly.



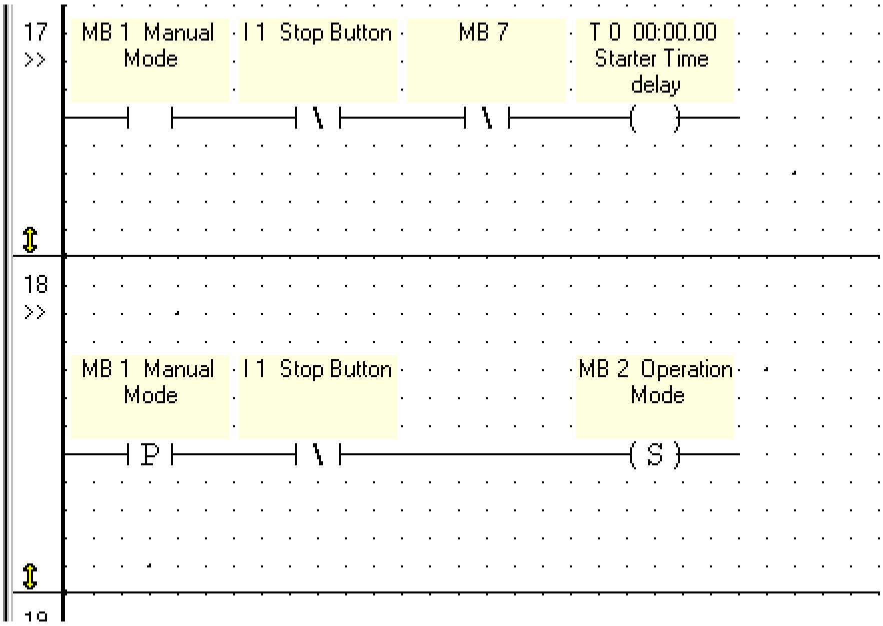
This net is constructed properly.



This net is improperly constructed and contains two rungs.



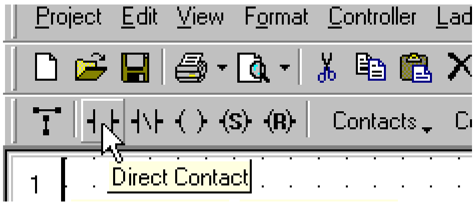
The rungs in the net below should be placed in two nets as shown below.:



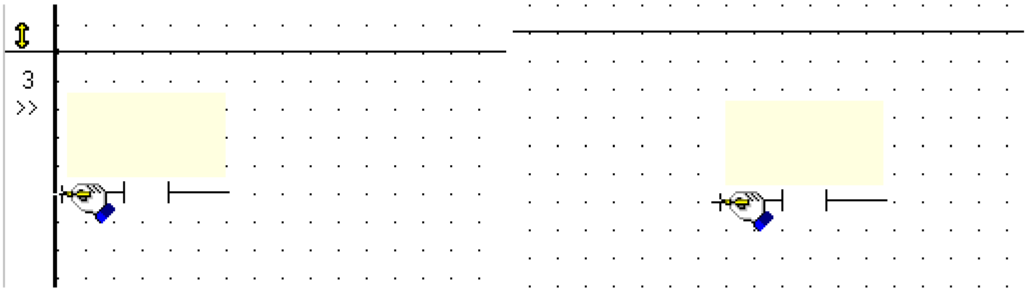
Placing Contacts & Coils

To place a Contact / Coil on a net:

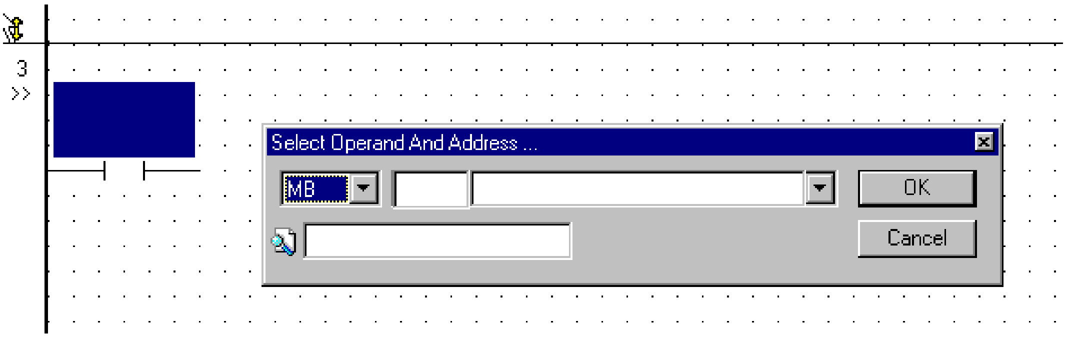
1. Click once to select the desired contact / coil.



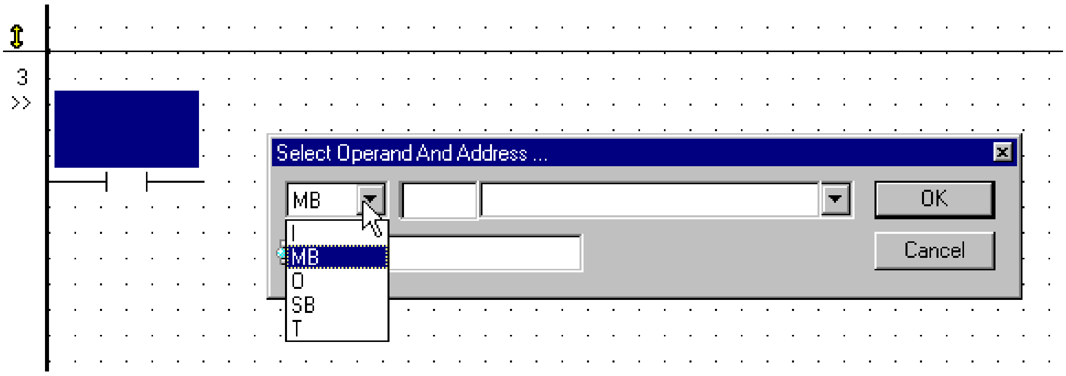
2. Move the element to the desired net position.



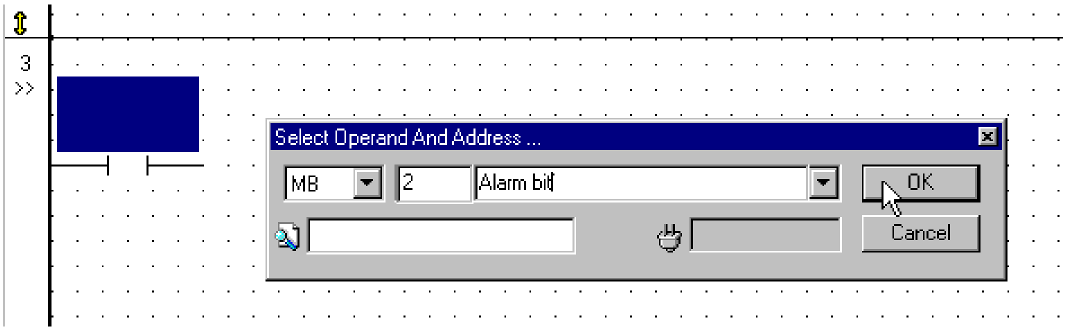
3. Click to place the element. The Operand and Address dialog box opens.



4. Select the Operand type from the drop - down menu.

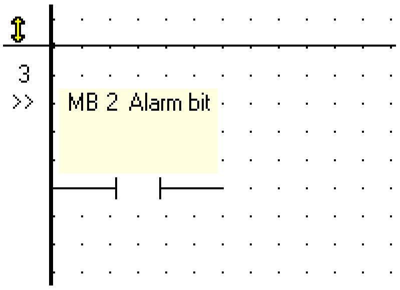


5. Enter the Operand Address and symbol. Click OK.



MB 2 Alarm bit

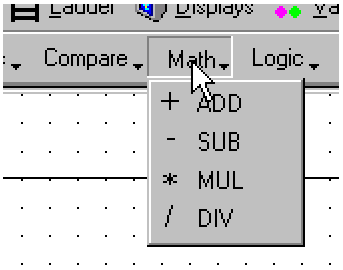
6. The element appears on the net with the selected Operand Address and symbol



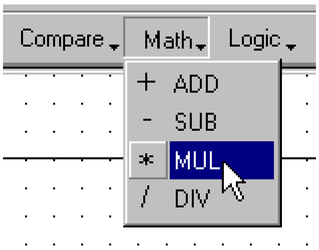
Placing a Function Block

To place a Compare / Math / Logic function block on a net:

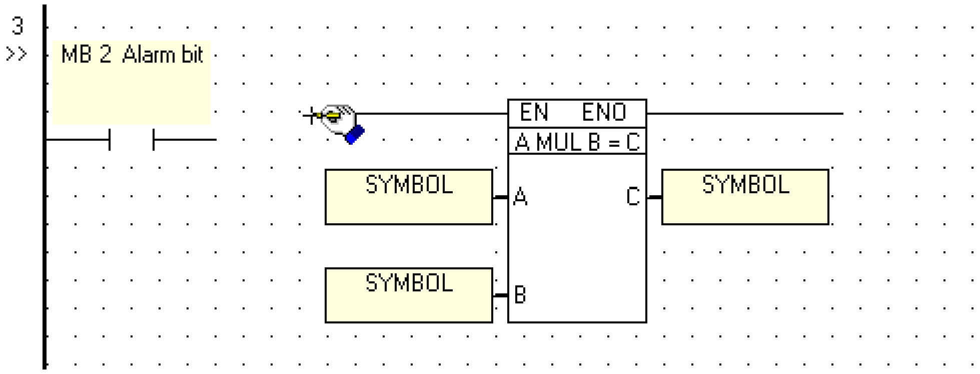
1. Click on the menu containing the desired type of function block,
   OR
   Right-click on a net to display the toolbar, then click on the desired menu; the menu opens.



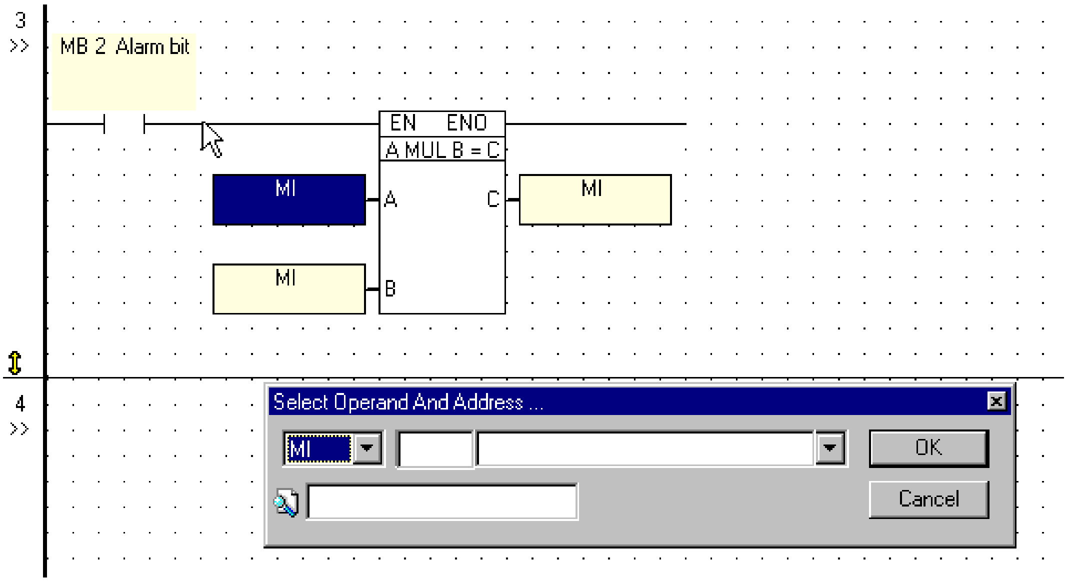
2. Select the desired operation.



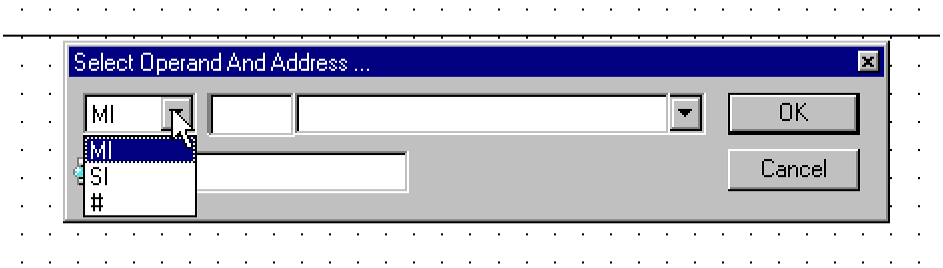
3. Move the function block to the desired net position.



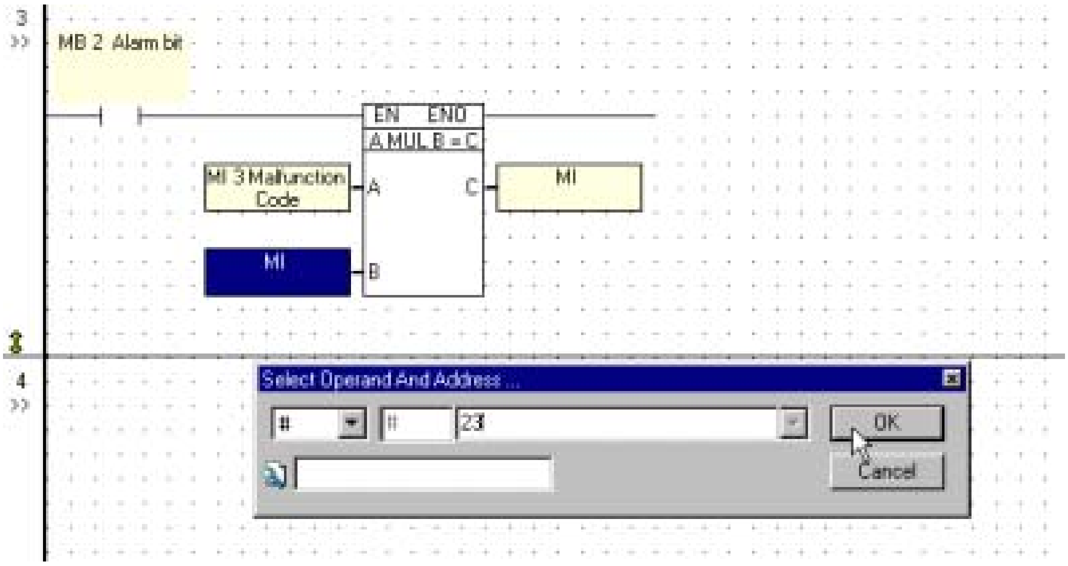
4. Click to place the element. The Operand Address and symbol dialog box opens.



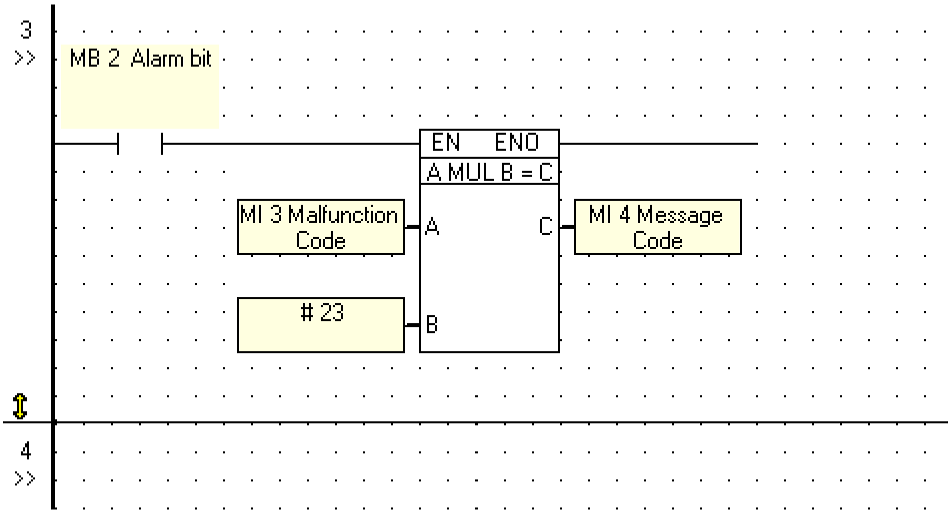
5. Select the desired Operand type.



6. Enter the Operand Address and symbol or constant value for each block variable. Click OK



7. The function block appears on the net with the selected block variable values and symbols.



Ladder Logic

You use Ladder Logic to write your project application. U90 Ladder is based on Boolean principals and follows IEC 1131-3 conventions.

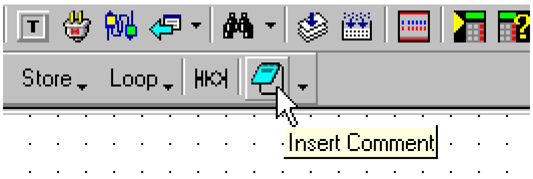
Ladder Diagrams are composed of different types of contact, coil and function block elements. In U90 Ladder, these elements are placed on nets.

In any Ladder Diagram, the contacts represent input conditions. They lead power from the left rail to the right rail. Coils represent output instructions. In order for output coils to be activated, the logical state of the contacts must allow the power to flow through the net to the coil.

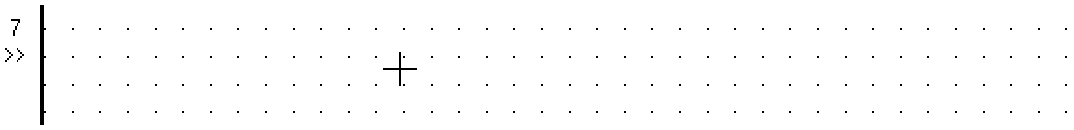
Comments

To insert comments:

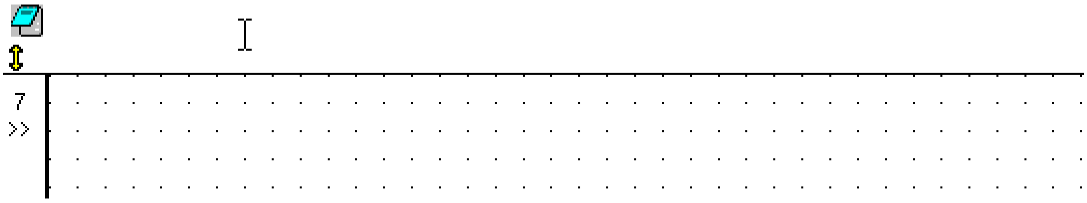
1. On the Ladder toolbar, click **Insert Comment** icon .



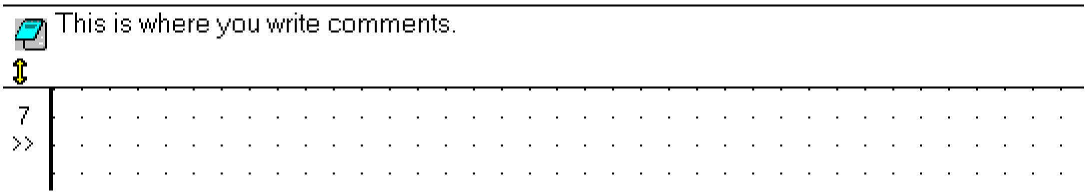
2. Move your cursor to the net in which you wish to insert a comment and click.



3. The Comment will appear above the net.



4. Type in your comments.

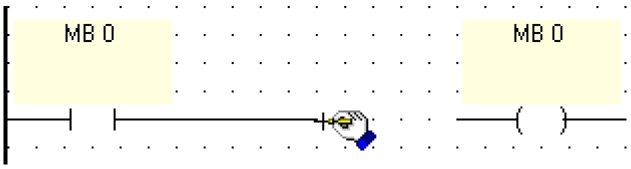


This is where you write comments.

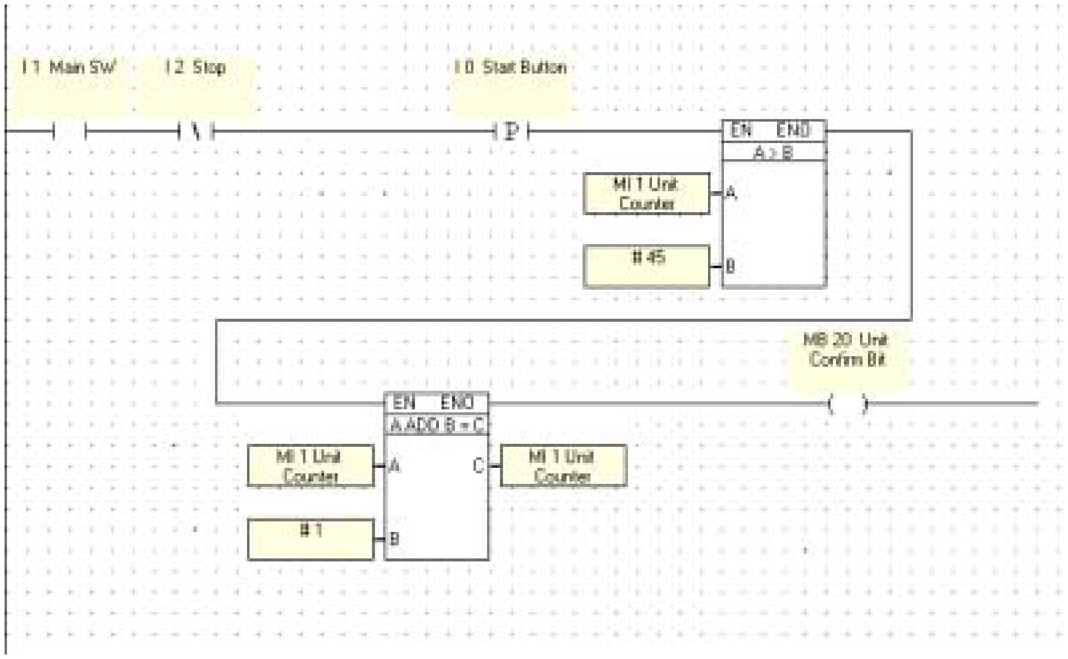
The length and content of your comments will have no effect on your project. They are not downloaded to the controller and do not affect the memory or word size of a project.

Connecting elements: Line Draw

Use the Line Draw tool to connect elements.

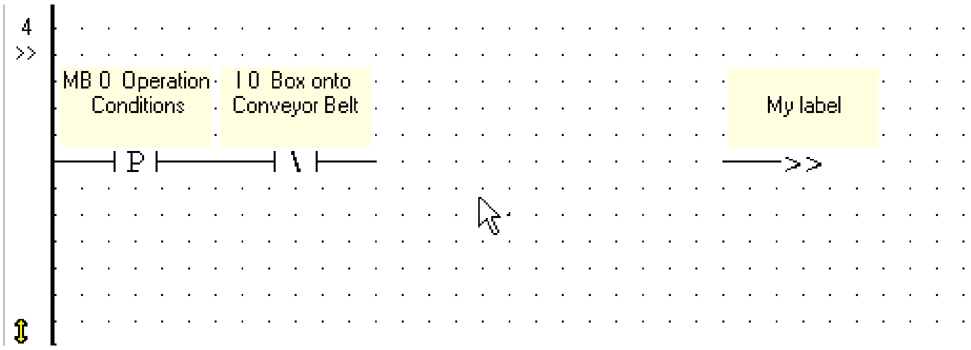


If you have a long series of elements in one net, you can use the Line Draw tool to extend the rung within the net.



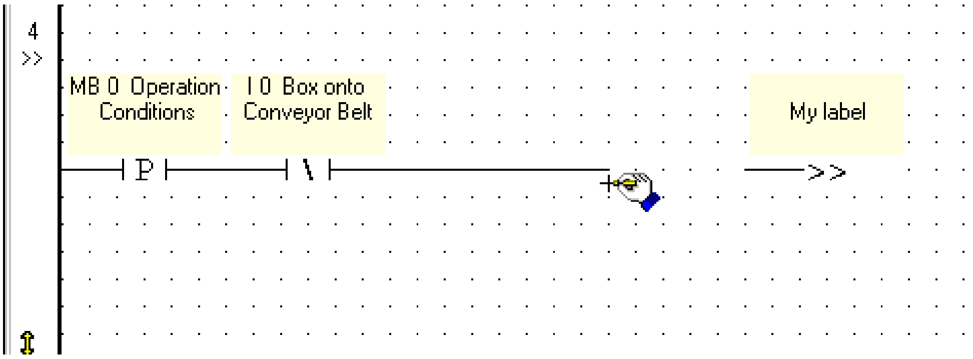
To use the Line Draw tool:

1. Place your cursor in the empty space in a net and double-click or click the Line Draw icon .



2. Your cursor changes into a drawing hand.

   Click and drag to draw the desired line to connect two elements in the net. Do not leave spaces between lines and elements. This may cause a compile error.



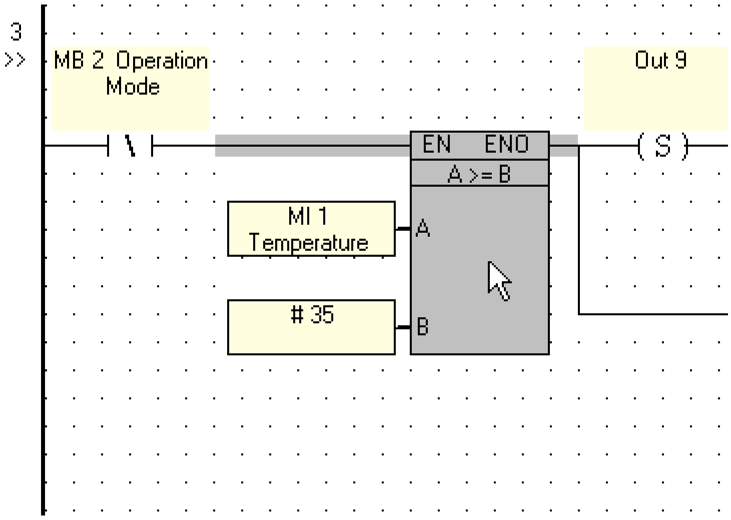
3.

Copy and Paste Elements

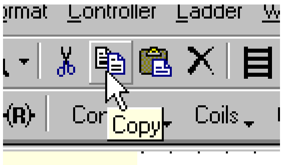
You can copy one or more elements from a net to paste into another net.

To Copy and Paste U90 Ladder elements in a net:

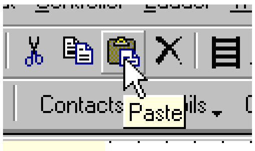
1. Select the element(s) you want to copy.



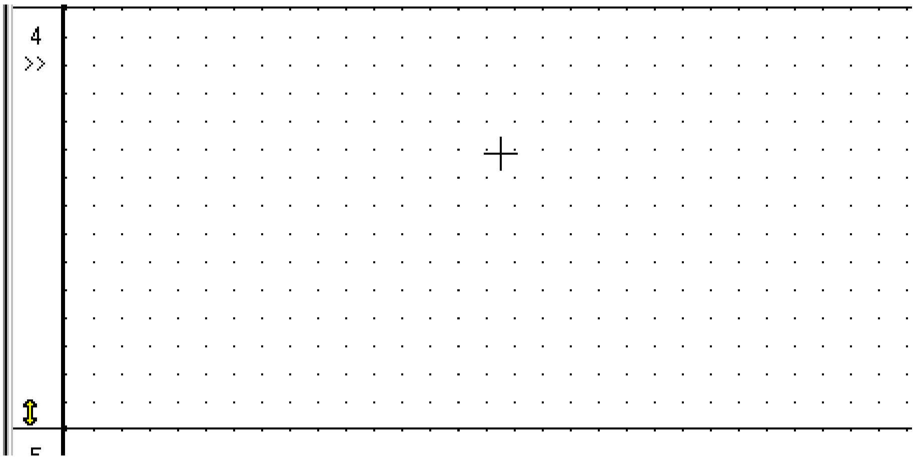
2. Select Copy.



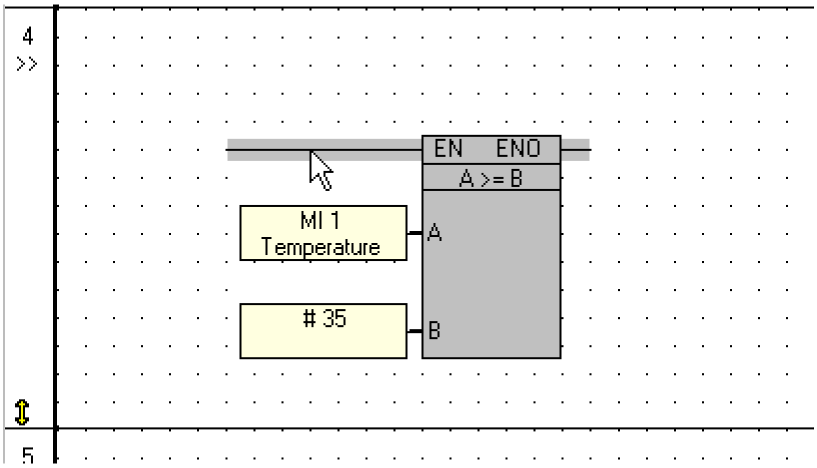
3. Select Paste.



4. Move the pointer to the net that you want to paste into



5. Click.

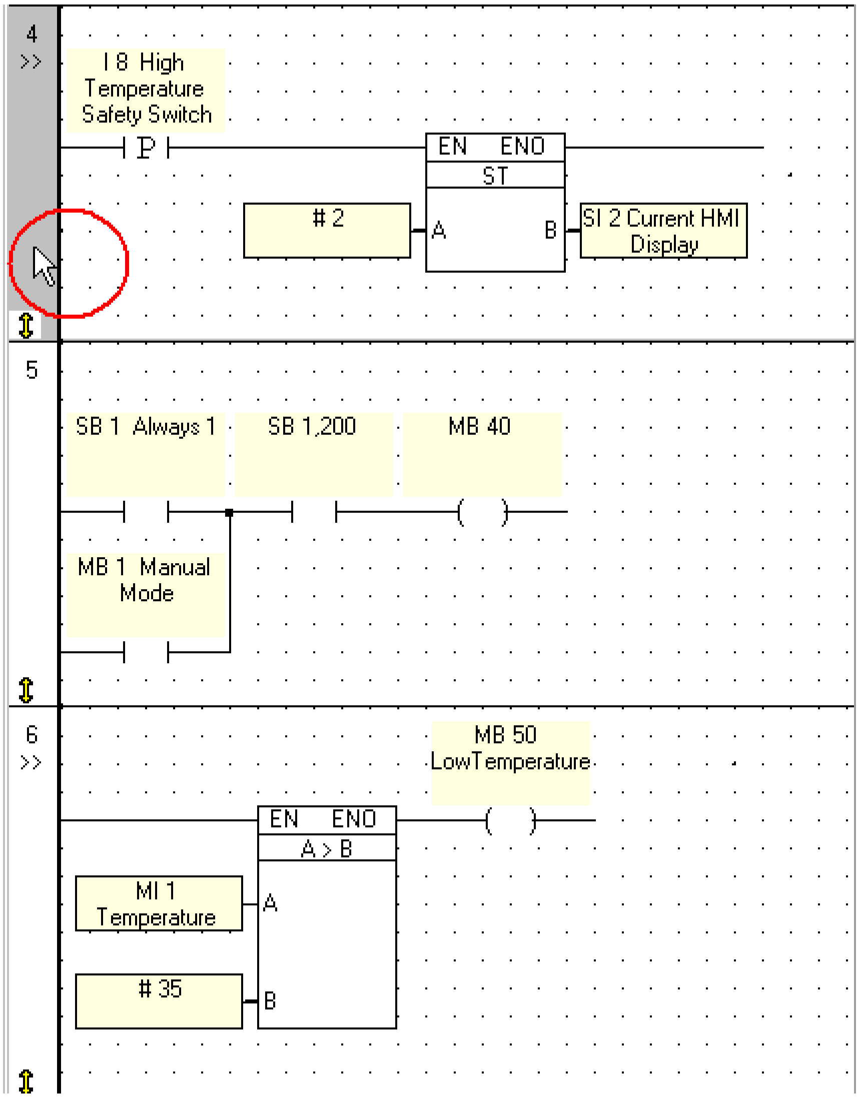


Note that the element(s) will appear in the same area in the new net as where they were in the copied net.

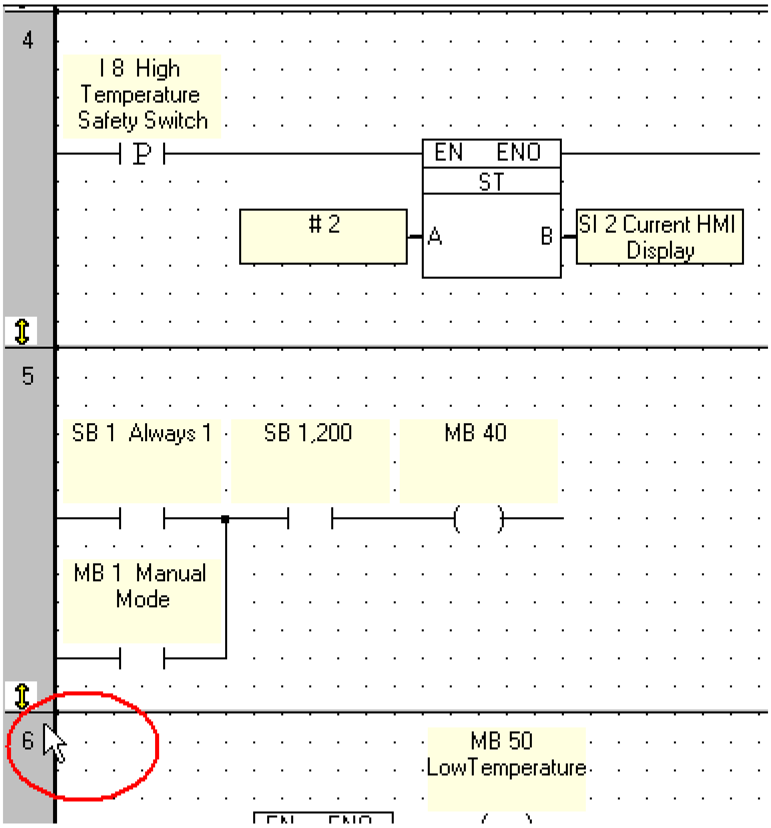
Copying multiple nets

To copy more than one net:

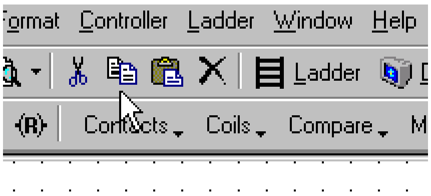
1. Select the first net by clicking on the left net bar.



2. Hold the **Shift** button and click on the last net in the range that you want to copy.



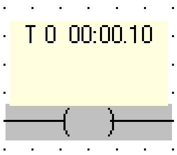
3. Click Copy on the Standard toolbar.



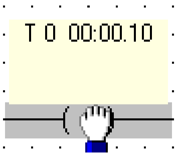
Moving Elements

To move an element within a net:

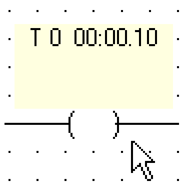
1. Select the element by single clicking on the element function (not the Operand and address area).



2. Hold the mouse button down. The cursor changes to a hand.



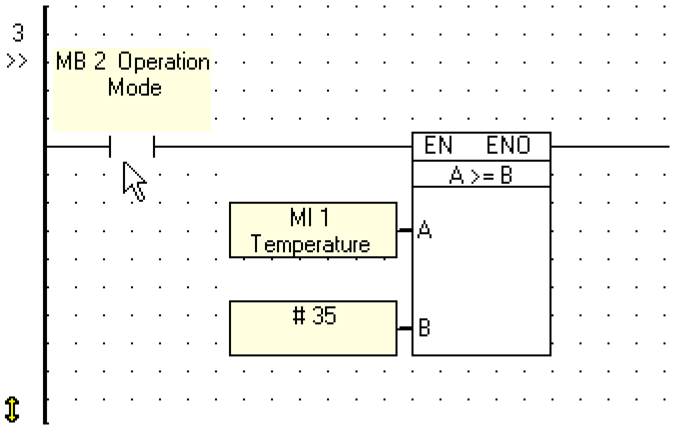
3. Move the mouse to re-position the element on the net. Release the mouse button.



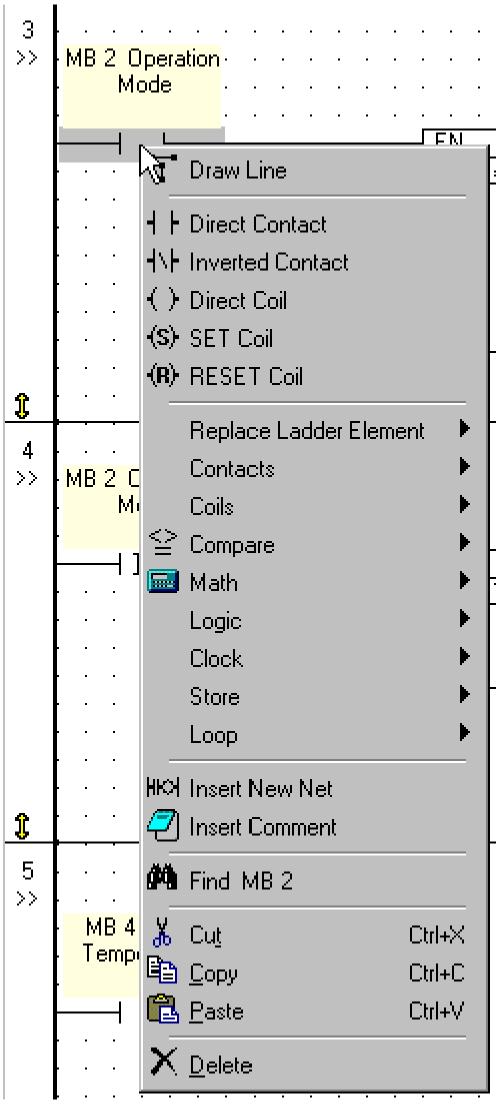
Replacing Ladder elements

To exchange one element for another within the same element family:

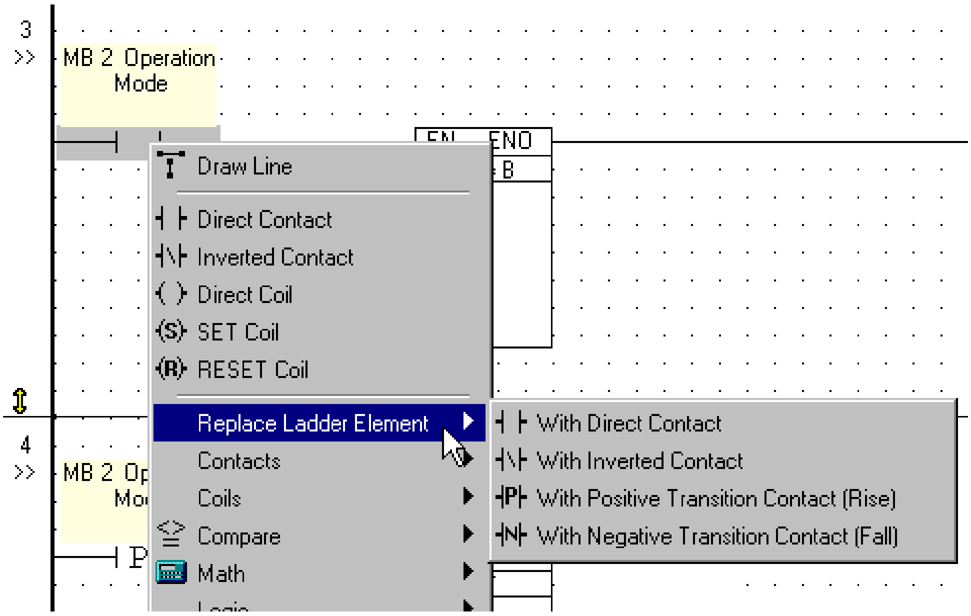
1. Select the element that you want to exchange.



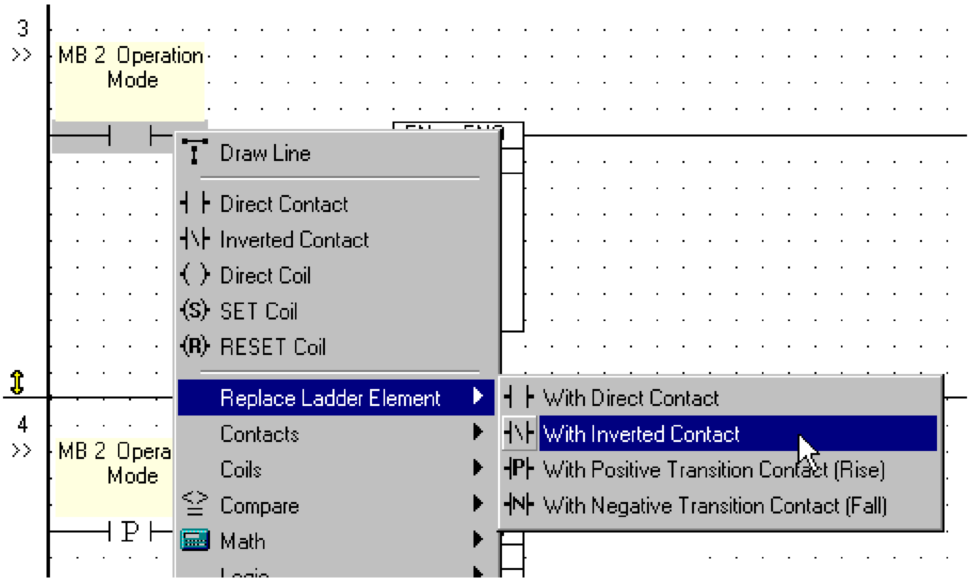
2. Right click to open the pop-up menu.



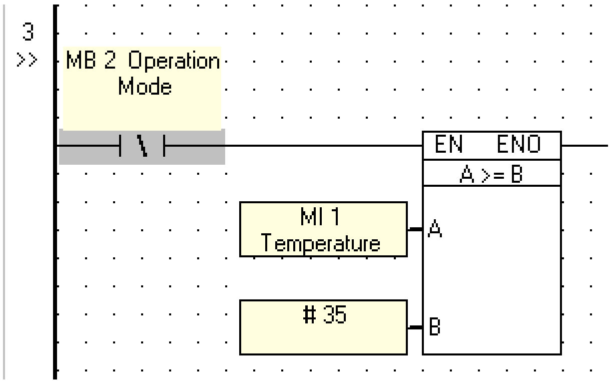
3. Select **Replace Ladder Element** option.



4. Select the desired replacement element type.

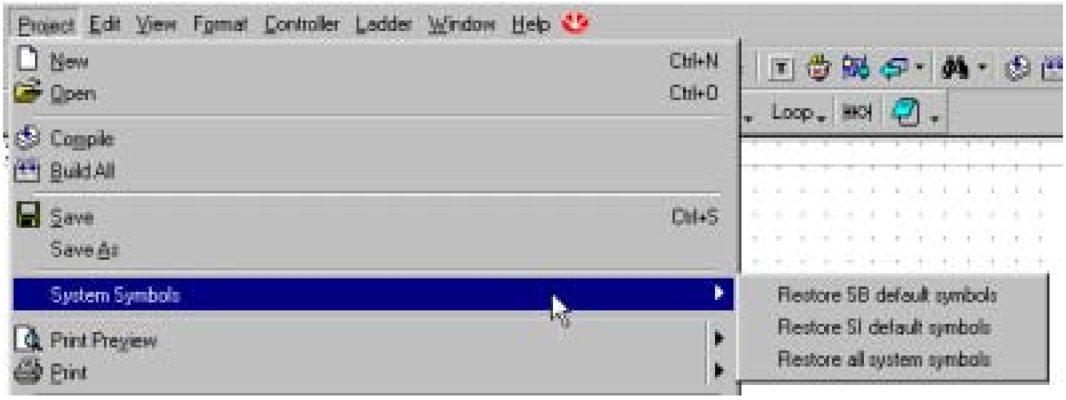


5. The element appears with the new element type.



Restoring System Symbols

To restore System Symbol values:


Project Edit View Format Controller Ladder Window Help
New Ctrl+N
Open Ctrl+O
Compile
Build All
Save Ctrl+S
Save As
System Symbols
Print Preview
Print
Loop
Restore SB default symbols
Restore SI default symbols
Restore all system symbols

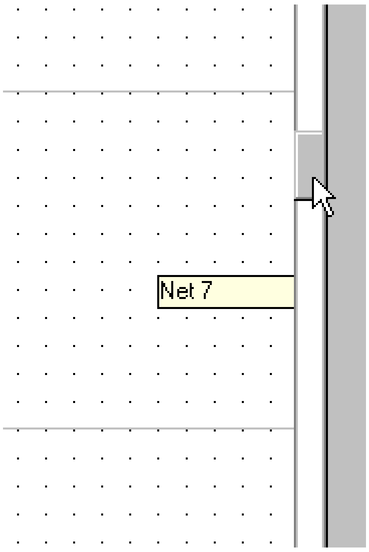
Keep in mind that there are SBs and SIs reserved for use by the system, such as SB 4 Divide by Zero or SI 4 Divide Remainder. Those SBs and SIs cannot be written into. If you accidentally write into them, you can recover their symbols.

Note that SBs and SIs are for system use. Even those currently 'blank' may be assigned a function in a later controller model. Writing into System Bits and System Integers is solely at the discretion of the programmer and the programmer is solely responsible for any problems that may arise as a result.

Scrolling between nets

To move quickly between nets:

1. Click on the scroll box.
2. Holding the mouse button down, drag the scroll box up or down to the desired net number.

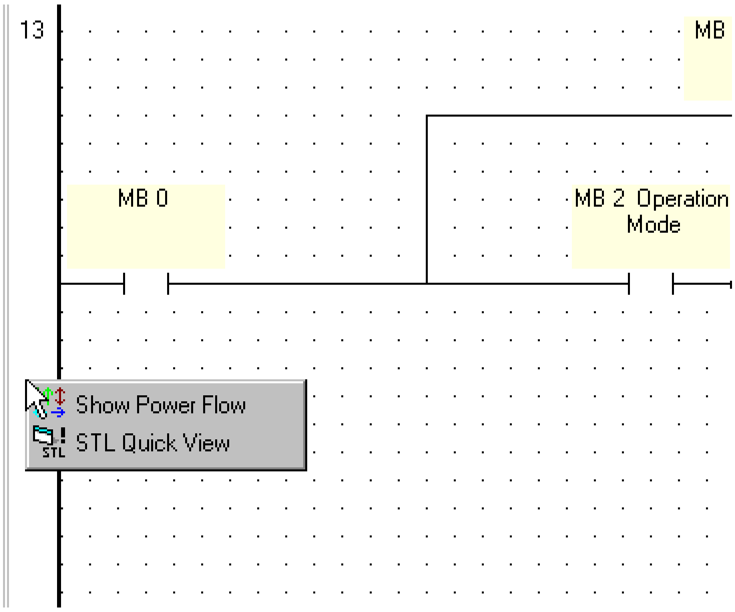


Viewing Logic Power Flow in a net

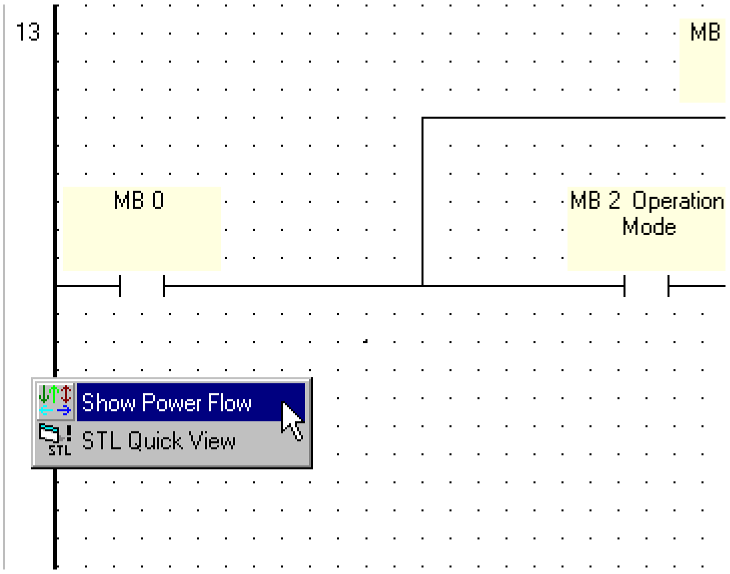
The **Show Power Flow** feature enables you to check the logic of a net you create.

You can see the Power Flow directions and, from these, how the net will work in the project.

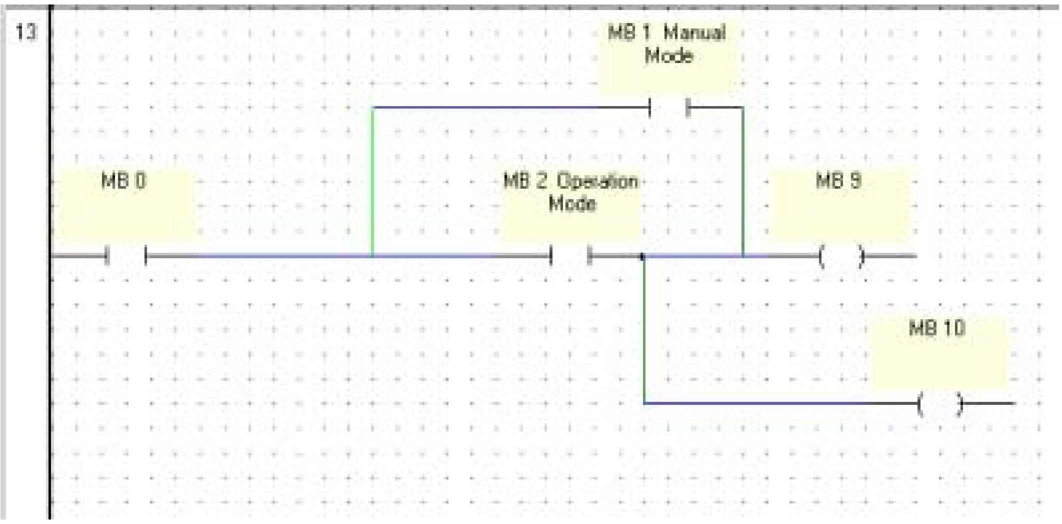
1. Right-click on the left Ladder bar. The Compiler Results menu appears.



2. Select Show Power Flow.



3. The net lines change color according to the power flow direction.



Each color represents a different direction of power flow.

Line Color	Power Direction
Dark Green	Down
Light Green	Up
Dark Blue	Left to Right
Light Blue	Right to Left
Red	Up or Down

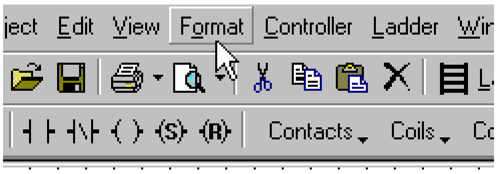
Displaying an Operand Symbol in the Ladder Diagram

In the Ladder Editor, you can view an element description as:

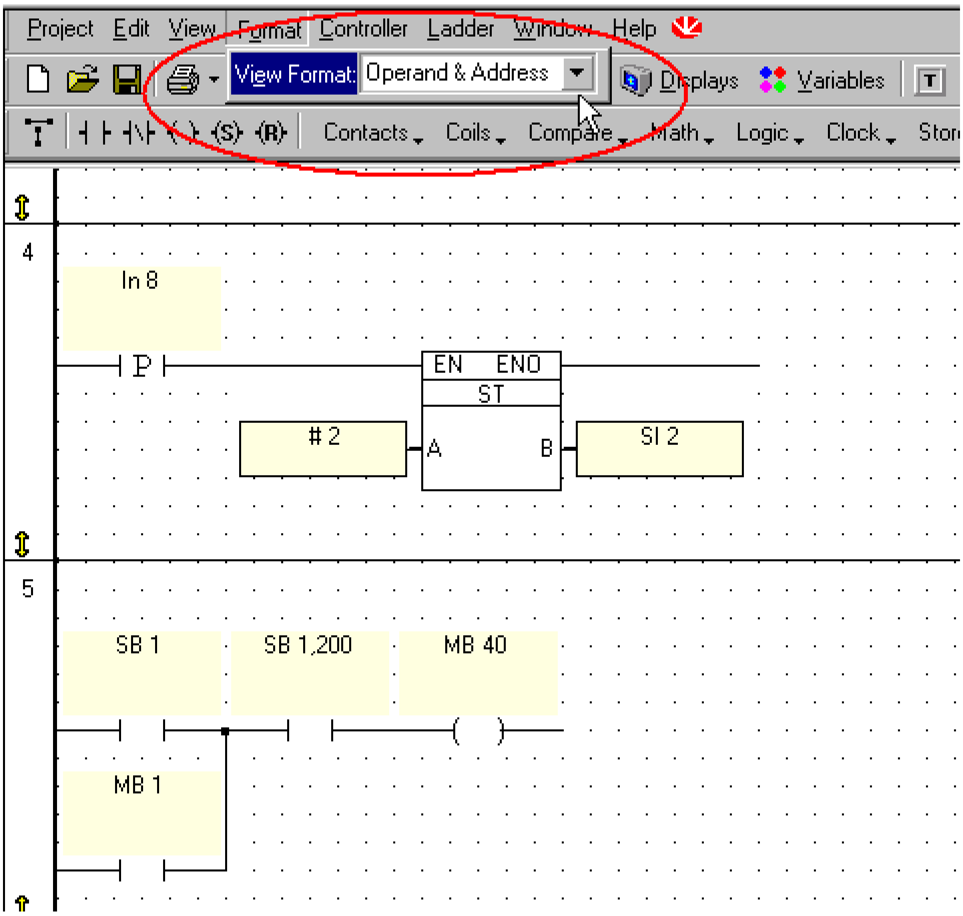
- An Operand and Address
- A Symbol
- Operand, Address and Symbol

To change the element description view format:

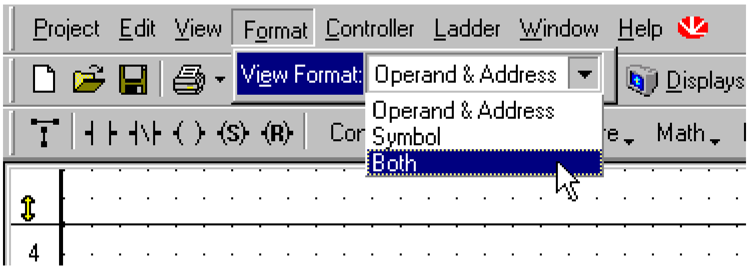
1. Click **Format** on the Standard menu bar.



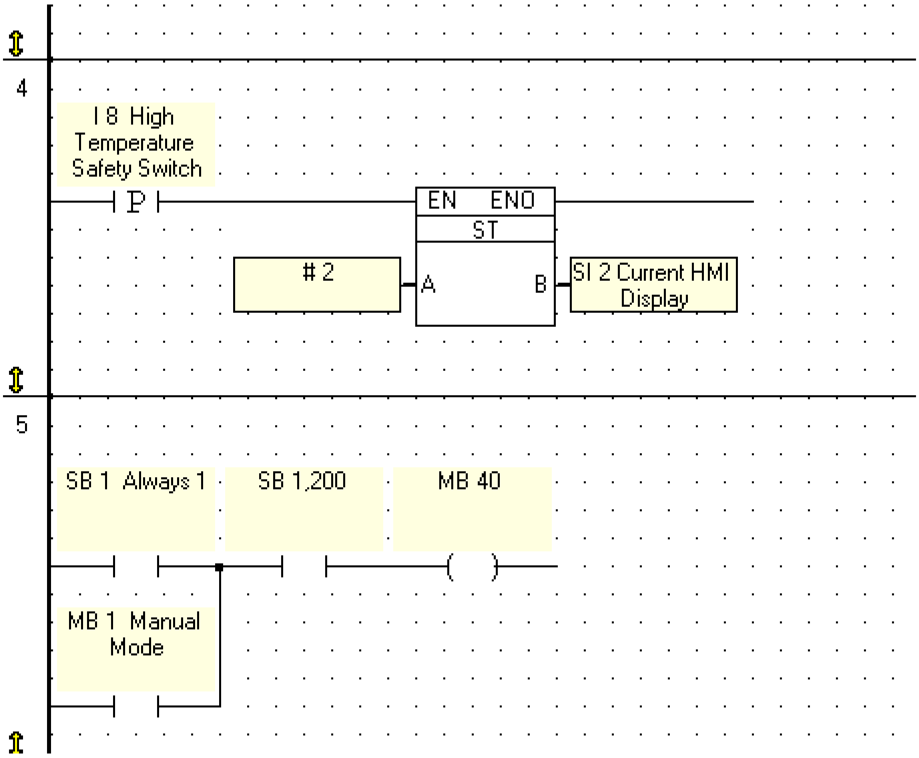
2. The **View Format** menu opens.


Project Edit View Format Controller Ladder Window Help
View Format: Operand & Address
Displays
Variables
Contacts Coils Compare Math Logic Clock Store
4
In 8
P
EN ENO
ST
# 2
A
B
SI 2
5
SB 1
SB 1,200
MB 40
MB 1

3. Select the desired view format.


Project Edit View Format Controller Ladder Window Help
View Format: Operand & Address
Displays
Operand & Address
Symbol
Both
Math
4

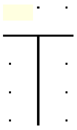
4. All of the Ladder elements appear with the selected view format.



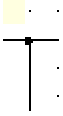
Intersecting lines: Junction

To check for junctions:

1. When you draw intersecting lines with the Line Draw tool, the line intersections are simply 90 degree angles.



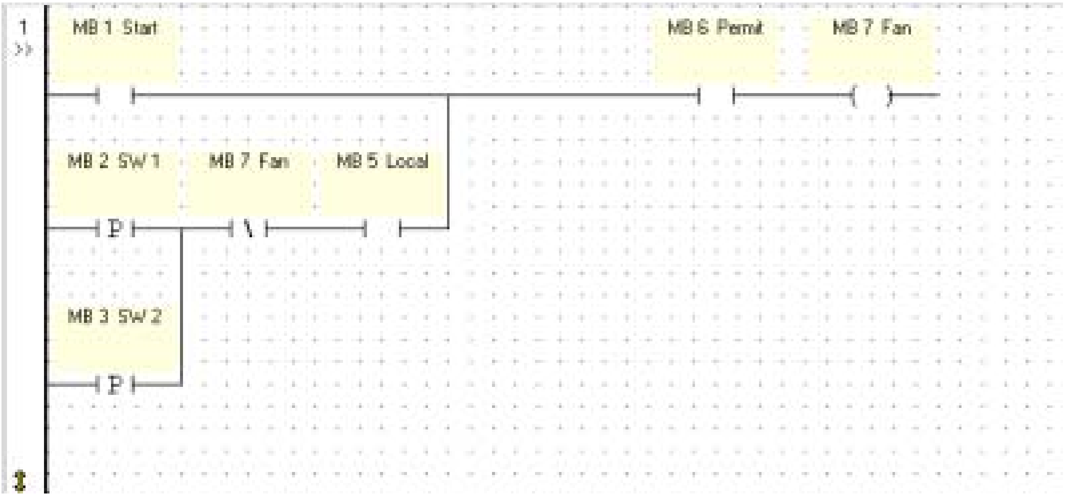
2. After compiling the project, there will be a small circle at each junction. This circle shows you that the compiler recognized these line intersections as junctions.



Ladder Nets with Feedbacks

According to IEC 1131 - 3, it is possible to create Ladder Diagram nets that contain feedback loops, i.e. where an element is used as both contact(s) and coil(s) in the same net.

In Ladder Diagram, all external input values such as those associated with contacts are gathered before each net is evaluated.



In the above example:

Where the net uses the state of its own output, the value of FAN (MB 7) coil associated with an inverted contact of MB 7 is always the value resulting from the previous evaluation.

However, if the value of FAN (MB 7) is used in any following nets, the latest evaluated state is used.

Elements

U90 Ladder Elements

Contacts	Icon
Direct Contact (NO)	┤ ├
Inverted Contact (NC)	┤\├
Positive Transition (Rise)	┤P├
Negative Transition (Fall)	┤N├

Coils	Icon
Direct Coil	( )
Inverted (negated) Coil	(\)
Set Coil	(S)
Reset Coil	(R)

Compare Functions	Icon
Greater Than	>

Greater/Equal	>=
Equal	=
Not Equal	<>
Less/Equal	<=
Less Than	<
Math Functions	Icon
Add	+
Subtract	-
Multiply	*
Divide	/
Logic Functions	
AND	
OR	
XOR	
Clock Functions	Icon
Time	🕒
Day Of Week	7
Day Of Month	31
Month	📅
Year	📅

Contacts

A contact represents an action or condition. A contact can be:

- Input
- Output
- Memory Bit
- System Bit
- Timer

Each contact condition in a net is loaded into the bit accumulator and evaluated to determine the coil (output or expression) condition. There are 4 types of contacts:

- Direct Contact
- Inverted Contact
- Positive Transition Contact (Rise or One Shot)
- Negative Transition Contact (Fall)

Contacts can be connected in both series and parallel on a U90 Ladder net.

To insert a Contact from the Ladder toolbar onto a Ladder net:

1. Click once to select the desired contact.
2. Move your mouse to the desired net position.
3. Click again.

There is no need to click and hold after selecting a contact.

Direct Contacts

A Direct Contact is a normally open contact condition. A Direct Contact condition can be:

- Input
- Output
- Memory Bit
- System Bit
- Timer

A Direct Contact condition can be an external input device (for example: a push button) or an internal input system element (for example: SB 50 Key **+/-** is pressed).

A door buzzer contains an example of a Direct Contact. When you push the buzzer, the buzzer sounds. When you release the buzzer, the sound stops.

During the system scan, the processor evaluates the program elements net by net.

If the Direct Contact address (the door buzzer) is OFF (logic 0): power will not flow through the Direct Contact. The door buzzer is silent.

If the Direct Contact address (the door buzzer) is ON (logic 1): power will flow through the Direct Contact. The door buzzer sounds.

Inverted Contacts

An Inverted Contact represents a normally closed contact condition. An Inverted Contact can be:

- Input
- Output
- Memory Bit
- System Bit
- Timer

An Inverted Contact condition can be from an external input device (for example: a push button) or from an internal input system element (for example: SB 50 Key **+/-** is pressed).

An emergency light contains an example of an Inverted Contact.

- Normally there is power flow through the emergency light's Inverted Coil and the light stays off.
- During an electric power outage, the power flow through the Inverted Coil stops and the emergency light comes on.

During the system scan, the processor evaluates the program elements net by net.

If the Inverted Contact address (power supply) is ON (logic 1): power will flow through the Inverted Contact. The emergency light will stay off.

If the Inverted Contact address (power supply) is OFF (logic 0): power will not flow through the Inverted Contact. The emergency light comes on.

If the power outage ends and power flow is returned to the Inverted Contact, it will close again and the emergency light will go off again.

Negative Transition Contact

A Negative Transition Contact gives a single one-shot pulse when its reference address falls from ON (logic 1) to OFF (logic 0). A Negative Transition Contact is registering the **fall** in status from ON to OFF.

A Negative Transition Contact condition can be:

- Input
- Output
- Memory Bit
- System Bit
- Timer

A computer ON/OFF button is an example of a Negative Transition Contact. The computer is ON.

If you push the ON/OFF button in without releasing it - the computer will not shut off. Only when you release the button will the system register a change in status from ON to OFF and the computer will shut OFF.

During the system scan, a Negative Transition Contact address is evaluated for a transition from ON to OFF. A transition allows power to flow through the Negative Transition Contact for one scan.

At the end of the one scan, the Negative Transition Contact is reset to OFF (logic 0). Only after the triggering signal turns from OFF to ON again is there the possibility for the Negative Transition Contact to be re-activated by the next falling transition from ON to OFF.

Positive Transition Contact

A Positive Transition Contact gives a single one shot pulse when its address reference rises from OFF (logic 0) to ON (logic 1). A Positive Transition Contact is registering the **change** in status from OFF to ON. The length of the ON status is not relevant.

A Positive Transition Contact condition can be:

- Input
- Output
- Memory Bit
- System Bit
- Timer

A cellular phone keypad key is an example of a Positive Transition Contact. When you push a key a number is displayed on the screen. It does not matter if you push the key quickly or hold it down for several seconds. The number will only appear once on the screen.

The cellular phone registers the transition from no key pressed to a key pressed. The **length** of time the key is pressed is not relevant. You must release the key and press it again to repeat the number on the cellular phone screen.

During the system scan, a Positive Transition Contact address is evaluated for a transition from OFF to ON. A transition allows power to flow through the Positive Transition Contact for one scan.

At the end of the one scan the Positive Transition Contact is reset to OFF (logic 0) even if the triggering signal stays on. Only **afte**r the triggering signal turns from ON back to OFF is there the possibility for the Positive Transition Contact to be activated again with a rise from OFF to ON.

Coils

A coil represents a result or expression of an action. A coil can be:

- Memory Bit
- System Bit
- Output
- Timer

Each contact condition is evaluated in a net to determine the coil (result or expression) condition. There are 4 types of coils:

- Direct Coil
- Inverted Coil
- Set Coil
- Reset Coil

Recommended: Do not energize a coil more than once in a program.

To insert a Coil from the Ladder toolbar onto a Ladder net:

1. Click once to select the desired coil.
2. Move your mouse to the desired net position.
3. Click again.

There is no need to click and hold after selecting a coil. Note that, while the Direct, Set and Reset Coils are available on every menu, the Inverted Coil is not.

Direct Coil

A Direct Coil represents a direct result instruction of the conditions (contacts and/or function blocks) on the Ladder net before the Direct Coil. A Direct Coil instruction can be:

- Output
- Memory Bit
- System Bit
- Timer

The coil result can go to an external output device (for example: a light) or an internal system element (for example: SB 2 Power Up Bit).

A door buzzer contains an example of a Direct Coil. When the door buzzer button (Direct Contact) is pushed the door buzzer (Direct Coil) sounds. When you release the buzzer the sound stops.

During the system scan, the processor evaluates all of the program elements on the Ladder net before the Direct Coil for power flow continuity.

If no power flow continuity exists in the net (the door buzzer button is not pushed): the Direct Coil address instruction is OFF (logic 0). The door buzzer does not sound.

If power flow continuity exists in the net (the door buzzer button is pressed): the Direct Coil address instruction is ON (logic 1). The door buzzer sounds.

Inverted Coil

An Inverted Coil represents the opposite result instruction of the conditions (contacts and/or function blocks) on the Ladder net before the Inverted Coil. An Inverted Coil instruction can be:

- Output
- Memory Bit

- System Bit

The result instruction can be to an external output device (for example: alarm bell) or to an internal system element (for example: SB 80 activate linearization).

During the system scan, the processor evaluates all of the program elements on the Ladder net before the Inverted Coil for power flow continuity.

If no power flow continuity exists in the net: the Inverted Coil address instruction is ON (logic 1).

If power flow continuity exists in the net: the Inverted Coil address is OFF (logic 0).

Set Coil

A set coil separates the coil from the action or condition that energized the coil. Once energized, a set coil's result is no longer dependant on the action that energized it. A set coil stays energized (latched) until its condition is reset (unlatched) by a reset coil.

A set coil can be:

- Memory Bit
- System Bit
- Output

An example of a set coil is an overhead light. When we turn on a light it stays lit until we turn it off (reset or unlatch it) or the light bulb burns out. Luckily, you do not have to hold the light switch to keep the light on.

An example of a coil that you do **not** want to be set (latched) is a car horn. You expect it to toot only when you press on the horn button and you expect it to stop when you stop pressing on the horn button.

Use set and reset coils to preserve a condition in a program.

Reset Coil

A reset coil turns off (unlatches) a set coil, provided that there is logic continuity to the reset coil. Once a set coil is energized it stays energized, **independent** of the original set condition, until a reset coil with the same address resets (unlatches) the coil condition.

A reset coil can be:

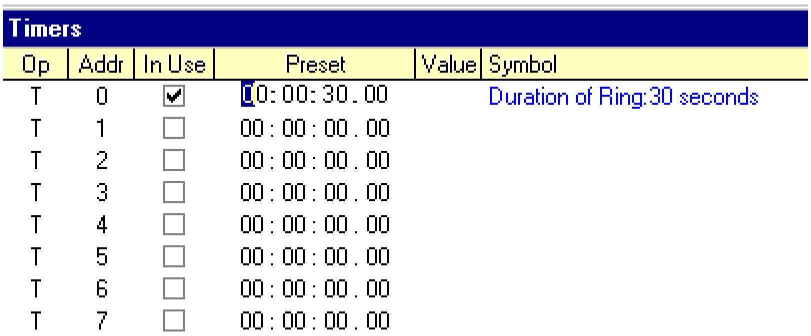
- Memory Bit
- System Bit
- Output

Do not use a set coil without a reset coil in a program.

Timers (T)

U90 Ladder offers 64 On Delay Timers. Timers have a preset value, a current value, and a bit value. Timers always count **down** from the Preset Value.

Click on the Timers folder in the Program Navigation pane to display the complete list of Timers. Scroll down to view the complete list.



Timers

Op	Addr	In Use	Preset	Value	Symbol
T	0	☑	00:00:30.00		Duration of Ring:30 seconds
T	1	□	00 : 00 : 00 . 00		
T	2	□	00 : 00 : 00 . 00		
T	3	□	00 : 00 : 00 . 00		
T	4	□	00 : 00 : 00 . 00		
T	5	□	00 : 00 : 00 . 00		
T	6	□	00 : 00 : 00 . 00		
T	7	□	00 : 00 : 00 . 00		

To place a Timer in your program, place a direct coil in a net, and select T.

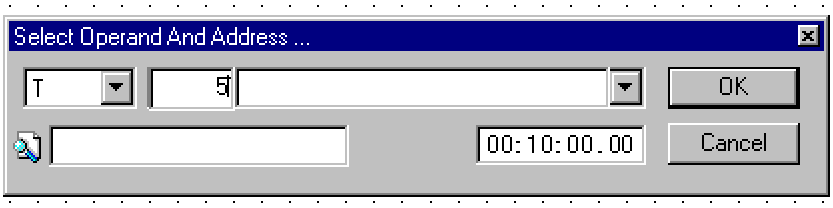
Note that a Timer value can be displayed in a Display as a current or elapsed value.

Setting Timers

To set a Timer's time:

After selecting the Timer's Address, the Timer value field is activated.

1. Enter the time.

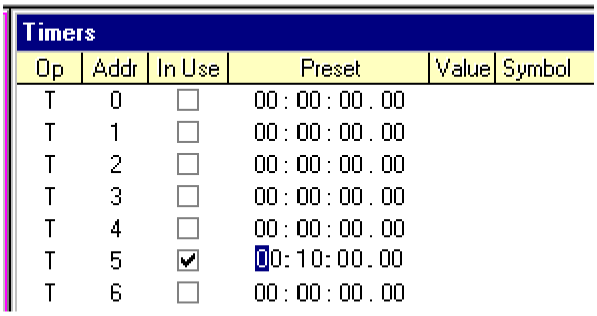


Select Operand And Address ...

T 5

00:10:00.00

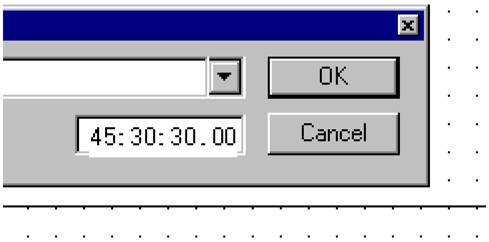
2. You can also write the time into a Timer via the Timer list window



Timers

Op	Addr	In Use	Preset	Value	Symbol
T	0	□	00 : 00 : 00 . 00		
T	1	□	00 : 00 : 00 . 00		
T	2	□	00 : 00 : 00 . 00		
T	3	□	00 : 00 : 00 . 00		
T	4	□	00 : 00 : 00 . 00		
T	5	☑	00:10:00.00		
T	6	□	00 : 00 : 00 . 00		

A Timer's maximum preset value is:



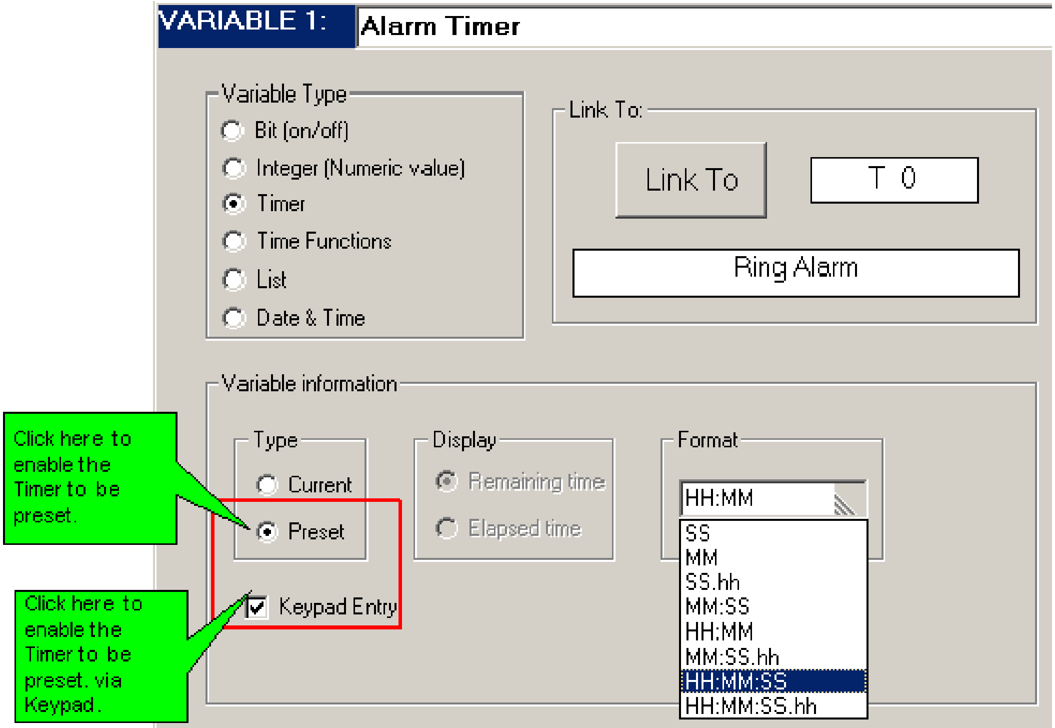
45:30:30.00

Note that the time format is: **HH:MM:SS.hh**.

A timer can also be set via the M90 keypad.

Presetting Timers via Keypad

You can choose to set a timer via the M90 keypad.



VARIABLE 1: Alarm Timer

Variable Type
- ○ Bit (on/off)
- ○ Integer (Numeric value)
- ● Timer
- ○ Time Functions
- ○ List
- ○ Date & Time

Link To:
Link To: T 0
Ring Alarm

Variable information

Type
- ○ Current
- ● Preset

☑ Keypad Entry

Display
- ● Remaining time
- ○ Elapsed time

Format: HH:MM
SS
MM
SS.hh
MM:SS
HH:MM
MM:SS.hh
HH:MM:SS
HH:MM:SS.hh

Click here to enable the Timer to be preset.

Click here to enable the Timer to be preset. via Keypad.

Operands

Operands

An element's Operand is the form in which information is stored and operated on in the U90 Ladder program.

Operand lists are organized in categories, according to operand type:

- Input: **I** (according to model and expansion)
- Output: **O** (according to model and expansion)
- Memory Bit: **MB** (0 - 255)
- Memory Integer: **MI** (0 - 255)
- System Bit: **SB** (0 - 255)
- System Integer: **SI** (0 - 255)
- Timer:**T** (0 - 63)

Every Operand has an Address and a Symbol.

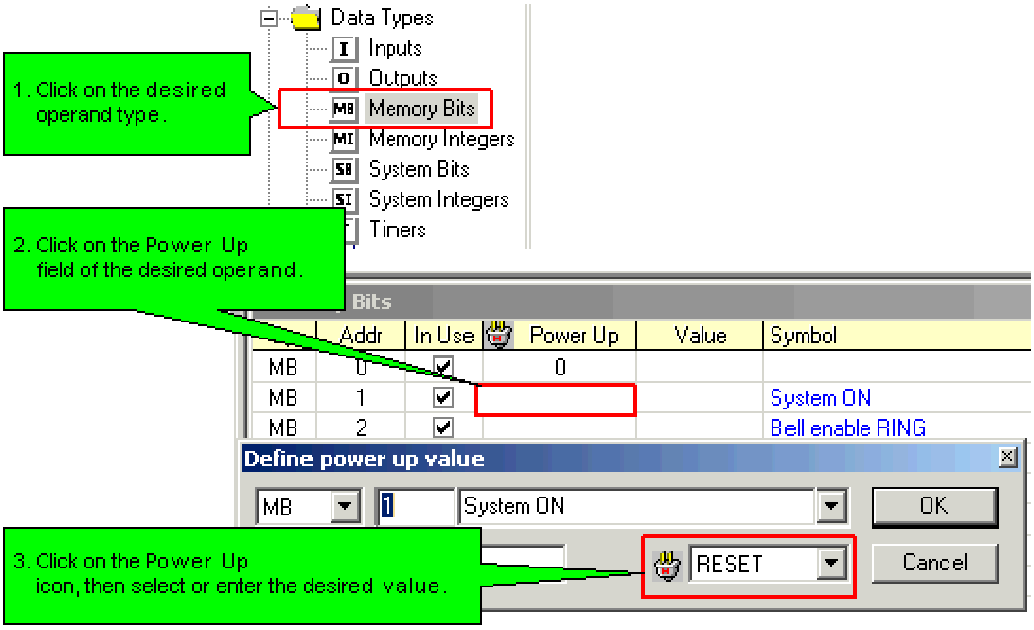
Symbols appear together with the operand every time the operand and address are used in the program. There are two types of symbols: preset and user-created.

- Preset symbols are descriptions that are connected to System Bits and System Integers.
- User-created symbols are descriptions that are written by the user for a specific project application. The user assigns a particular description to a particular operand.

Power-up

You can assign Power Up values to most Data Types. These values are written into the operand by the program when the controller is turned on. Outputs, MBs, SBs can be set or reset; integer values can be written into MIs and SIs.

You can assign Power Up values when you place an element into a net, or by opening a Data Type list as shown below.



	Addr	In Use	Power Up	Value	Symbol
MB	0	☑	0		
MB	1	☑			System ON
MB	2	☑			Bell enable RING

MB 1 System ON

1. Click on the desired operand type.

2. Click on the Power Up field of the desired operand.

3. Click on the Power Up icon, then select or enter the desired value.

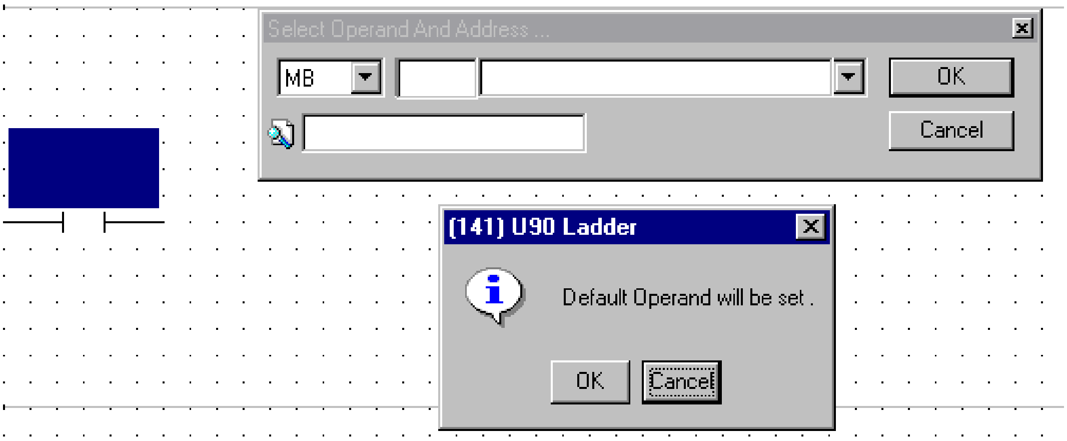
Operand Addressing

An Operand Address is the physical location in the M90 memory where the element information is stored.

For example:

- MB 10 - "10" is the address of the MB Operand
- MI 35 - "35" is the address of the MI Operand
- T 12 - "12" is the address of the Timer Operand

U90 Ladder allows you to create your own symbols before you write your program. This feature can help you to organize your project properly from the very beginning. You can also create symbols as you write your program. Symbols can be edited after you create them. Note that there is a default address setting for each operand type. The Default message box will appear if you do not specify an address:



Default Operand will be set .

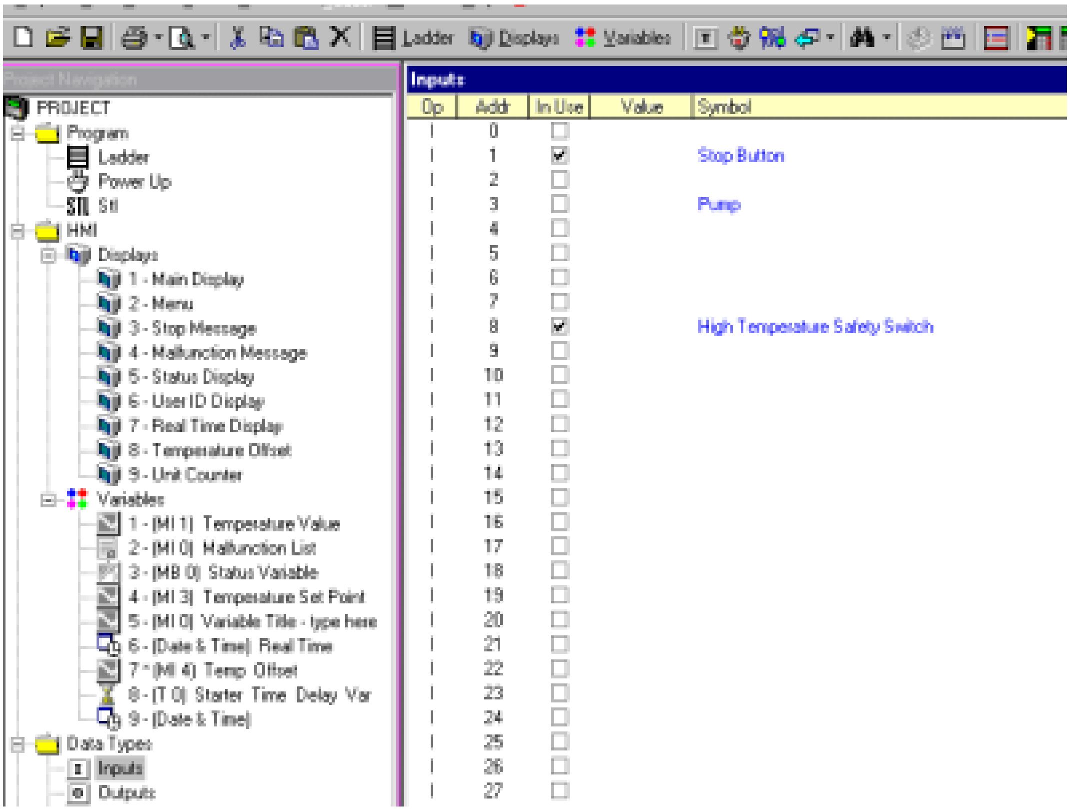
Inputs (I)

Inputs are one Operand type available for writing a project application.

The number of Inputs varies according to the M90 model and I/O expansion rack.

An Input is an actual hardwired input connection into the controller.

Click on the Inputs folder in the Program Navigation pane to display the complete list of Inputs. Scroll down to view the complete list



PROJECT
- Program
  - Ladder
  - Power Up
  - Stl
- HMI
  - Displays
    - 1 - Main Display
    - 2 - Menu
    - 3 - Stop Message
    - 4 - Malfunction Message
    - 5 - Status Display
    - 6 - User ID Display
    - 7 - Real Time Display
    - 8 - Temperature Offset
    - 9 - Unit Counter
  - Variables
    - 1 - (MI 1) Temperature Value
    - 2 - (MI 0) Malfunction List
    - 3 - (MB 0) Status Variable
    - 4 - (MI 3) Temperature Set Point
    - 5 - (MI 0) Variable Title - type here
    - 6 - (Date & Time) Real Time
    - 7 * (MI 4) Temp Offset
    - 8 - (T 0) Starter Time Delay Var
    - 9 - (Date & Time)
- Data Types
  - Inputs
  - Outputs

Inputs

Op	Addr	In Use	Value	Symbol
I	0	☐		
I	1	☑		Stop Button
I	2	☐		
I	3	☐		Pump
I	4	☐		
I	5	☐		
I	6	☐		
I	7	☐		
I	8	☑		High Temperature Safety Switch
I	9	☐		
I	10	☐		
I	11	☐		
I	12	☐		
I	13	☐		
I	14	☐		
I	15	☐		
I	16	☐		
I	17	☐		
I	18	☐		
I	19	☐		
I	20	☐		
I	21	☐		
I	22	☐		
I	23	☐		
I	24	☐		
I	25	☐		
I	26	☐		
I	27	☐		

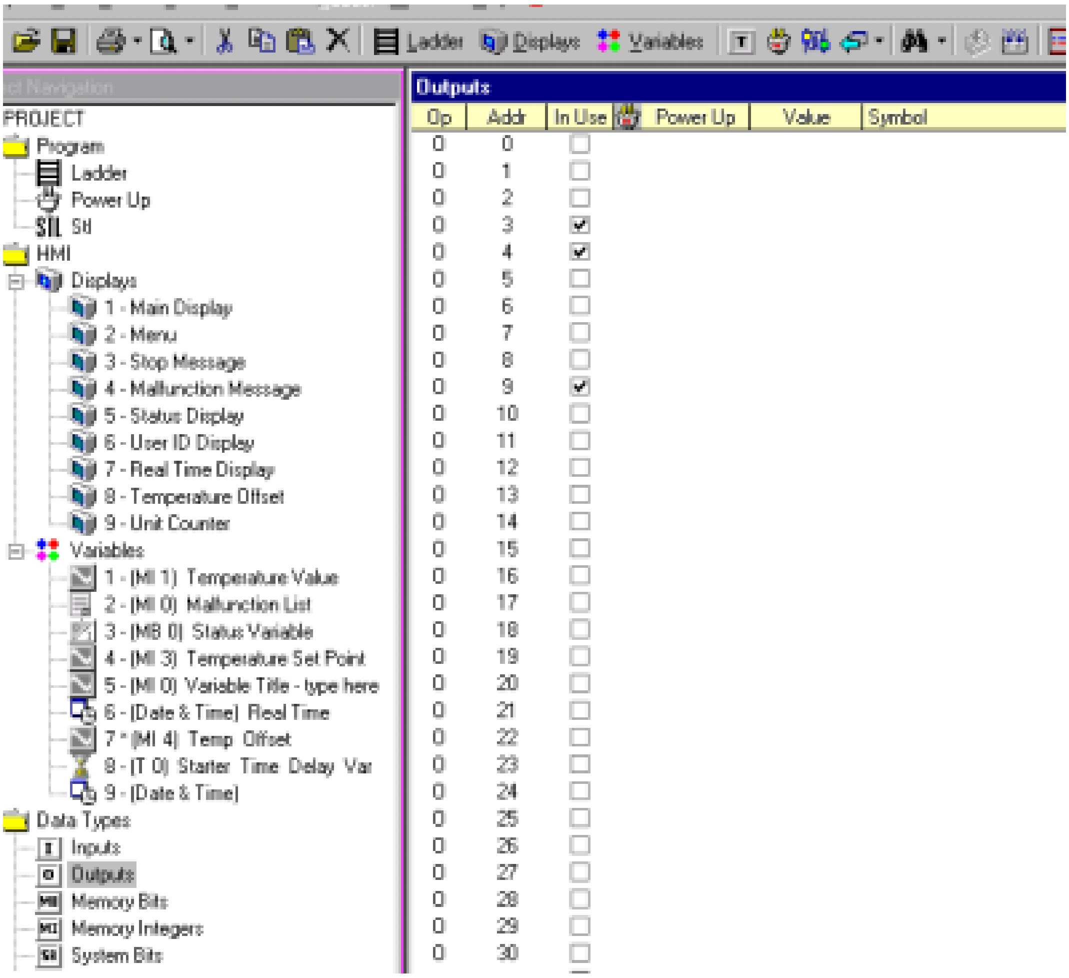
Outputs (O)

Outputs are one Operand type available for writing a project application.

The number of Outputs varies according to the M90 model and I/O expansion module.

An Output is an actual hardwired output connection from the controller.

Click on the Outputs folder in the Program Navigation pane to display the complete list of Outputs. Scroll down to view the complete list



- PROJECT
  - Program
    - Ladder
    - Power Up
    - Stl
  - HMI
    - Displays
      - 1 - Main Display
      - 2 - Menu
      - 3 - Stop Message
      - 4 - Malfunction Message
      - 5 - Status Display
      - 6 - User ID Display
      - 7 - Real Time Display
      - 8 - Temperature Offset
      - 9 - Unit Counter
    - Variables
      - 1 - [MI 1) Temperature Value
      - 2 - [MI 0) Malfunction List
      - 3 - [MB 0] Status Variable
      - 4 - [MI 3) Temperature Set Point
      - 5 - [MI 0) Variable Title - type here
      - 6 - [Date & Time] Real Time
      - 7 - [MI 4] Temp Offset
      - 8 - [T 0] Starter Time Delay Var
      - 9 - [Date & Time]
  - Data Types
    - Inputs
    - Outputs
    - Memory Bits
    - Memory Integers
    - System Bits

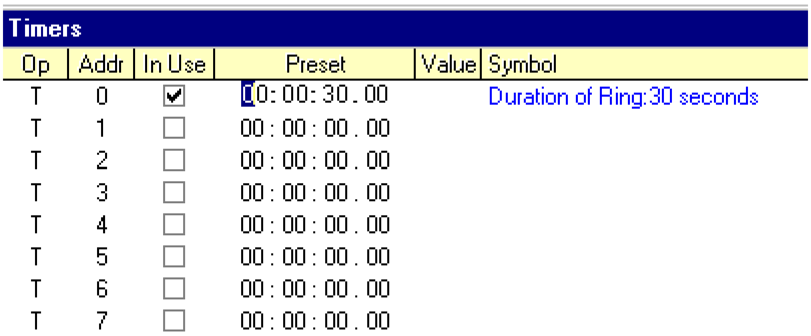
Outputs

Op	Addr	In Use	Power Up	Value	Symbol
O	0	□			
O	1	□			
O	2	□			
O	3	☑			
O	4	☑			
O	5	□			
O	6	□			
O	7	□			
O	8	□			
O	9	☑			
O	10	□			
O	11	□			
O	12	□			
O	13	□			
O	14	□			
O	15	□			
O	16	□			
O	17	□			
O	18	□			
O	19	□			
O	20	□			
O	21	□			
O	22	□			
O	23	□			
O	24	□			
O	25	□			
O	26	□			
O	27	□			
O	28	□			
O	29	□			
O	30	□			

Timers (T)

U90 Ladder offers 64 On Delay Timers. Timers have a preset value, a current value, and a bit value. Timers always count **down** from the Preset Value.

Click on the Timers folder in the Program Navigation pane to display the complete list of Timers. Scroll down to view the complete list.



Timers

Op	Addr	In Use	Preset	Value	Symbol
T	0	☑	00: 00: 30. 00		Duration of Ring:30 seconds
T	1	□	00 : 00 : 00 . 00		
T	2	□	00 : 00 : 00 . 00		
T	3	□	00 : 00 : 00 . 00		
T	4	□	00 : 00 : 00 . 00		
T	5	□	00 : 00 : 00 . 00		
T	6	□	00 : 00 : 00 . 00		
T	7	□	00 : 00 : 00 . 00		

To place a Timer in your program, place a direct coil in a net, and select T.

Note that a Timer value can be displayed in a Display as a current or elapsed value.

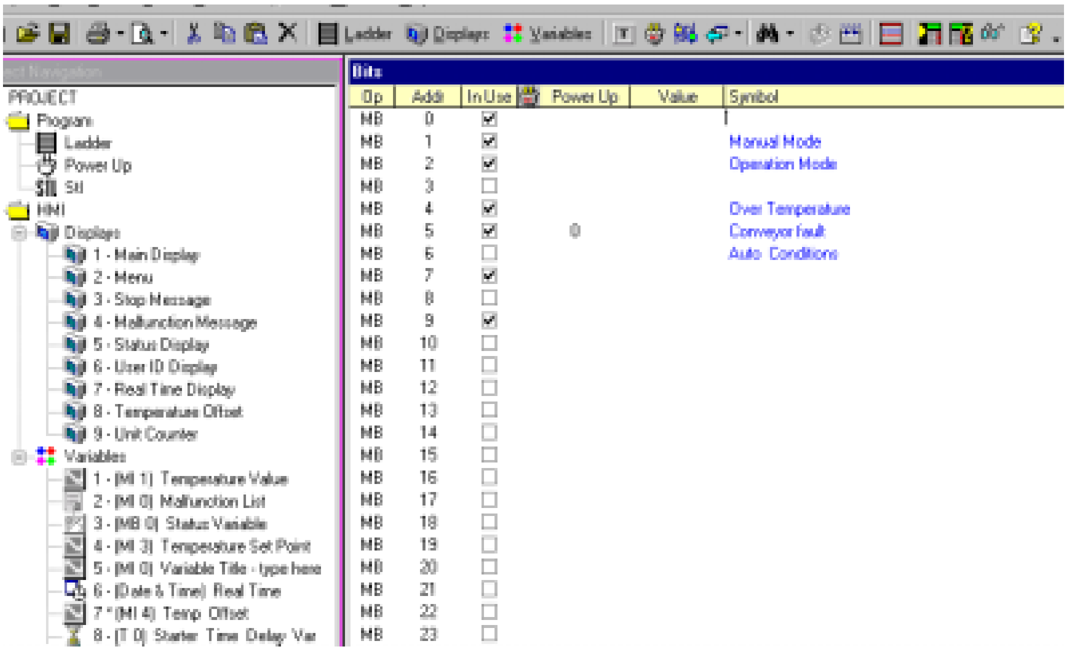
Memory Bits (MB)

Memory Bits are one Operand type available for writing a project application.

There are 256 MBs (Address MB 0 - MB 255).

Memory Bits hold a bit value (0 or 1).

Click on the Memory Bits folder in the Program Navigation pane to display the complete list of Memory Bits. Scroll down to view the complete list



- PROJECT
  - Program
    - Ladder
    - Power Up
    - Stl
  - HMI
    - Displays
      - 1 - Main Display
      - 2 - Menu
      - 3 - Stop Message
      - 4 - Malfunction Message
      - 5 - Status Display
      - 6 - User ID Display
      - 7 - Real Time Display
      - 8 - Temperature Offset
      - 9 - Unit Counter
    - Variables
      - 1 - [MI 1] Temperature Value
      - 2 - [MI 0] Malfunction List
      - 3 - [MB 0] Status Variable
      - 4 - [MI 3] Temperature Set Point
      - 5 - [MI 0] Variable Title - type here
      - 6 - [Date & Time] Real Time
      - 7 *[MI 4] Temp Offset
      - 8 - [T 0] Starter Time Delay Var

Bits

Op	Addr	In Use	Power Up	Value	Symbol
MB	0	☑			
MB	1	☑			Manual Mode
MB	2	☑			Operation Mode
MB	3	□			
MB	4	☑			Over Temperature
MB	5	☑	0		Conveyor fault
MB	6	□			Auto Conditions
MB	7	☑			
MB	8	□			
MB	9	☑			
MB	10	□			
MB	11	□			
MB	12	□			
MB	13	□			
MB	14	□			
MB	15	□			
MB	16	□			
MB	17	□			
MB	18	□			
MB	19	□			
MB	20	□			
MB	21	□			
MB	22	□			
MB	23	□			

System Bits

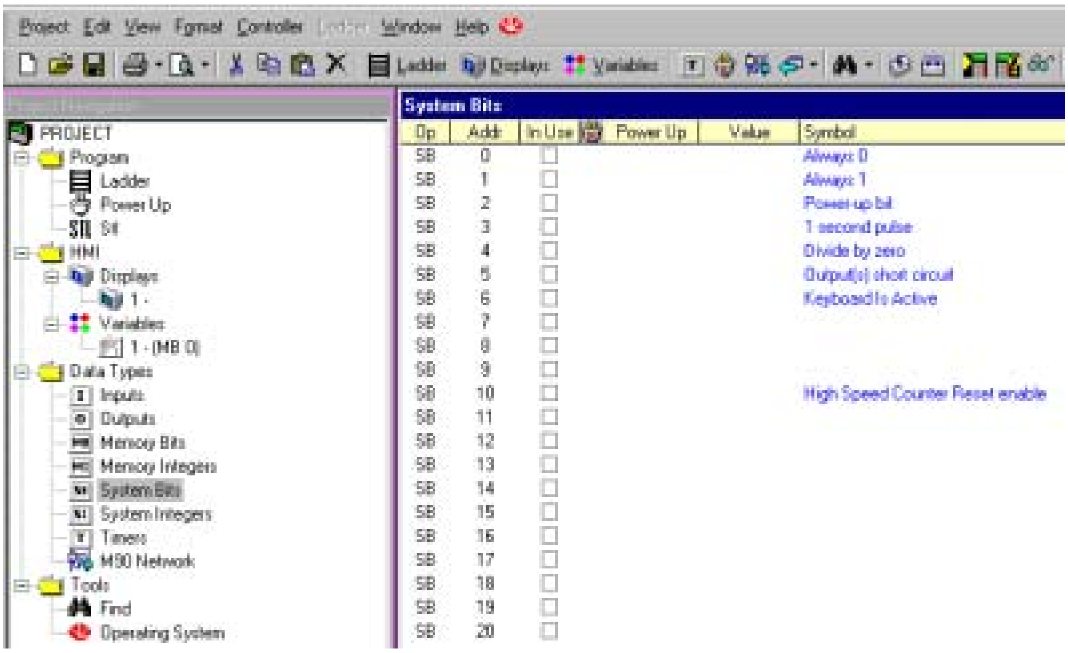
System Bits are the Operating System interface to the user writing the application. System Bits are reserved by the Operating System for particular functions. Some System Bits, for example, are connected to the M90's keypad keys.

There are 256 SBs (Address SB 0 - SB 255).

Only certain SBs may be written into by the programmer:

- SB 80: Activate Linearization
- SB 200 -SB 215: M90 Network Operand

Click on the System Bits folder in the Program Navigation pane to display the complete list of System Bits. Scroll down to view the complete list



System Bits

Op	Addr	In Use	Power Up	Value	Symbol
SB	0	□			Always: 0
SB	1	□			Always: 1
SB	2	□			Power-up bit
SB	3	□			1 second pulse
SB	4	□			Divide by zero
SB	5	□			Output(s) short circuit
SB	6	□			Keyboard Is Active
SB	7	□			
SB	8	□			
SB	9	□			
SB	10	□			High Speed Counter Reset enable
SB	11	□			
SB	12	□			
SB	13	□			
SB	14	□			
SB	15	□			
SB	16	□			
SB	17	□			
SB	18	□			
SB	19	□			
SB	20	□			

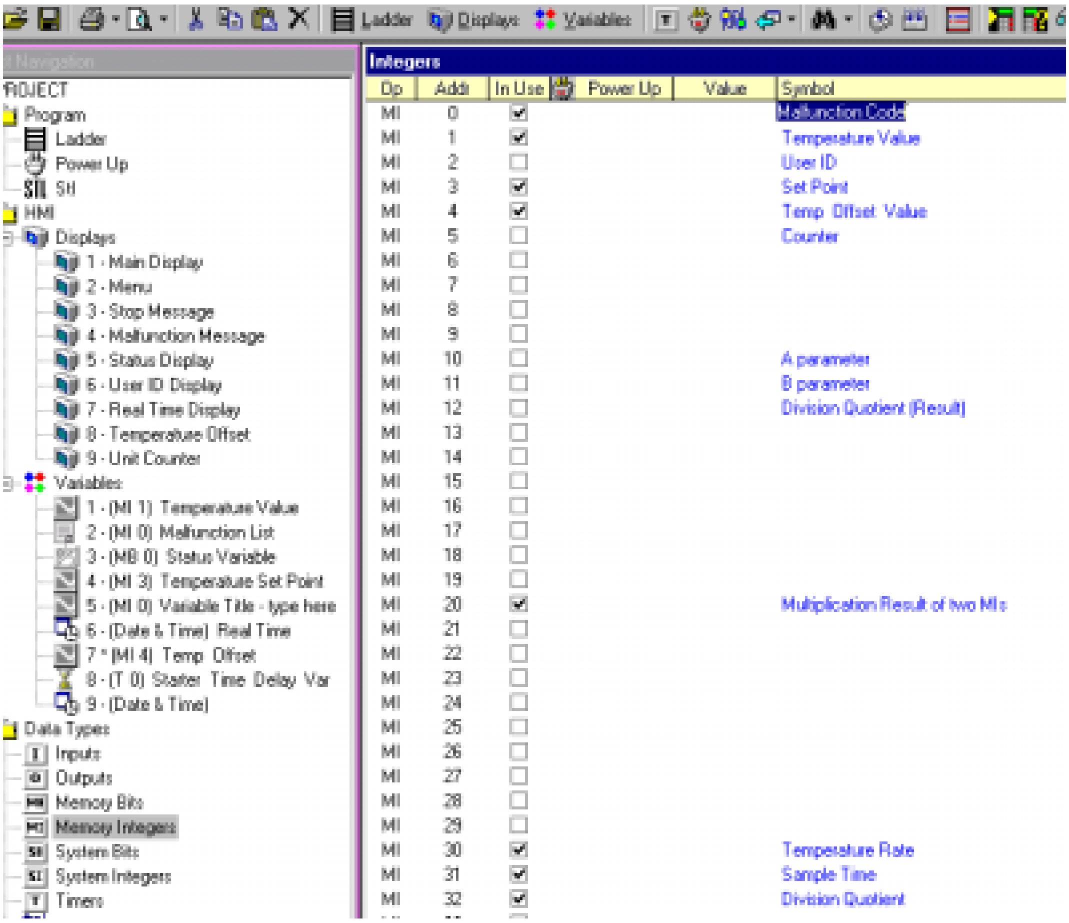
Memory Integers (MI)

Memory Integers are one Operand type available for writing a project application.

There are 256 MIs (Address MI 0 - MI 255).

Memory Integers hold an integer value (-32768 to +32767).

Click on the Memory Integers folder in the Program Navigation pane to display the complete list of Memory Integers. Scroll down to view the complete list



Integers

Op	Addr	In Use	Power Up	Value	Symbol
MI	0	☑			Malfunction Code
MI	1	☑			Temperature Value
MI	2	□			User ID
MI	3	☑			Set Point
MI	4	☑			Temp Offset Value
MI	5	□			Counter
MI	6	□			
MI	7	□			
MI	8	□			
MI	9	□			
MI	10	□			A parameter
MI	11	□			B parameter
MI	12	□			Division Quotient (Result)
MI	13	□			
MI	14	□			
MI	15	□			
MI	16	□			
MI	17	□			
MI	18	□			
MI	19	□			
MI	20	☑			Multiplication Result of two MIs
MI	21	□			
MI	22	□			
MI	23	□			
MI	24	□			
MI	25	□			
MI	26	□			
MI	27	□			
MI	28	□			
MI	29	□			
MI	30	☑			Temperature Rate
MI	31	☑			Sample Time
MI	32	☑			Division Quotient

System Integers (SI)

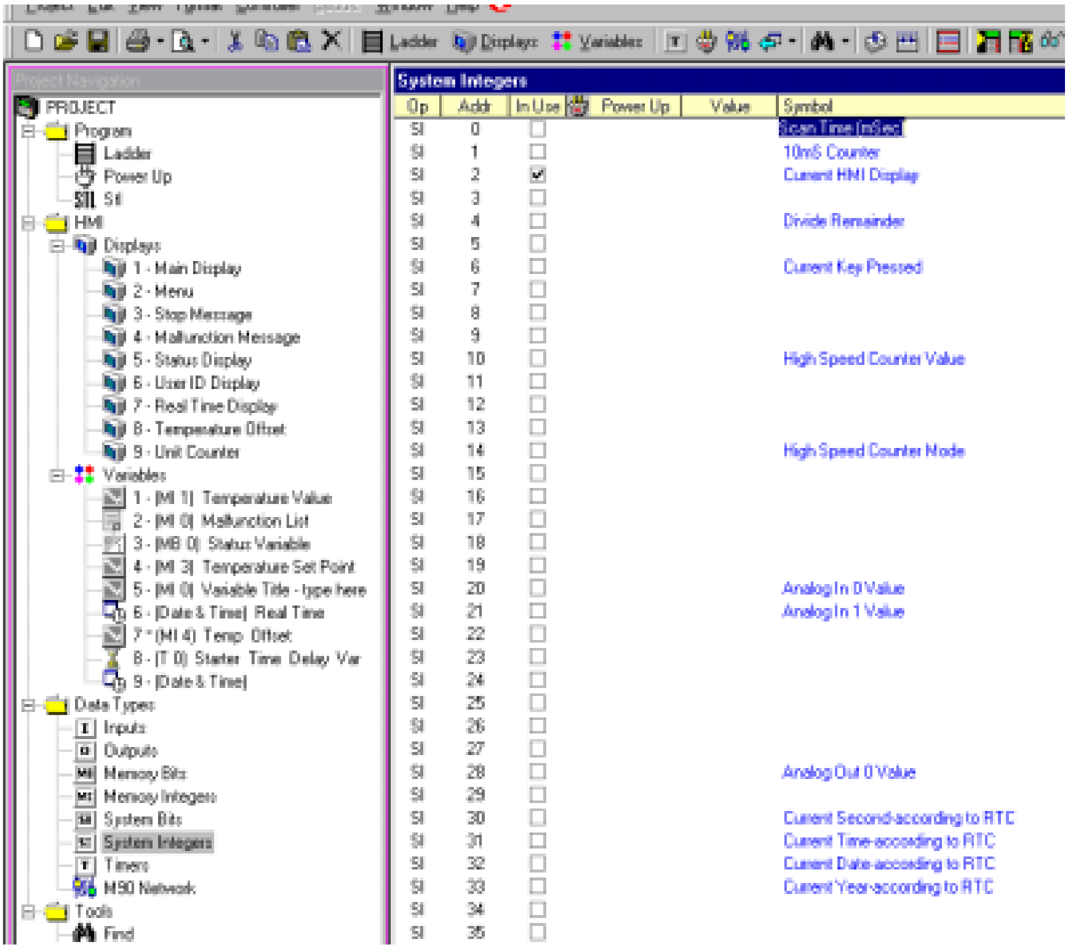
System Integers are the Operating System interface to the user writing the application. System Integers are reserved by the Operating System for particular functions. Specific System Integers, for example, are connected to the M90's high speed counter/shaft-encoder.

There are 256 SIs (Address SI 0 - SI 255).

Only certain SIs may be written into by the programmer:

- SI 2: Current HMI Display
- SI 80 - SI 84: Linearization Parameters
- SI 200 , SI 201: M90 Network Operand

Click on the System Integers folder in the Program Navigation pane to display the complete list of System Integers. Scroll down to view the complete list



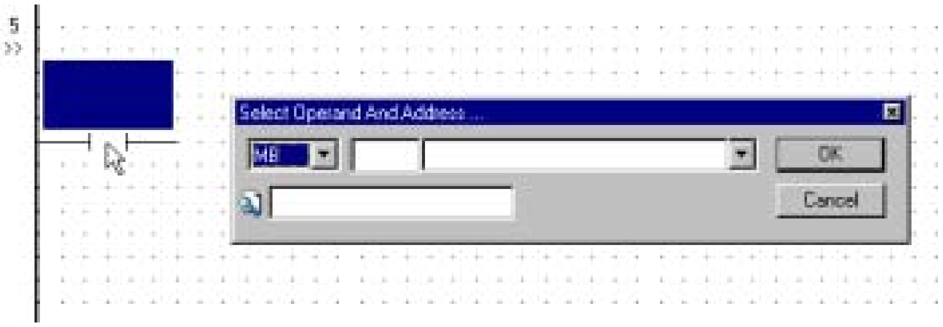
- PROJECT
  - Program
    - Ladder
    - Power Up
    - Stl
  - HMI
    - Displays
      - 1 - Main Display
      - 2 - Menu
      - 3 - Stop Message
      - 4 - Malfunction Message
      - 5 - Status Display
      - 6 - User ID Display
      - 7 - Real Time Display
      - 8 - Temperature Offset
      - 9 - Unit Counter
    - Variables
      - 1 - [MI 1] Temperature Value
      - 2 - [MI 0] Malfunction List
      - 3 - [MB 0] Status Variable
      - 4 - [MI 3] Temperature Set Point
      - 5 - [MI 0] Variable Title - type here
      - 6 - [Date & Time] Real Time
      - 7 * [MI 4] Temp Offset
      - 8 - [T 0] Starter Time Delay Var
      - 9 - [Date & Time]
  - Data Types
    - Inputs
    - Outputs
    - Memory Bits
    - Memory Integers
    - System Bits
    - System Integers
    - Timers
    - M90 Network
  - Tools
    - Find

System Integers

Op	Addr	In Use	Power Up	Value	Symbol
SI	0	□			Scan Time (mSec)
SI	1	□			10mS Counter
SI	2	☑			Current HMI Display
SI	3	□			
SI	4	□			Divide Remainder
SI	5	□			
SI	6	□			Current Key Pressed
SI	7	□			
SI	8	□			
SI	9	□			
SI	10	□			High Speed Counter Value
SI	11	□			
SI	12	□			
SI	13	□			
SI	14	□			High Speed Counter Mode
SI	15	□			
SI	16	□			
SI	17	□			
SI	18	□			
SI	19	□			
SI	20	□			Analog In 0 Value
SI	21	□			Analog In 1 Value
SI	22	□			
SI	23	□			
SI	24	□			
SI	25	□			
SI	26	□			
SI	27	□			
SI	28	□			Analog Out 0 Value
SI	29	□			
SI	30	□			Current Second-according to RTC
SI	31	□			Current Time-according to RTC
SI	32	□			Current Date-according to RTC
SI	33	□			Current Year-according to RTC
SI	34	□			
SI	35	□			

Assigning an Operand Address by Symbol

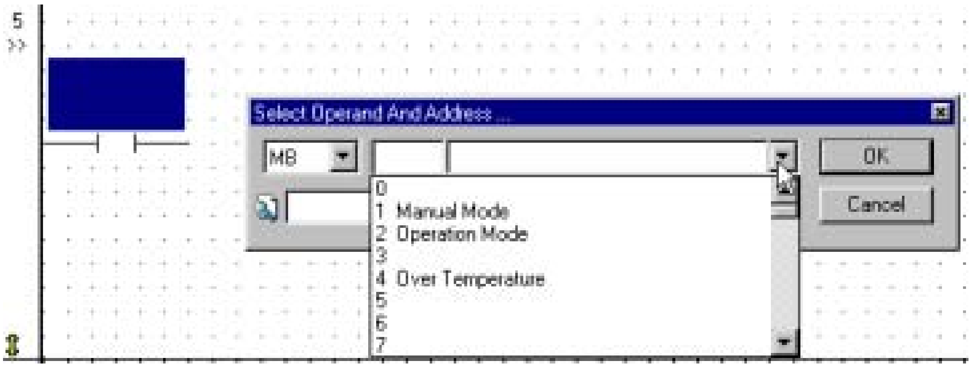
1. After placing the element on the net, the Select Operand and Address dialog box opens.



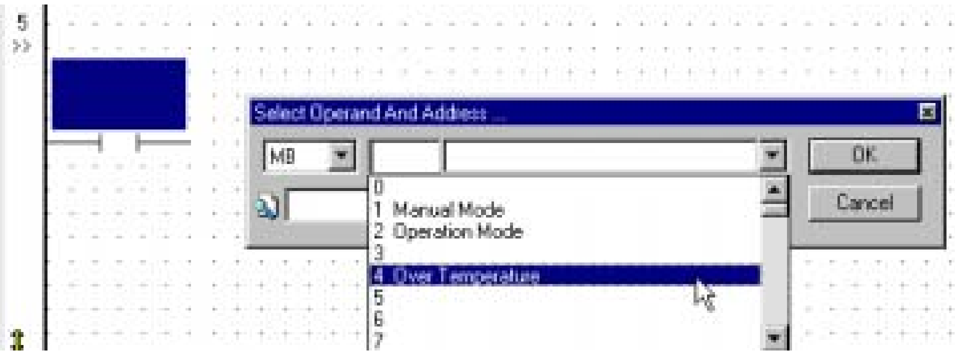
Select Operand And Address ...

MB

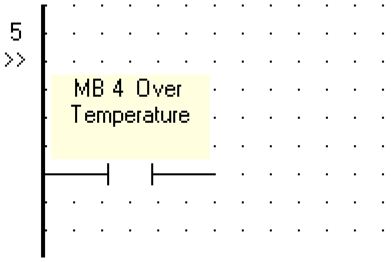
2. Click on the Symbol drop-down menu.



3. Select the desired Address.



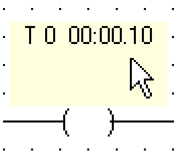
4. The element appears with the selected Address and Symbol.



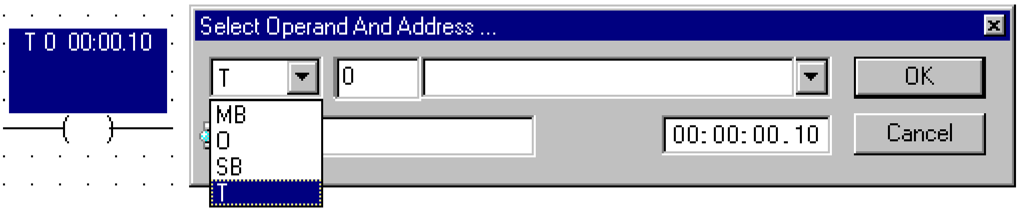
Changing an Operand type

To change an Operand type:

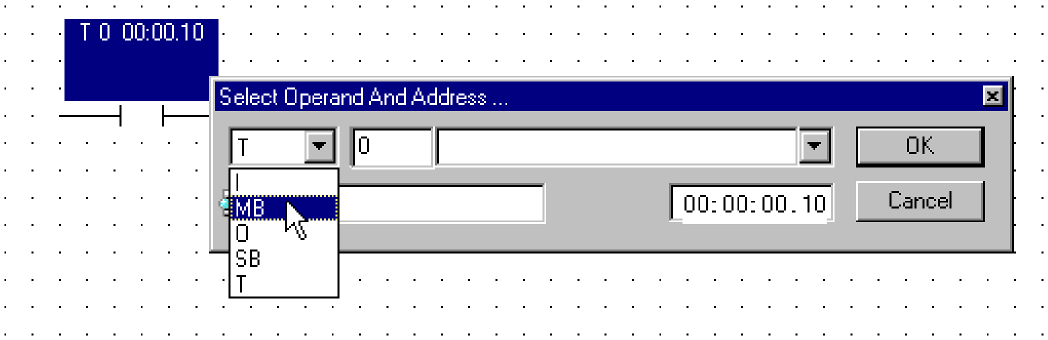
1. Double click on the element's Operand.



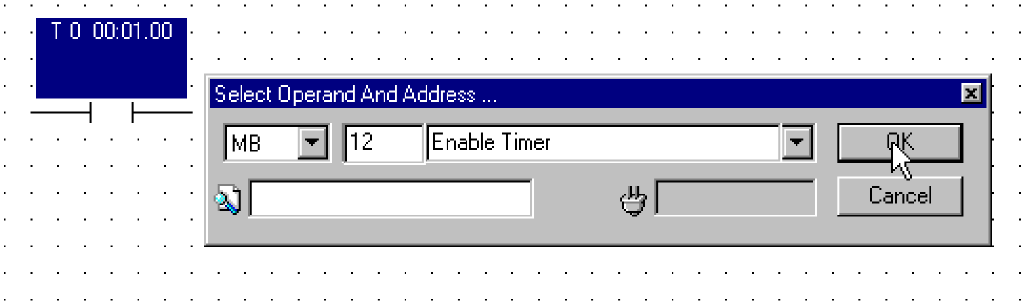
2. The Operand and Address dialog box opens.



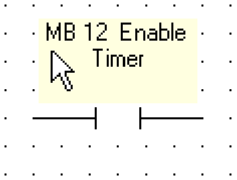
3. Select the new Operand type.



4. Enter the new Operand Address and symbol.



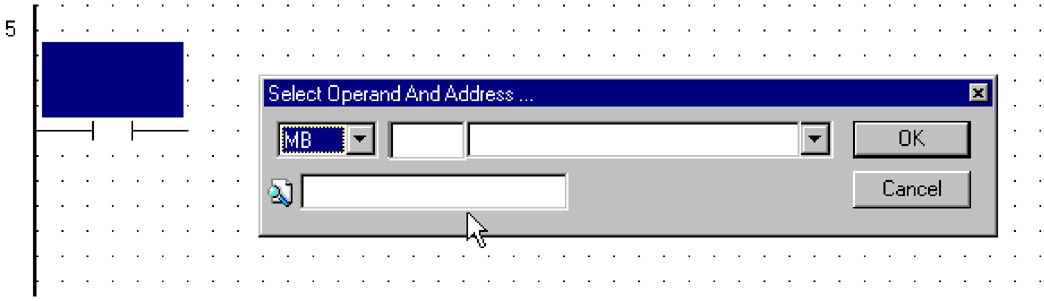
5. Click OK. The element appears on the net with the new Operand, Address and symbol.



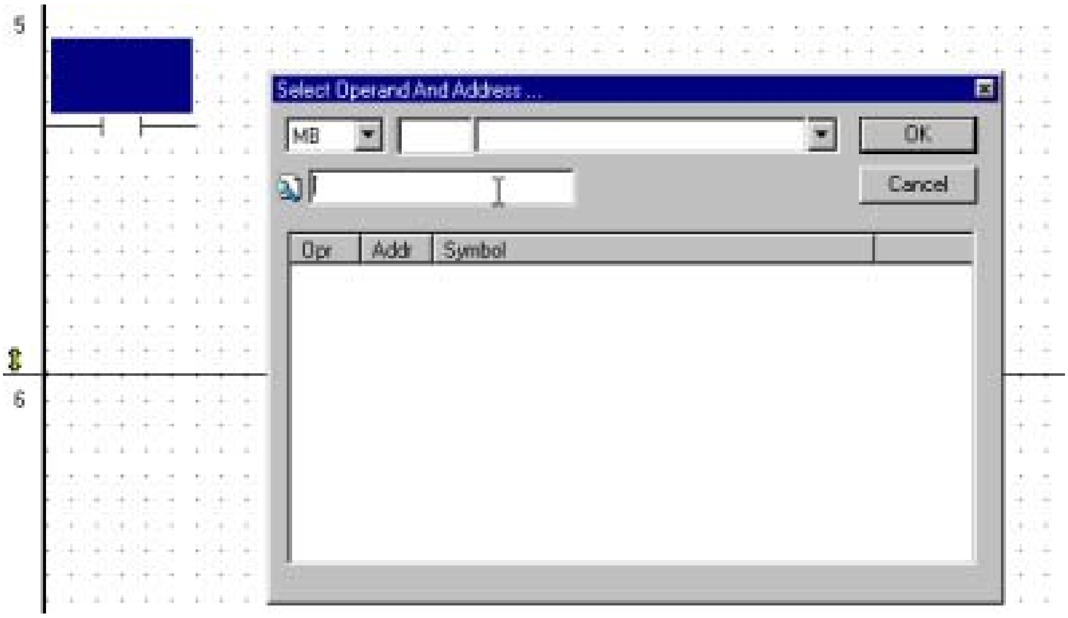
Finding an Operand by symbol

To find an Operand by its symbol when placing an element on a net:

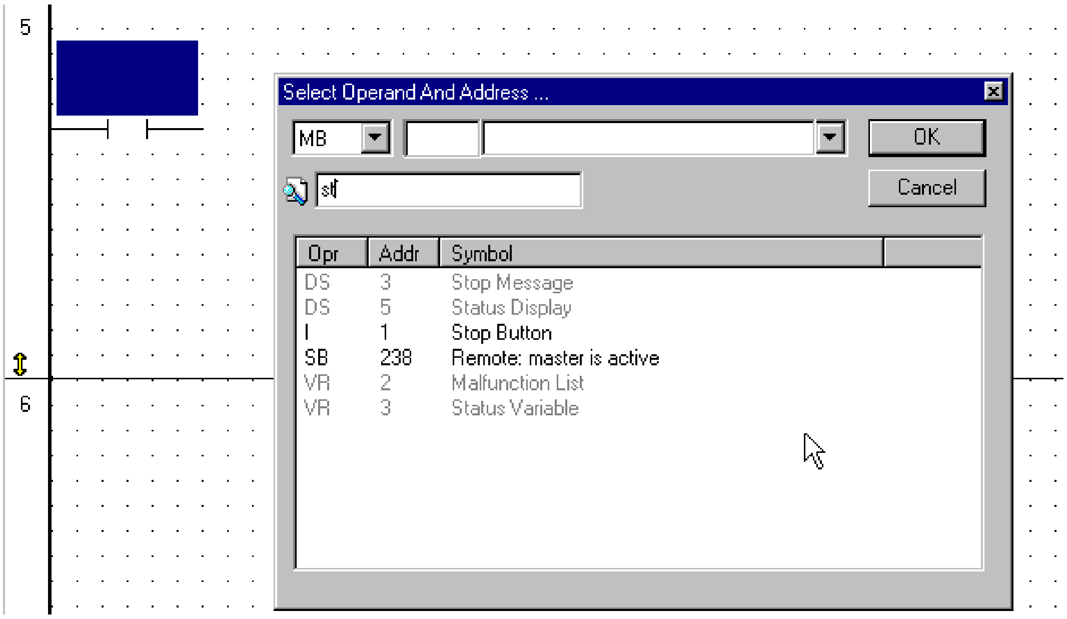
1. Click in the **Symbol Search** box in the Select Operand and Address dialog box.



2. The **Symbol Search** dialog box opens.



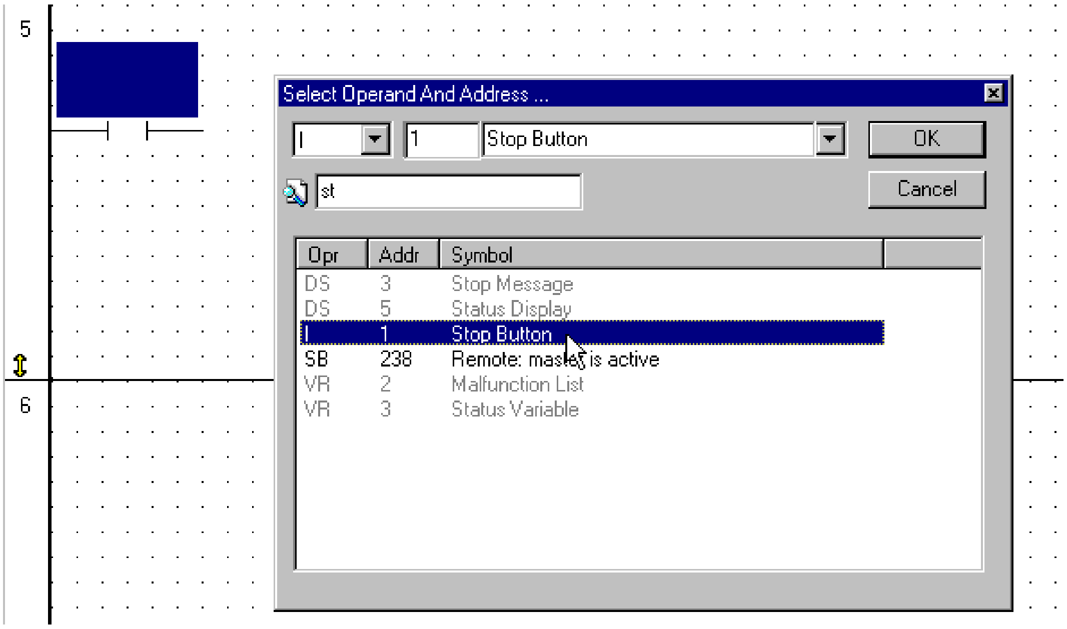
3. Begin entering the Symbol name for which you are searching. The list will become more specific the more letters you enter.



st

Opr	Addr	Symbol
DS	3	Stop Message
DS	5	Status Display
I	1	Stop Button
SB	238	Remote: master is active
VR	2	Malfunction List
VR	3	Status Variable

4. Select the desired Operand from the Symbol Search list.



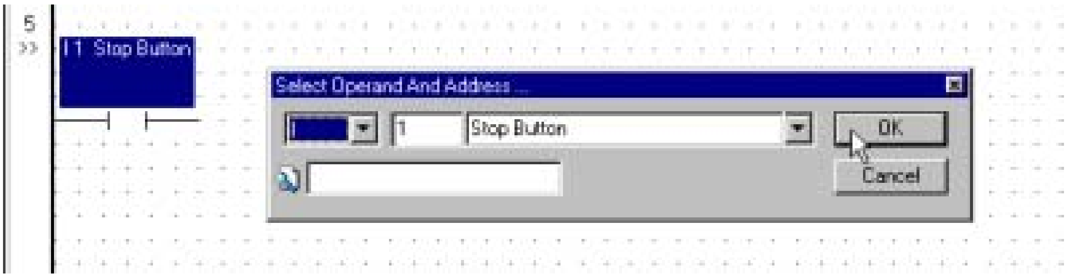
Select Operand And Address ...

I | 1 | Stop Button | OK

st | Cancel

Opr	Addr	Symbol
DS	3	Stop Message
DS	5	Status Display
I	1	Stop Button
SB	238	Remote: master is active
VR	2	Malfunction List
VR	3	Status Variable

5. Click OK. The selected element appears on the net with the desired Operand and Address.



Select Operand And Address ...

I | 1 | Stop Button | OK

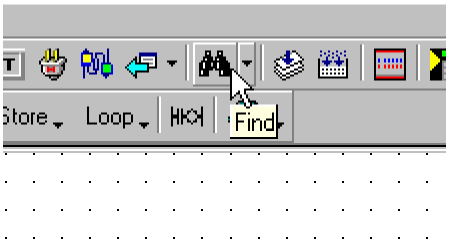
Cancel

Operand Locations List

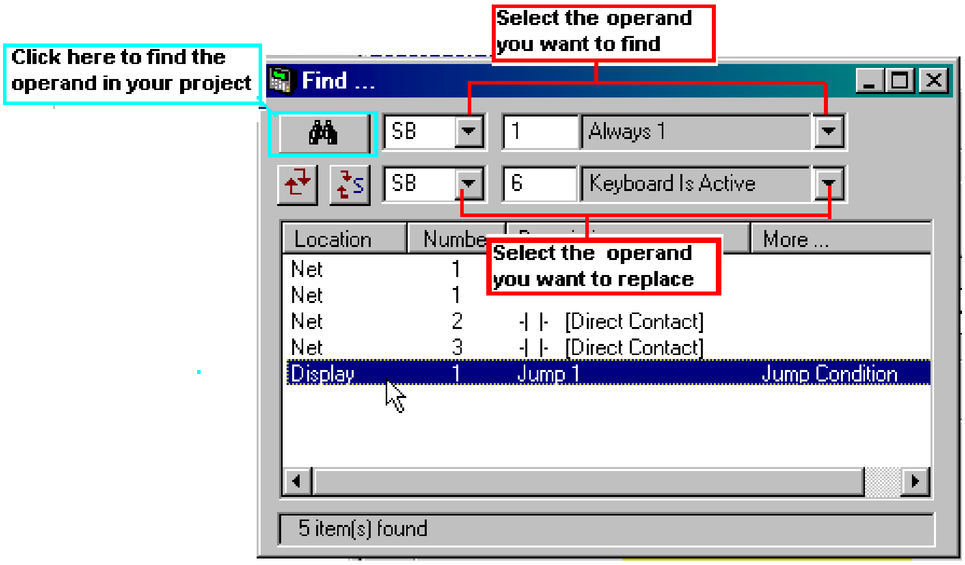
To get a list of Operand locations:

If you already have one location where you know the Operand exists, you can select the Operand and then open the Find dialog box. A list of all locations of the selected Operand will appear.

1. Click on the **Find** icon in the Standard toolbar.



2. The Find function opens.
3. Select the name and address of the operand you wish to find.
4. Click the Find button shown below; a list appears showing every time that operand is used in the project.
5. Select the name and address of the operand you wish to replace as shown below.



Click here to find the operand in your project

Select the operand you want to find

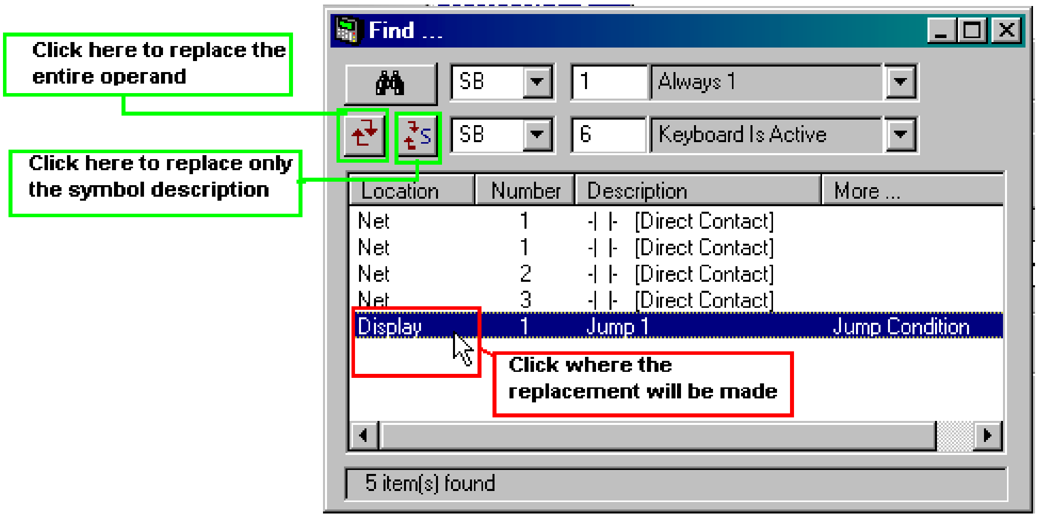
Select the operand you want to replace

Location	Numbe		More ...
Net	1		
Net	1		
Net	2	-| |- [Direct Contact]	
Net	3	-| |- [Direct Contact]	
Display	1	Jump 1	Jump Condition

5 item(s) found

6. Select the location of the operand or description you wish to replace by clicking it within the list.

7. Replace operands or their descriptions by clicking the buttons shown below.



Click here to replace the entire operand

Click here to replace only the symbol description

Click where the replacement will be made

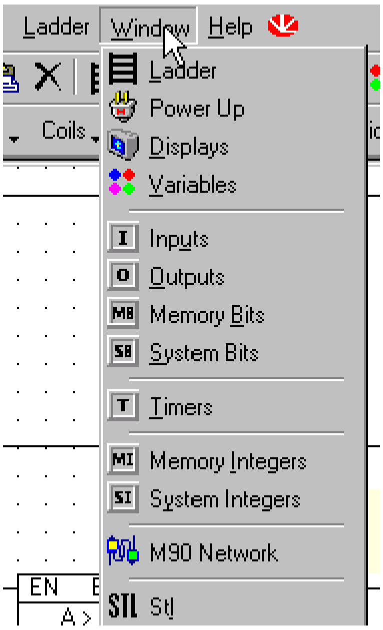
Location	Number	Description	More ...
Net	1	-| |- [Direct Contact]	
Net	1	-| |- [Direct Contact]	
Net	2	-| |- [Direct Contact]	
Net	3	-| |- [Direct Contact]	
Display	1	Jump 1	Jump Condition

5 item(s) found

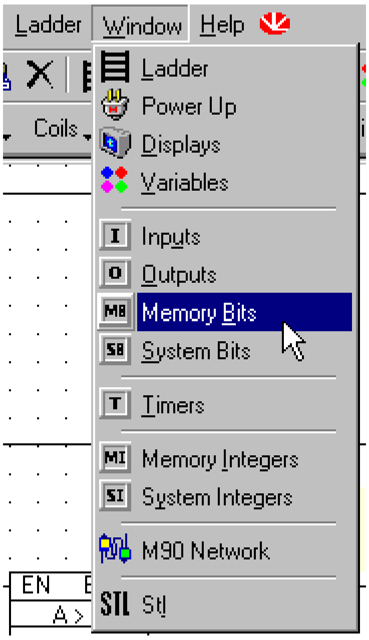
Operands in use

To check what Operands are being used in a project:

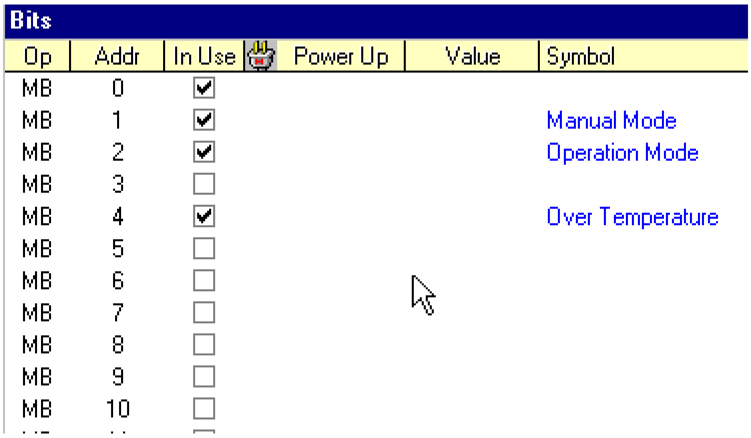
1. Open the Window Menu on the Main menu bar.



2. Select the Operand type you wish to check.



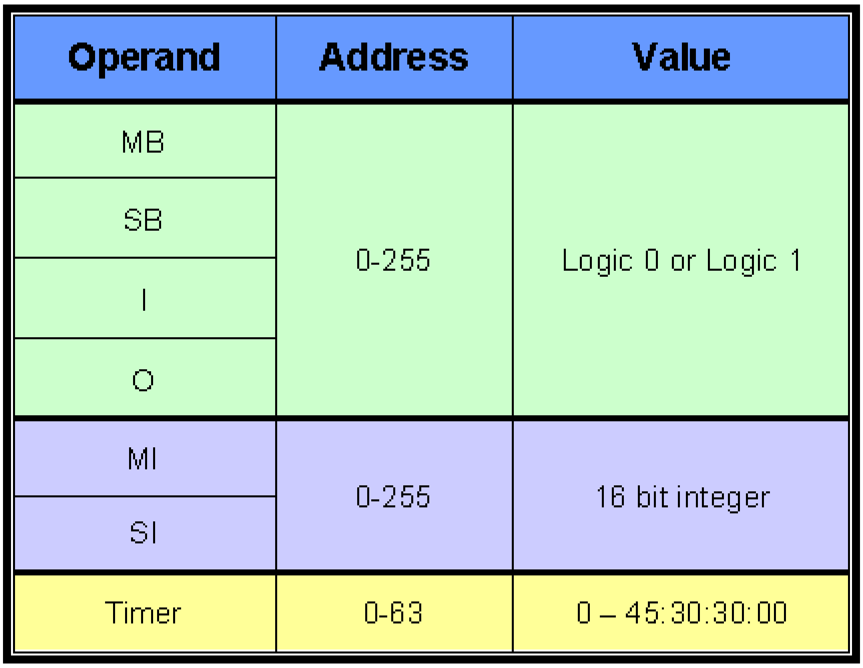
3. The Operand List window opens. The Operands in use are marked with a check mark in the **In Use** box.



Bits

Op	Addr	In Use	Power Up	Value	Symbol
MB	0	☑			
MB	1	☑			Manual Mode
MB	2	☑			Operation Mode
MB	3	□			
MB	4	☑			Over Temperature
MB	5	□			
MB	6	□			
MB	7	□			
MB	8	□			
MB	9	□			
MB	10	□			

Operand Values:



Operand	Address	Value
MB	0-255	Logic 0 or Logic 1
SB		
I		
O		
MI	0-255	16 bit integer
SI		
Timer	0-63	0 – 45:30:30:00

The integer value range is 2^{16}– 1: that is +32767 to -32768.

Keep this integer range in mind when creating function blocks.

For example: MI 75 + #50 = MI 76

If MI 75 goes beyond 32626, the integer value returned in MI 76 will be a **negative** number!

Functions

The following types of Function Blocks can be used in your program:

- Compare Functions
- Logic Functions
- Math Functions
- Store functions
- Clock Functions

- Loops: Jump to Label

Functions without Ladder elements

VisiLogic contains functions that are not represented by Ladder Elements. You can perform these functions by storing values into the System Integers listed here.

To select the function type, first store the number of the function in SI 140, then use SI 141 to 146 to contain the data to be used in the function.

Note that when you run Test (Debug) Mode, the current value in SI 140 will **not** be displayed.

SI	Description
140	Select Function
141	Function Operand #1
142	Function Operand #2
143	Function Operand #3
144	Function Operand #4
145	Function Operand #5
146	Function Operand #6

- Communication Utilities
- Interrupt
- Access indirectly addressed registers: Using the Database
- Load Indirect
- Load Timer Preset/Current Value
- Store Timer's Preset/Current Value
- SMS phone number: via MI Pointer
- Shift Register
- Copy Vector
- Copy MI to Output vector, Input vector to MI
- Fill Vector
- Convert MB to MI, MI to MB
- Linearization
- Find Mean, Maximum, and Minimum Values
- A*B/C
- Square Root

Compare Functions

A compare function represents a data manipulation instruction. U90 Ladder uses function blocks to operate compare functions. Each function block takes 2 inputs (MI, SI or a constant integer) and manipulates them according to the function block instruction.

If the function block instructions are true (logic 1): power flows through the block.

If the function block instructions are false (logic 0): power does not flow through the block.

There are 6 types of Compare Functions:

- Greater Than
- Greater Than or Equal To
- Equal To
- Not Equal To
- Less Than or Equal To
- Less Than

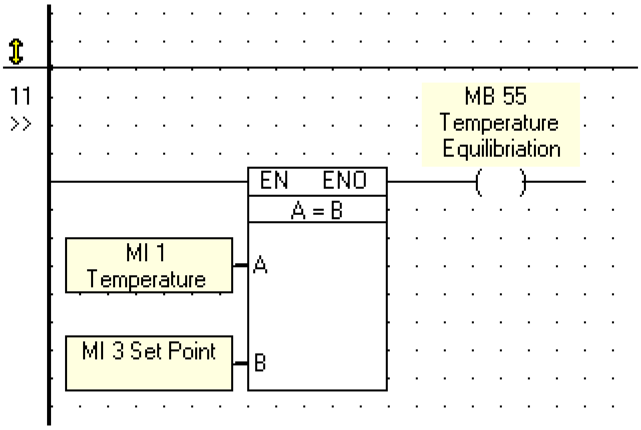
Equal

The Equal function block evaluates input A to see if its constant integer value is equal to input B.

If input A is equal to input B : power will flow through the function block.

If input A is not equal to input B: power will not flow through the function block.

Input Operands A & B must be integer values: MI, SI or # constant integer value.



According to the above example:

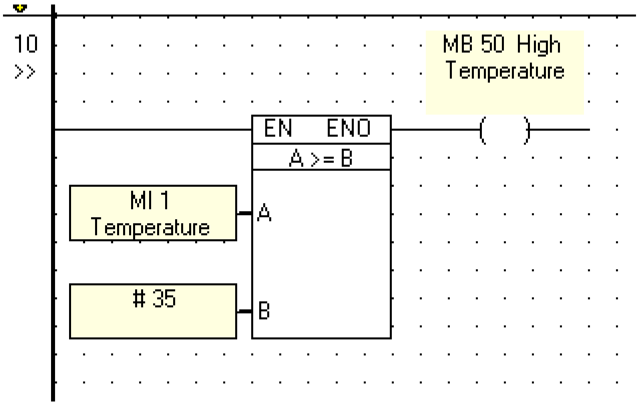
- If MI 1 is equal to MI 3; then MB 55 will go to logic "1" (ON).
- If MI 1 is not equal to MI 3; then MB 55 will go to logic "0" (OFF).

Greater or Equal

The Greater Than or Equal function block evaluates input A to see if its integer value is greater than or equal to input B.

If input A is greater than or equal to input B: power will flow through the function block.

If input A is not greater than or not equal to input B: power will not flow through the function block.



According to the above example:

- If MI 1 value is greater or equal to constant integer 35; then MB 50 will go to logic "1" (ON).

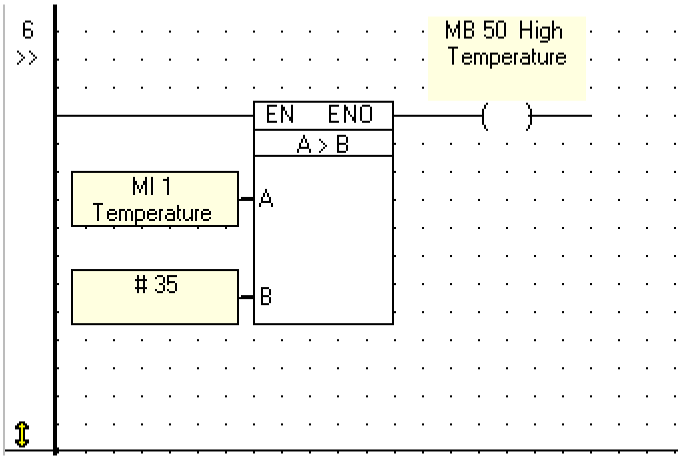
- If MI 1 value is not greater or equal to constant integer 35; then MB 50 will go to logic "0" (OFF).

Greater Than >

The Greater Than function block evaluates input A to see if its current value is greater than input B.

If input A is greater than input B: power will flow through the function block.

If input A is not greater than input B: power will not flow through the function block.



According to the above example:

- If MI 1 value is greater than 35; then MB 50 will go to logic "1" (ON).
- If MI 1 not greater than 35; MB 50 will go to logic "0".

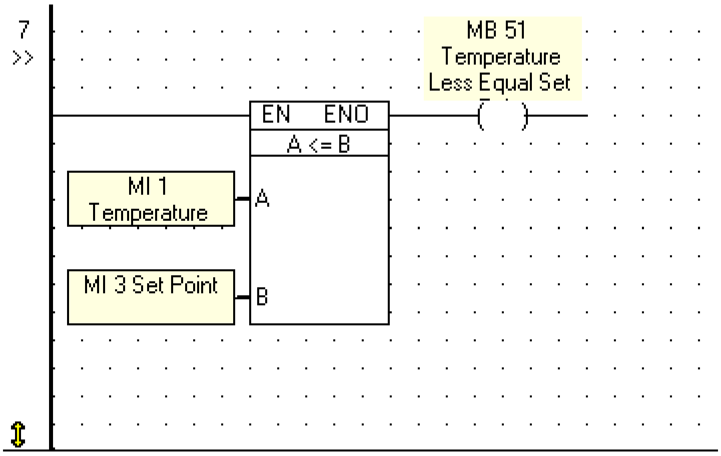
Care must be taken when using greater and less than function blocks. Do not create a program with instructions for Greater Than and Less Than but without an instruction block for how to proceed in a situation where input A equals input B.

Less or Equal <=

The Less Than or Equal To function block evaluates input A to see if its current value is less than or equal to input B.

If input A is less than or equal to input B: power will flow through the function block.

If input A is not less than or equal to input B: power will not flow through the function block.



According to the above example:

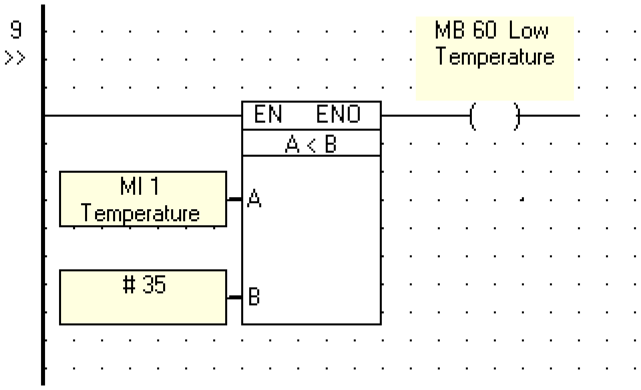
- If MI 1 value is greater than the MI 3 value; then MB 51 will go to logic "1" (ON).
- If MI 1 not greater than the MI 3 value; MB 51 will go to logic "0".

Less Than <

The Less Than function block evaluates input A to see if its integer value is less than input B.

If input A is less than input B: power will flow through the function block.

If input A is not less than input B: power will not flow through the function block.



According to the above example:

- If MI 1 value is less than constant integer 35; then MB 60 will go to logic "1" (ON).
- If MI 1 value is not less than constant integer 35; MB 60 will go to logic "0" (OFF).

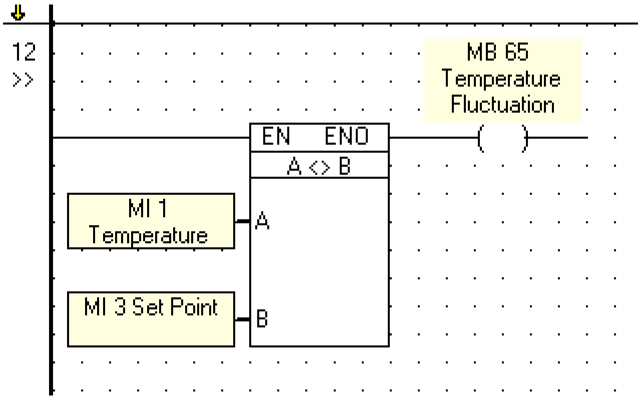
Not Equal <>

The Not Equal function block evaluates input A to see if its integer value is not equal to input B.

If input A is not equal to input B: power will flow through the function block.

If input A is equal to input B: power will not flow through the function block.

Input Operands A & B must be integer values: MI, SI or # constant integer value.



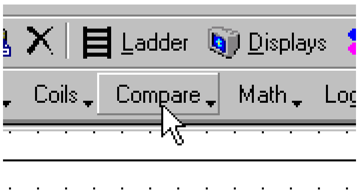
According to the above example:

- If MI 1 is not equal to MI 3; then MB 65 will go to logic "1" (ON).
- If MI 1 is equal to MI 3; then MB 65 will go to logic "0" (OFF).

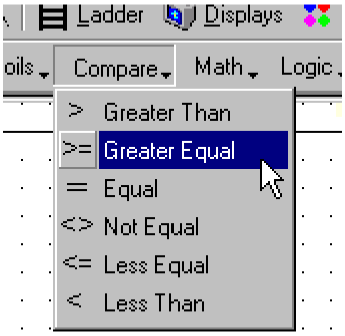
Using the Compare function

To use the Compare function:

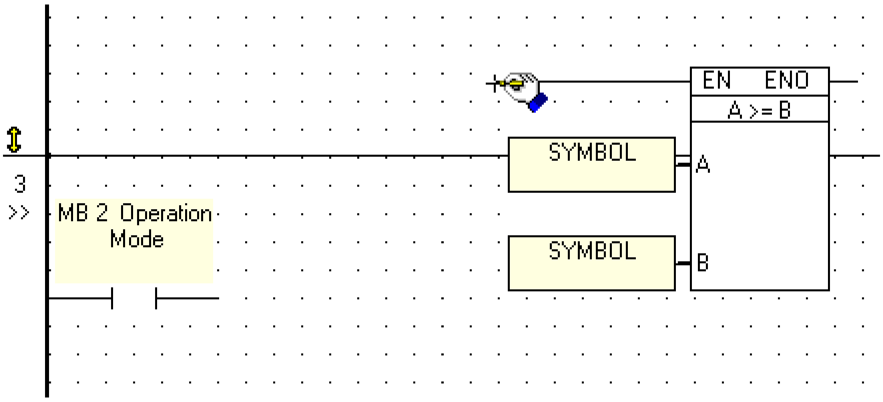
1. Click **Compare** on the Standard toolbar. The Compare function list opens.



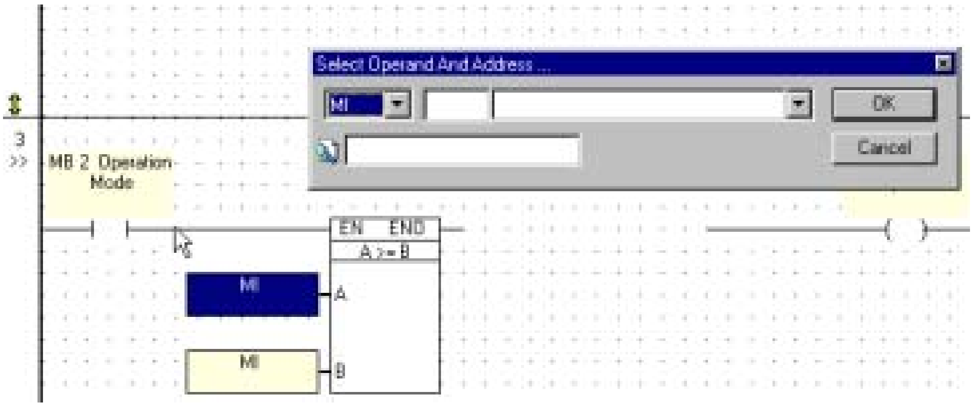
2. Select the desired Compare function.



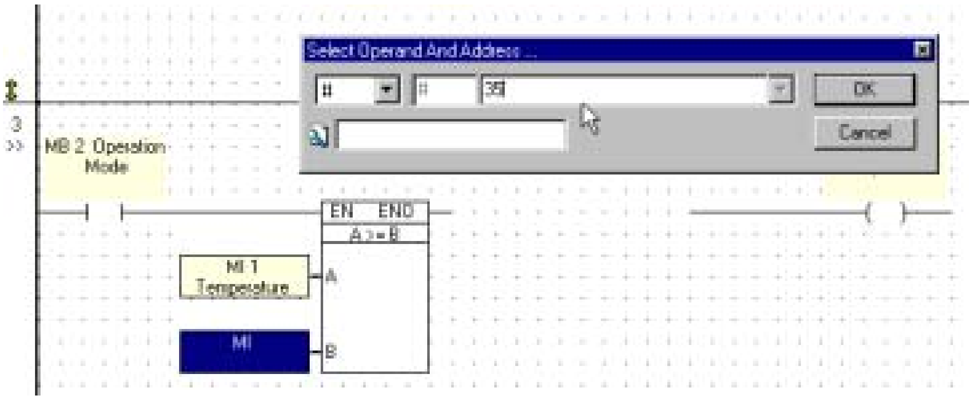
3. Move the function block that appears to the desired net location.



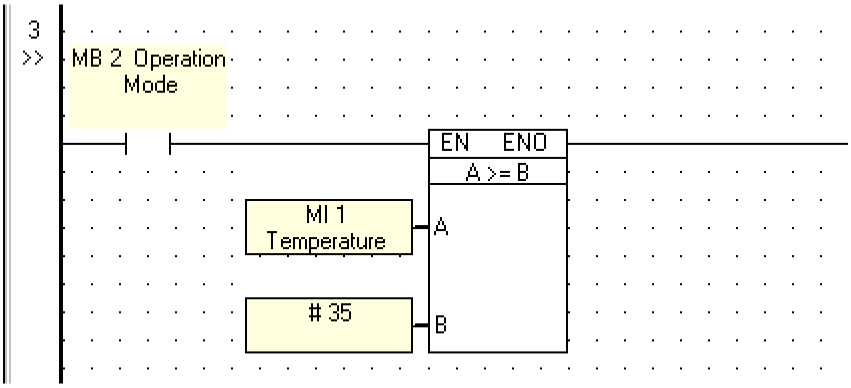
4. Click to place the function block. The Select Operand and Address dialog box opens.



5. Enter the Operands and Addresses in the dialog boxes and click OK.



6. The Compare function block appears with the selected Operands and Addresses.



Logic Function

You perform logical functions in U90 Ladder by using logic function blocks. Function blocks are provided for:

- AND
- OR
- XOR

The internal operation of a function block is transparent to the user. You input the two operands. The result is automatically output by the function block

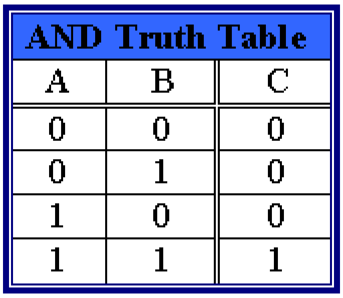
Input Operands A & B must be integer values: MI , SI or # constant integer value.

Output Operand C may be a Memory Integer or a System Integer.

AND

Example

The AND logic function block can evaluate the state of two integers. If a bit is true (logic 1) in both input A and B then the output C will be true (logic 1). If input A and B is false (logic 0) - the output C will be false (logic 0). If either input A or B is false (logic 0) - the output C will be false (logic 0).



AND Truth Table		
A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

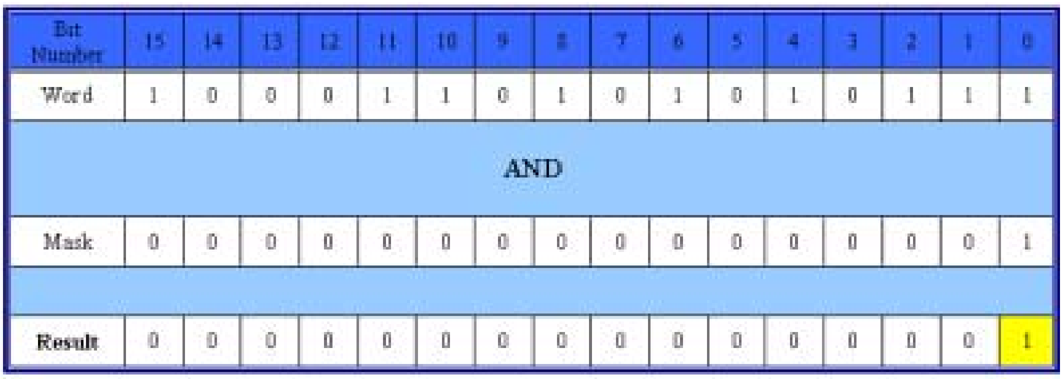
Input Operands A & B must be integer values: MI , SI or # constant integer value.

Output Operand C may be an MI or a SI.

AND can be used to mask out certain bits of an input integer not relevant to a given function.

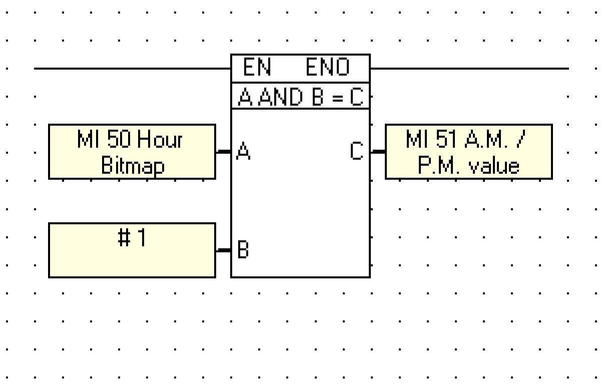
Example:

If a clock function block uses the first bit of a 16-bit word to decide if a given time is A.M. or P.M., you can mask out the other 15 bits. This will tell you if the current time is A.M. or P.M.



Bit Number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	1	0	0	0	1	1	0	1	0	1	0	1	0	1	1	1
AND																
Mask	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Result	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

All of the non-relevant bits will be turned off (logic 0) expect the A.M. / P.M. bit.

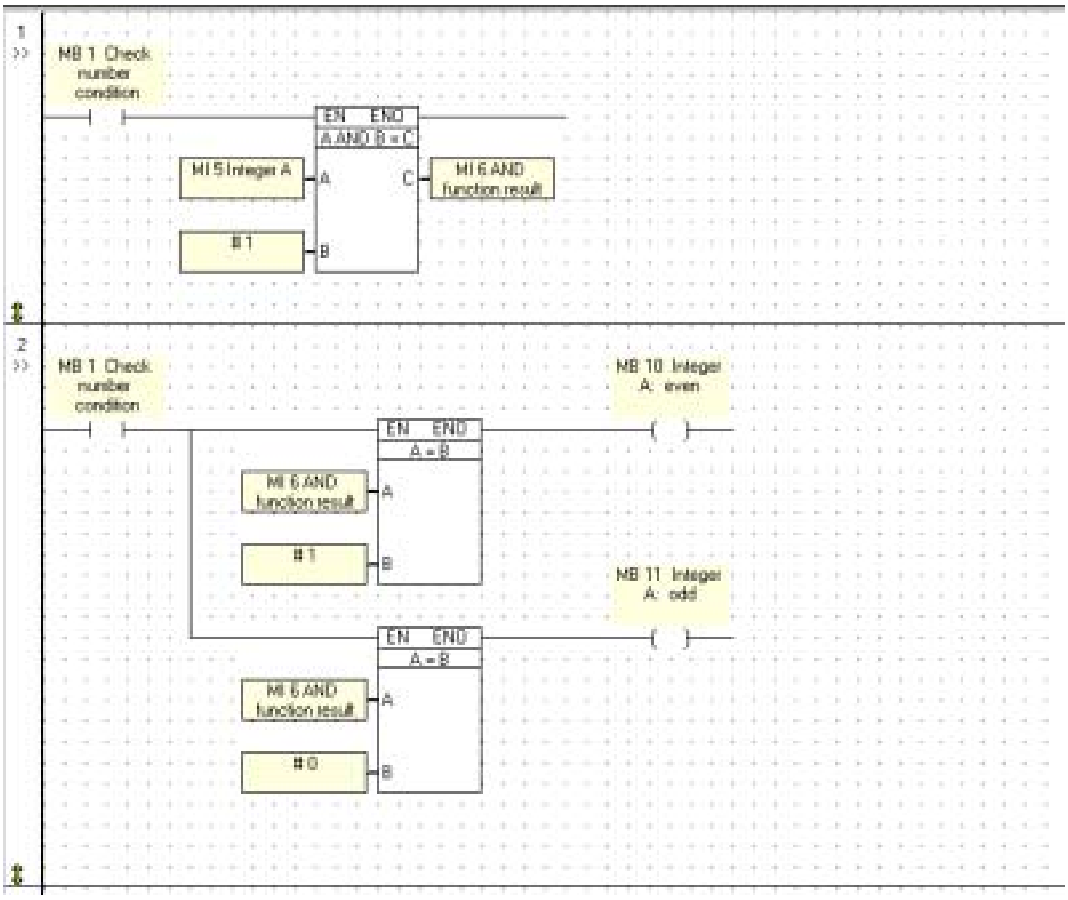


AND Example

You want to determine if an MI / SI value is an odd or an even number in your application.

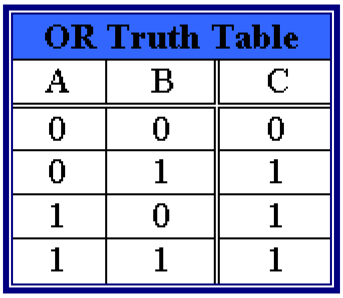
An AND function between an integer A and #1:

- If integer A is an even number then the result of the AND operation = #1.
- If integer A is an odd number then the result of the AND operation = #0



OR

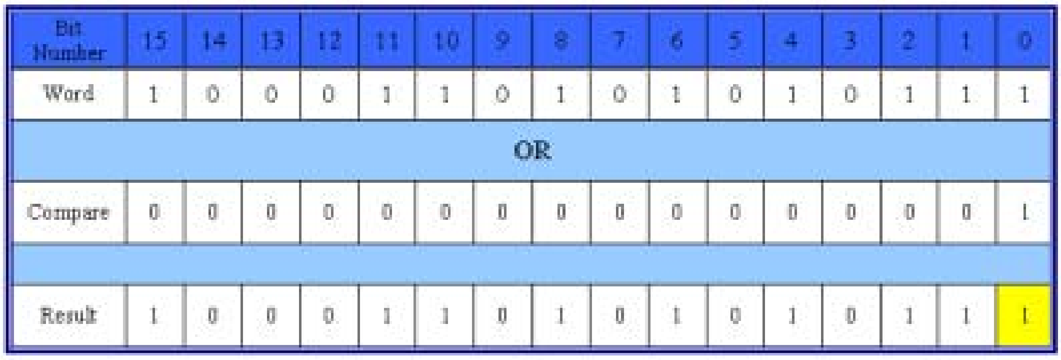
The OR logic function block can evaluate the state of two integers to see if either input A or B is true. If input A OR B is true - the output C will be true (logic 1). If both input A and B are true (logic 1) - the output C will also be true (logic 1).



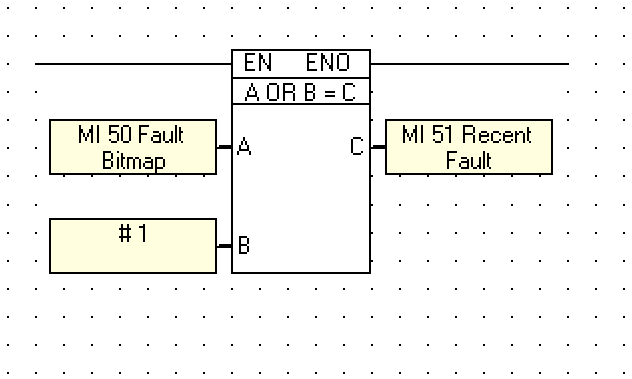
OR Truth Table		
A	B	C
0	0	0
0	1	1
1	0	1
1	1	1

Input Operands A & B must be integer values: MI , SI or # constant integer value.

Output Operand C may be a Memory Integer or a System Integer.

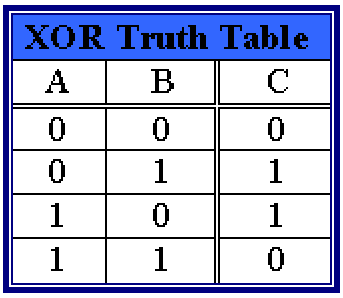


Bit Number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	1	0	0	0	1	1	0	1	0	1	0	1	0	1	1	1
OR																
Compare	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Result	1	0	0	0	1	1	0	1	0	1	0	1	0	1	1	1



XOR

The XOR logic function block can evaluate the state of two integers to see if input A and B are equal. If either input A OR B is true - the output C will be true (logic 1). If both input A and B are true (logic 1) - the output C will be false (logic 0). If both input A and B are false (logic 0) - the output C will be false (logic 0).

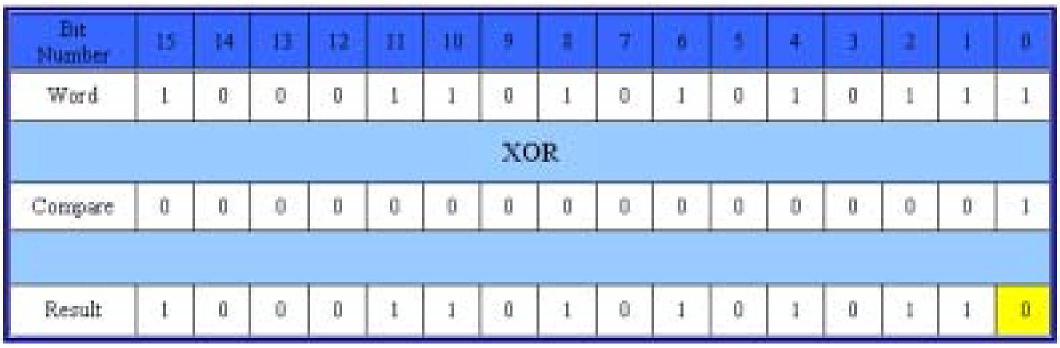


XOR Truth Table		
A	B	C
0	0	0
0	1	1
1	0	1
1	1	0

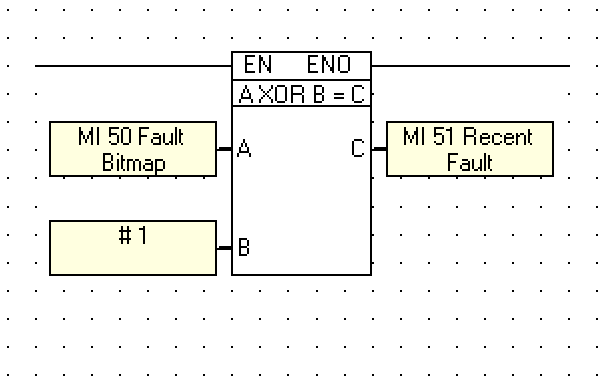
Input Operands A & B must be integer values: MI, SI or # constant integer value.

Output Operand C may be a Memory Integer or a System Integer.

Use XOR to recognize changes in an integer to check for integer bit corruption. If 2 integers are equal: the result will return logic 0. If there has been bit corruption: the corrupted bit will return logic 1.



Bit Number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	1	0	0	0	1	1	0	1	0	1	0	1	0	1	1	1
XOR																
Compare	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Result	1	0	0	0	1	1	0	1	0	1	0	1	0	1	1	0



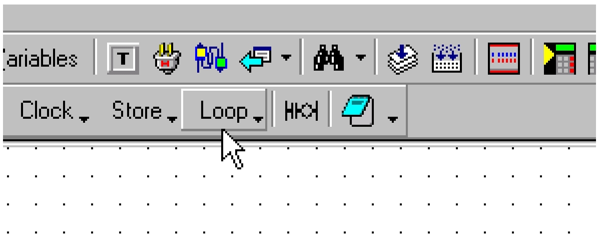
Loops: Jump to Label

Loops in a Ladder project cause the program to jump over certain net(s), according to specific logic conditions.

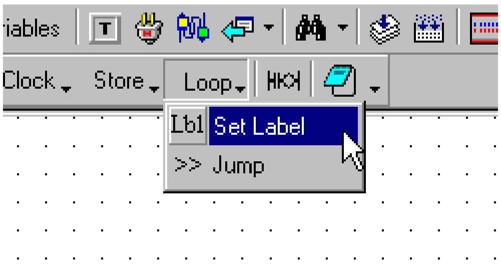
A Loop contains a Jump element and a Label. When the Jump condition(s) is true, the project jumps to the associated Label.

To create a Loop in your project:

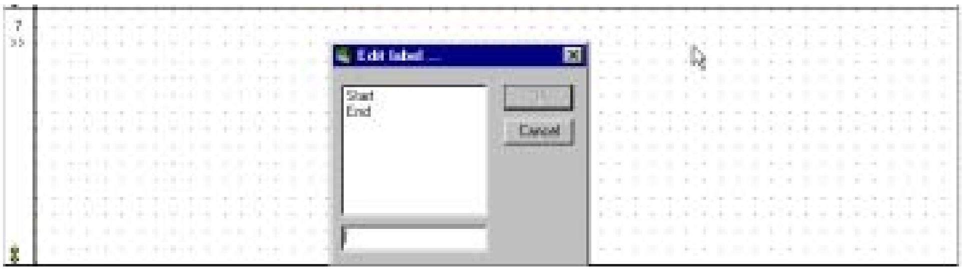
1. Click **Loop** on the Ladder toolbar.



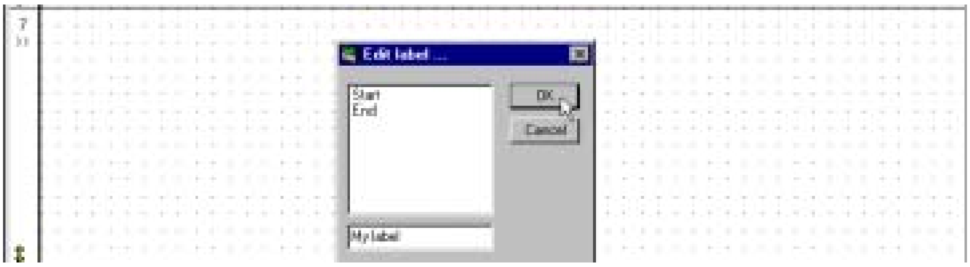
2. Select **Set Label** from the **Loop** menu. Place the cursor in the desired net and click.



3. The **Edit Label** box opens.

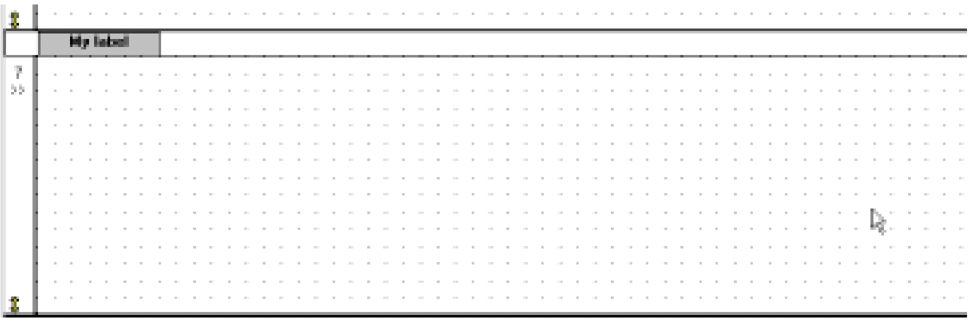


4. Enter a Label name of up to **eight** characters.

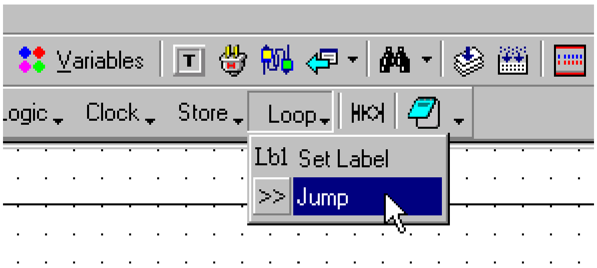


My label

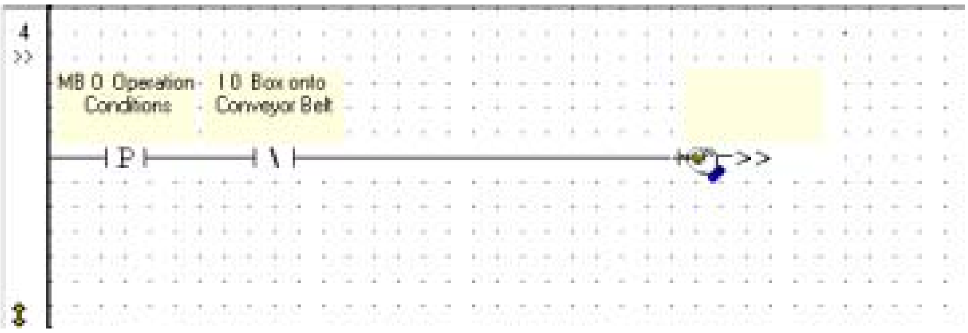
5. The Label appears above the net.



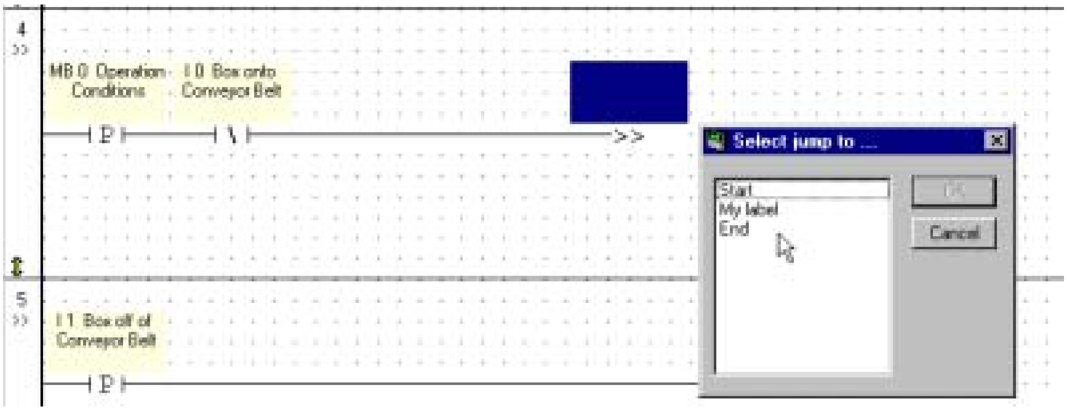
6. Select **Jump** from the Loop menu.



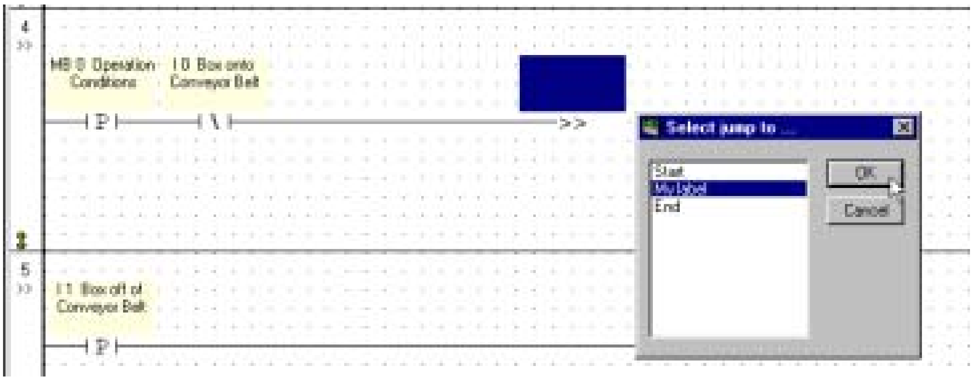
7. Place the Jump in the desired place on the desired net.



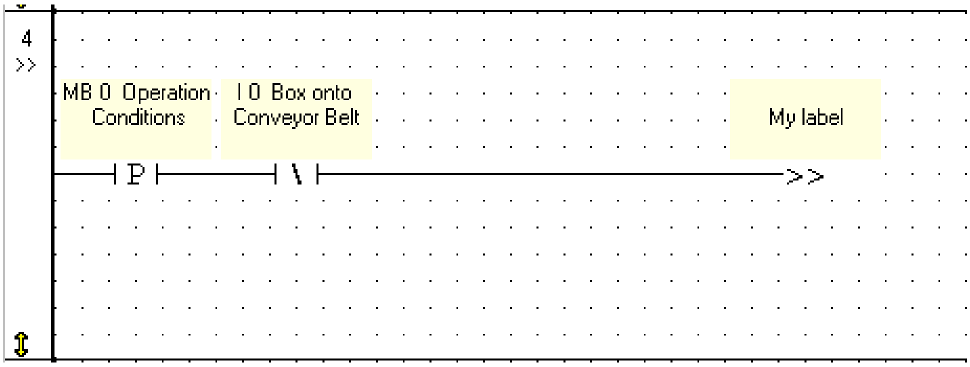
8. **Select Jump to...** window appears.


4
MB 0 Operation Conditions
I 0 Box onto Conveyor Belt
P
>>
5
I 1 Box off of Conveyor Belt
Select jump to ...
Start
My label
End
OK
Cancel

9. Select the desired Label name to which you want to jump. Click OK.


4
MB 0 Operation Conditions
I 0 Box onto Conveyor Belt
P
>>
5
I 1 Box off of Conveyor Belt
Select jump to ...
Start
My label
End
OK
Cancel

9. The Jump element appears with the selected Label name on the net.



According to the above example, if Ladder logic is true for net 4, the program will jump over nets 5 and 6 and continue from net 7.

Important note: You must take care when creating Loops not to create an endless Loop. While you can place Labels before a Jump condition and you can refer to a Label more than once, repeated referrals to a Label above a Jump element can create an endless loop which will cause the controller to stop with an error message "PROGRAM LOOP."

Loop functions are featured in the sample application, such as the applications ' Shortening scan time-jump'. This application may be found by selecting Sample U90 Projects from the Help Menu.

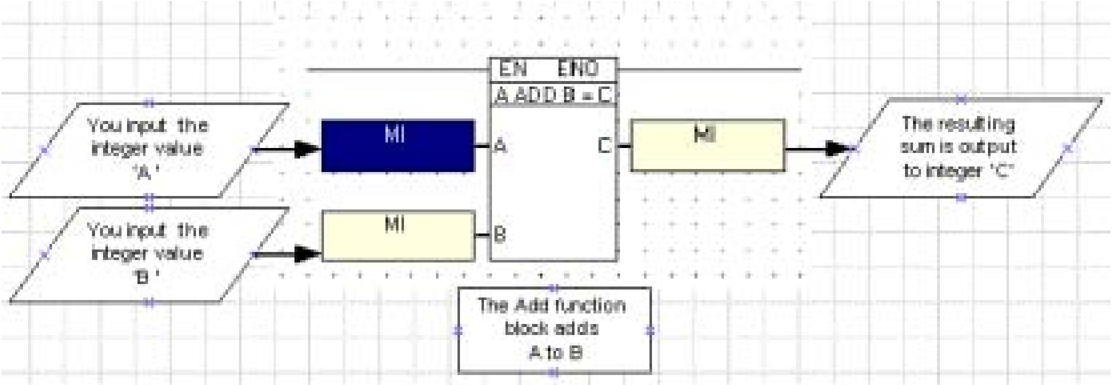
Math Functions

You perform mathematical functions in U90 Ladder by using math function blocks. Function blocks are provided for:

- Addition
- Subtraction
- Multiplication
- Division

The internal operation of a function block is transparent to the user. You simply input the two operands. The result is automatically output by the function block.

The example below shows the Add function block.


EN ENO
A ADD B = C
You input the integer value 'A'
You input the integer value 'B'
MI
MI
A
B
C
MI
The resulting sum is output to integer 'C'
The Add function block adds A to B

Input Operands A & B must be integer values: MI , SI or # constant integer value.

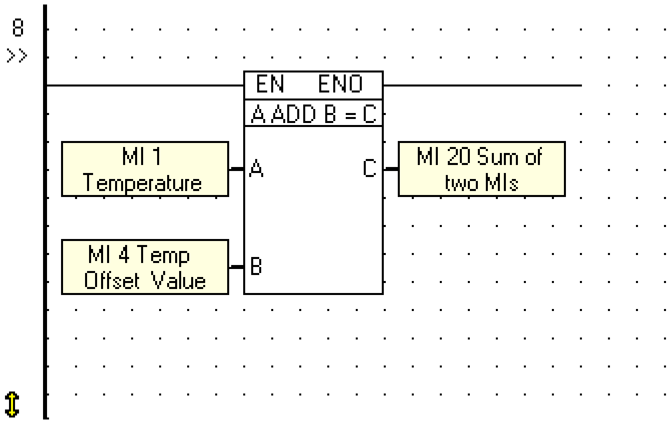
Output Operand C may be a Memory Integer or a System Integer.

You can use an Add function block to assign a real number value to an MI or SI.

Add ＋

Example

The math function add is executed by the Add function block shown below.

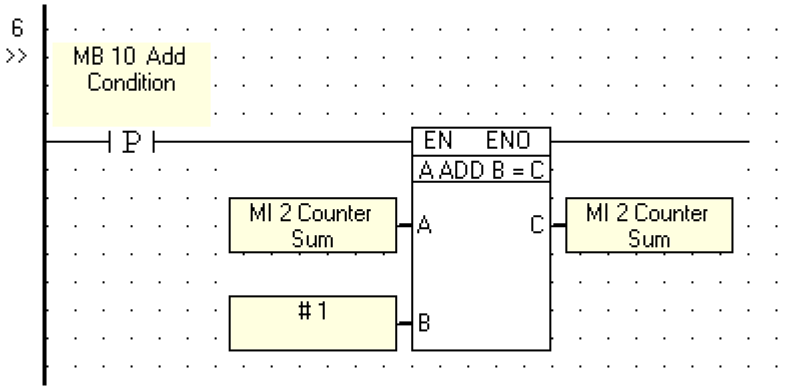


Input Operands A & B must be integer values: MI , SI or #constant integer value.

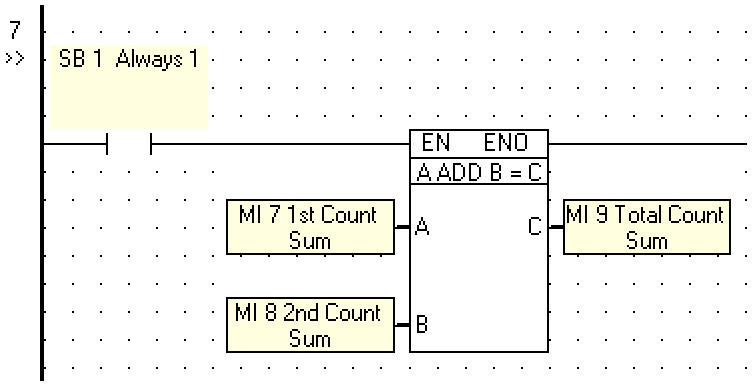
Output Operand C may be a Memory Integer or a System Integer.

Add Examples

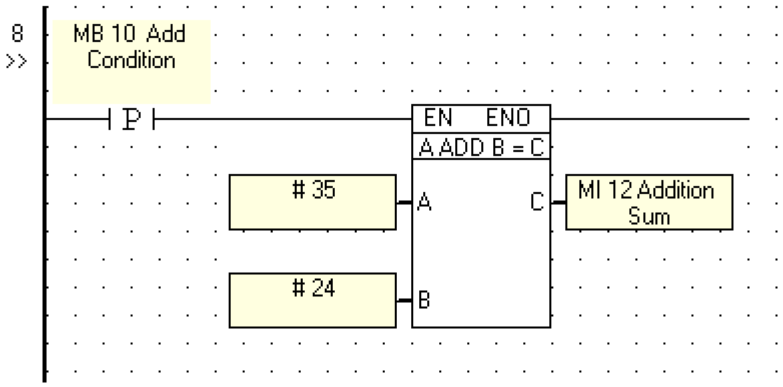
You can use the Add function to add an MI value to an integer value.



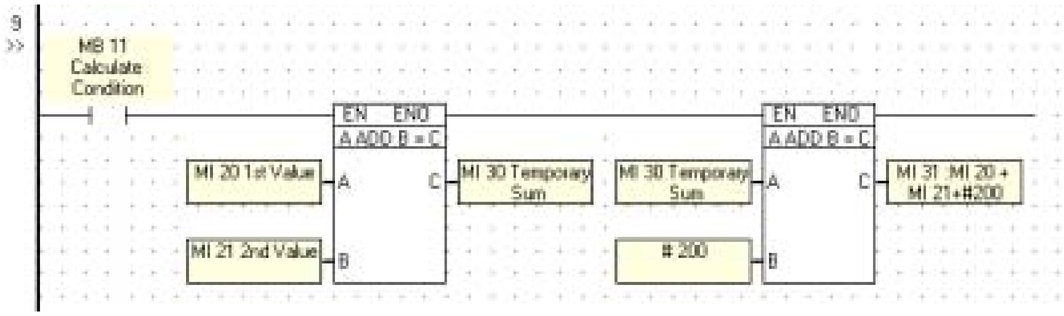
You can use the Add function to add two MI values.



You can use the Add function to add two integer values.



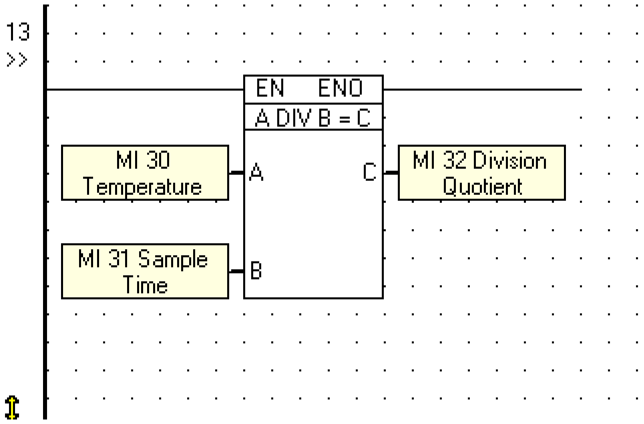
You can use Add function blocks in series.



Divide /

Examples

The math function divide is executed by the Divide function block shown below.



Input Operands A & B must be integer values: MI , SI or #constant integer value.

Output Operand C may be a Memory Integer or a System Integer.

The Divide function can only return whole numbers. The M90 does not support floating point integers. Examples: 7.2 and 9.5.

Use System Integer 4 (**SI 4** - Divide Remainder) to find the exact integer value of a division function that may involve a remainder.

Note that you must use the remainder value in SI 4 immediately after the division function. SI 4 will be written over with the next division function and the specific remainder value will be lost.

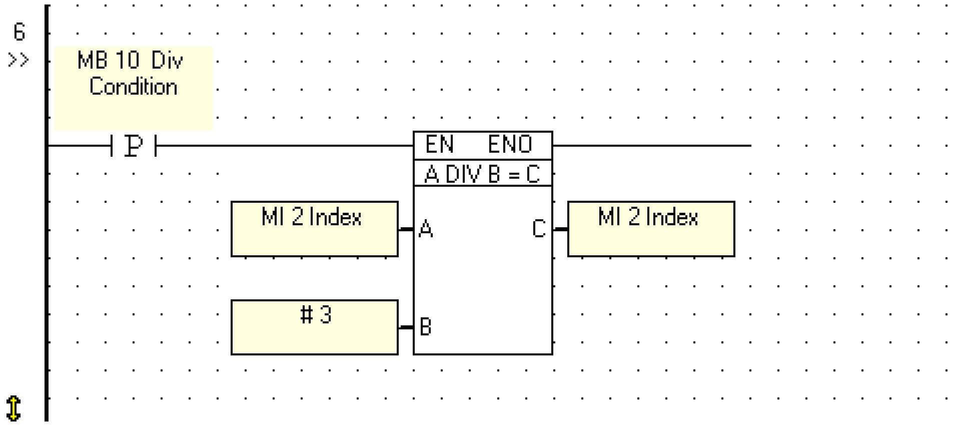
System Bit 4 (**SB 4** - Divide by Zero) will activate if the division operation will inadvertently result in a division by zero and return zero in Operand C.

Division Examples

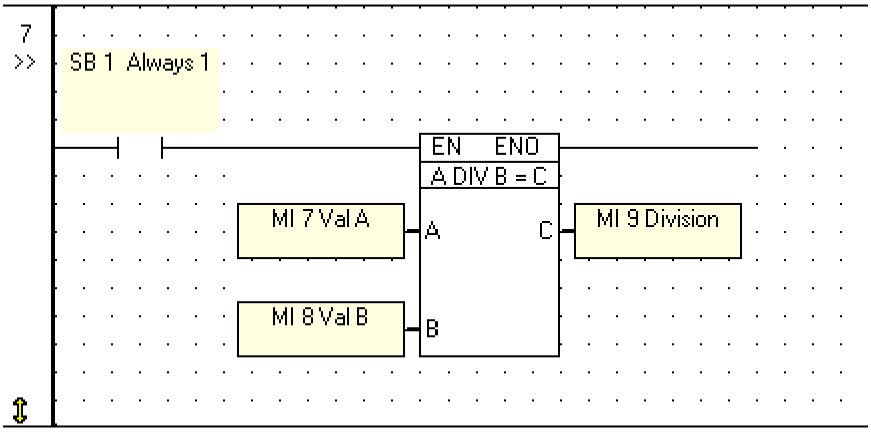
Remember that any remainder of a Division function will be written into **SI 4**. You must use any remainder value immediately after the Division function because **SI 4** will be written over with the next division function and the specific remainder value will be lost.

SB 4 (Divide by Zero) will activate if the division operation will inadvertently result in a division by zero and it will return zero in Operand C.

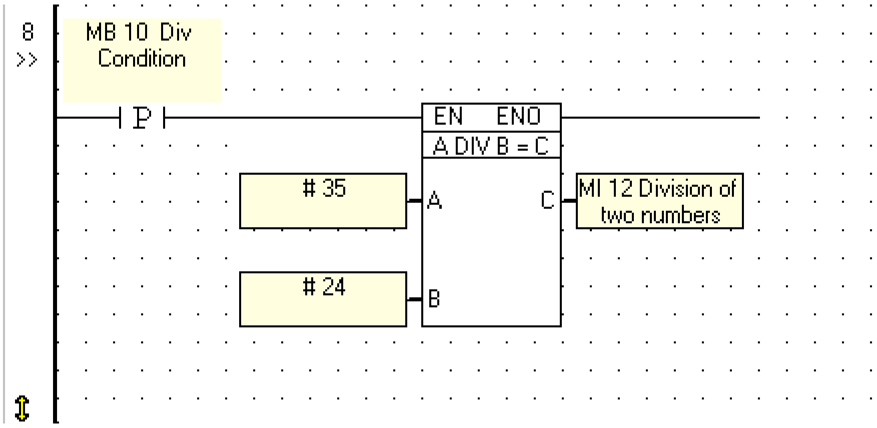
You can use the Division function to divide an MI value and integer value.



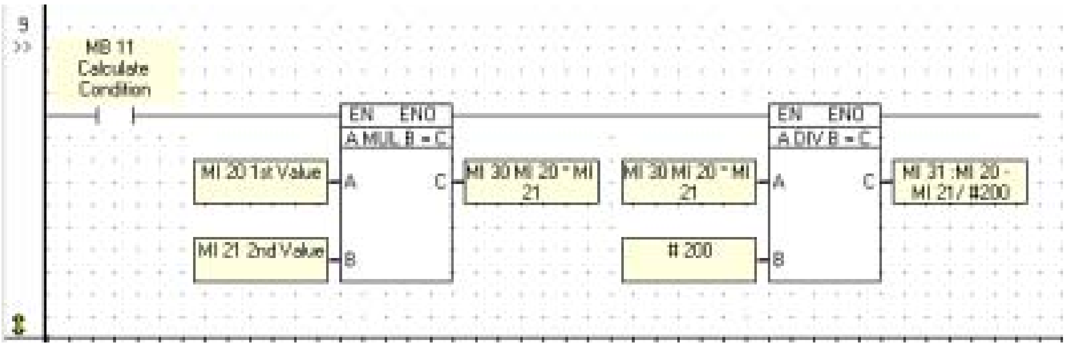
You can use the Division function to divide two MI values.



You can use the Division function to divide two integer values.



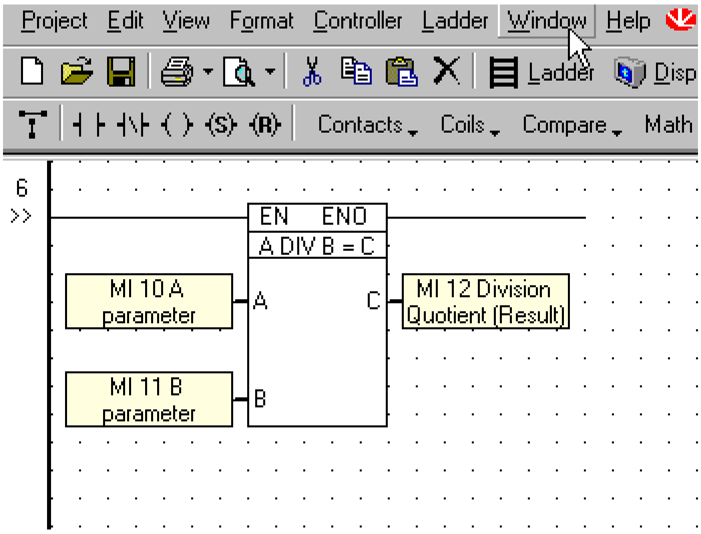
You can use Math function blocks in series.



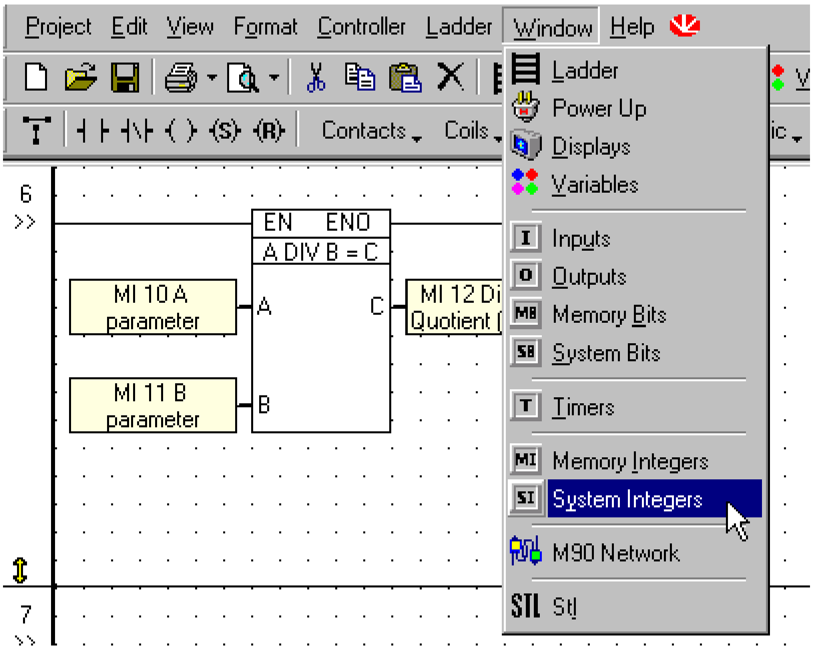
Division Function: Remainder values

To get the remainder value of a Division function:

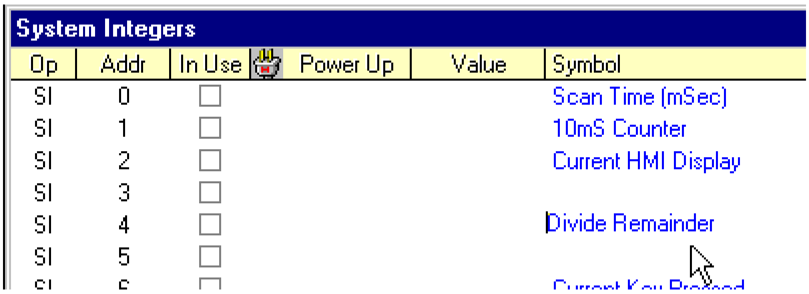
1. Enter the desired Operands into the Division function block.



2. Select **System Integers** from the Window Menu on the Standard Menu bar.



3. SI 4 holds the Remainder value for the most recent Division operation.



System Integers

Op	Addr	In Use	Power Up	Value	Symbol
SI	0	□			Scan Time (mSec)
SI	1	□			10mS Counter
SI	2	□			Current HMI Display
SI	3	□			
SI	4	□			Divide Remainder
SI	5	□			
SI	6	□			Current Key Pressed

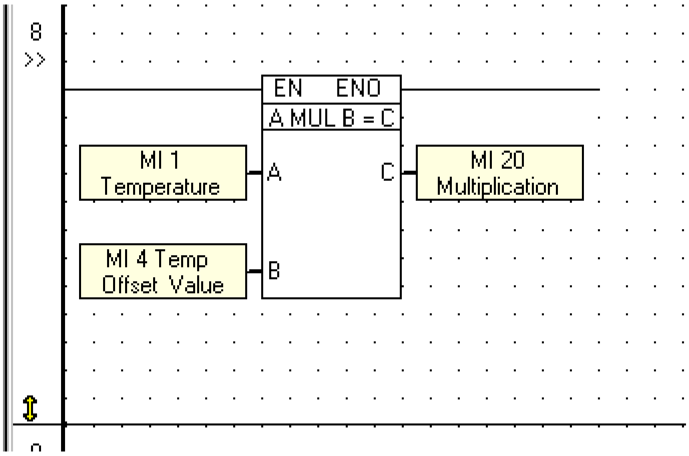
According to the above example:

If MI 10 = 7 and MI 11 = 2, then MI 12 = 3 and SI 4 = 1

Multiply *

Examples

The math function Multiply is executed by the Multiply function block shown below.

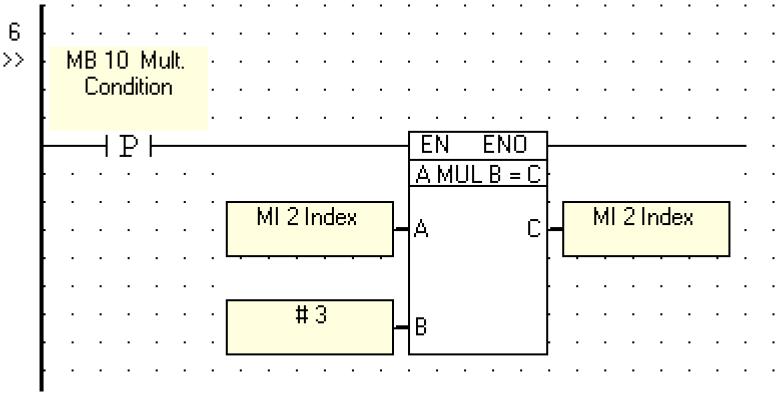


Input Operands A & B must be integer values: MI, SI or #.

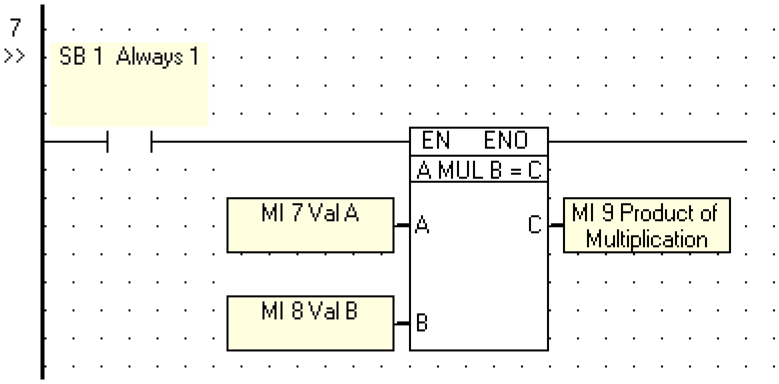
Output Operand C may be a Memory Integer or a System Integer.

Multiplication Examples

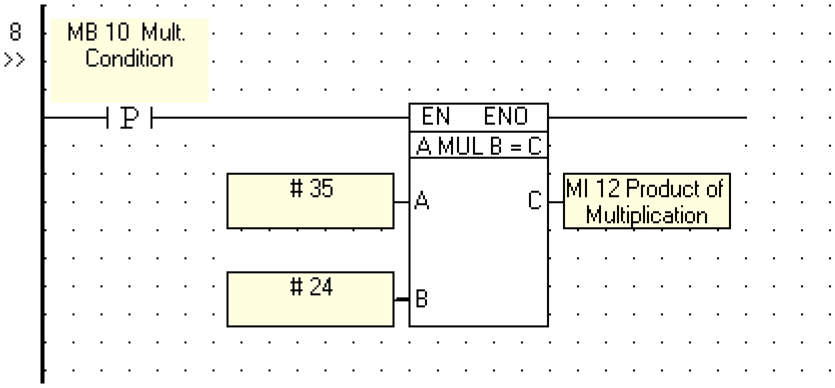
You can use the Multiplication function to multiply an MI with an integer value.



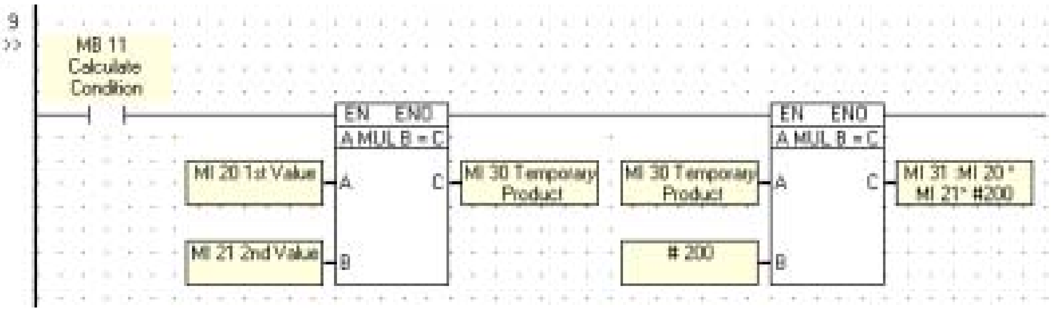
You can use the Multiplication function to multiply two MI values.



You can use the Multiplication function to multiply two integer values.



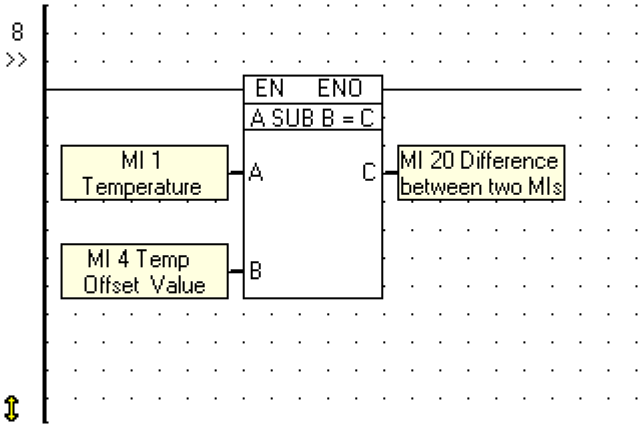
You can use Multiplication function blocks in series.



Subtract -

Examples

The math function subtract is executed by the Sub function block shown below.

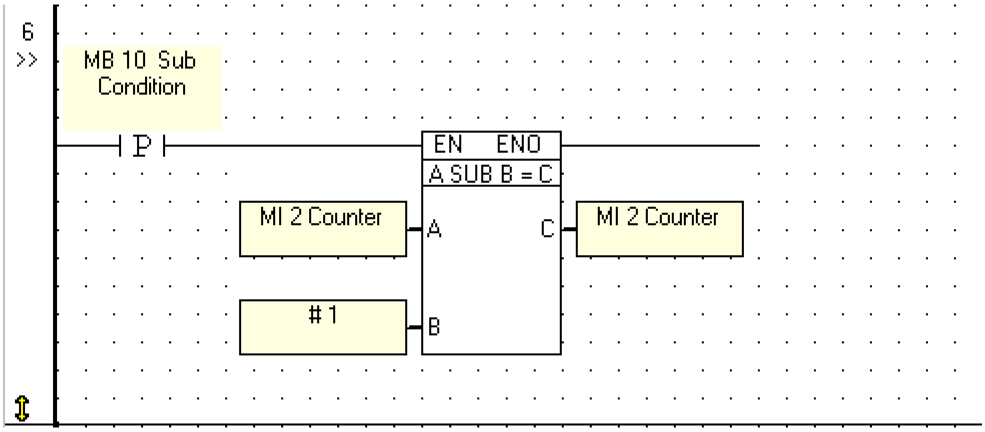


Input Operands A & B must be integer values: MI , SI or # constant integer value.

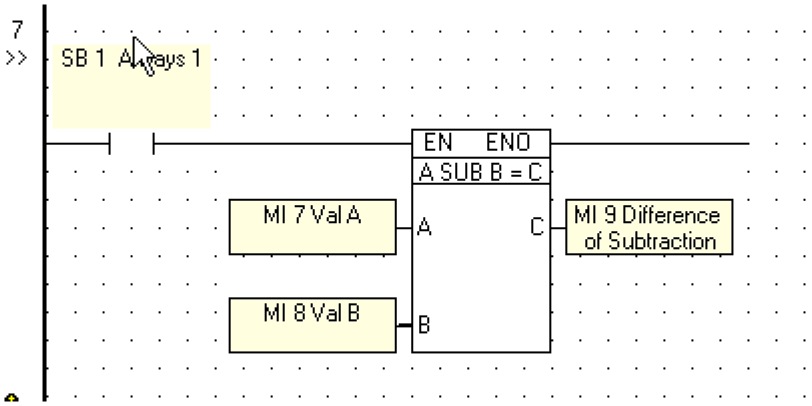
Output Operand C may be a Memory Integer or a System Integer.

Subtraction Examples

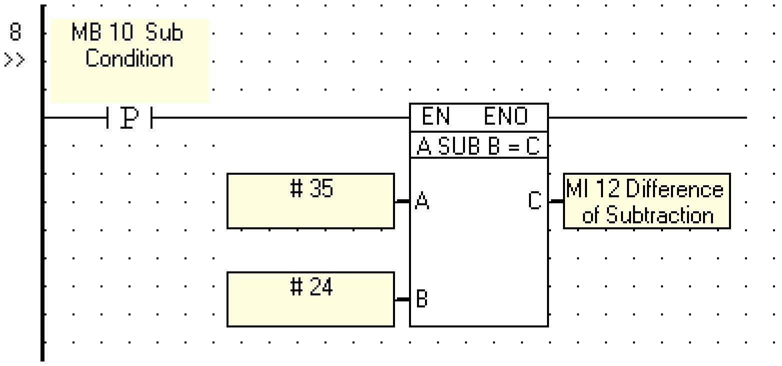
You can use the Subtraction function to subtract between an MI value and an integer value.



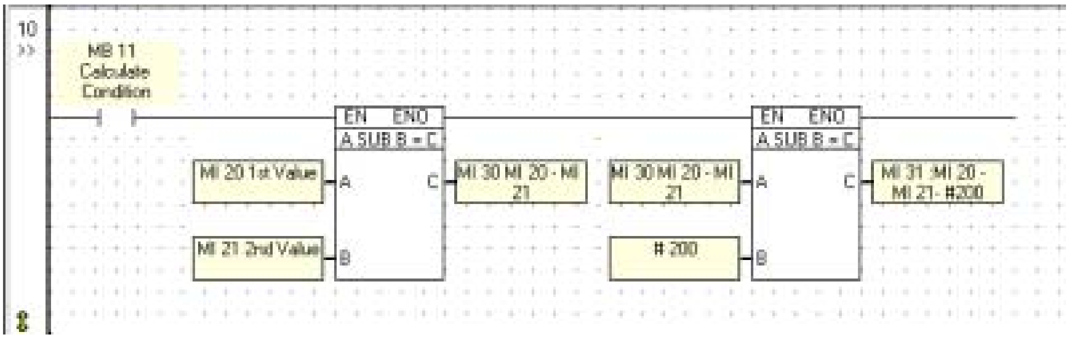
You can use the Subtraction function to subtract between two MI values.



You can use the Subtraction function to subtract between two integer values.



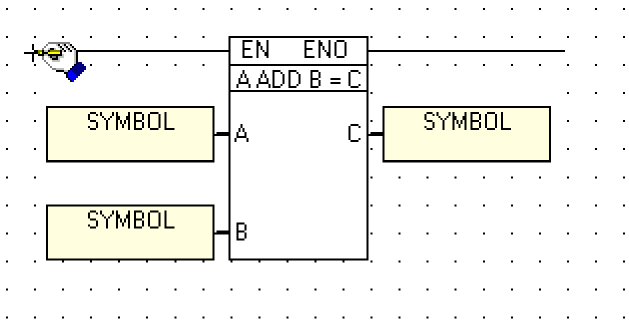
You can use Subtraction function blocks in series.



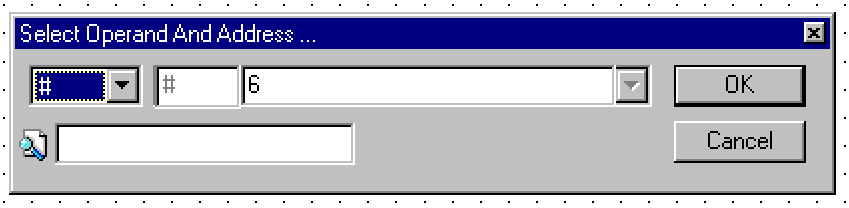
Math functions: Constant integers, MI,or SI

To execute a math function between an integer and MI/SI:

Each Math function has 3 elements: 2 input values and 1 output value. Each of these 3 elements has the possibility of being an integer (as well as a MI or SI).



From the Select Operand and Address dialog box select # for the Operand type and Address. Enter the integer (number) value in the Symbol box.


Select Operand And Address ...

#	#	6

Store Functions

An MI Operand contains an integer value (-32768 to +32767).

There are two ways to store an integer value in an MI:

- Store Direct
- Store Indirect

The last integer value written into a specific MI will overwrite any previous integer value stored there before.

Example:

MI 6 = 35. You then write the value 37 into MI 6, the value 35 will be replaced by the value 37.

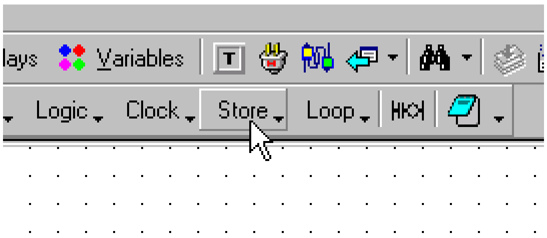
Store functions are featured in several sample applications, such as the application ' History of Events'. These applications may be found by selecting Sample U90 Projects from the Help Menu.

Store Direct function

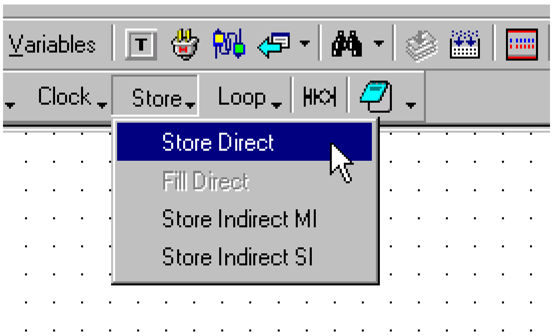
Store Direct allows you to write a constant, MI or SI value into another MI or SI.

To use the Store Direct function:

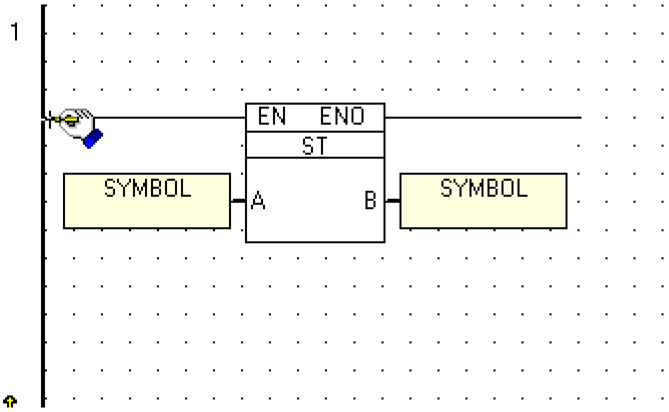
1. Click **Store** on the Ladder Toolbar.



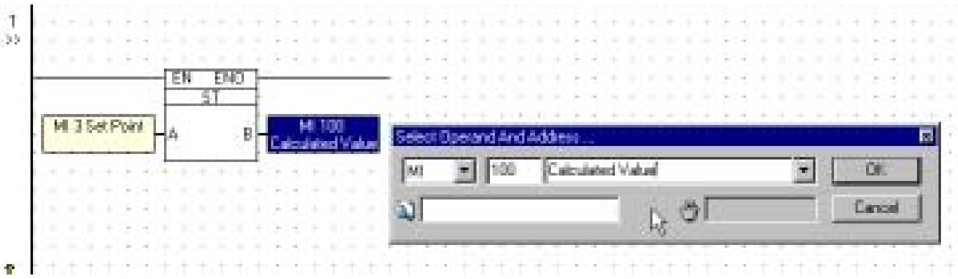
2. Select **Store Direct** from the Store Menu.



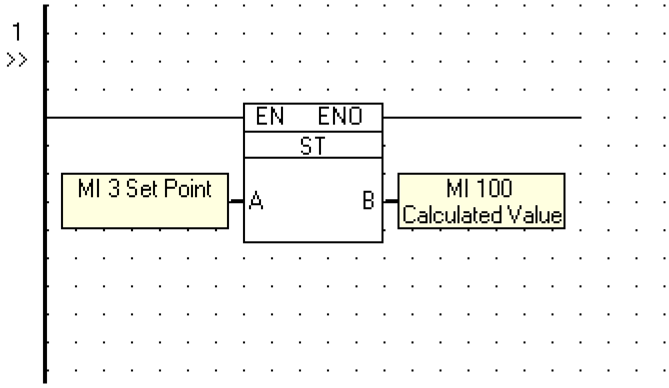
3. Move the Store Direct element to the desired net.



4. Enter the desired Operands and Addresses.



5. The Store Direct element appears on the net with the set Operands and Addresses.



According to the above example, the value in MI 3 will be stored in MI 100. The previous value in MI 100 is **lost**. The current value in MI 3 remains **unchanged**.

Store Indirect function

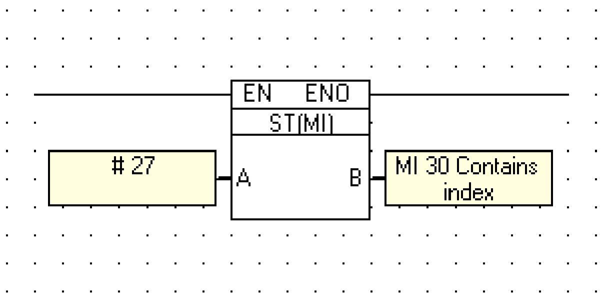
Store Indirect allows you to write an integer value (constant, MI or SI) into another MI or SI using indirect addressing.

For example:

When using the Store Indirect MI, if the value stored in the B parameter is 5; then MI 5 is the address where the value will be stored.

When using the Store Indirect SI, if the value stored in the B parameter is 2; then SI 2 is the address where the value will be stored.

For example:



According to the above example:

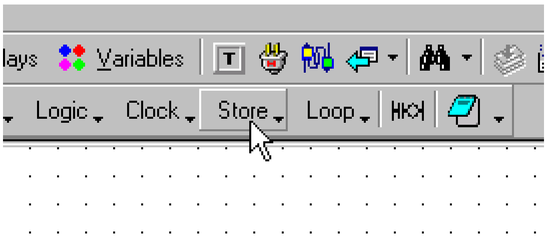
- If MI 30 contains the constant 5; then #27 will be stored in MI 5.
- If MI 30 contains the constant 35; then #27 will be stored in MI 35.

There are two types of Store Indirect function:

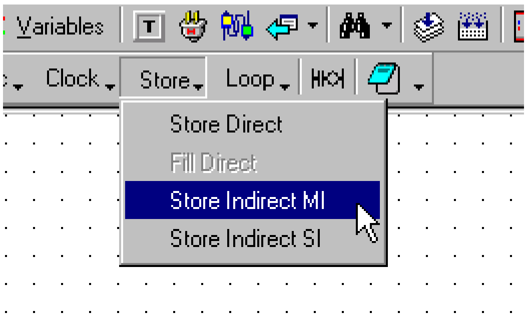
- The Store Indirect MI function relates to the MI address.
- The Store Indirect SI function relates to the SI address.

To use the Store Indirect function:

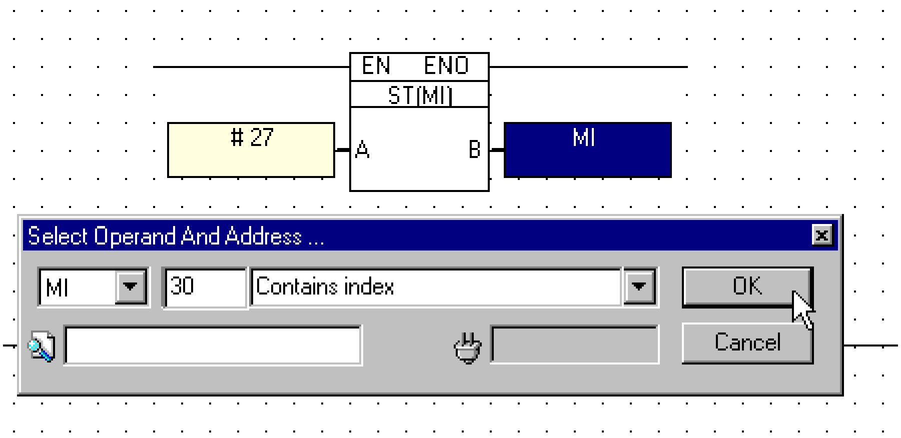
1. Click **Store** on the Ladder Toolbar.



2. Select **Store Indirect MI** on the Store menu.



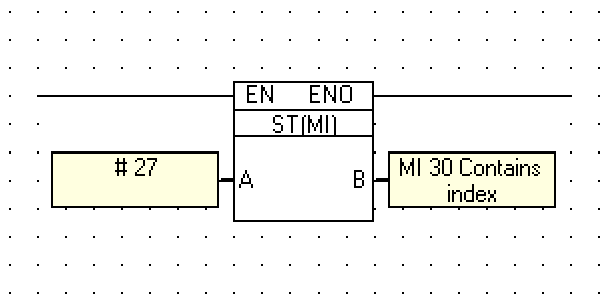
3. Enter the desired Operands, Addresses and Symbols. Click OK.



Select Operand And Address ...

MI | 30 | Contains index

4. The **Store Indirect MI** element appears on the net.



Time Functions

Clock Functions

You perform clock and calendar functions in the U90 Ladder with Clock function blocks. Function blocks are provided for:

- Time
- Day of the Week
- Day of the Month
- Month
- Year

You activate these functions through the **Clock** drop-down menu of the Ladder toolbar.

The U90 Ladder provides 2 methods for executing Clock functions:

- Direct
- Indirect

You set the value of **Direct Clock** functions when you write your project.

The **user** sets the value of an **Indirect Clock** function from the M90 via the keypad.

Clock functions are featured in several sample applications, such as the applications ' School Bell Direct', 'Database Log', and 'Print & Time'. These applications may be found by selecting Sample U90 Projects from the Help Menu.

Direct Clock function

The Direct Clock function allows the programmer to write a Ladder program using calendar conditions for:

- Time of Day
- Day of the Week
- Day of the Month
- Month
- Year

These functions are located on the **Clock** drop-down menu of the Ladder toolbar.

You set the value of **Direct Clock** functions when you write your project.

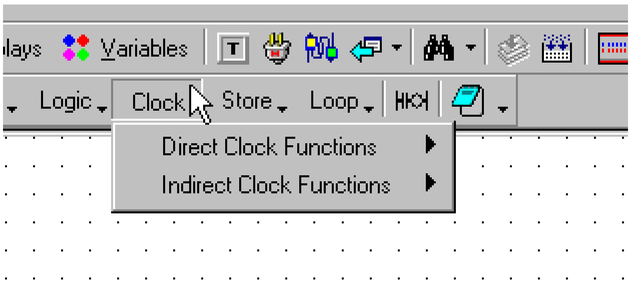
You must use the **Indirect Clock** functions if you want the user to set the value of a Clock function from the M90.

Direct Clock function example

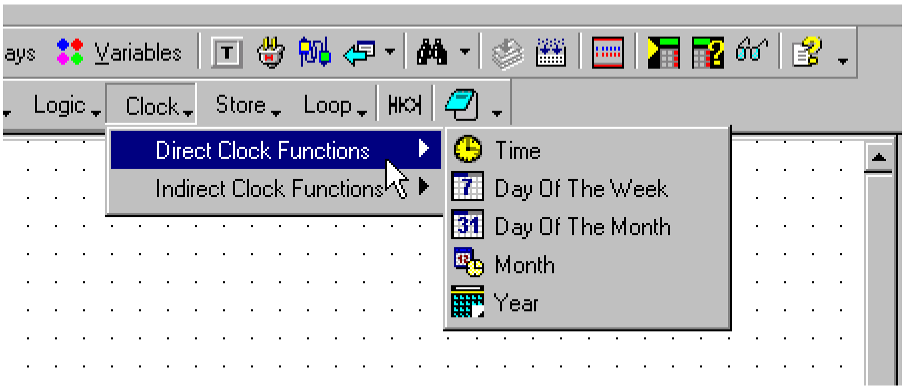
You want to create a project where a machine is working

- in January and March
- beginning on the 12th day of a month, until and including the 20th
- in the years 2000 and 2001
- between the hours 10:30 and 12:15.

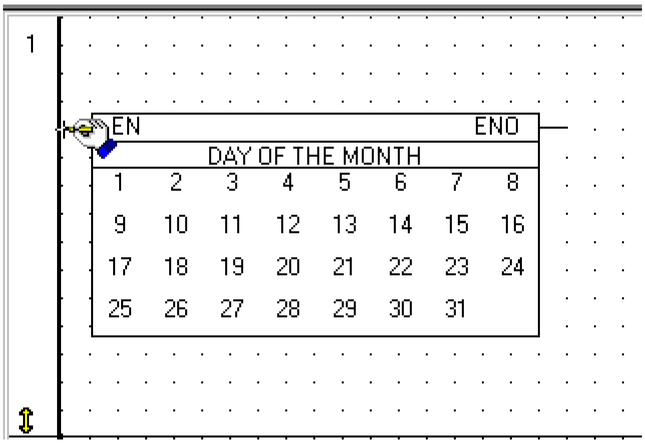
1. Click **Clock** on the Ladder toolbar.



2. Select **Direct Clock Functions.** The Direct Clock Functions menu opens.

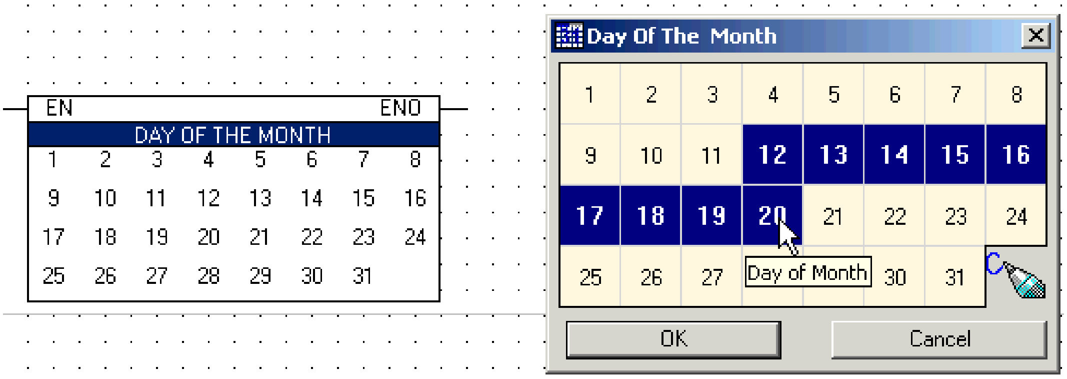


3. Select **Day of the Month** and place it in the desired place on the net.

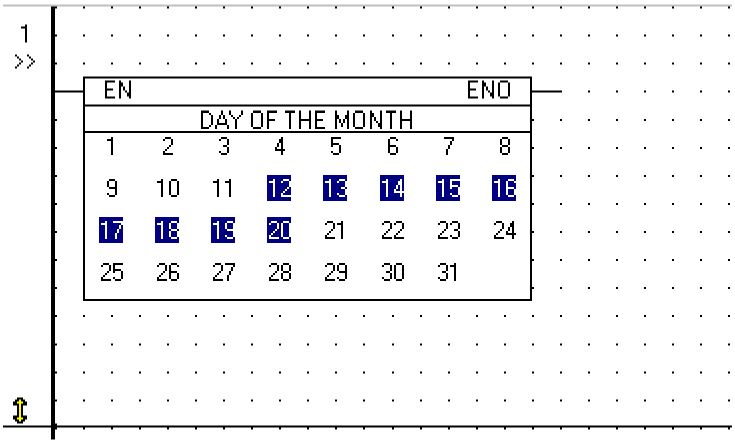


4. The Day of the Month menu opens.

5. Click the desired days of the month.

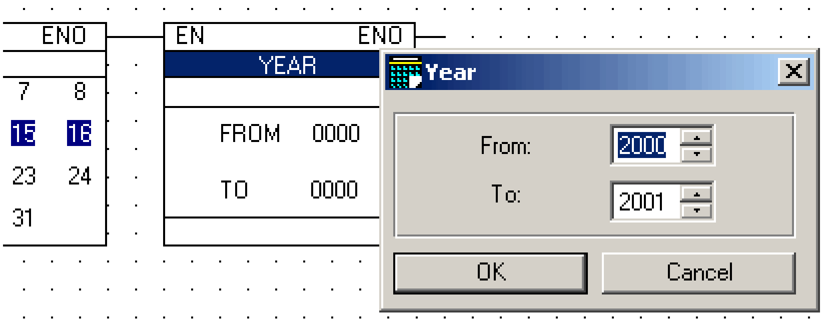


6. The **Days of the Month** function appears on the net with the selected days of the week highlighted.

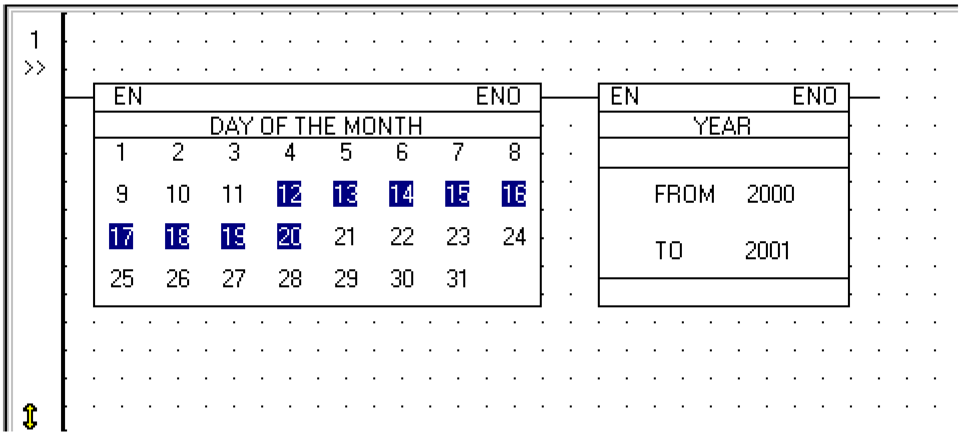


7. Select the **Year** function. The Year menu opens.

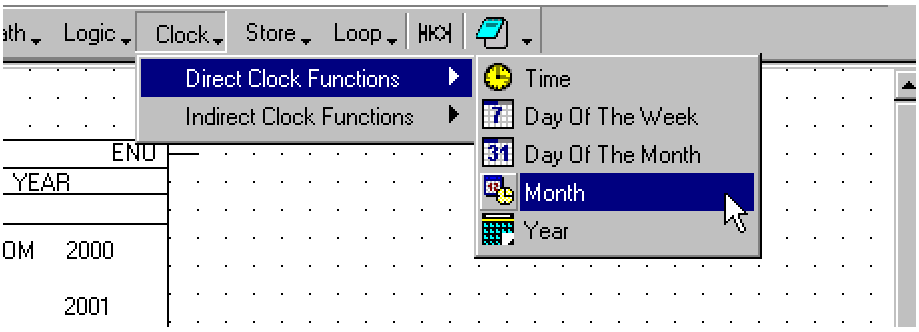
8. Enter the desired **Year** range.



9. The Year function appears with the desired values.

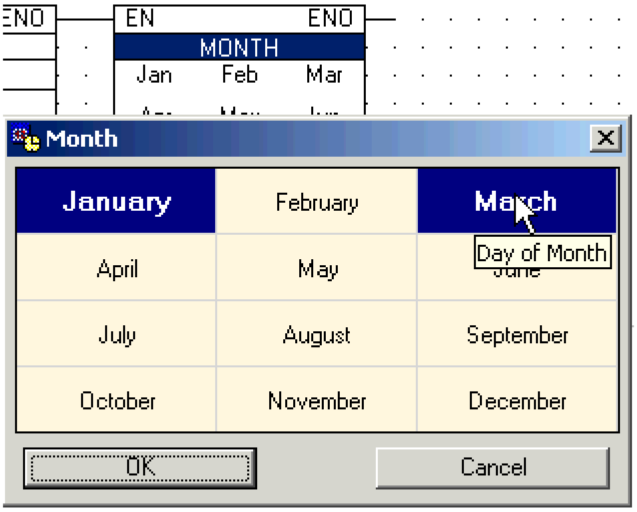


10. Select **Month** on the **Direct Clock** Functions menu.

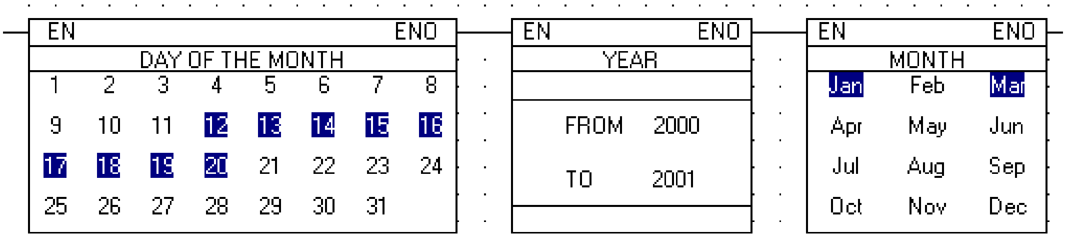


11. The **Month** menu opens.

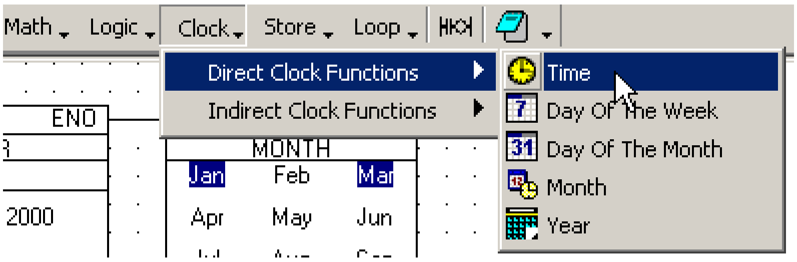
12. Select the desired Months. Click OK.



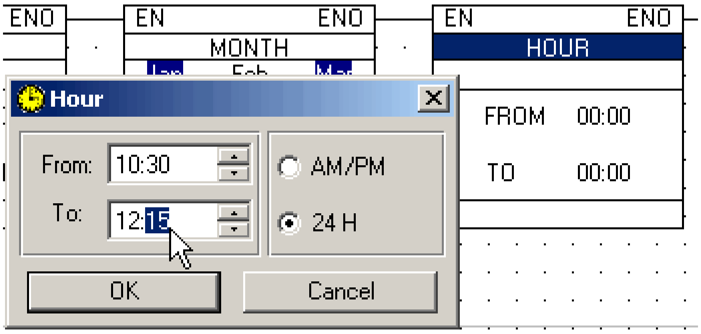
13. The **Month of the Year** function appears with the desired Months highlighted.



14. Select **Time** from the Direct Clock Functions menu.

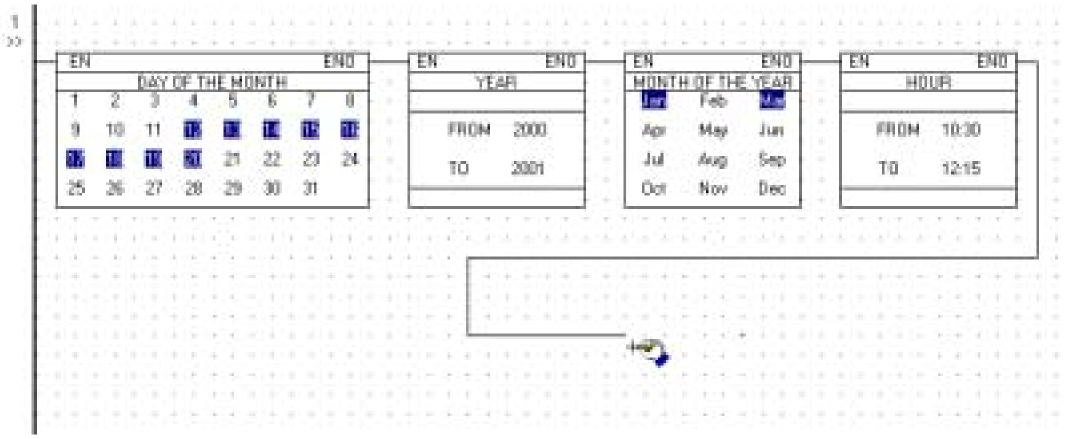


15. Enter the desired Time range in the **Hour** menu. Click OK.

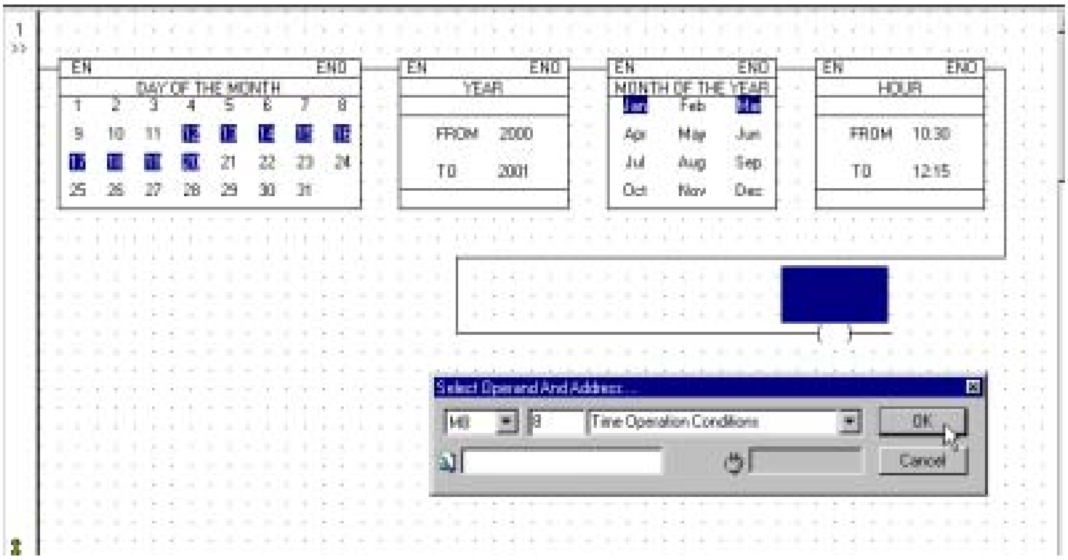


From: 10:30
To: 12:15
○ AM/PM
● 24 H

16. Expand the net rung as needed in the net using the **Line Draw** tool.

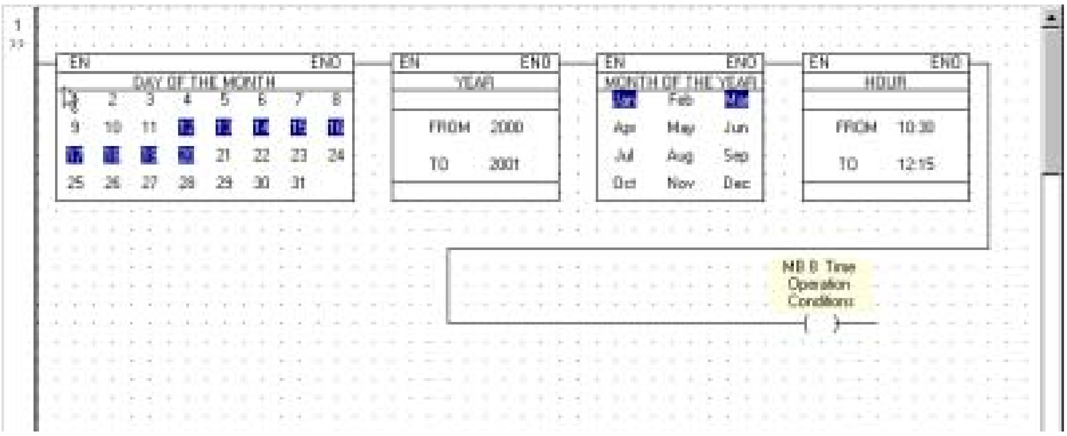


17. Select and place a Direct Coil on the net. Enter the desired Operand, Address and Symbol.



MB 8 Time Operation Conditions

18. The net appears as shown below.



Indirect Clock function

Indirect Clock functions allows the programmer to write a Ladder program where the user will enter the time value via the M90 keypad. Functions are provided for::

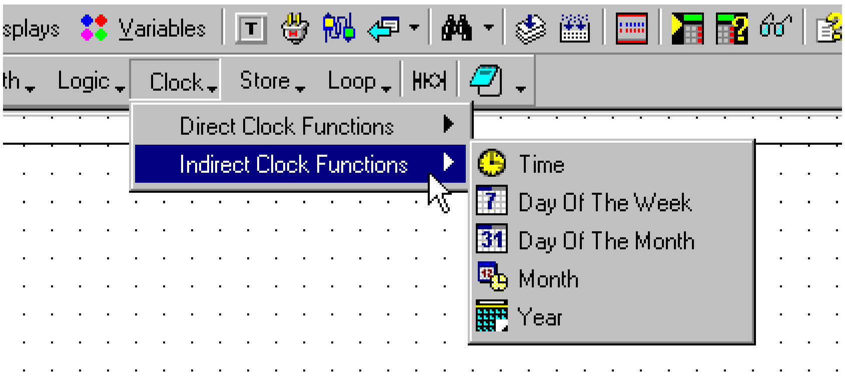
- Indirect Time of Day
- Indirect Day of Week
- Indirect Day of Month
- Indirect Month
- Indirect Year

These functions are located on the **Clock** drop-down menu of the Ladder toolbar.

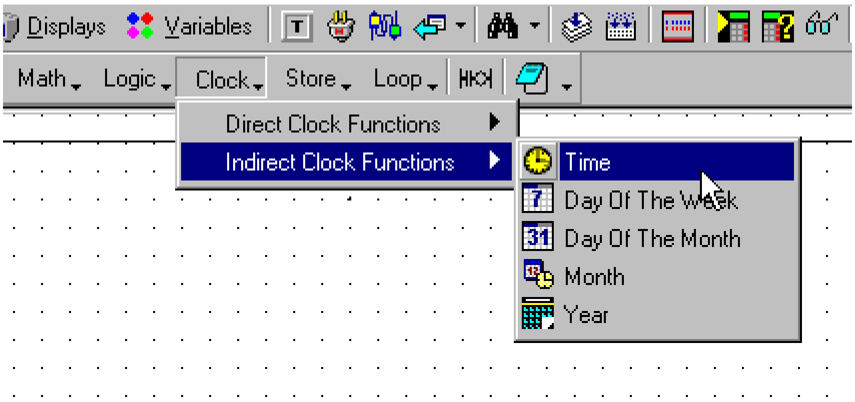
Indirect Clock function example

You want to create a project where a machine is working according to a time and date entered by the user via the M90 keypad.

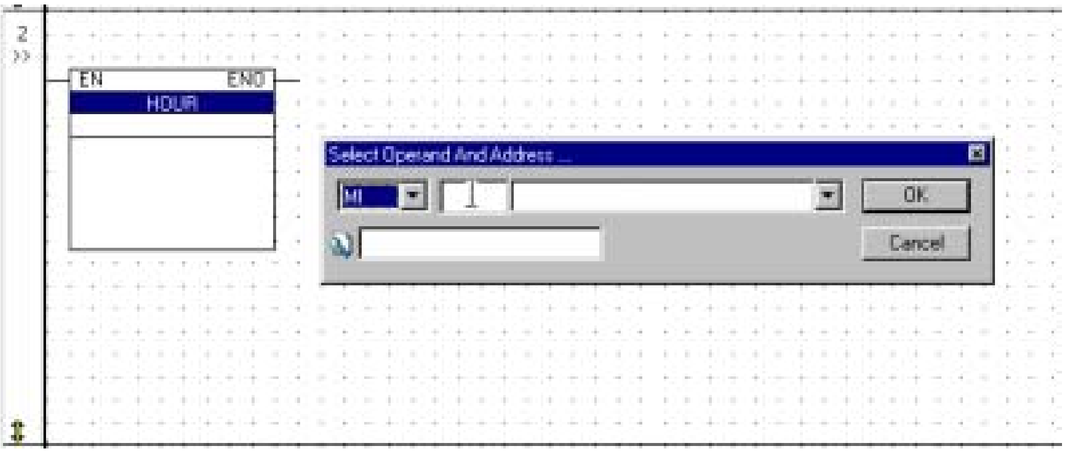
1. Select **Indirect Clock Functions** from the Clock menu of the Ladder toolbar.



2. Select **Time** from the Indirect Clock Functions menu.

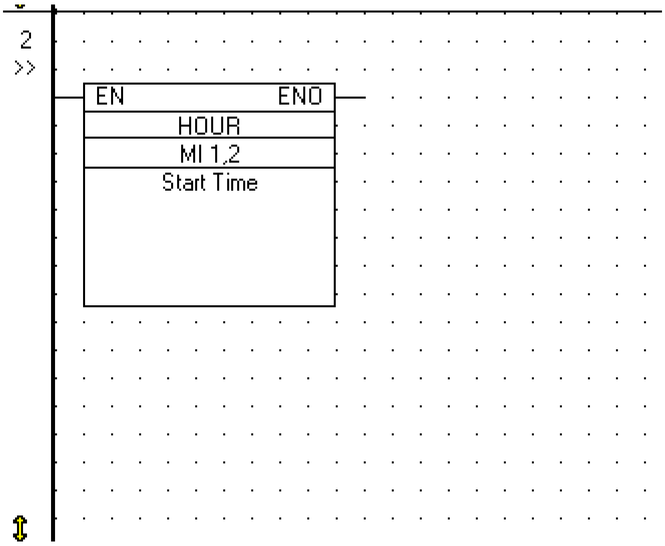


3. Enter the desired Operand, Address and Symbol.

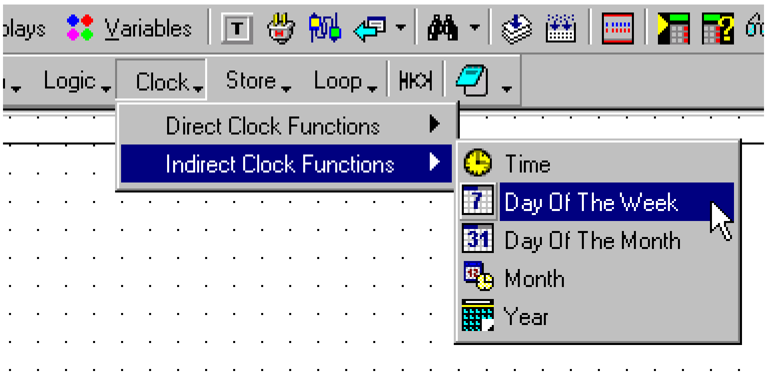


4. The Hour function appears with the selected Operand and Address. Note that the hour function is checking a range between two MIs / SIs. Therefore, two Operands are needed: the beginning and the end of the range.

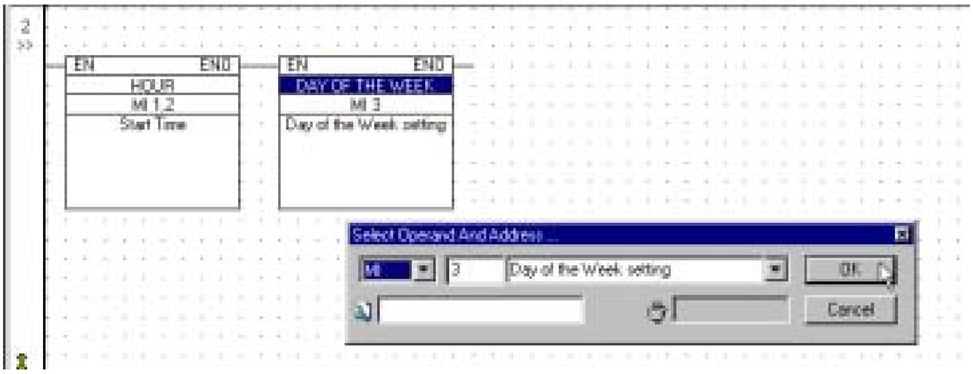
   The program automatically takes the **next** Operand from the one you enter. According to the following example, you enter MI 1 and the program assigns the end of the range to MI 2, the **next** MI.



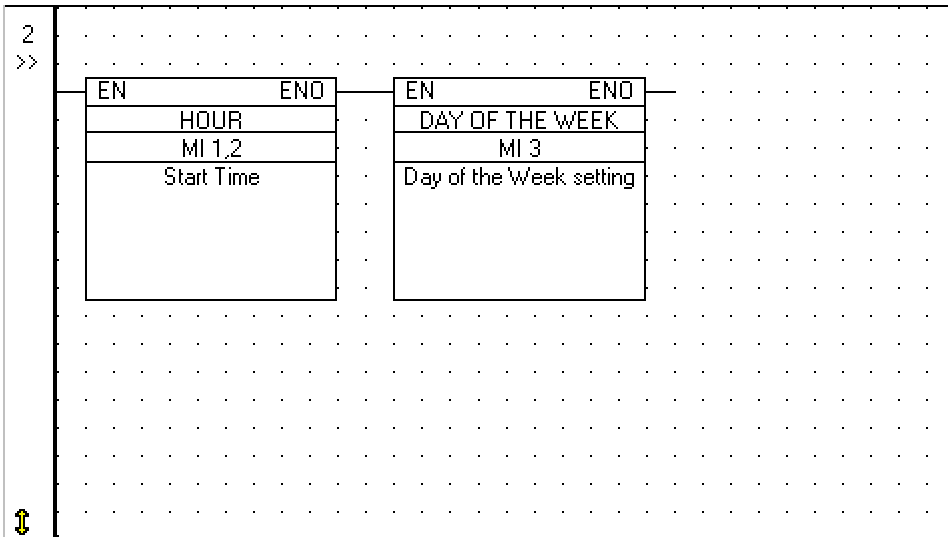
5. Select **Day Of The Week** from the **Indirect Clock Function** menu.



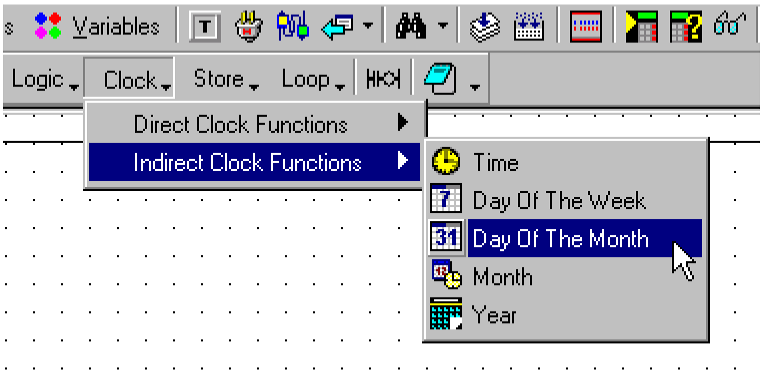
6. Place the Day Of The Week function on the desired net. Enter the desired Operand, Address and Symbol.



7. The **Day Of The Week** function appears with the selected Operand, Address and Symbol on the net.

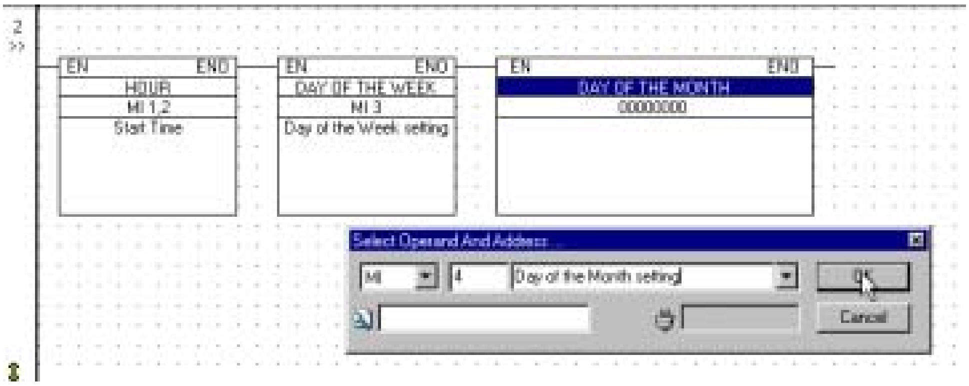


8. Select **Day of the Month** from the **Indirect Clock Function** menu.



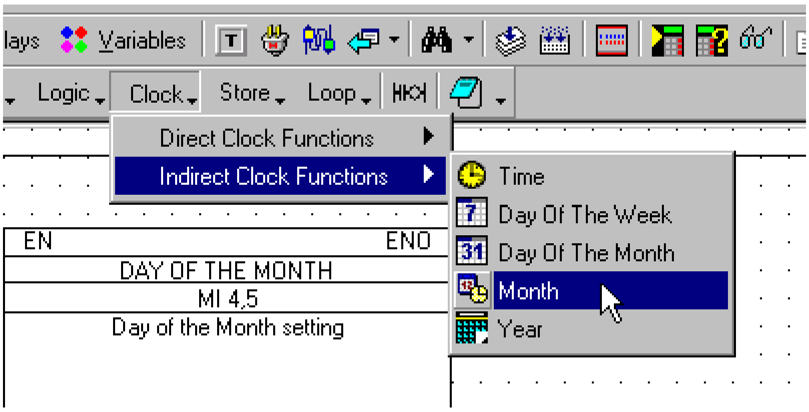
9. Enter the desired Operand, Address and Symbol. The Day of the Month function is a 32 - bit Bit map. Therefore it requires two MIs / SIs.

   The program automatically takes the **next** Operand from the one you enter. According to the following example, you enter MI 4 and the program assigns the end of the range to MI 5, the **next** MI.

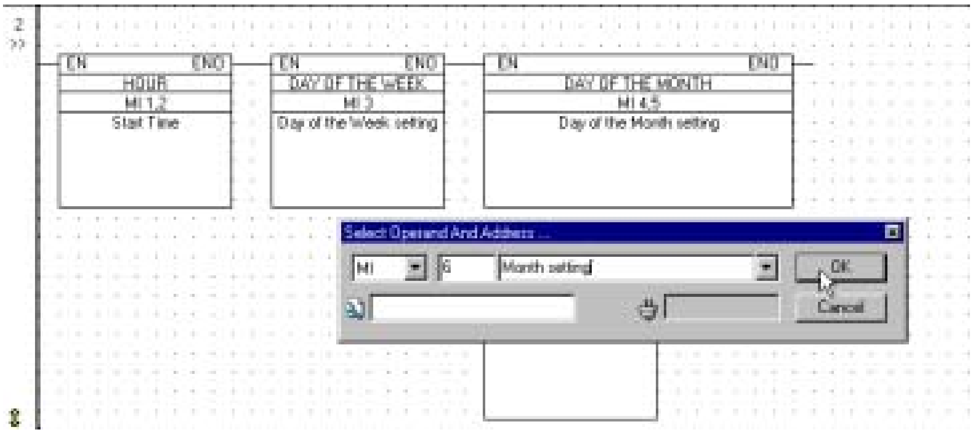


MI 4 Day of the Month setting

10. Select **Month** from the Indirect Clock Functions menu.

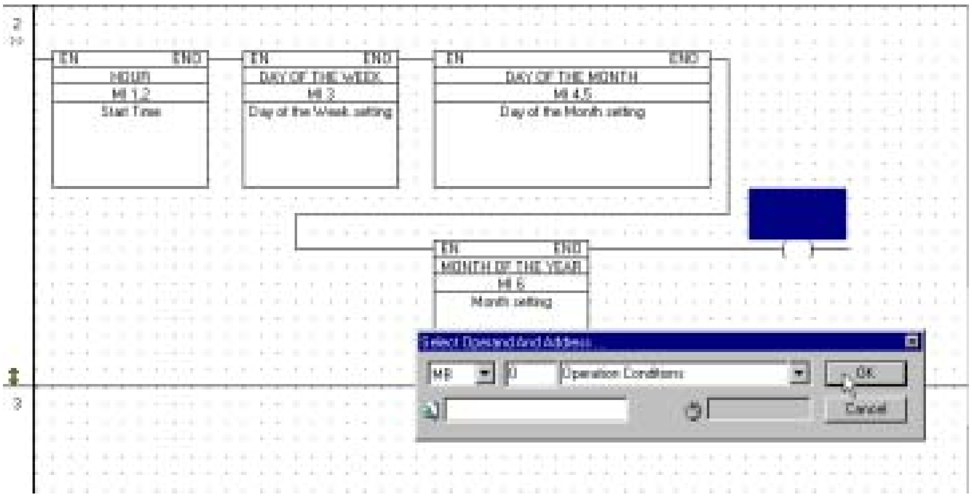


11. Enter the desired Operand, Address and Symbol.

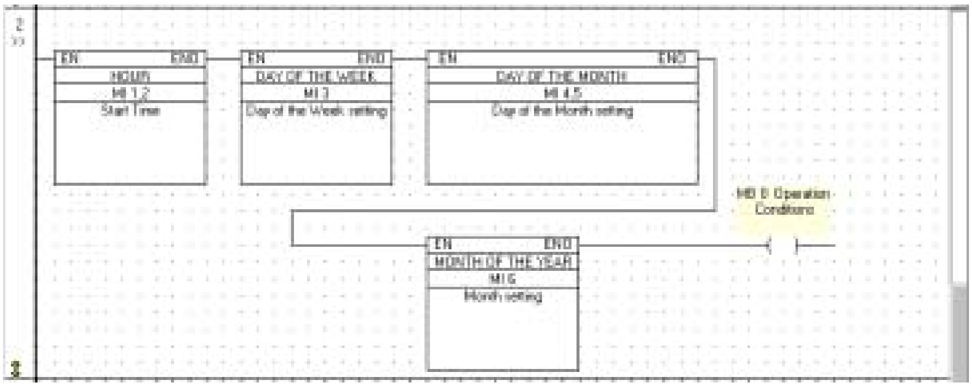


MI 6 Month setting

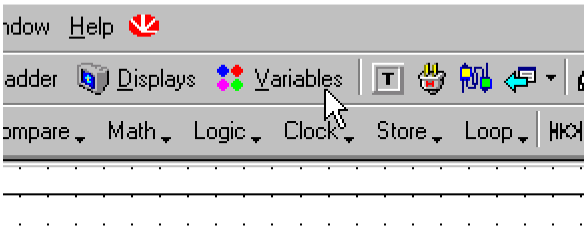
12. Place a **Direct Coil** at the end of the rung. Enter the desired Operand, Address and Symbol.



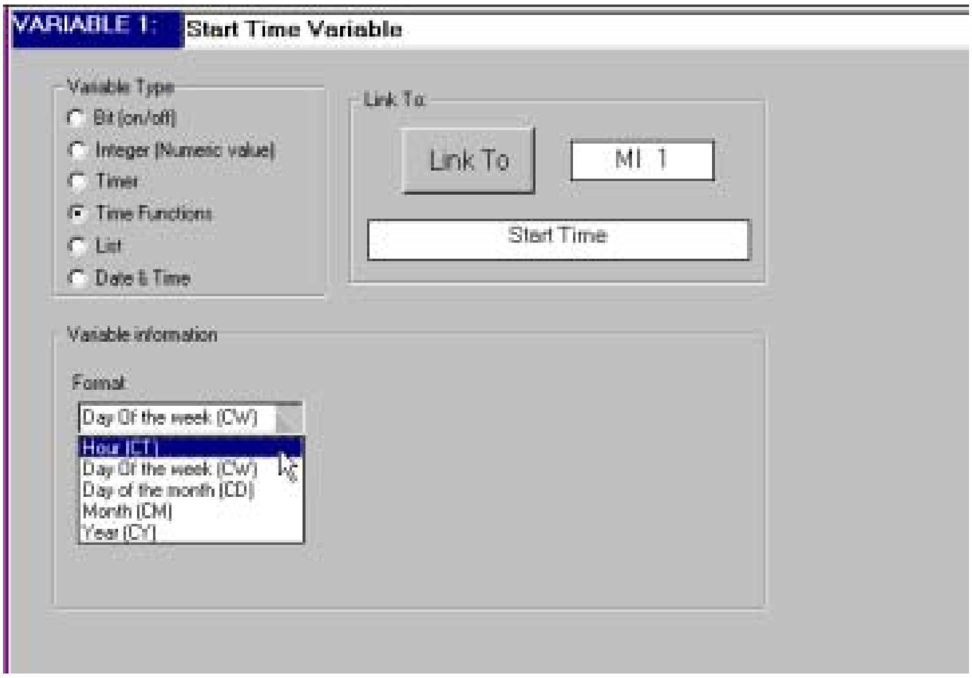
13. The net appears as such:



14. To enable the user to view and modify the Indirect Clock function values, you must now create HMI Displays and Variables. Click **Variables** on the Standard toolbar.



15. The Variable Editor opens. Select **Time Functions** for each Variable. Link the Variable to the appropriate MI. Select the appropriate Variable Information Format for the time. Below is the Start Time Variable for the time in hours.



VARIABLE 1: Start Time Variable

Variable Type
- ○ Bit (on/off)
- ○ Integer (Numeric value)
- ○ Timer
- ● Time Functions
- ○ List
- ○ Date & Time

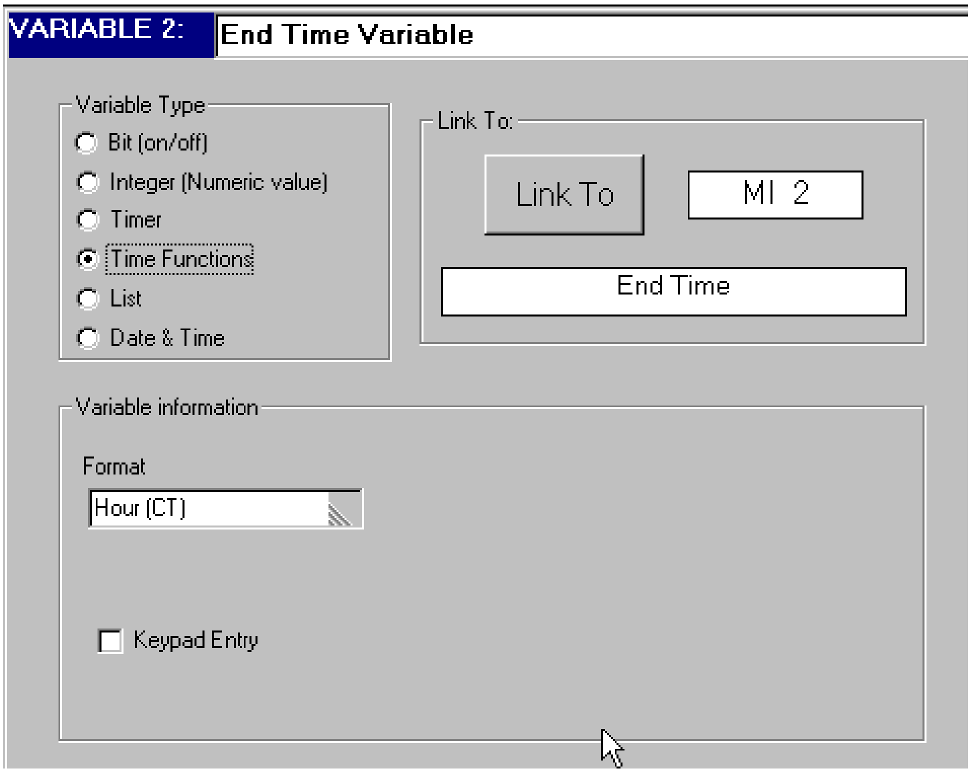
Link To:

MI 1
Start Time

Variable information

Format: Day Of the week (CW)
- Hour (CT)
- Day Of the week (CW)
- Day of the month (CD)
- Month (CM)
- Year (CY)

16. The End Time Variable for the time in hours.



VARIABLE 2: End Time Variable

Variable Type
- ○ Bit (on/off)
- ○ Integer (Numeric value)
- ○ Timer
- ● Time Functions
- ○ List
- ○ Date & Time

Link To:

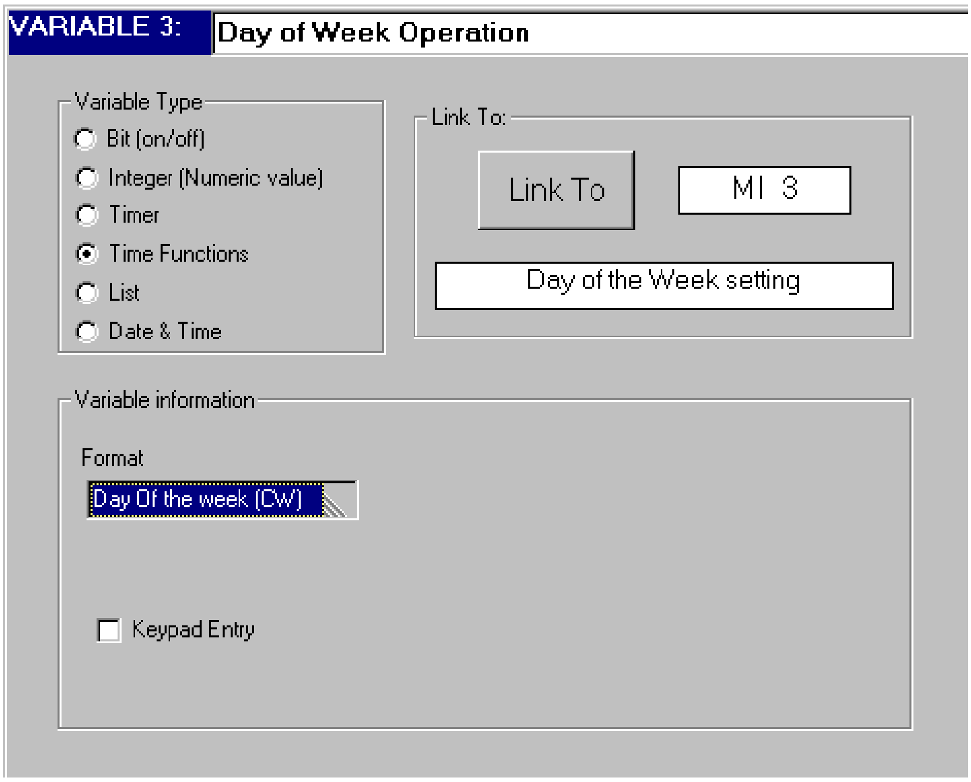
MI 2
End Time

Variable information

Format: Hour (CT)

☐ Keypad Entry

17. The Day of the Week Variable.



VARIABLE 3: Day of Week Operation

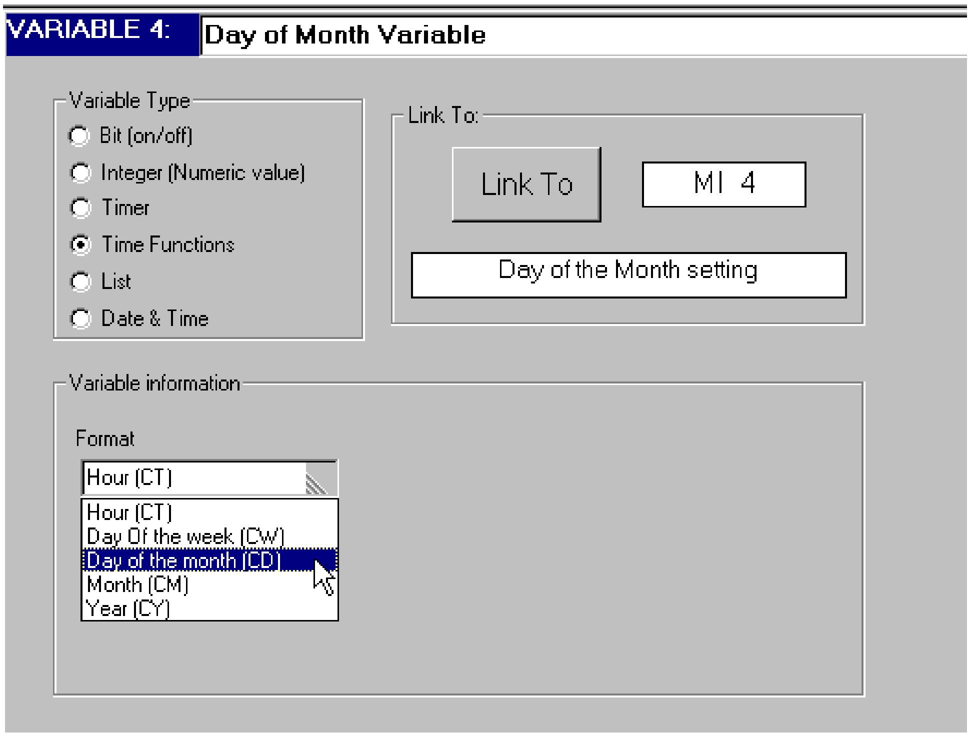
Variable Type
○ Bit (on/off)
○ Integer (Numeric value)
○ Timer
● Time Functions
○ List
○ Date & Time

Link To:
Link To
MI 3
Day of the Week setting

Variable information
Format
Day Of the week (CW)

□ Keypad Entry

18. The Day of Month Variable.



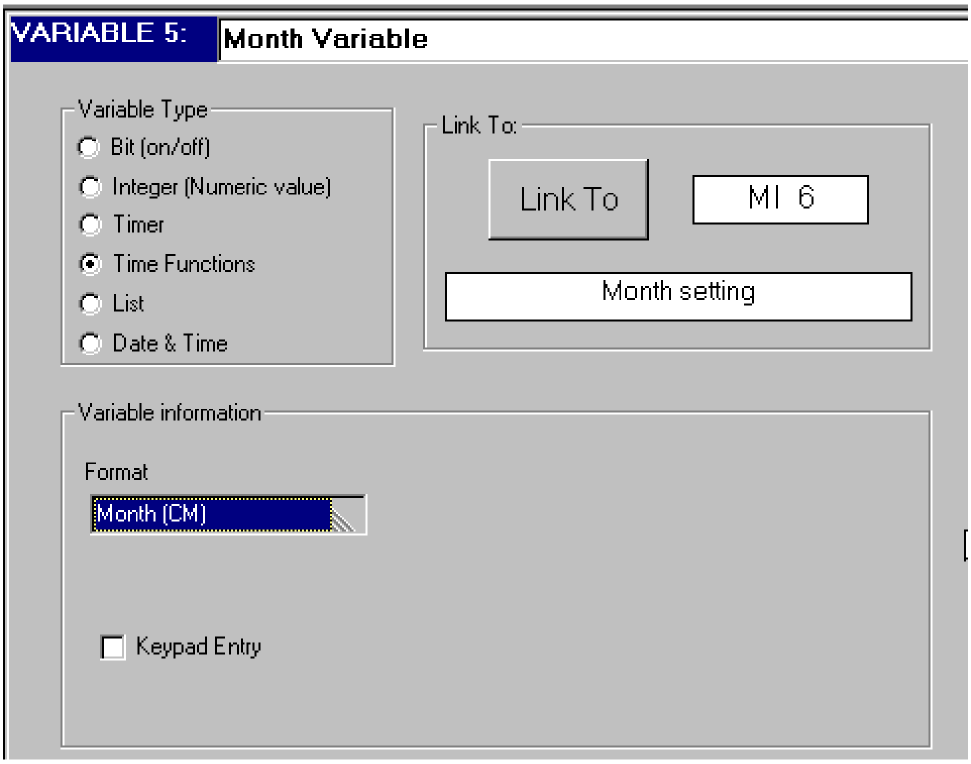
VARIABLE 4: Day of Month Variable

Variable Type
○ Bit (on/off)
○ Integer (Numeric value)
○ Timer
● Time Functions
○ List
○ Date & Time

Link To:
Link To
MI 4
Day of the Month setting

Variable information
Format
Hour (CT)
Hour (CT)
Day Of the week (CW)
Day of the month (CD)
Month (CM)
Year (CY)

19. The Month Variable.



VARIABLE 5: Month Variable

Variable Type

- ○ Bit (on/off)
- ○ Integer (Numeric value)
- ○ Timer
- ● Time Functions
- ○ List
- ○ Date & Time

Link To:

MI 6

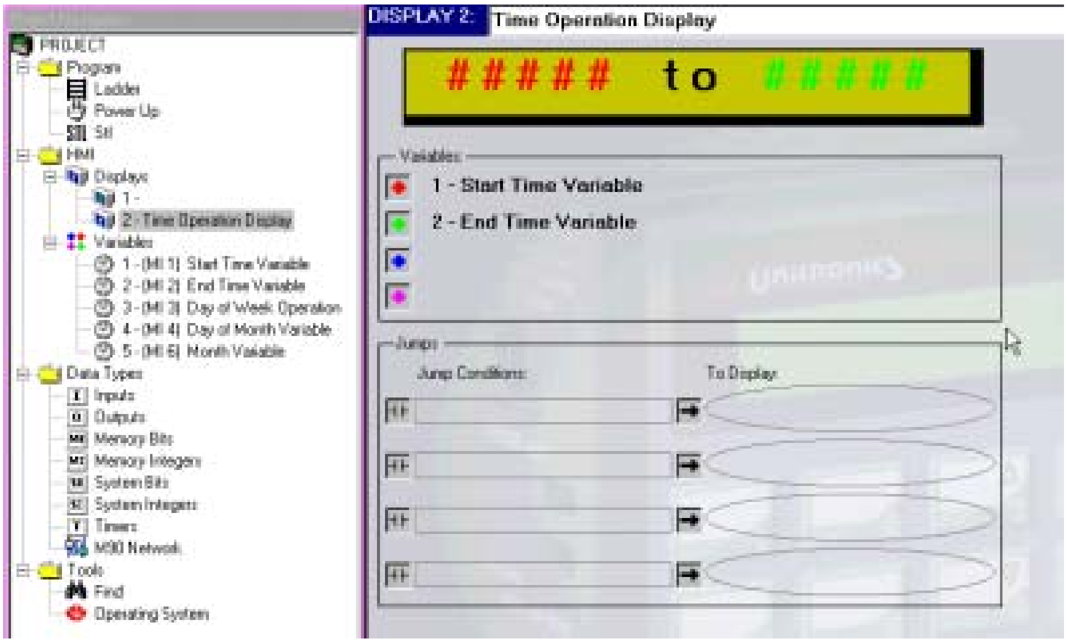
Month setting

Variable information

Format: Month (CM)

□ Keypad Entry

20. Create the Displays for the Variables. Below is an example for viewing the time range in hours.



- PROJECT
  - Program
    - Ladder
    - Power Up
    - Stl
  - HMI
    - Displays
      - 1 -
      - 2 - Time Operation Display
    - Variables
      - 1 - (MI 1) Start Time Variable
      - 2 - (MI 2) End Time Variable
      - 3 - (MI 3) Day of Week Operation
      - 4 - (MI 4) Day of Month Variable
      - 5 - (MI 6) Month Variable
  - Data Types
    - Inputs
    - Outputs
    - Memory Bits
    - Memory Integers
    - System Bits
    - System Integers
    - Timers
    - M90 Network
  - Tools
    - Find
    - Operating System

DISPLAY 2: Time Operation Display

to

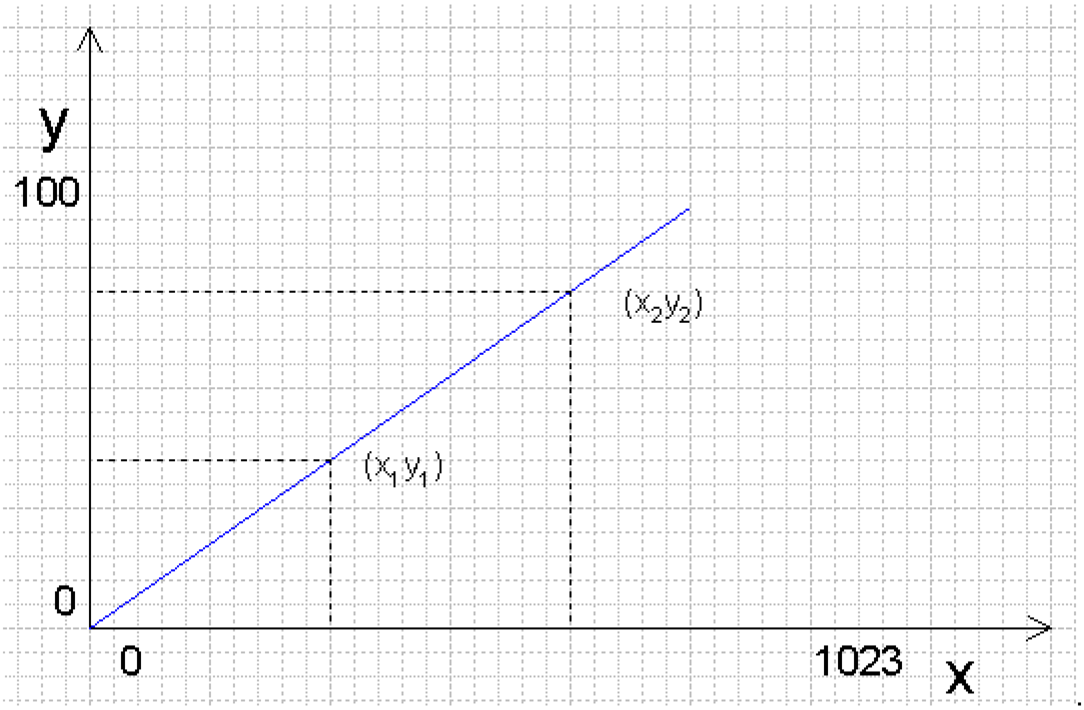
Variables:

- 1 - Start Time Variable
- 2 - End Time Variable

Functions without Elements

Linearization

Linearization can be used to convert analog values from I/Os into decimal or other integer values. An analog value from a temperature probe, for example can be converted to degrees Celsius and displayed on the controller's display screen.



Linearize values for Display

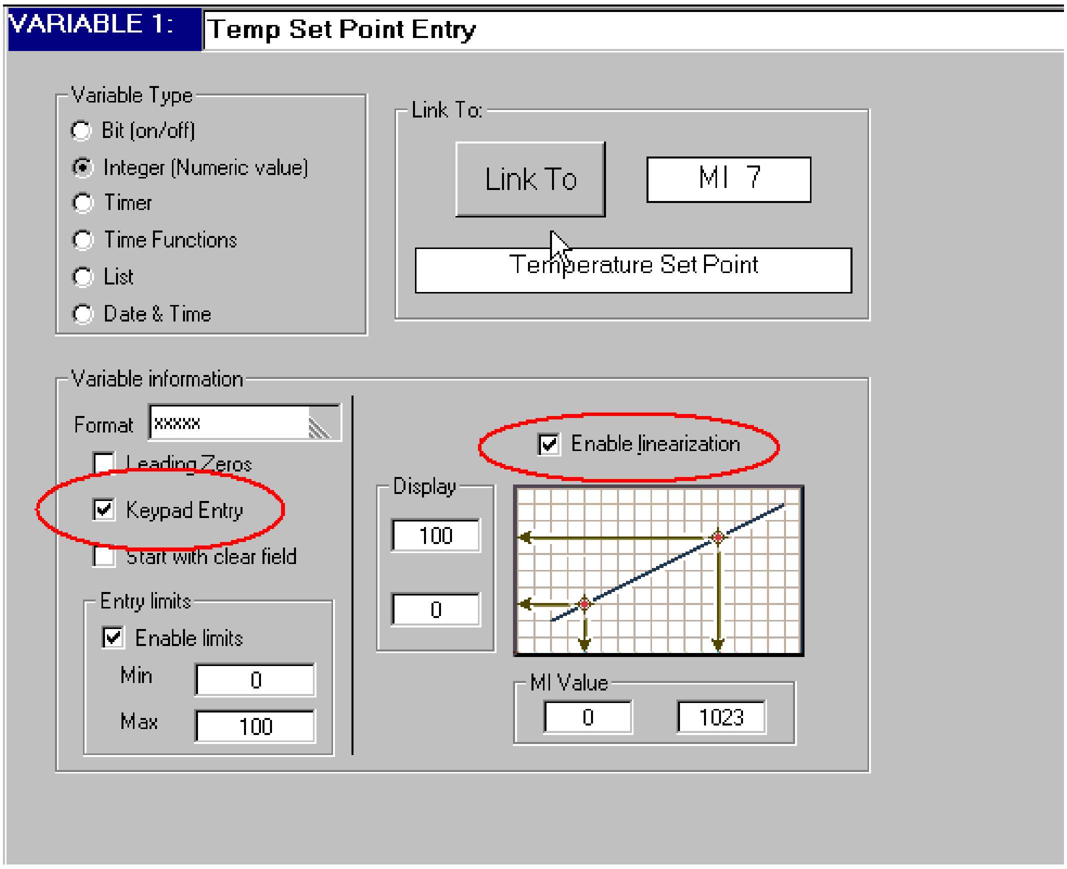
Note that the linearized value created in this way may be displayed-- **but** the value **cannot** be used anywhere else within the project for further calculations or operations.

You can enter an Analog value, such as temperature, via the M90 keypad, then convert that value into a Digital value for comparison with a digital value from a temperature probe by selecting **Enable Linearization** in the linked Variable.

This conversion process is Reverse Linearization.

To enable Analog to Digital conversion:

1. Create a Display for entering the analog value.
2. Create an Integer Variable.
3. Select **keypad entry** and **enable linearization**.
4. Enter the linearization values for the x and y axes.



VARIABLE 1: Temp Set Point Entry

Variable Type
○ Bit (on/off)
● Integer (Numeric value)
○ Timer
○ Time Functions
○ List
○ Date & Time

Link To:
Link To: MI 7
Temperature Set Point

Variable information
Format: xxxxx
□ Leading Zeros
☑ Keypad Entry
□ Start with clear field

Entry limits
☑ Enable limits
Min: 0
Max: 100

☑ Enable linearization

Display: 100, 0

MI Value: 0, 1023

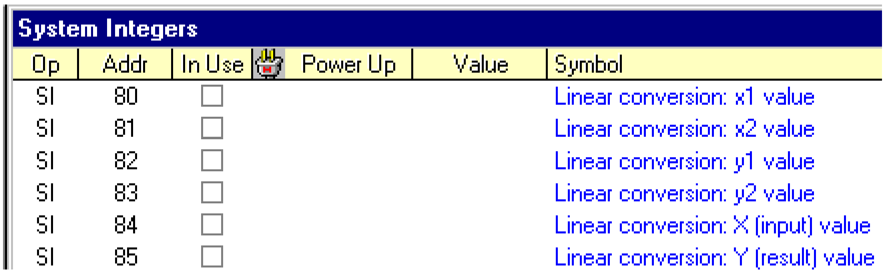
According to the above example:

- A temperature entry of 100^0 C will be converted to 1023 Digital value.
- A temperature entry of 50^0 C will be converted to 512 Digital value.

Linearize values in the Ladder

You can also linearize values in your Ladder and display them on the M90's LCD.

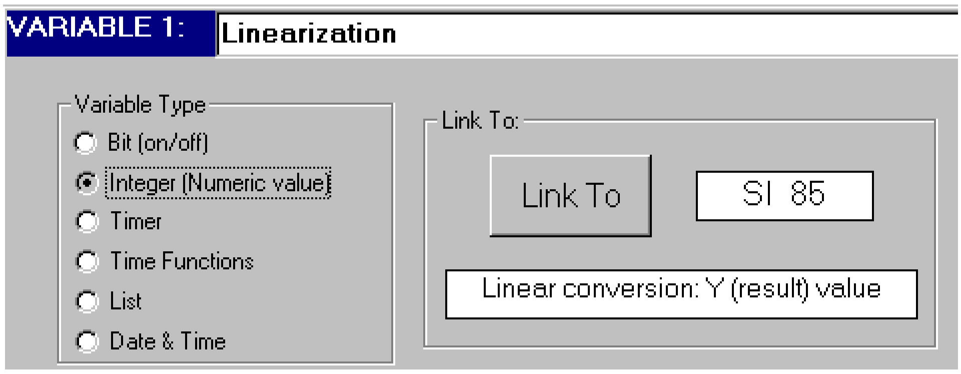
1. In your Ladder project, use SI 80 - 85 to set the (x,y) variable ranges. Use SB 80 to activate the **Linearization** function.



System Integers

Op	Addr	In Use	Power Up	Value	Symbol
SI	80	□			Linear conversion: x1 value
SI	81	□			Linear conversion: x2 value
SI	82	□			Linear conversion: y1 value
SI	83	□			Linear conversion: y2 value
SI	84	□			Linear conversion: X (input) value
SI	85	□			Linear conversion: Y (result) value

The linearization values created here can be displayed by linking SI 85 to a Display;the value **can** be used elsewhere within the project for further calculations or operations.

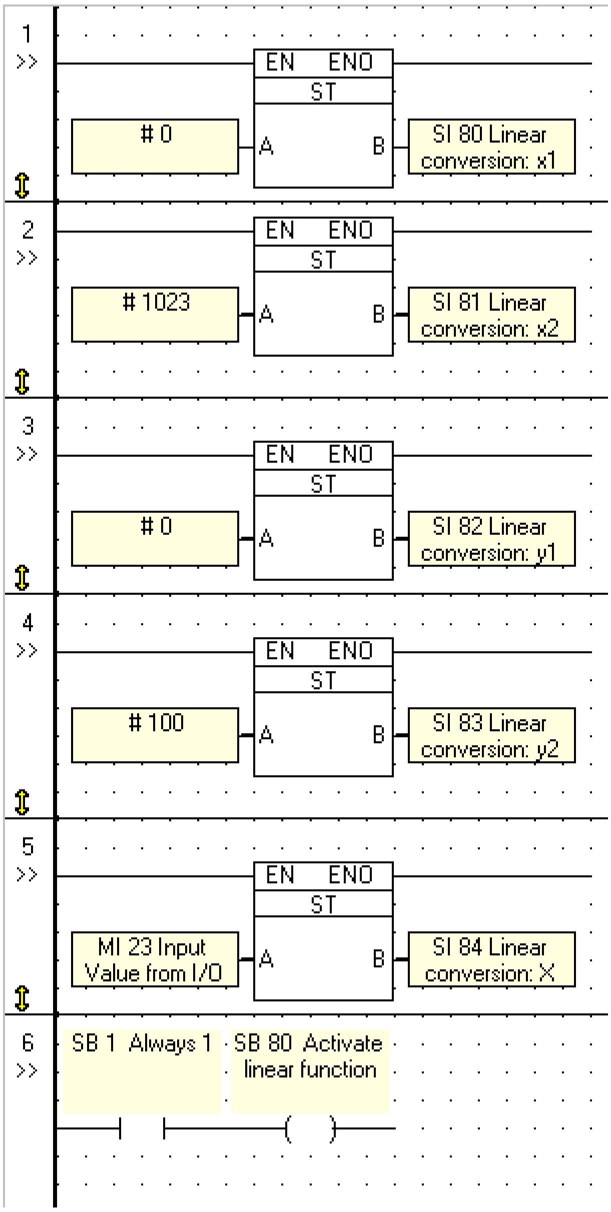


VARIABLE 1: Linearization

Variable Type
○ Bit (on/off)
● Integer (Numeric value)
○ Timer
○ Time Functions
○ List
○ Date & Time

Link To:
Link To
SI 85
Linear conversion: Y (result) value

Example: write the variable ranges into SI 80 - 83, then writing an analog input into SI 84:


1
EN ENO
ST
# 0
A
B
SI 80 Linear conversion: x1
2
EN ENO
ST
# 1023
A
B
SI 81 Linear conversion: x2
3
EN ENO
ST
# 0
A
B
SI 82 Linear conversion: y1
4
EN ENO
ST
# 100
A
B
SI 83 Linear conversion: y2
5
EN ENO
ST
MI 23 Input Value from I/O
A
B
SI 84 Linear conversion: X
6
SB 1 Always 1
SB 80 Activate linear function

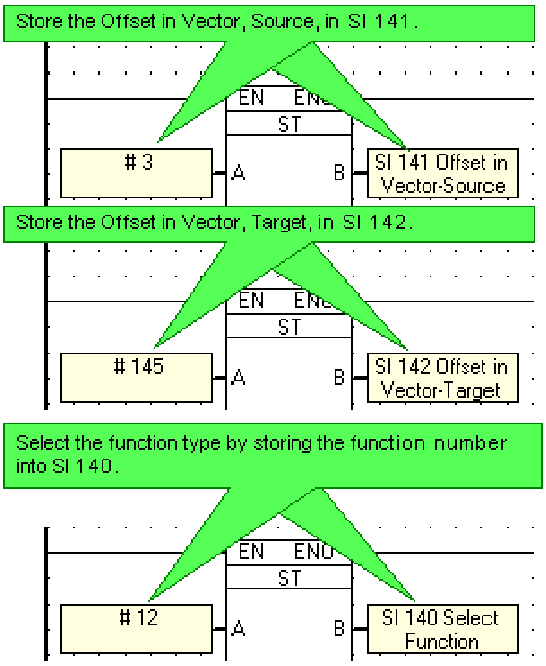
Load Indirect

Load Indirect allows you to take a value contained in a **source** operand and load that value into a **target** operand using indirect addressing. Note that since there is no Ladder element for this function; you perform it by storing values into:

- SI 141 to determine the data source,
- SI 142 to determine the load target,
- SI 140 to select the type of function. Storing the function number calls the function. In your application, call the function **after** you have entered all of the other parameters.

To use Load Indirect:

•


Store the Offset in Vector, Source, in SI 141.

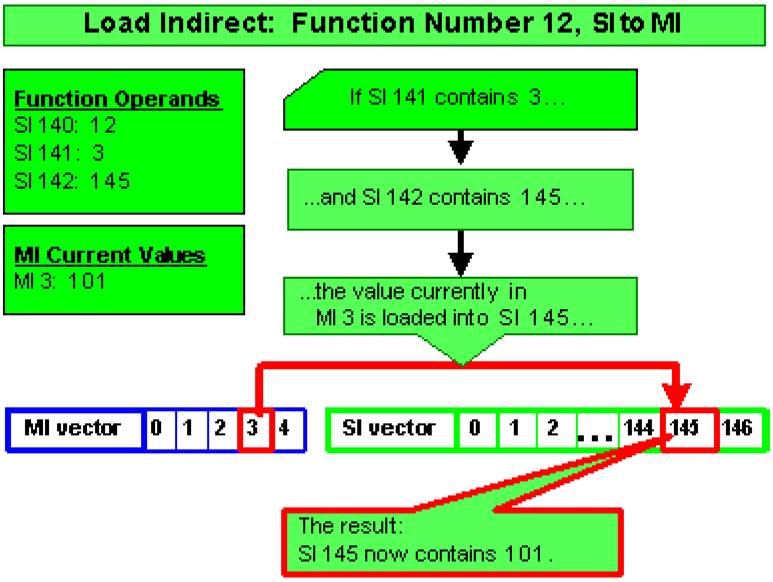
Store the Offset in Vector, Target, in SI 142.

Select the function type by storing the function number into SI 140.

Function Number (SI 140)	Offset in Vector, Source (SI 141)	Offset in Vector, Target (SI 142)
10	MI	MI
11	SI	MI
12	MI	S
13	SI	S

Note that when you run Test (Debug) Mode, the current value in SI 140 will **not** be displayed.

•



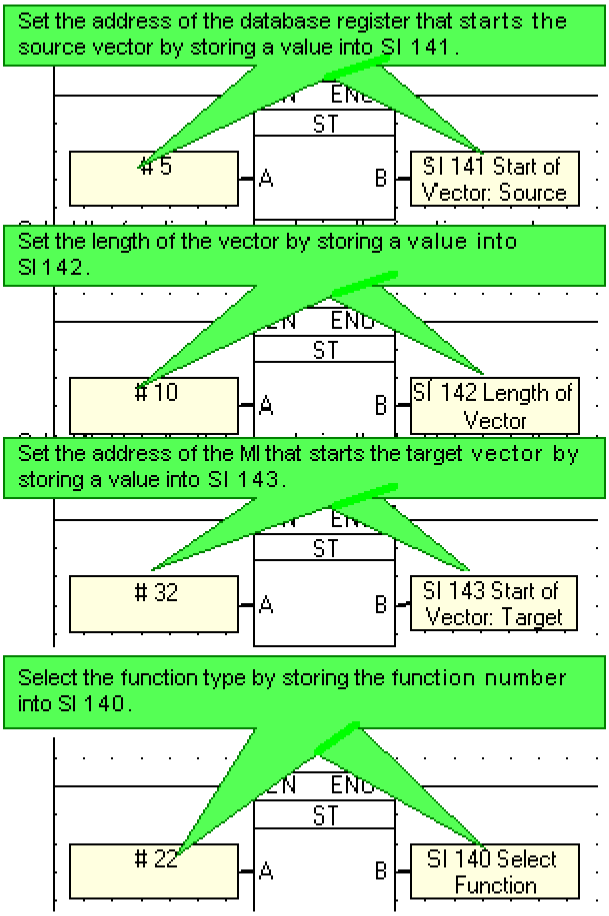
Copy Vector

Vector Copy enables you to set a range of operands, copy the values of each operand within that range **(source)**, then write those values into a corresponding range of operands of the same length **(target)**. You can copy from/to a vector of MI registers or Database registers by selecting the appropriate function.

Note that since there is no Ladder element for this function; you perform it by storing values into:

- SI 141 to determine the source vector,
- SI 142 to determine the length of the vector,
- SI 143 to determine the target vector,
- SI 140 to select the type of function. Storing the function number calls the function. In your application, call the function **after** you have entered all of the other parameters.

To use Copy Vector:



Set the address of the database register that starts the source vector by storing a value into SI 141.

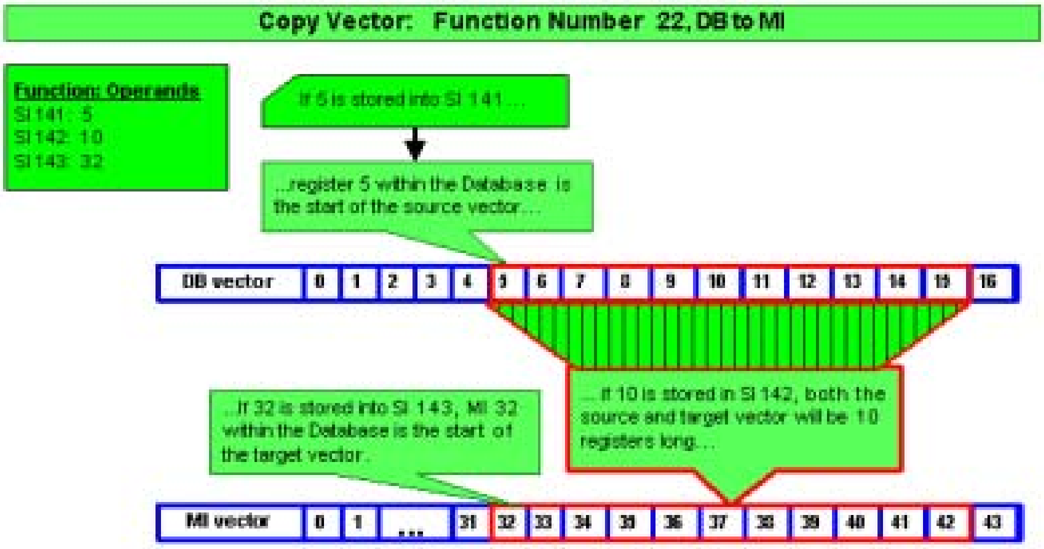
Set the length of the vector by storing a value into SI 142.

Set the address of the MI that starts the target vector by storing a value into SI 143.

Select the function type by storing the function number into SI 140.

Function Number (SI 140)	Source Vector, (SI 141)	Target Vector, (SI 142)
20	MI	MI
21	MI	DB
22	DB	MI
23	DB	DB

Note that when you run Test (Debug) Mode, the current value in SI 140 will **not** be displayed.



Copy Vector: Function Number 22, DB to MI

Function: Operands
SI 141: 5
SI 142: 10
SI 143: 32

If 5 is stored into SI 141...

...register 5 within the Database is the start of the source vector...

...If 32 is stored into SI 143, MI 32 within the Database is the start of the target vector.

...If 10 is stored in SI 142, both the source and target vector will be 10 registers long...

Fill Vector

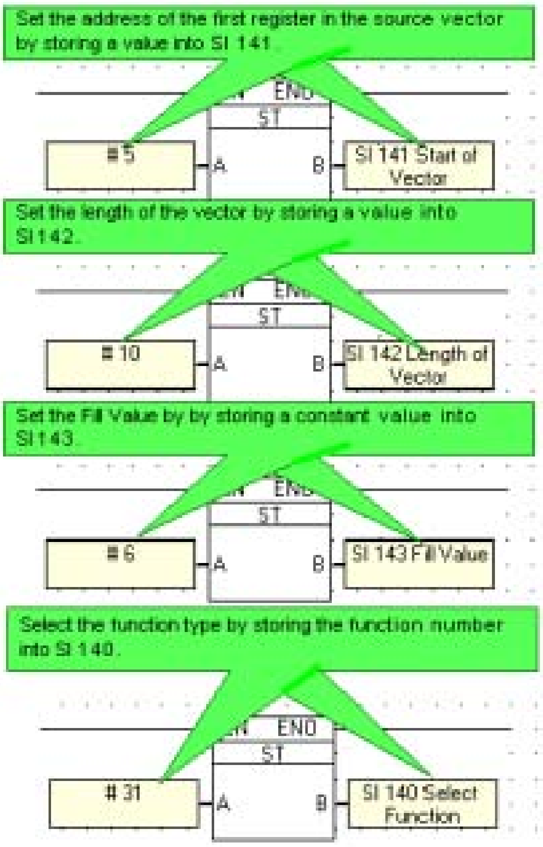
Fill Direct enables you to set a range of registers. The function copies a value from a desired operand or constant value **(source)**, then writes that value into every operand within the range **(target vector)**.

You can fill a vector of MI registers or Database registers by selecting the appropriate function.

Note that since there is no Ladder element for this function; you perform it by storing values into:

- SI 141 to determine the start of the target vector,
- SI 142 to determine the length of the target vector,
- SI 143 to select the Fill Value; the register whose value will be written into each register within the target vector,
- SI 140 to select the type of function. Storing the function number calls the function. In your application, call the function **after** you have entered all of the other parameters.

To use Fill Vector:



Set the address of the first register in the source vector by storing a value into SI 141.

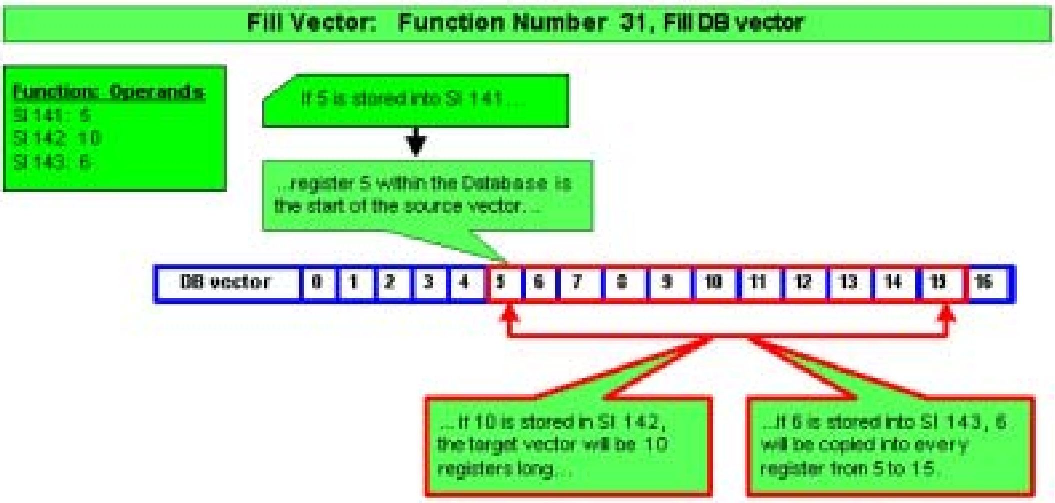
Set the length of the vector by storing a value into SI 142.

Set the Fill Value by by storing a constant value into SI 143.

Select the function type by storing the function number into SI 140.

Function Number (SI 140)	Description
30	Fill MI Vector
31	Fill DB Vector

Note that when you run Test (Debug) Mode, the current value in SI 140 will **not** be displayed.



Fill Vector: Function Number 31, Fill DB vector

Functions: Operands
SI 141: 5
SI 142: 10
SI 143: 6

If 5 is stored into SI 141...

...register 5 within the Database is the start of the source vector...

... If 10 is stored in SI 142, the target vector will be 10 registers long...

...If 6 is stored into SI 143, 6 will be copied into every register from 5 to 15.

Find Mean, Maximum, and Minimum Values

This function enables you to take a vector of registers and find the:

- Mean of all the values in the vector,
- Minimum value in the vector,
- Maximum value in the vector.

You can base the function on a vector of MI registers or Database registers by selecting the appropriate function.

Note that since there is no Ladder element for this function; you perform it by storing values into:

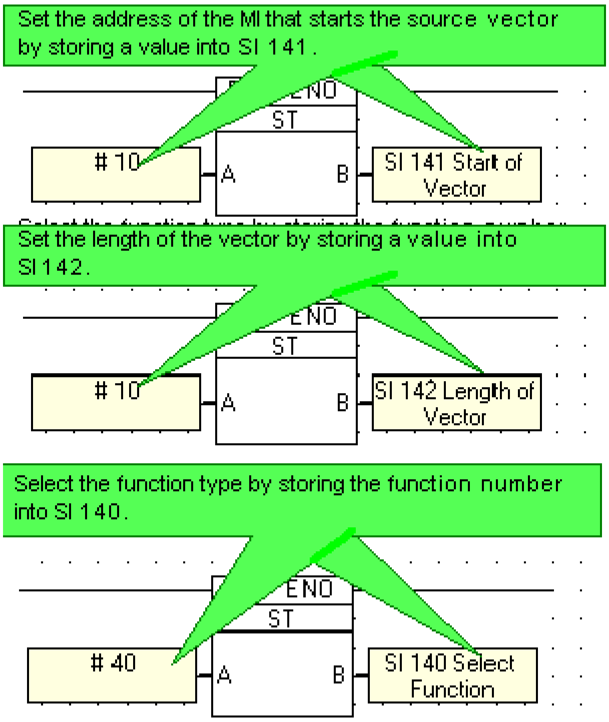
- SI 141 to determine the start of the vector,
- SI 142 to determine the length of the vector,
- SI 140 to select the type of function. Storing the function number calls the function. In your application, call the function **after** you have entered all of the other parameters.

The results will be placed in:

- SI 143: Mean
- SI 144: Minimum
- SI 145: Maximum

Note that if a remainder value results from the division operation used to calculate the Mean, that remainder value will be place in SI 4, Divide Remainder.

To use this function:



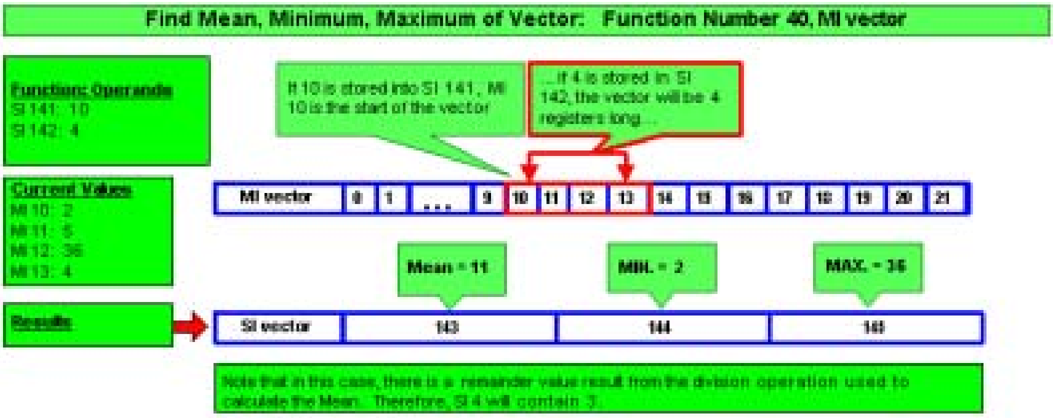
Set the address of the MI that starts the source vector by storing a value into SI 141 .

Set the length of the vector by storing a value into SI 142.

Select the function type by storing the function number into SI 140.

Function Number (SI 140)	Description
40	Find Mean, Minimum, Maximum in MI vector
41	Find Mean, Minimum, Maximum in DB vector

Note that when you run Test (Debug) Mode, the current value in SI 140 will **not** be displayed.



If 10 is stored into SI 141 , MI 10 is the start of the vector

...if 4 is stored in SI 142, the vector will be 4 registers long...

Note that in this case, there is a remainder value result from the division operation used to calculate the Mean. Therefore, SI 4 will contain 3.

A*B/C

This function enables you to :

- Multiply 2 operand values,
- Divide the product by a third operand.

The product of the multiplication operation is temporarily stored in a long integer to avoid overflow problems.

Since there is no Ladder element for this function; you perform it by storing values into:

- SI 141 to provide Operand A (multiplicand),
- SI 142 to provide Operand B (multiplicand),
- SI 143 to provide Operand C (divisor),

Store 100 into SI 140 to call the function. In your application, call the function **after** you have entered all of the other parameters.

The results will be placed in:

- SI 144,
- SI 4: Divide Remainder.

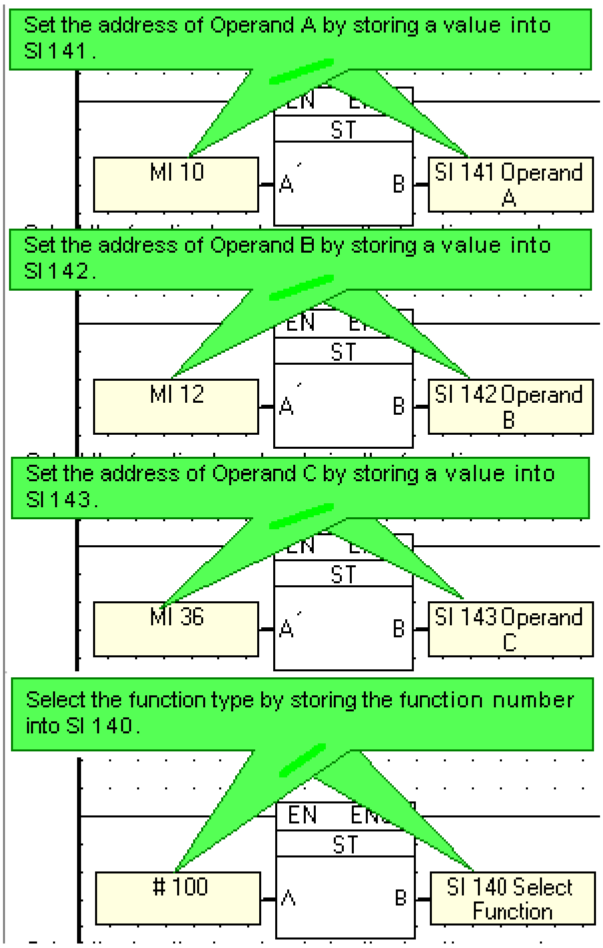
If the result is out of the integer range:

- SB 141 will turn ON.

If the value contained in Operand C (divisor) is 0:

- SB 4: Divide by 0, will turn ON.

To use this function:



Set the address of Operand A by storing a value into SI 141.

Set the address of Operand B by storing a value into SI 142.

Set the address of Operand C by storing a value into SI 143.

Select the function type by storing the function number into SI 140.

Function Number (SI 140)	Description
100	Multiply A x B, Divide by C

Note that when you run Test (Debug) Mode, the current value in SI 140 will **not** be displayed.

Square Root

This function enables you to find the square root of a number.

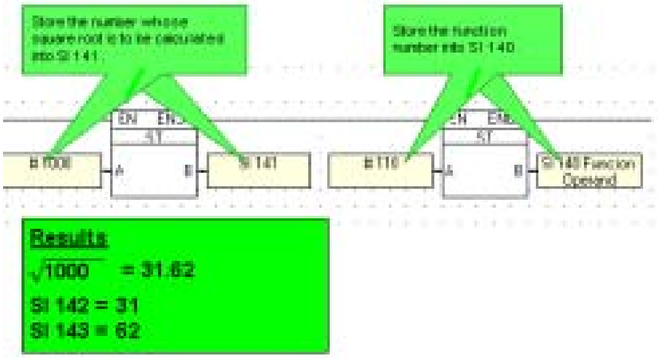
Since there is no Ladder element for this function; you perform it by storing the number whose square root is to be calculated into SI 141.

Store 110 into SI 140 to call the function. In your application, call the function **after** you have entered all of the other parameters.

The results will be placed in:

- SI 142. This contains the whole number result.
- SI 143. If the result is not a whole number, this contains up to 2 digits to the left of the decimal point.

To use this function:



Store the number whose square root is to be calculated into SI 141

Store the function number into SI 140

Results

$\sqrt{1000} = 31.62$

SI 142 = 31

SI 143 = 62

Function Number (SI 140)	Description
110	Calculate square root

Note that when you run Test (Debug) Mode, the current value in SI 140 will **not** be displayed.

Store Timer's Preset/Current Value

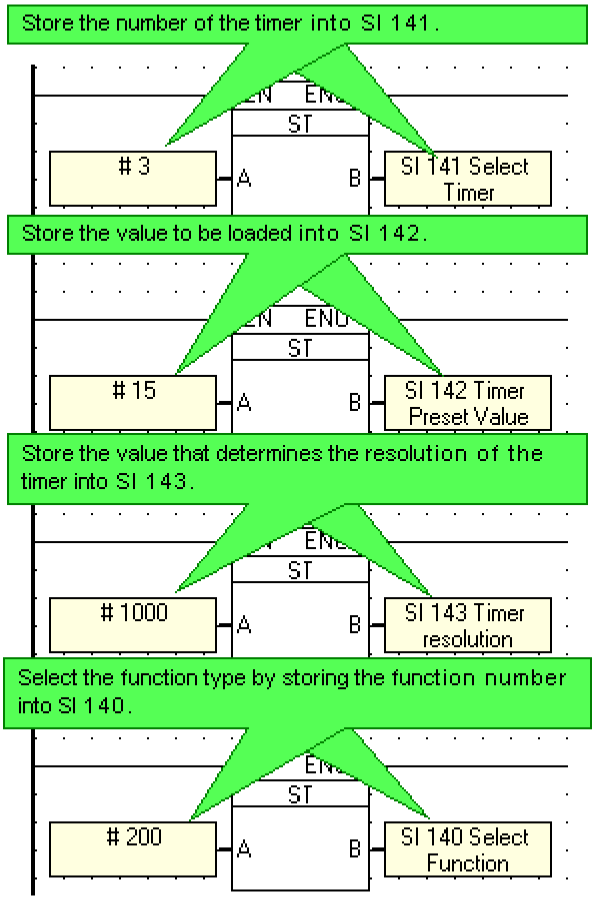
This function allows you to take a value and store it into a timer to change the preset or current timer value. Since there is no Ladder element for this function; you perform it by storing values into :

- SI 141 to select the timer; 0-63,
- SI 142 to determine the timer value,
- SI 143 to select the timer's resolution (timer units, or 'ticks'),
- SI 140 to select the type of function. Storing the function number calls the function. In your application, call the function **after** you have entered all of the other parameters.

Take into account that:

- Since you cannot change the resolution of a timer when the application is running, SI 143 is not used in a Store Timer's Current Value function.
- A timer's current value can be changed at any time, including when the timer is active. The new value can be either greater or smaller than the previous value; storing 0 into a timer's current value stops it immediately.
- A change of Timer Preset value without changing the resolution will take effect when the timer restarts.
- Changing the resolution of the timer's preset value does not affect the current resolution; it is therefore recommended that the resolution not be changed while the timer is active.

To use this function:



Store the number of the timer into SI 141.

Store the value to be loaded into SI 142.

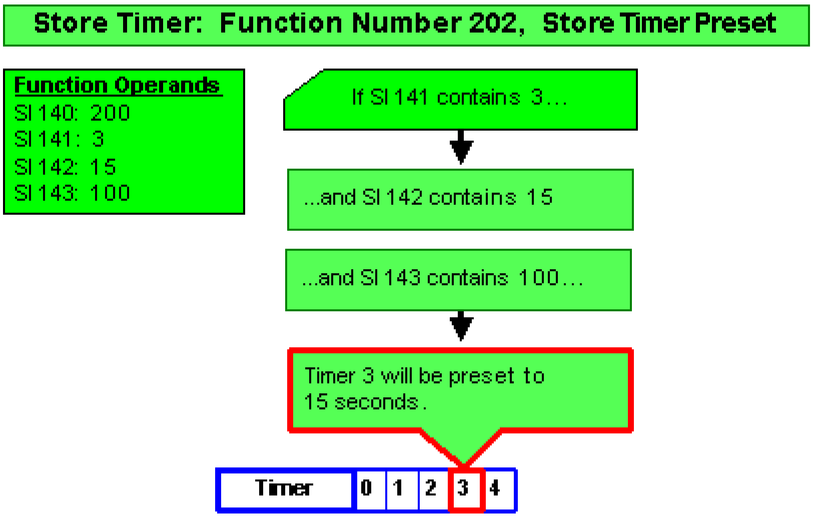
Store the value that determines the resolution of the timer into SI 143.

Select the function type by storing the function number into SI 140.

Function Number (SI 140)	Description
200	Store Timer Preset
201	Store Timer Current

Note that when you run Test (Debug) Mode, the current value in SI 140 will **not** be displayed.

Timer Resolution (stored into SI 143)	
Value	**Resolution**
0	Maintain Timer Resolution
1	10mS (0.01S)
10	100mS (001S)
100	1000mS (1S)
1000	10000mS (10S)



Store Timer: Function Number 202, Store Timer Preset

Function Operands
SI 140: 200
SI 141: 3
SI 142: 15
SI 143: 100

If SI 141 contains 3...

...and SI 142 contains 15

...and SI 143 contains 100...

Timer 3 will be preset to 15 seconds.

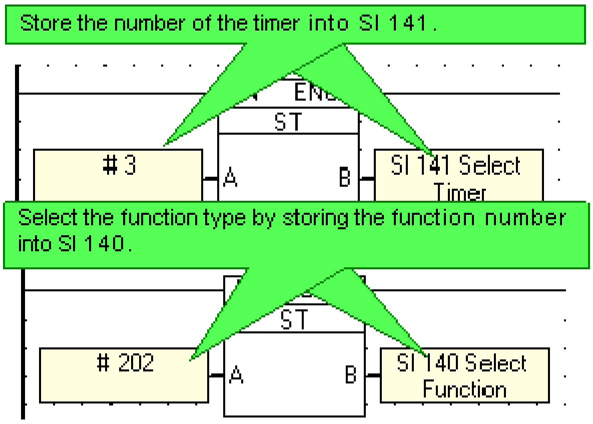
Note that the timer value is 14 bits.

Load Timer Preset/Current Value

This function allows you to take a preset or current timer value and load it into another operand. Note that since there is no Ladder element for this function; you perform it by storing values into:

- SI 141 to select the timer; 0-63,
- SI 140 to select the type of function. Storing the function number calls the function. In your application, call the function **after** you have entered all of the other parameters.

To use this function:

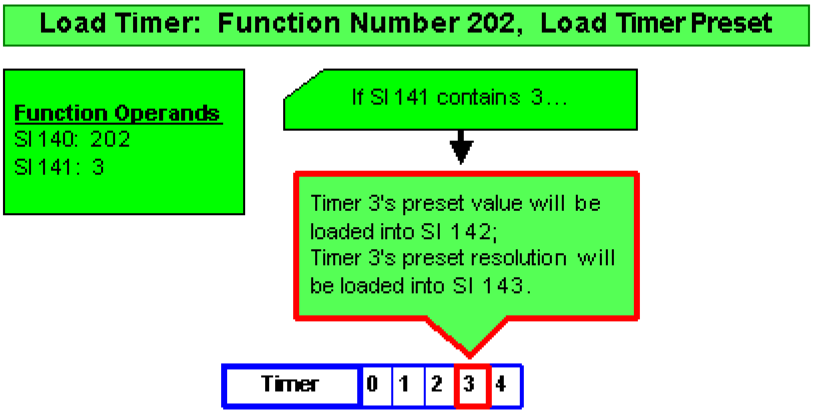

Store the number of the timer into SI 141.

Select the function type by storing the function number into SI 140.

Function Number (SI 140)	Description
202	Load Timer Preset
203	Load Timer Current

Note that when you run Test (Debug) Mode, the current value in SI 140 will **not** be displayed.

Timer Resolution (stored into SI 143)	
Value	**Resolution**
1	10mS (0.01S)
10	100mS (001S)
100	1000mS (1S)
1000	10000mS (10S)


Load Timer: Function Number 202, Load Timer Preset

Function Operands
SI 140: 202
SI 141: 3

If SI 141 contains 3...

Timer 3's preset value will be loaded into SI 142;
Timer 3's preset resolution will be loaded into SI 143.

Communication Utilities

Use this utility to enable your controller to receive data from external devices, such as bar-code readers, via an RS232 port. Since there is no Ladder element for this function; you perform it by storing values into SIs.

Note that the communication settings stored into theseSIs only take effect at power-up.

SI	Parameter	Value to Store	Notes
141	STX (Start of Text)	0-255(ASCII) -1: No Start of Text (not recommended)	The STX parameter indicates where the data block begins. ● Note that the ASCII character '/' (backslash) cannot be used to indicate the start of the

			data block.
142	ETX (End of Text)	0-255(ASCII) -1: ETX marked by Length -2: ETX marked by 'Silence'	The ETX parameter indicates where the data block ends. When the ETX is registered by the function, SB 60 turns ON. • If you use an ASCII character (0-255), note that if this character occurs after the Length parameter defined in SI 143, SB 60 turns ON. • Selecting -1 causes the function to use the length of a data block alone to determine its end. • Selecting -2 causes the function to use the duration of silent time following the STX to determine the end of a data block.
143	ETX Length or Silent	Length: up to 128 Silent: up to 24000	• This defines both the length of text, or silence, that signal the end of text. • Note that the duration of a silent 'counter' unit is approximately 2.509 mS. The 'silent' value should be lower than the M90 TimeOut value. • When defined as length, SI 143 cannot exceed SI 144.
144	Maximum Length	Up to 128	• This is the maximum legal length for received text. • When the maximum length is exceeded, the Receive Buffer is automatically cleared, and SB 60 is turned OFF, enabling new data to be received. • This can be used to detect buffer overflow.
145	Start Address: Receive Buffer	MI Address	This MI contains the start address for the vector of registers that serves as the Receive Buffer.
60	Number of Bytes currently in Receive Buffer	Read only	SI 60 indicates how many bytes of data are currently in the Receive Buffer.
61	Number of Bytes in Receive Buffer when SB 60=1	Read only	SI 61 indicates how many bytes of data are in the Receive Buffer when SB 60 turns ON.
146	Copy Data: Format	0: copy each received byte 1: copy in groups of 4 received bytes.	• 0 causes each separate byte to be copied to a separate register including STX and ETX. • 1 causes every 4 bytes to be copied to a single register, without the STX and ETX. This is used when the received data is in numeric format. For example 12345 would be copied to 2 consecutive MIs. The first MI would contain 1234, the second would contain 5.
140	Start receiving	300	In your application, use this to call the function after you have entered all of the other parameters. Note that when you run Test (Debug)

SB	Description	Notes
		Mode, the current value in SI 140 will not be displayed.
SB	Description	Notes
60	Data Successfully Received	Read only. Turns ON when the ETX condition is registered by the system.
61	Copy Data in Receive Buffer to MI Vector	Write only. • Turning this SB ON causes the buffer contents to be copied to the MI vector defined in SI 145. The data will be copied according to the format defined in SI 146. • If SI 146 is set to 0, this SB can be set at any time. If SI 146 is set to 1, this SB can be set after SB 60 turns ON.
62	Clear Receive Buffer, Clear SI 60, Clear SI 61, Reset SB 60	• This SB must be turned ON to enable a new message, or data block, to be received. • Turn this SB ON to enable data to be received **before** the maximum length, defined in SI 144, is exceeded.

Note that if no data is received for a period exceeding the M90 TimeOut, you will lose the data in the buffer.

To see how to use the Communications Utility, check the sample application **Read Card - Display Number Value.U90**. This may be found by accessing Sample U90 Projects from the Help menu.

This application demonstrates how to read a magnetic card number using an "IDTECH" card reader, then display that number on the M90's screen. The card reader transmits the number in ASCII characters in this format:

< %?[CR];xxxxx?[CR] > where **xxxxx** is the card number.

The ASCII character used to mark the Start Of Text (STX) is < **;** > (semicolon). End Of Text (ETX) is marked with the character < **?** > .

Since the card number is 5 digits long, the card number is copied to 2 separate MIs. The MIs are linked to 2 variables that are shown on the M90's screen in 2 separate Displays.

The parameters must be written into their respective operands using one scan condition. For this purpose, it is recommended to use SB 2 Power-up bit, as shown in the sample application.

Interrupt

This function is time-based. You call an interrupt routine by storing 500 into SI 140. The interrupt function causes:

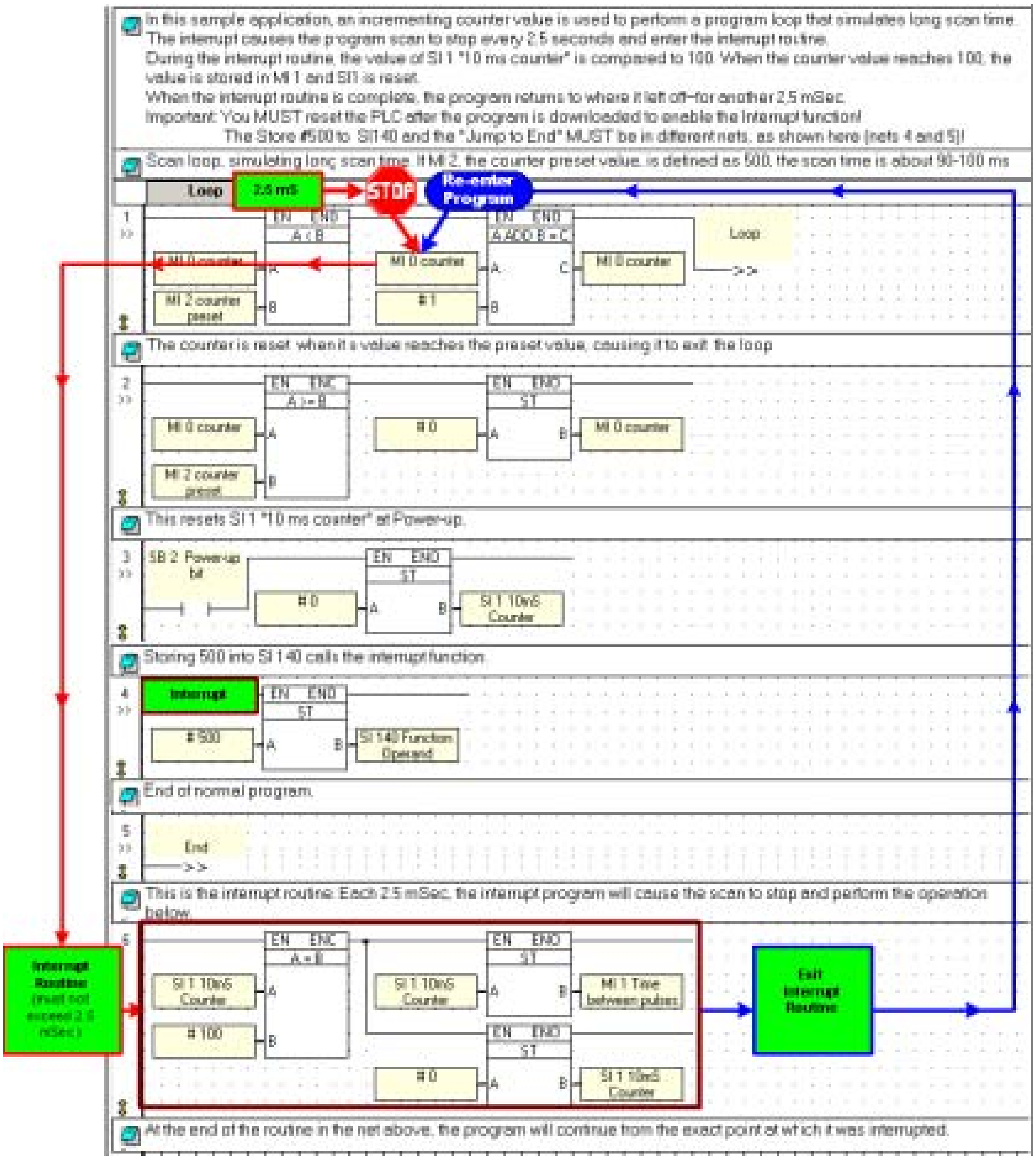
- The program scan to pause every 2.509 mSec. The interrupt causes the program to stop immediately without regard to the program scan, even if it occurs in the middle of a net.
- A jump to the net which follows the interrupt. The nets following the interrupt comprise the interrupt routine. Note that the interrupt routine should be as short as possible, and must not exceed approximately 0.5 mSec.
- When the interrupt routine is finished, the program continues from where it left off.

Note that the nets containing the Interrupt routine must be the last ones in the program. The format must be as shown in the example below:

- Store 500 into SI140 to call the function
- Jump to End
- The nets containing the actual interrupt routine.

Note that when you run Test (Debug) Mode, the current value in SI 140 will **not** be displayed.

Example



In this sample application, an incrementing counter value is used to perform a program loop that simulates long scan time.
The interrupt causes the program scan to stop every 2.5 seconds and enter the interrupt routine.
During the interrupt routine, the value of SI 1 "10 ms counter" is compared to 100. When the counter value reaches 100, the value is stored in MI 1 and SI1 is reset.
When the interrupt routine is complete, the program returns to where it left off--for another 2,5 mSec.
Important: You MUST reset the PLC after the program is downloaded to enable the Interrupt function!
The Store #500 to SI140 and the "Jump to End" MUST be in different nets, as shown here (nets 4 and 5)!

Scan loop, simulating long scan time. If MI 2, the counter preset value, is defined as 500, the scan time is about 90-100 ms

The counter is reset when it s value reaches the preset value, causing it to exit the loop

This resets SI 1 "10 ms counter" at Power-up.

Storing 500 into SI 140 calls the interrupt function

End of normal program.

This is the interrupt routine: Each 2.5 mSec, the interrupt program will cause the scan to stop and perform the operation below.

At the end of the routine in the net above, the program will continue from the exact point at which it was interrupted.

Convert MB to MI, MI to MB

An M90 register is built of 16 bits.

Using the MB to MI function, you can convert 16 bits or more into a integer value. Conversely, you can convert an integer value into 16 bits or more using the MI to MB function.

Note that if the converted values exceed 16 bits, the function will write the value to consecutive registers. Any values in those registers will be overwritten.

To apply the functions, use the following System Integers (SI) and System Bits (SB)

SI	Description	SB	
SI170	Address of MI containing integer value	SB170	MB to MI
SI171	Start address of MB array (vector)	SB171	MI to MB
SI172	Amount of MBs		

You can use this function, for example to send an SMS when there is a change in the status of the M90's inputs:

1. Represent the status of the M90's inputs using MBs.
2. Convert these MBs into an MI
3. Perform a XOR operation on the result.

When there is a change in input status, the XOR operation will return a value different than 0, which may then be used to trigger the sending of an SMS.

Examples

Example 1:

1. Store the value 7 into SI 170, 10 into SI 171 and 9 into SI 172.
2. Set SB 170 to ON.

The program will calculate the binary value of a 9 bit array which starts with MB 10. The resulting value will be placed into MI 7.

Example 2:

1. Store the value 7 into SI 170, 10 into SI 171 and 9 into SI 172.
2. Set SB 171 to ON

The program will calculate the binary value of the value contained in MI 7. The result will be scattered on a 9 bit array which starts with MB 10.

Copy MI to Output vector, Input vector to MI

Using this function, you can:

- Copy a vector of Inputs (**I**) to a register.
- Copy a register value to a vector of Outputs (**O**).

Note that an M90 register contains 16 bits. If the converted values exceed 16 bits, the function will write the value to consecutive registers. Any values in those registers will be overwritten. When a register value is copied to outputs, the function will store the register value in consecutive outputs.

Input to Register

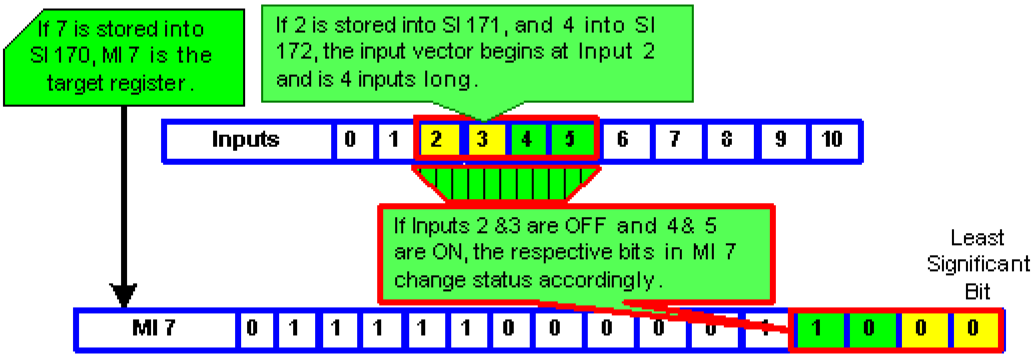
SI	Description	SB	Function
SI170	Address of MI containing integer value	SB172	I to MI
SI171	Start address of bit array (vector)	SB173	MI to O
SI172	Amount of bits		

Example: Input to MI, SB 172

1. Store the value 7 into SI 170, 2 into SI 171 and 4 into SI 172.
2. Set SB 172 to ON.

The program takes the status of I2 to I5, and changes the status of the respective bits in MI 7.

Bits in the target register that are outside of the defined range are not affected.


If 7 is stored into SI 170, MI 7 is the target register.
If 2 is stored into SI 171, and 4 into SI 172, the input vector begins at Input 2 and is 4 inputs long.
Inputs
0
1
2
3
4
5
6
7
8
9
10
If Inputs 2 &3 are OFF and 4& 5 are ON, the respective bits in MI 7 change status accordingly.
Least Significant Bit
MI 7
0
1
1
1
1
1
0
0
0
0
0
1
1
0
0
0

Example: MI to Output, SB 173

1. Store the value 7 into SI 170, 3 into SI 171 and 7into SI 172.
2. Set SB 173 to ON.

The program will take the binary value of the MI 7, and change the status of the respective outputs in the defined vector, O3 to O7.

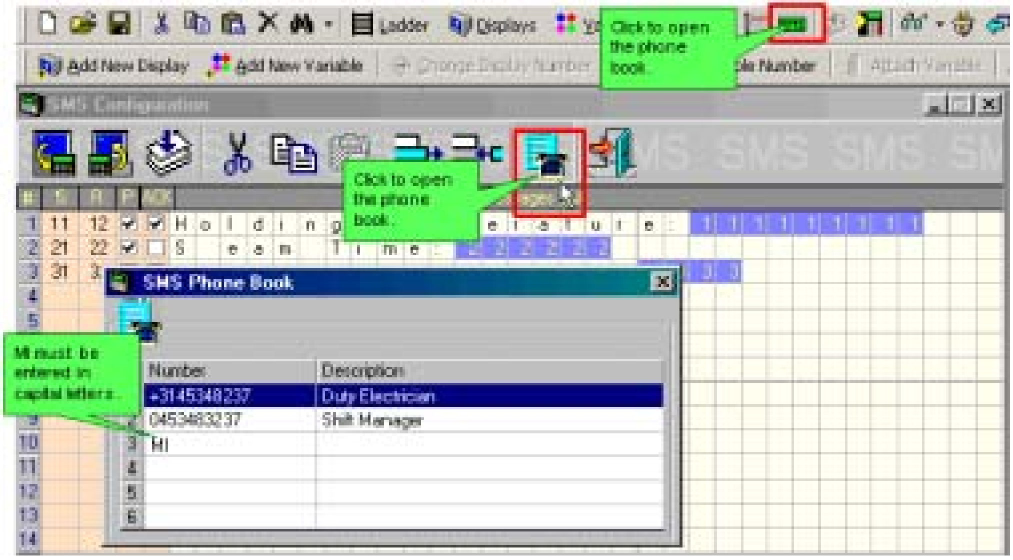
SMS Phone Number: via MI Pointer

Use this utility to use an MI vector as one of the phone numbers in the SMS phone book. This allows you to:

- Enable a number to be dialed via the M90's keypad.
- Exceed the 6 number limit of the SMS phone book.

Note that since there is no Ladder element for this function; you perform it by:

- Storing the start address of the MI vector needed to contain the phone number into SI 141,
- Entering the characters MI, in capital letters, in the *SMS phone book*,



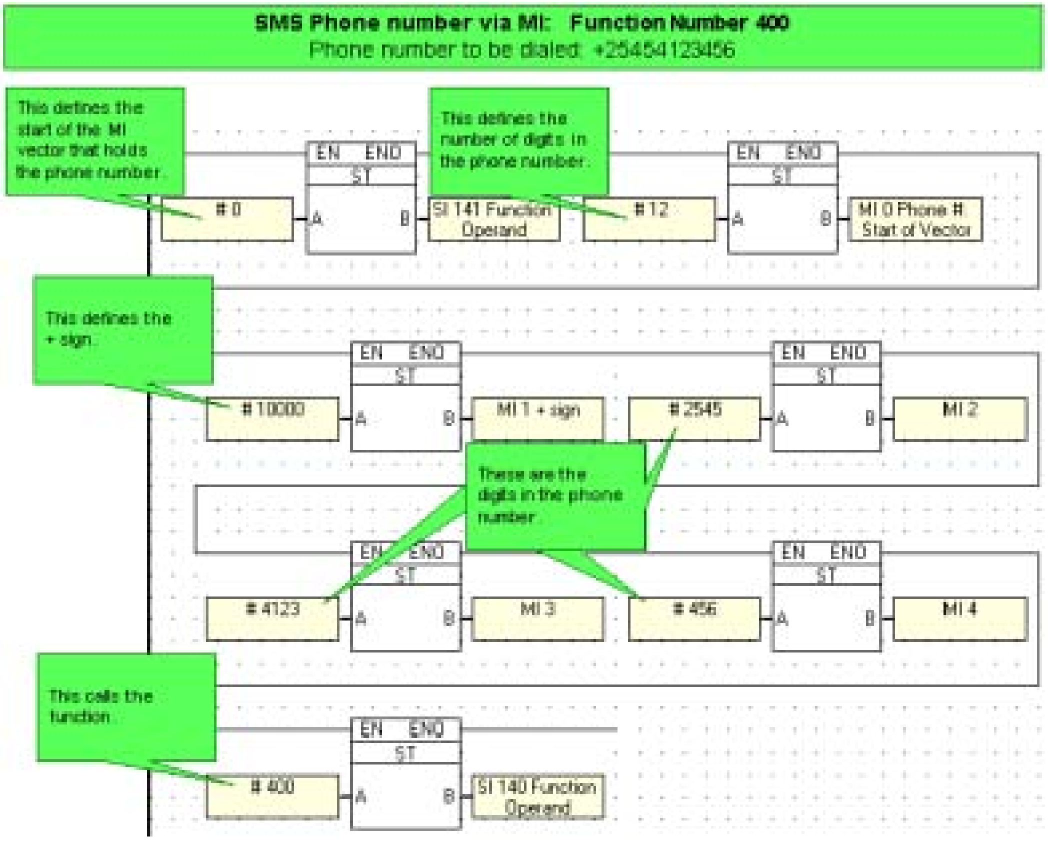
Click to open the phone book.

Click to open the phone book.

MI must be entered in capital letters.

	Number	Description
1	+3145348237	Duty Electrician
2	0453483237	Shift Manager
3	MI	
4		
5		
6		

- Using the index number of that line to call the number, which enables the number in the MI vector to be called,
- Storing 400 into SI 140 to select the function. Storing the function number calls the function. In your application, call the function **after** you have entered all of the other parameters. Note that when you run Test (Debug) Mode, the current value in SI 140 will **not** be displayed.


SMS Phone number via MI: Function Number 400
Phone number to be dialed: +25454123456
This defines the start of the MI vector that holds the phone number.
EN ENO
ST
# 0
A
B
SI 141 Function Operand
This defines the number of digits in the phone number.
# 12
MI 0 Phone #: Start of Vector
This defines the + sign.
# 10000
MI 1 + sign
# 2545
MI 2
These are the digits in the phone number.
# 4123
MI 3
# 456
MI 4
This calls the function.
# 400
SI 140 Function Operand

Shift Register

You can use the following SIs and SBs to perform Shift Left and Shift Right Functions.

SI	Symbol	Description
87	Shift Value	This register contains the number to be shifted.
88	Shift By	This register contains the number of bits to be shifted (Default is 1 bit).

SB	Symbol
87	Shift Left
88	Shift Right

Example : Shift Left

To shift the number 64 **left** by 1 bit:

1. Use a Store function to write the number 64 into SI 87.
2. Use a Store function to write the number 1 into SI 88.
3. Turn SB 87 ON.

Once the function is performed SI 87 will contain 128.

In binary:
Start value: 0000000001000000 = 64
After Shift Left : 0000000010000000 =128

Example : Shift Right

To shift the number 64 **right** by 1 bit:

1. Use a Store function to write the number 64 into SI 87.

2. Use a Store function to write the number 1 into SI 88.
3. Turn SB 88 ON.

Once the function is performed SI 87 will contain 32.

In binary:
Start value: 0000000001000000 = 64
After Shift Right: 0000000000100000 =32

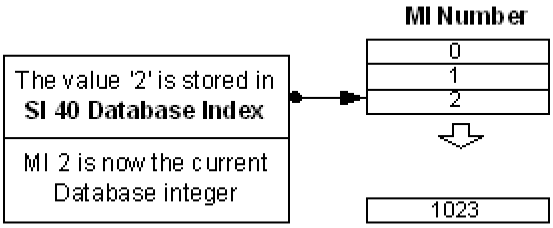
Access indirectly addressed registers: Using the Database

You can access and use integers 0 through 1023 within the M90 OPLC's memory as a database, via SI 40 and SI41.

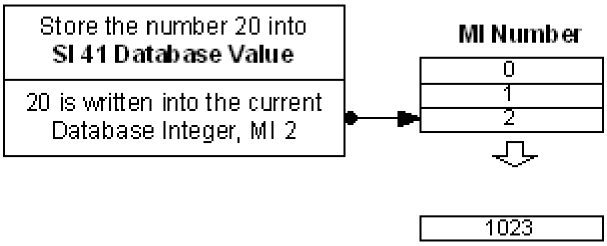
Note that when you run Test (Debug) Mode, the current value in SI 140 will **not** be displayed.

Writing Values

1. Use SI 40 Database Index to access a particular MI.
   For example, to access MI 2 you store the number 2 into SI 40.

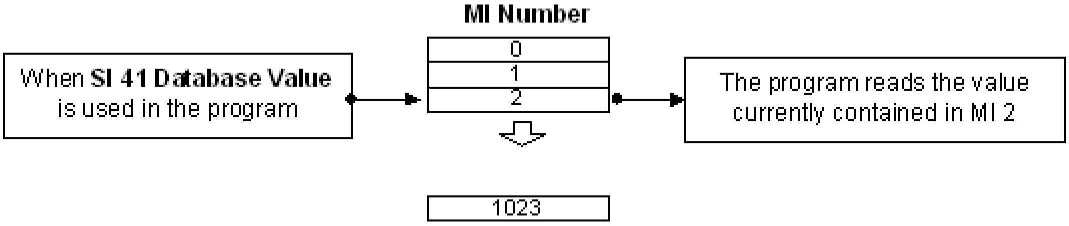


2. Use SI 41 Database Value to write a value into MI 2.
   For example, you can store a number value into SI 41.



Reading Values

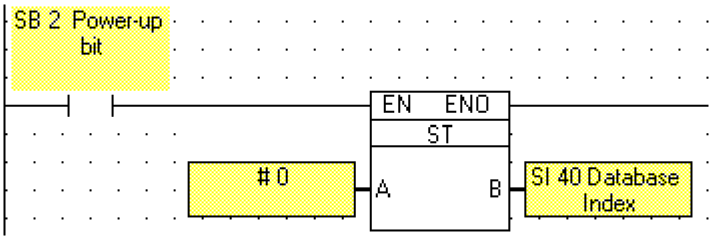
When you use SI 41 Database Value in your program, the program actually reads the MI that is referenced by SI 40 Database Index.


MI Number
When SI 41 Database Value
is used in the program
0
1
2
1023
The program reads the value
currently contained in MI 2

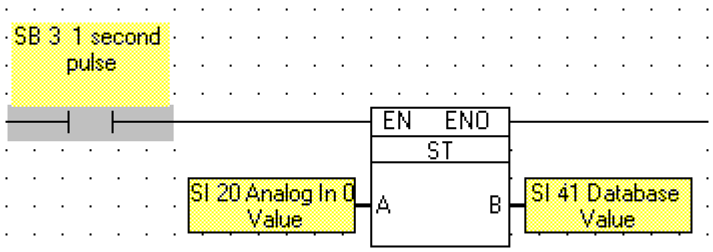
Examples

Example 1: Write

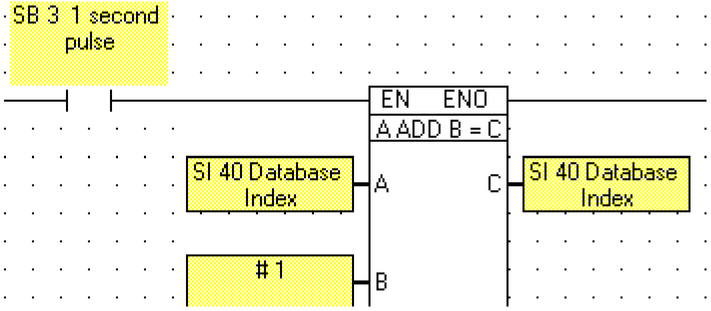
In the net below, 0 is stored in SI 40 when the M90 OPLC is powered up. This means that integer 0 is now the current 'database' integer.



In the net below, the analog value contained in SI 20 is stored in SI 41 every second. According to the net above, the current 'database' integer is 0. The analog value is therefore stored in integer 0.



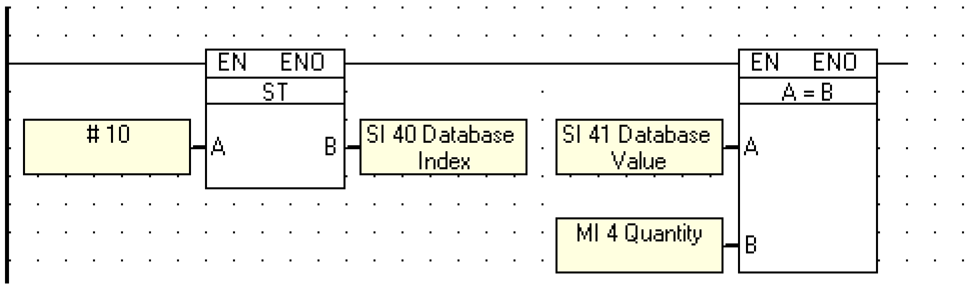
In the next net, the value in SI 40 is incremented by 1every second, changing the current database integer. This means that the first analog value will be stored in integer 0, the second analog value in integer 1, and so on.



Example 2: Read

In the first part of the net below, 10 is stored into SI 40. Integer 10 is the 'database' integer. In the second part of the net, the value in SI 41 is compared to the value in integer 4.

The value in SI 41 is the value actually in integer 10—the current database integer.



Counter

Building a Counter

If you want to use a counter in your application, you build it using:

- Math function
- Compare function
- Store function

Use a Positive / Negative Transition contact on the event operand to activate the counter.

Example:

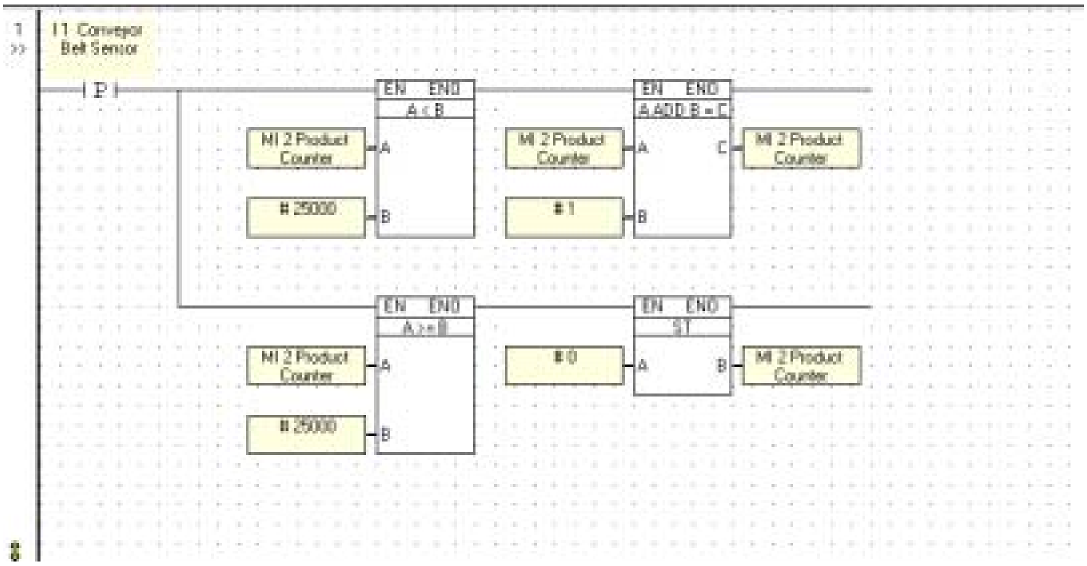
You want to count the gross number of a product traveling across a conveyor belt. There is a sensor (e.g. photocell, limit switch or proximity switch) at a specific point across the conveyor belt which senses the product as it passes.

The sensor is connected to an M90 Input. The Positive Transition from this Input will advance the counter by one.

When the counter value reaches the maximum defined value, the counter will reset to 0.

Counter Ladder example:

- Input 1 is the sensor
- MI 2 is the Counter
- The maximum defined value is 25000.



Keep in mind when building your counter that adding a number to 32767 will return a negative number.

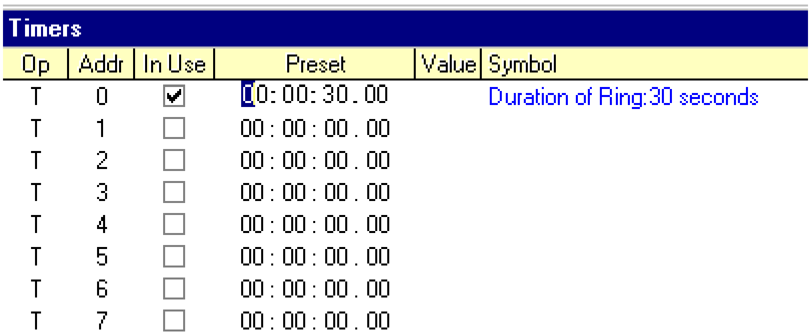
Counters are featured in several sample applications, such as the applications ' Time Interval- SI 1', 'Outputs-activate in sequence', and 'Logging analog values'. These applications may be found by selecting Sample U90 Projects from the Help Menu.

Timers

Timers (T)

U90 Ladder offers 64 On Delay Timers. Timers have a preset value, a current value, and a bit value. Timers always count **down** from the Preset Value.

Click on the Timers folder in the Program Navigation pane to display the complete list of Timers. Scroll down to view the complete list.



Timers

Op	Addr	In Use	Preset	Value	Symbol
T	0	☑	00: 00: 30. 00		Duration of Ring:30 seconds
T	1	□	00 : 00 : 00 . 00		
T	2	□	00 : 00 : 00 . 00		
T	3	□	00 : 00 : 00 . 00		
T	4	□	00 : 00 : 00 . 00		
T	5	□	00 : 00 : 00 . 00		
T	6	□	00 : 00 : 00 . 00		
T	7	□	00 : 00 : 00 . 00		

To place a Timer in your program, place a direct coil in a net, and select T.

Note that a Timer value can be displayed in a Display as a current or elapsed value.

Store Timer's Preset/Current Value

This function allows you to take a value and store it into a timer to change the preset or current timer value. Since there is no Ladder element for this function; you perform it by storing values into :

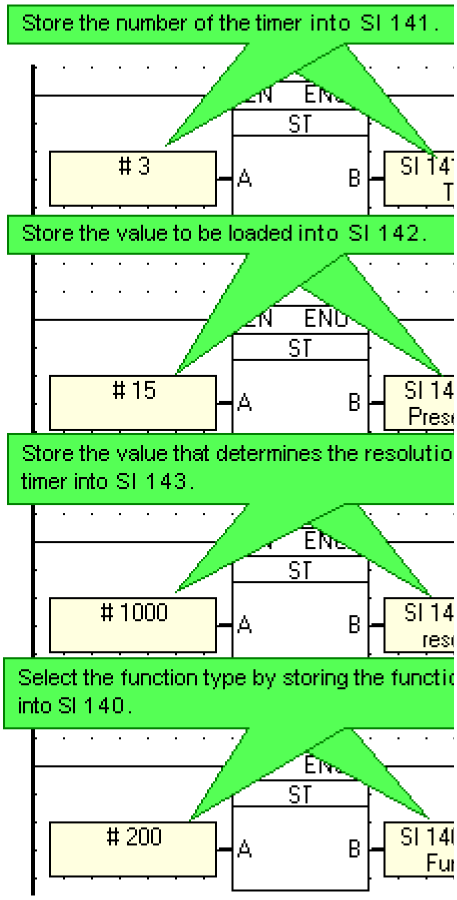
- SI 141 to select the timer; 0-63,
- SI 142 to determine the timer value,
- SI 143 to select the timer's resolution (timer units, or 'ticks'),
- SI 140 to select the type of function. Storing the function number calls the function. In your application, call the function **after** you have entered all of the other parameters.

Take into account that:

- Since you cannot change the resolution of a timer when the application is running, SI 143 is not used in a Store Timer's Current Value function.
- A timer's current value can be changed at any time, including when the timer is active. The new value can be either greater or smaller than the previous value; storing 0 into a timer's current value stops it immediately.
- A change of Timer Preset value without changing the resolution will take effect when the timer restarts.
- Changing the resolution of the timer's preset value does not affect the current resolution; it is therefore recommended that the resolution not be changed while the timer is active.

To use this function:

-



Store the number of the timer into SI 141.

Store the value to be loaded into SI 142.

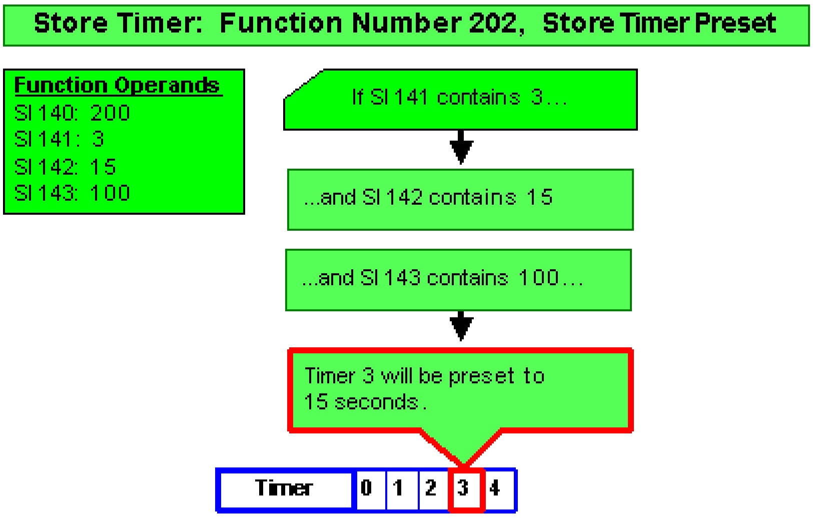
Store the value that determines the resolutio timer into SI 143.

Select the function type by storing the functio into SI 140.

Function Number (SI 140)	Description
200	Store Timer Preset
201	Store Timer Current

Note that when you run Test (Debug) Mode, the current value in SI 140 will **not** be displayed.

Timer Resolution (stored into SI 143)	
Value	**Resolution**
0	Maintain Timer Resolution
1	10mS (0.01S)
10	100mS (001S)
100	1000mS (1S)
1000	10000mS (10S)



Store Timer: Function Number 202, Store Timer Preset

Function Operands
SI 140: 200
SI 141: 3
SI 142: 15
SI 143: 100

If SI 141 contains 3...

...and SI 142 contains 15

...and SI 143 contains 100...

Timer 3 will be preset to 15 seconds.

Note that the timer value is 14 bits.

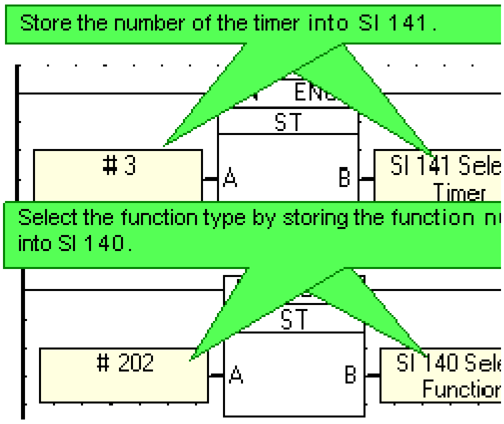
Load Timer Preset/Current Value

This function allows you to take a preset or current timer value and load it into another operand. Note that since there is no Ladder element for this function; you perform it by storing values into:

- SI 141 to select the timer; 0-63,

- SI 140 to select the type of function. Storing the function number calls the function. In your application, call the function **after** you have entered all of the other parameters.

To use this function:



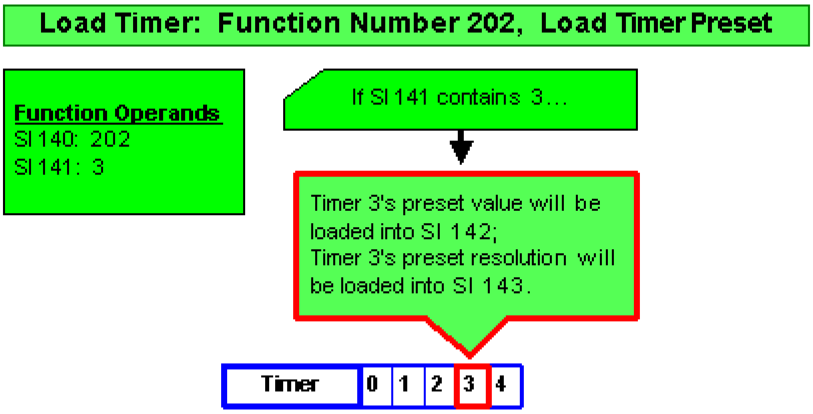
Store the number of the timer into SI 141.

Select the function type by storing the function nu into SI 140.

Function Number (SI 140)	Description
202	Load Timer Preset
203	Load Timer Current

Note that when you run Test (Debug) Mode, the current value in SI 140 will **not** be displayed.

Timer Resolution (stored into SI 143)	
Value	**Resolution**
1	10mS (0.01S)
10	100mS (001S)
100	1000mS (1S)
1000	10000mS (10S)



Load Timer: Function Number 202, Load Timer Preset

Function Operands
SI 140: 202
SI 141: 3

If SI 141 contains 3...

Timer 3's preset value will be loaded into SI 142;
Timer 3's preset resolution will be loaded into SI 143.

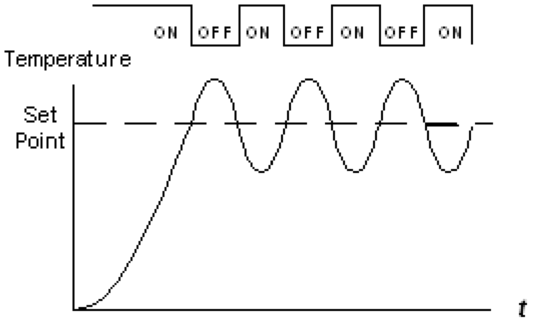
PID

The PID function uses system feedback to continuously control a dynamic process. The purpose of PID control is to keep a process running as close as possible to a desired Set Point.

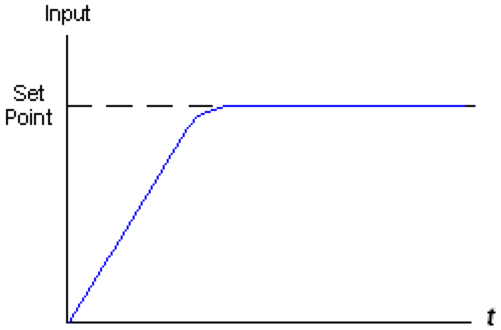
The M90 can run 4 closed PID loops.

About PID and Process Control

A common type of control is On-Off control. Many heating systems work on this principle. The heater is off when the temperature is above the Set Point, and turns on when the temperature is below the Set Point. The lag in the system response time causes the temperature to overshoot and oscillate around the Set Point.



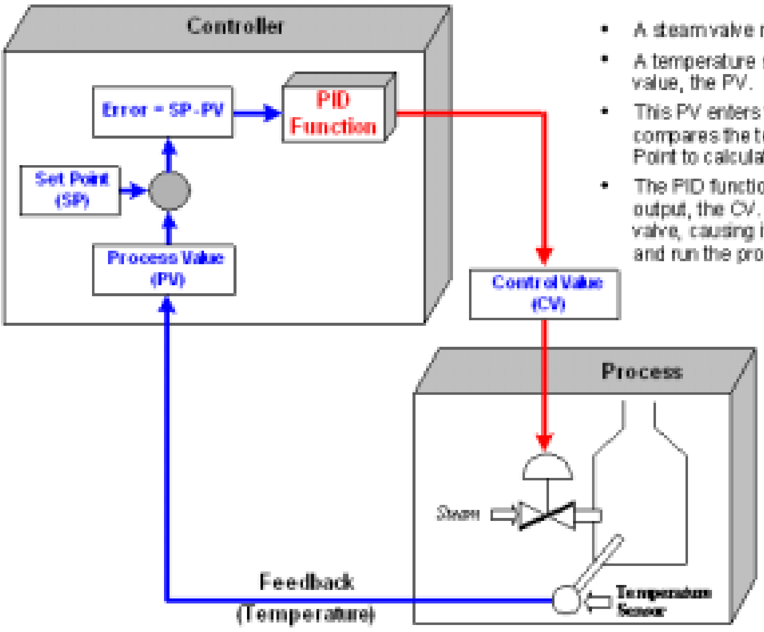
PID control enables you to minimize overshoot and damp the resulting oscillations.



PID enables your controller to automatically regulate your process by:

1. Taking the output signal from the process, called the Process Variable (PV),
2. Comparing this output value with the process Set Point. The difference between the output Process Variable and the Set Point is called the Error signal.
3. Using the Error signal to regulate the controller output signal, called the Control Variable (CV), to keep the process running at the Set Point. Note that this output signal may be an analog or time-proportional variable value.

In the figure below, a system is regulated according to temperature.



- A steam valve releases heat into the system.
- A temperature sensor outputs a temperature value, the PV.
- This PV enters the controller. The PID function compares the temperature value with the Set Point to calculate the Error signal.
- The PID function regulates the controller output, the CV. This output controls the steam valve, causing it to vary the release of steam and run the process at the desired Set Point.

Inside the PID Function

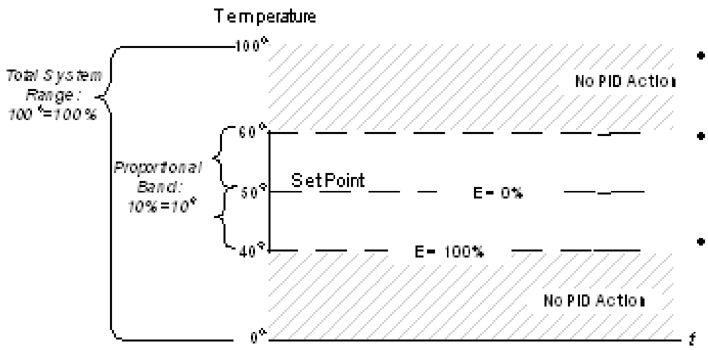
The PID function is based on 3 actions, Proportional, Integral, and Derivative. The PID output is the combined output of all 3 actions.

All of the PID functions are activated by changes in the process Error, the difference between the Process Value and the process Set Point value (E = SP – PV).

Proportional Band

The proportional band is a range defined around the Set Point. It is expressed as a percentage of the total Process Value (PV). When the PV is within this range, the PID function is active.

Note that the proportional band may exceed 100%. In this case, PID control is applied over the entire system range.



- The Process Value (PV) of this system is 0°-100°, a total range of 100°.
- The proportional band is set at 10 %. This means that the range of the proportional band is 40°-60°.
- When the Temperature is outside of the proportional band, the PID function is not active.

Proportional Action

Proportional action begins after the PV enters the proportional band; at this point, the Error is 100%. The action outputs a value that is in **direct linear proportion to the size of the Error value**.

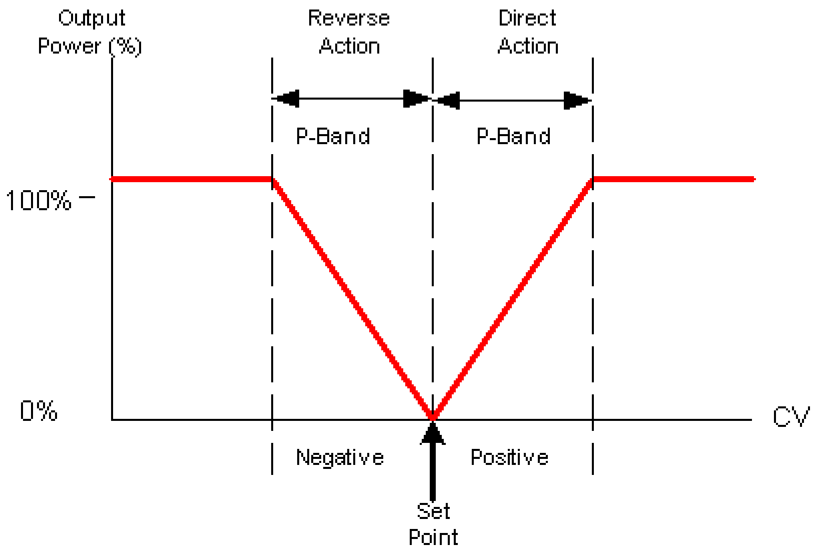
A broad proportional band causes a more gradual initial response from the controller. Typically, Set Point overshoot is low; but when the system stabilizes, oscillations around the Set Point tend to be greater.

A narrow band causes a rapid response that typically overshoots the Set Point by a greater margin. However, the system does tend to stabilize closer to the set point. Note that a proportional band set at 0.0% actually forces the controller into On-Off mode.

The drawback of proportional control is that it can cause the system to stabilize below set point. This occurs because when the system is at set point, Error is zero and the control value output is therefore pegged at zero as well. The majority of systems require continuous power to run at set point. This is achieved by integrating integral and derivative control into the system.

Direct and Reverse Action

Direct action causes the output to change in the same direction as the change in Error, meaning that a positive change in Error causes a positive change in the proportional band's output. Reverse action creates an inverse change in the output, meaning that a positive change in Error causes a negative change in output.

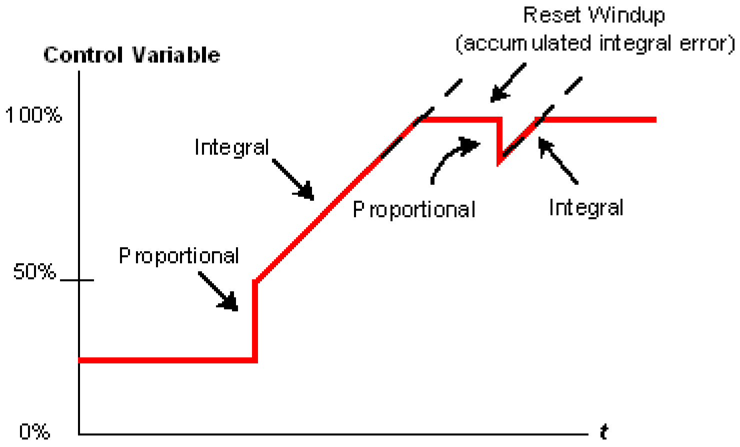


Integral Action

Integral action responds to the rate of change in the controller's CV output relative to the change in Error. The integral time you set is the amount of time, as calculated by the controller, required to bring the process to Set Point. Note that if you set a short integral time, the function will respond very quickly and may overshoot the Set Point. Setting a larger integral time value will cause a slower response. Integral time is sometimes called Reset.

The controller's CV output may reach and remain at 100%, a condition called saturation. This may occur, for example, if the process is unable to reach Set Point. This causes the Error signal to remain stuck in either the positive or negative range. In this situation, the integral action will grow larger and larger as the Error accumulates over time. This is called integral "wind up", which can cause the controller to overshoot the set point by a wide margin.

This situation can be prevented by setting an MB to clear the accumulated Integral error when saturation occurs.



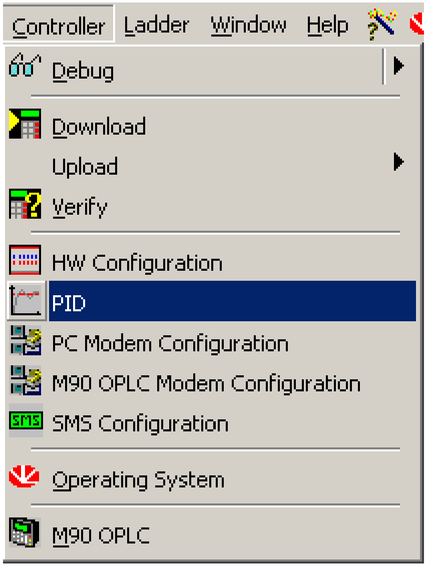
Derivative Action

Derivative action responds to the rate and direction of change in the Error. This means that a fast change in error causes a strong response from the controller.

The derivative action 'anticipates' the PV's value in relation to the Set Point and adjusts the controller's CV output accordingly, thus shortening the PID function's response time.

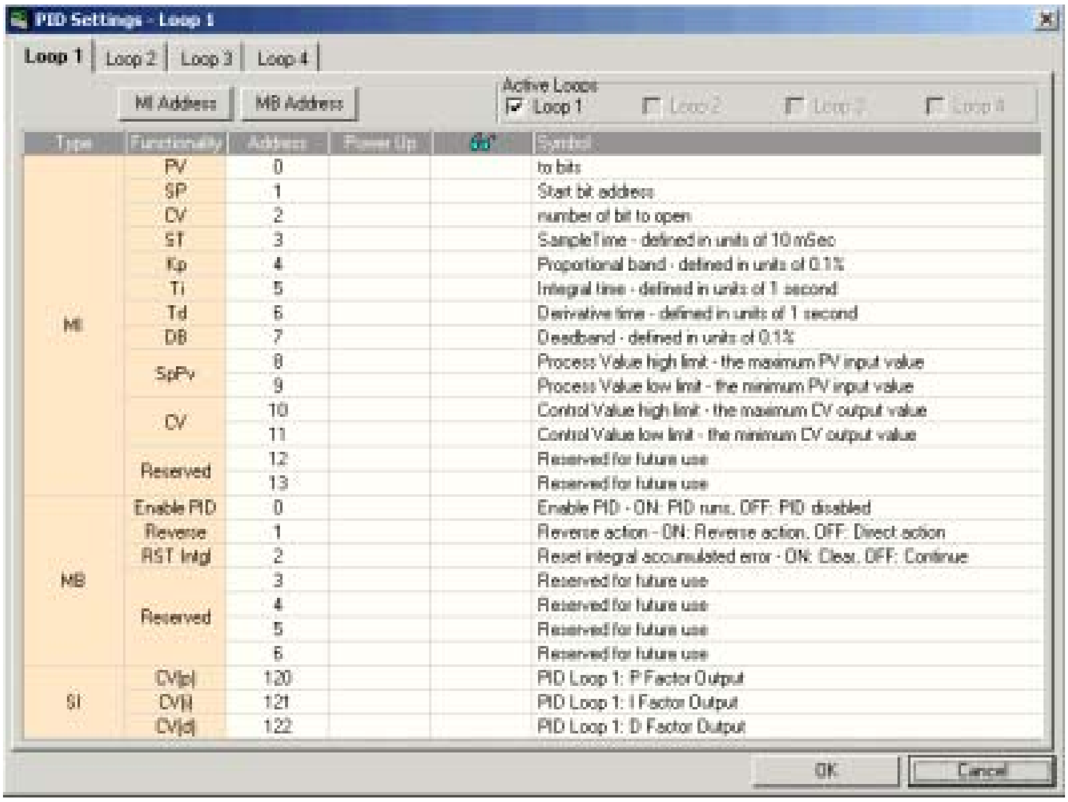
Defining a PID function

1. Select PID from the Controller menu.



The PID parameter box opens as shown below. The parameters are arranged in three groups. Each group is linked to a vector of operands.

2. Link operands to the PID parameters by:
-Clicking the MI Address or MB Address buttons,
OR
-Clicking a parameter;
the Select Operand & Address box opens.
3. Enter a vector's Start Address, then click OK; the parameters are linked to operands in that vector.
4. Repeat the procedure for each of the four PID loops.
5. Before you can use a PID loop, you must activate it by clicking the appropriate check box under Active Loops.



Type	Functionality	Address	Power Up		Symbol
MI	PV	0			to bits
	SP	1			Start bit address
	CV	2			number of bit to open
	ST	3			SampleTime - defined in units of 10 mSec
	Kp	4			Proportional band - defined in units of 0.1%
	Ti	5			Integral time - defined in units of 1 second
	Td	6			Derivative time - defined in units of 1 second
	DB	7			Deadband - defined in units of 0.1%
	SpPv	8			Process Value high limit - the maximum PV input value
		9			Process Value low limit - the minimum PV input value
	CV	10			Control Value high limit - the maximum CV output value
		11			Control Value low limit - the minimum CV output value
	Reserved	12			Reserved for future use
		13			Reserved for future use
MB	Enable PID	0			Enable PID - ON: PID runs, OFF: PID disabled
	Reverse	1			Reverse action - ON: Reverse action, OFF: Direct action
	RST Intgl	2			Reset integral accumulated error - ON: Clear, OFF: Continue
	Reserved	3			Reserved for future use
		4			Reserved for future use
		5			Reserved for future use
		6			Reserved for future use
SI	CV[p]	120			PID Loop 1: P Factor Output
	CV[i]	121			PID Loop 1: I Factor Output
	CV[d]	122			PID Loop 1: D Factor Output

PID Function Parameters

Operand Type	Parameters	Function
MI	PV: Process Value	PV is the feedback from the process. PV is output from the process and input to the PID function. In a heating system, the temperature measured by a temperature sensor provides the PV.
	SP: Set Point	SP is the target value for the process. In a heating system, this is the temperature value set for the system. Note that the Set Point and Process value must be given in the same type of units (degrees Celsius, bars, meters per second, etc.)
	CV: Control Value	CV is the output from the PID function. CV is output from the PID function and input to the process. Note that this output signal may be an analog or time-proportional variable value.
	ST: Sample Time	Use this parameter to define the intervals between PID function updates, in units of 10mSecs.
	Kp: Proportional Band	Use this parameter to define the proportional band, in units of 0.1%. The proportional band is a percentage of the total Process Value (PV). It is a range defined around the Set Point. When the PV is within this range, the PID function is active.
	Ti: Integral Time	Use this parameter to define the integral time, in units of 1 second. Integral action responds to the rate of change in the controller's CV output relative to the change in Error. The integral time you set is the amount of time, as calculated by the controller, required to bring the process to Set Point.
	Td: Derivative Time	Use this parameter to define the derivative time, in units of 1 second. Derivative action responds to the rate and direction of change in the Error. This means that a fast change in error causes a strong response from the controller. The derivative action 'anticipates' the PV's value in relation to the Set Point and adjusts the CV accordingly, thus shortening the PID function's response time.
	DB: Dead Band	Use this parameter to define the dead band, in units of 0.1%. Note that the dead band is a percentage of the proportional band. When values are within the dead band range, the PID function suspends action; the

		controller's CV output is not changed.
	SPPV: Set Point for Process Value	High: Use this parameter to define the upper limit for the Process Value.
		Low: Use this parameter to define the lower limit for the Process Value.
	CV: Set Point for Control Value	High: Use this parameter to define the upper limit for the Control Value.
		Low: Use this parameter to define the lower limit for the Control Value.
	Reserved	Reserved for future use.
MB	Enable PID	Use this parameter in your program to turn the PID loop on and off. ON activates PID action: OFF deactivates PID action.
	Reverse	Use this parameter in your program to control PID output direction. Off activates Reverse Action, ON activates Direct Action. Direct action causes the output value to change in the same direction as the change in PV. Reverse action causes the output value to change in the opposite direction as the change in PV. **Note** ♦ In the case of a temperature control application, **Reverse** Action is **heating**, **Direct** Action is **cooling.**
	RST INTGL: Reset Integral Error	Use this parameter to clear integral error. If the system does not reach setpoint within the time defined in the parameter **Intgl. Time**, Integral error occurs and may increase. Use this parameter to prevent the error from growing large enough to interfere with the Integral operation.
	Reserved	Reserved for future use.
	Reserved	Reserved for future use.
	Reserved	Reserved for future use.
SI	CV(P): Proportional Value	This is the Proportional component of the PID function, calculated by the controller.
	CV(I): Integral Value	This is the Integral component of the PID function, calculated by the controller.
	CV(P): Derivative Value	This is the Derivative component of the PID function, calculated by the controller.

PID Loop Tuning Tips

Here is a common method that may be used to manually tune PID loops.

Kp	1. Set the parameter Kp to 1000 (100%), and the parameters Ti and Td to 0. 2. Check where the PV stabilizes. At this stage, it is to be expected that this may take a long time; note that the PV will probably not reach setpoint. The goal of this operation is, via adjusting the Kp value, to stabilize the PV in the shortest possible time--without overshoot. Note that in general, adjusting the Kp value alone will **not** enable the application to reach setpoint. 3. If the PV rises too slowly, lower the Kp value;if the PV rises too sharply and overshoots, raise the Kp value. In most cases, the Kp value will be between 50-500.

Ti Once the Kp value has been optimized, adjust the Ti value. In most applications, once the Kp has been optimized, the system can reach setpoint via adjusting the Ti parameter.

The goal is to use this parameter to reach the setpoint as quickly as possible without overshoot.

In many applications, a starting value of 300 (Sec) is appropriate. If the time interval before the system reaches reach setpoint is too long, decrease the Ti. If the time interval is too short, overshoot will occur. In this event, increase the Ki value.

In most cases, the Ti value will be between 50-250.

Td If your system requires a rapid PID response, such as to sharp changes in temperature, use the Td parameter.

The higher the value, the more quickly the system will react. A high value may cause overshoot. In most applications, a Td value of 0-30 (sec) is appropriate.

In most cases, the Td value will be between 0-20.

Note ♦ The recommended value for the ST (Sample Time) parameter is 100 (ST=100).

Utilities

Information Mode

The M90's Information Mode allows you to display and edit data, and to perform certain preset actions. The system data is displayed on the M90 LCD screen and edited via the M90 keypad.

You can enter Information Mode at any time, without regard to what is currently displayed on the HMI screen. Viewing data does not affect the M90 program. Note that when you are in Information Mode, the keypad is dedicated to that purpose. The keypad cannot be used for normal application functions until you exit Information Mode.

To enter Information Mode, press the <i> key for several seconds. You navigate through the main menu to reach the category of data you want. Selecting a category opens a submenu.

The list below shows the categories of information that are available for viewing.

Using Information Mode, you can access:

- I/O status
- Analog Inputs: Operating range and current value
- Counter values
- MB and SB Status
- MI and SI current values
- Timers: Current timer value, preset value, and timer status
- M90 ID number
- RS232 Parameters
- Time and Date
- System Information

You can also restart your program, as well as initialize MBs and MIs.

A full description of Information mode is included is the M90 User Guide.

Update Real-Time-Clock (RTC)

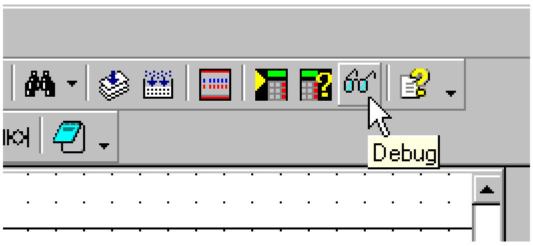
You can update the RTC by storing values into the following SIs.

SI	Description	Values to Store
SI 32	Current Date - according to RTC	Store the day and month as 4 digits. For example, 0402 is February4th; 3012 is December 30th
SI 33	Current Year - according to RTC	Store the year. For example, 1961, or 2002.
SI 34	Current Day of Week - according to RTC	1 to 7, where 1= Sunday, 7 = Saturday

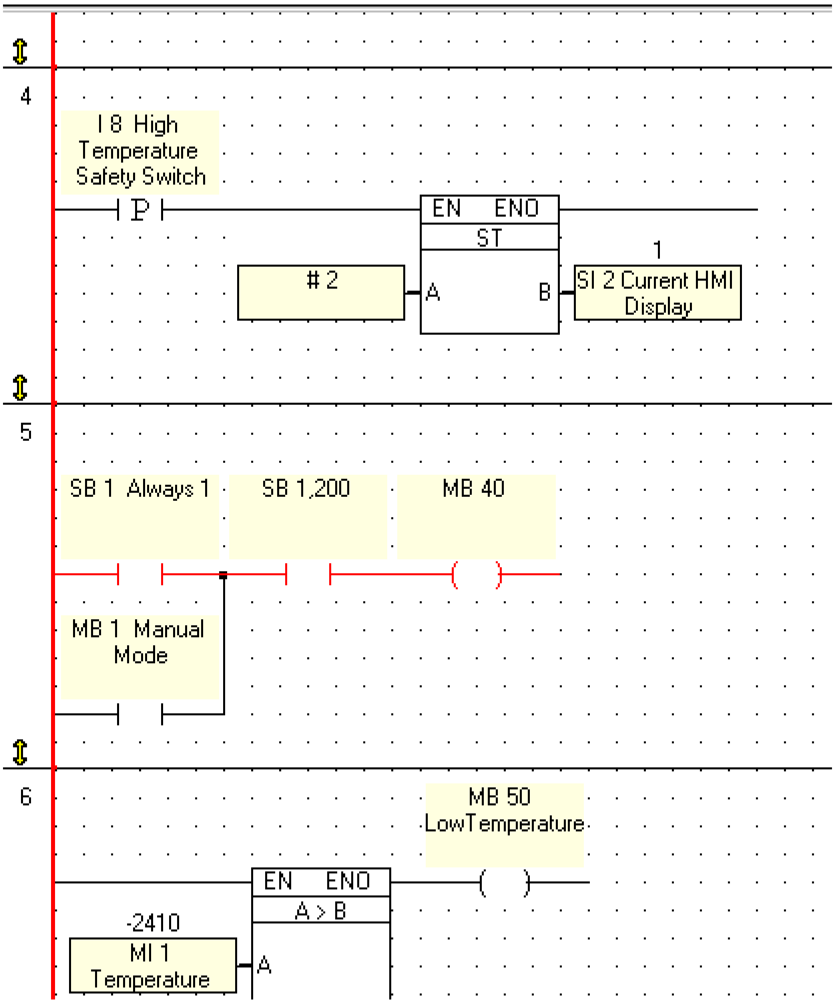
Testing your project (Debug mode)

To test a project:

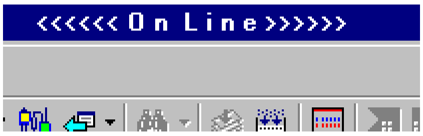
1. Connect the M90 to your PC using the communication cable provided with the software package.
2. Download your program into the M90 from your PC.
3. Click the **Test** icon on the Standard toolbar.



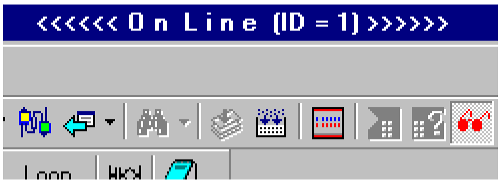
4. The left Ladder bar and any net with Logic flow will appear red. The current values of all MIs and SIs appear above the Operand Symbol.



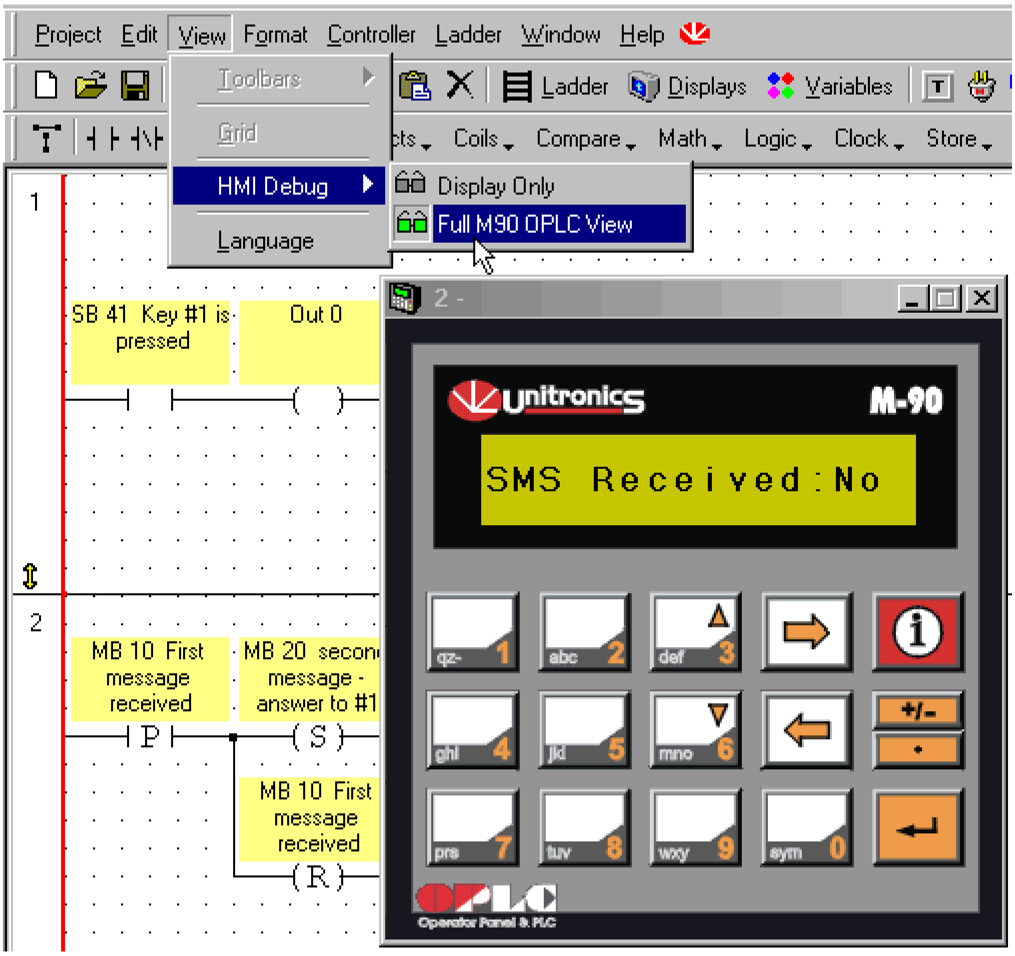
5. During **Test** mode the Title Bar notifies you that you are **On-Line**.



If you are working in a M90 network, the unit ID number appears as well.



You can also view a fully functional, working representation of the M90 OPLC, by selecting Debug HMI from the View menu as shown below. You can choose to see only the current HMI display, or the complete M90, complete with keypad keys. You can test the keypad keys by clicking them, or by using the corresponding numeric keys on your PC's keyboard.


Project Edit View Format Controller Ladder Window Help
Toolbars
Grid
HMI Debug
Language
Display Only
Full M90 OPLC View
Ladder Displays Variables
Coils Compare Math Logic Clock Store
SB 41 Key #1 is pressed
Out 0
MB 10 First message received
MB 20 second message - answer to #1
MB 10 First message received
unitronics M-90
SMS Received:No
qz- 1
abc 2
def 3
ghi 4
jkl 5
mno 6
prs 7
tuv 8
wxy 9
sym 0
+/-
OPLC Operator Panel & PLC

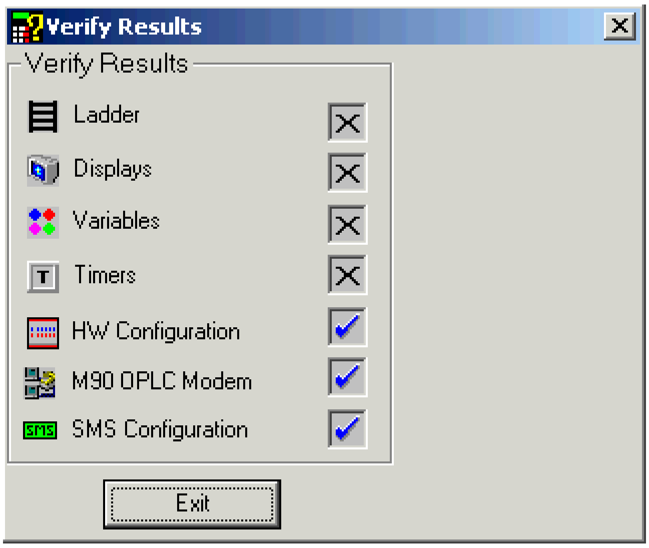
Verify Project

The Verify utility shows the differences between the project open in your PC and the program currently installed in the controller.

To use Verify:

1. Connect your PC to the controller using a program download cable
2. Select Verify from the Connection menu.

Verify marks different sections with an **X**, as shown below.



Verify Results

Verify Results

Ladder ☒

Displays ☒

Variables ☒

Timers ☒

HW Configuration ☑

M90 OPLC Modem ☑

SMS Configuration ☑

M90 Downloader

The M90 Downloader utility is included in Unitronics Remote Access software, which is located on the M90 Software Package CD. The M90 Downloader makes it possible to install .d90 files in local or remote M90/91 controllers without using U90 Ladder.

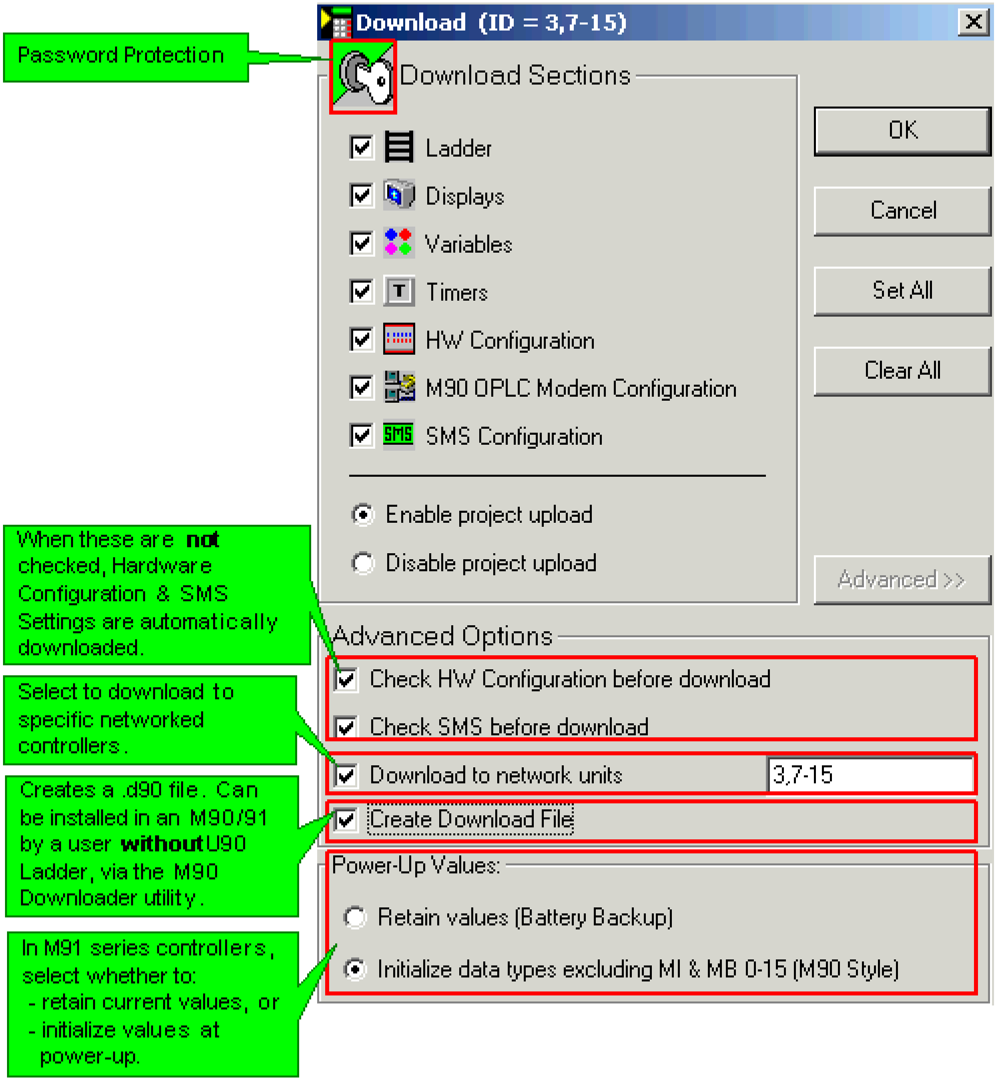
.d90 files are complete M90 applications in a compressed format. They are created when you download U90 programs to an M90.

Creating Download files

Notes ♦ Both the M90 used to make the download file (source), and the M90 that is installed with the .d90 file (target) should be installed with the same OS version.

♦ To avoid errors in the .d90 file, the Download process must run smoothly, without being aborted or affected by PC faults.

1. Click Download, then click the Select All button.
2. Click the Advanced button and *Check Create Download file*.



Password Protection

Download (ID = 3,7-15)

Download Sections

☑ Ladder

☑ Displays

☑ Variables

☑ Timers

☑ HW Configuration

☑ M90 OPLC Modem Configuration

☑ SMS Configuration

● Enable project upload

○ Disable project upload

Advanced Options

☑ Check HW Configuration before download

☑ Check SMS before download

☑ Download to network units 3,7-15

☑ Create Download File

Power-Up Values:

○ Retain values (Battery Backup)

● Initialize data types excluding MI & MB 0-15 (M90 Style)

When these are **not** checked, Hardware Configuration & SMS Settings are automatically downloaded.

Select to download to specific networked controllers.

Creates a .d90 file. Can be installed in an M90/91 by a user **without** U90 Ladder, via the M90 Downloader utility.

In M91 series controllers, select whether to:
- retain current values, or
- initialize values at power-up.

3. A dialog box opens, enabling you to select a Save location. Select a location, then click OK, a .d90 file is created.

Checking the integrity of the Download file

Although you do not need to have the M90 Downloader installed on your PC in order to create .d90 files, you need to install it in order to check .d90 files.

1. After you have created the .d90 file, save the U90 Ladder project from which it was downloaded.
2. Open a new, empty project and download it--using the Select All option--to the M90.
3. Start Remote Access, and start the M90 Downloader which is located on the Remote Access Tools menu.
4. Using the M90 Downloader, navigate to the .d90 file and download it into the M90.
5. Reopen the original U90 Ladder project used to create the .d90 file.
6. Select Verify from the Controller menu; the Verify process will compare the U90 project in your PC with the .d90 application installed in the M90.
7. If the Verify process is successful, the .d90 file is valid.

For more information regarding the M90 Downloader, check the Remote Access Help.

Battery Back-up values

M90/91 controllers have an internal battery back-up for certain values during a power failure.

In M90 models, the battery backs up values from:

- MI 0 - MI 15
- MB 0 - MB 15
- RTC values

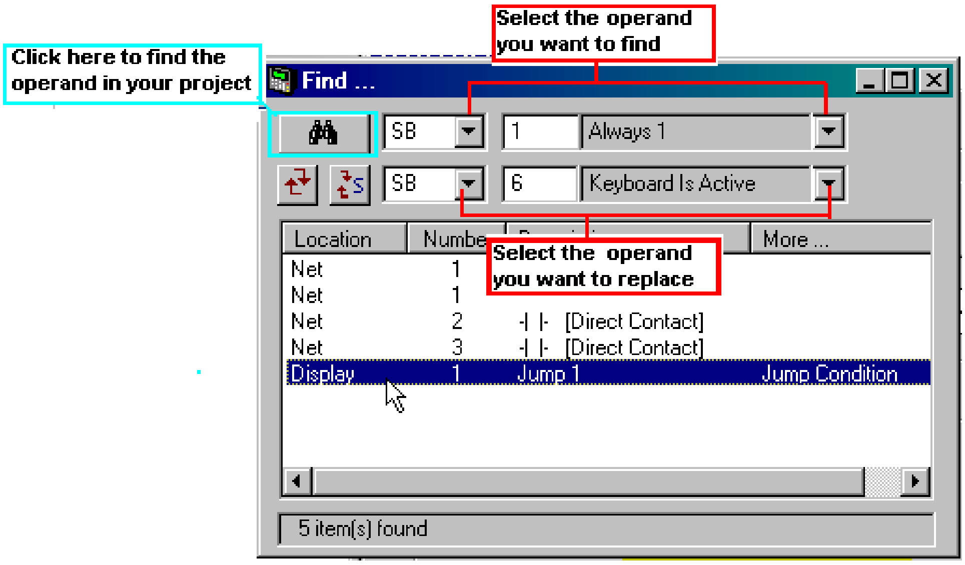
Therefore, any Operand that must retain its value during a power failure must be written into one of the above Operands.

In M91 models, **all** system data and RTC values are backed up.

Find and Replace Elements

To use Find and Replace:

1. Open the **Find** function by clicking on the Find button on the U90 Ladder toolbar.
2. The Find function opens.
3. Select the name and address of the operand you wish to find.
4. Click the Find button shown below; a list appears showing every time that operand is used in the project.
5. Select the name and address of the operand you wish to replace as shown below.



Select the operand you want to find

Click here to find the operand in your project

Select the operand you want to replace

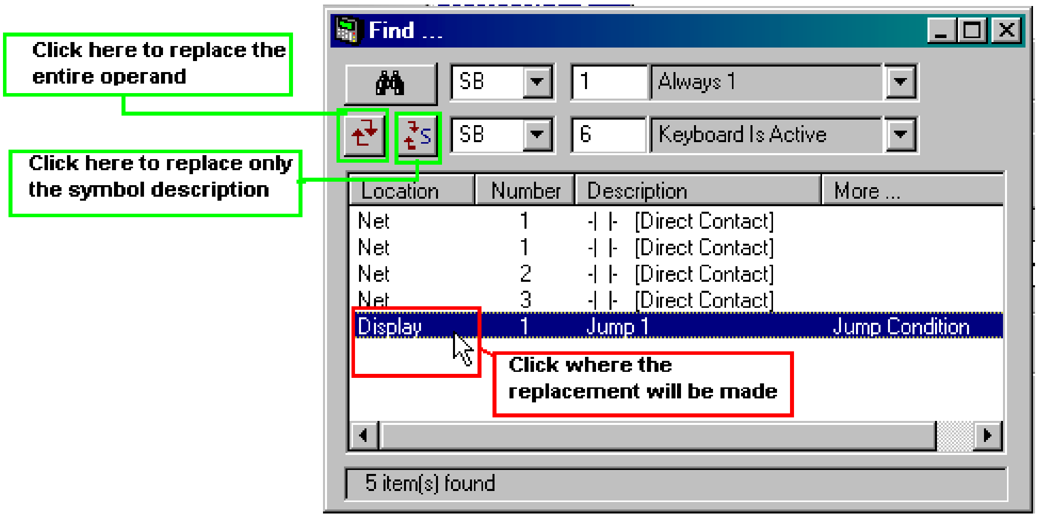
SB 1 Always 1

SB 6 Keyboard Is Active

Location	Numbe		More ...
Net	1		
Net	1		
Net	2	-| |- [Direct Contact]	
Net	3	-| |- [Direct Contact]	
Display	1	Jump 1	Jump Condition

5 item(s) found

6. Select the location of the operand or description you wish to replace by clicking it within the list.
7. Replace operands or their descriptions by clicking the buttons shown below.



Click here to replace the entire operand

Click here to replace only the symbol description

Click where the replacement will be made

SB 1 Always 1

SB 6 Keyboard Is Active

Location	Number	Description	More ...
Net	1	-| |- [Direct Contact]	
Net	1	-| |- [Direct Contact]	
Net	2	-| |- [Direct Contact]	
Net	3	-| |- [Direct Contact]	
Display	1	Jump 1	Jump Condition

5 item(s) found

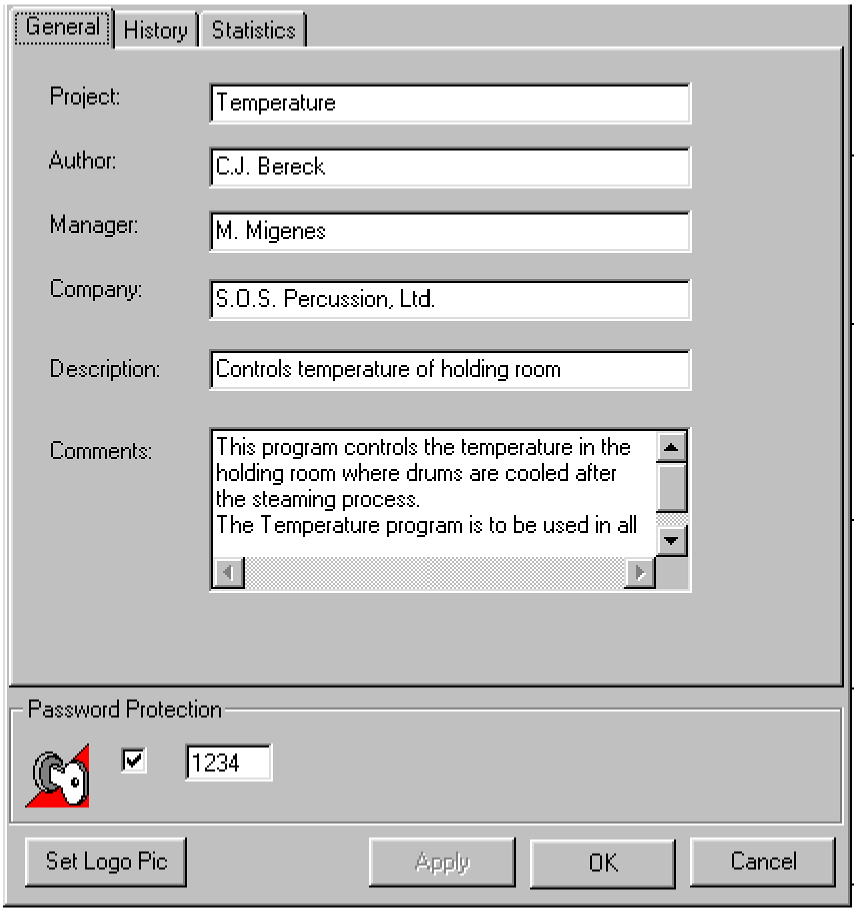
Program Password Protection

When you download a password-protected project into the M90:

- The project cannot be uploaded without the password.
- Project sections cannot be downloaded without the password.

Applying a password

1. Display project properties by selecting Properties from the Project menu. The project Properties box opens.
2. Enable the password field by clicking on the Upload Password check box. When the box is checked, the keys turns and the field turns red. Note that if the box is not checked, you cannot access the password.
3. Enter the password. It must contain 4 digits as shown below--no symbols.



Project: Temperature

Author: C.J. Bereck

Manager: M. Migenes

Company: S.O.S. Percussion, Ltd.

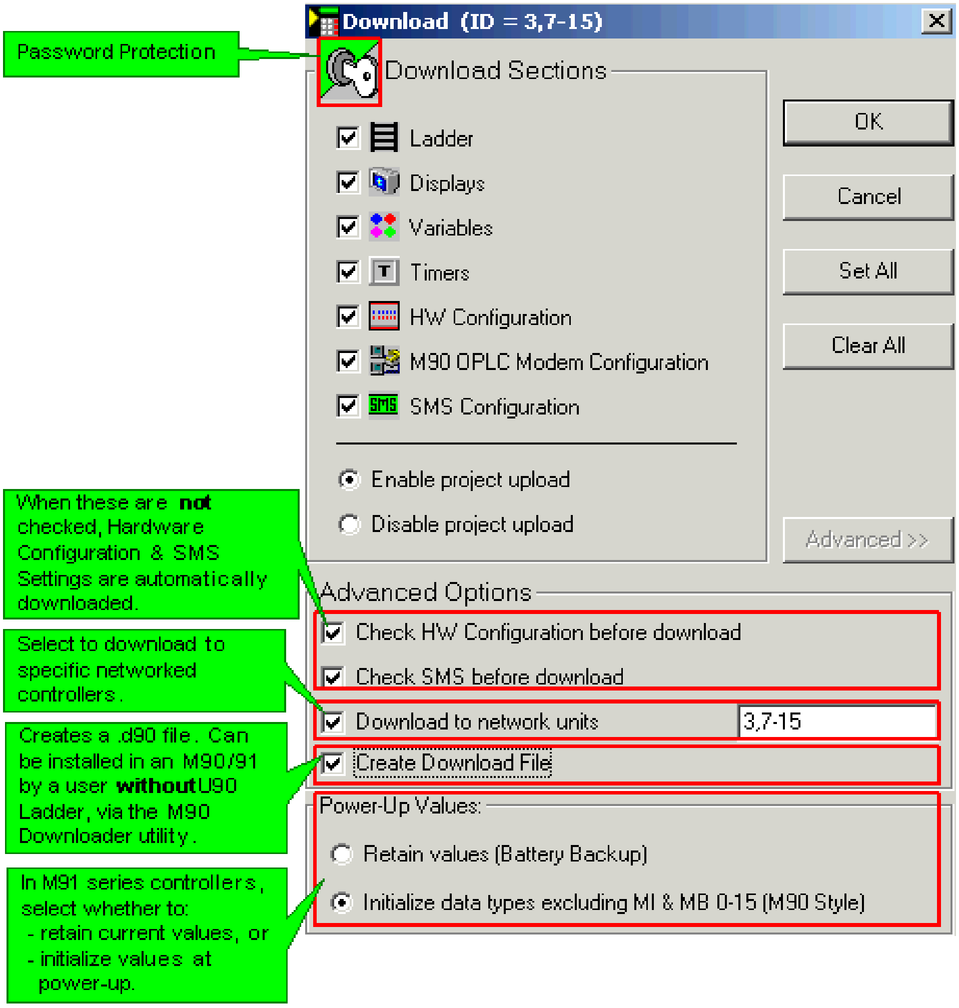
Description: Controls temperature of holding room

Comments: This program controls the temperature in the holding room where drums are cooled after the steaming process.
The Temperature program is to be used in all

Password Protection

☑ 1234

4. Click the **Download** icon on the Standard toolbar. The Download Window opens showing Download Sections. Note the ' password protected' key symbol.
5. Click on Set All. All of the sections are automatically checked as shown below.



Download (ID = 3,7-15)

Password Protection

Download Sections

- ☑ Ladder
- ☑ Displays
- ☑ Variables
- ☑ Timers
- ☑ HW Configuration
- ☑ M90 OPLC Modem Configuration
- ☑ SMS Configuration

● Enable project upload

○ Disable project upload

Advanced Options

- ☑ Check HW Configuration before download
- ☑ Check SMS before download
- ☑ Download to network units 3,7-15
- ☑ Create Download File

Power-Up Values:

○ Retain values (Battery Backup)

● Initialize data types excluding MI & MB 0-15 (M90 Style)

When these are **not** checked, Hardware Configuration & SMS Settings are automatically downloaded.

Select to download to specific networked controllers.

Creates a .d90 file. Can be installed in an M90/91 by a user **without** U90 Ladder, via the M90 Downloader utility.

In M91 series controllers, select whether to:
- retain current values, or
- initialize values at power-up.

6. Click OK. The project downloads.

Note that :

- This process resets the M90, and initializes all bit and integer values.
- If the M90 already contains a password-protected project, you need to supply the password to download sections.
- If the M90 already contains a password-protected program, you cannot upload the program without a password.

Display Integer values as ASCII or Hexadecimal

You can:

- Display the values in an MI vector as ASCII characters.
- Display a register value in hexadecimal format.

To do this, attach a numeric Variable to a Display. The variable uses linearization to display the value(s) in the desired format.

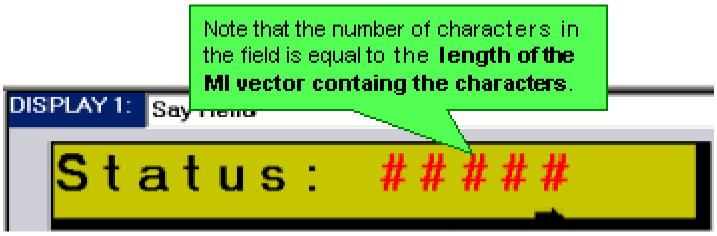
Note that non-supported ASCII characters will be shown as <space> characters.

ASCII -Hexadecimal character table

Vector as ASCII

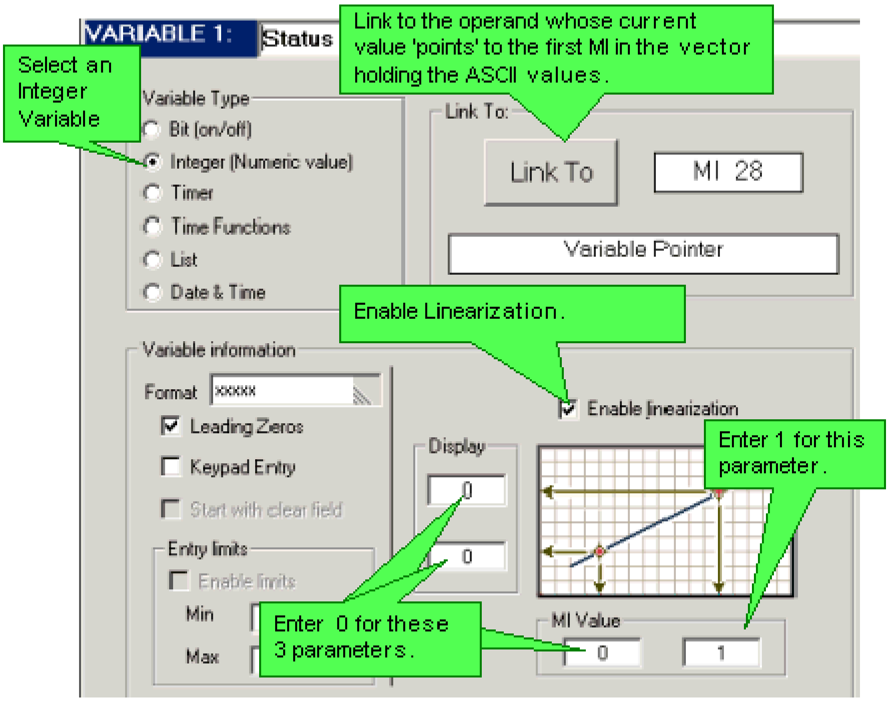
When the application shown in the example below is downloaded, the ASCII characters 'Hello' will be displayed on the M90 screen when Key #3 is pressed.

1. Create a Variable Field in a Display, then attach a Variable.



Note that the number of characters in the field is equal to the **length of the MI vector containg the characters**.

2. Define the Variable as shown below.



Select an Integer Variable

Link to the operand whose current value 'points' to the first MI in the vector holding the ASCII values.

Enable Linearization.

Enter 1 for this parameter.

Enter 0 for these 3 parameters.

VARIABLE 1: Status

Variable Type
- ○ Bit (on/off)
- ● Integer (Numeric value)
- ○ Timer
- ○ Time Functions
- ○ List
- ○ Date & Time

Link To: MI 28

Variable Pointer

Variable information

Format: xxxxx

☑ Leading Zeros

☐ Keypad Entry

☐ Start with clear field

Entry limits: ☐ Enable limits

Min

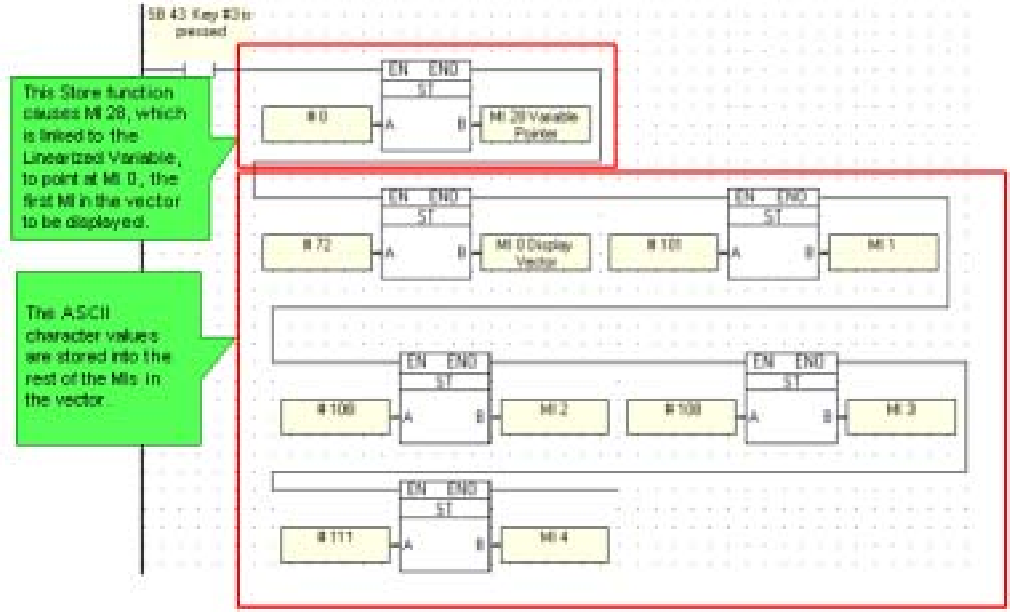
Max

☑ Enable linearization

Display: 0, 0

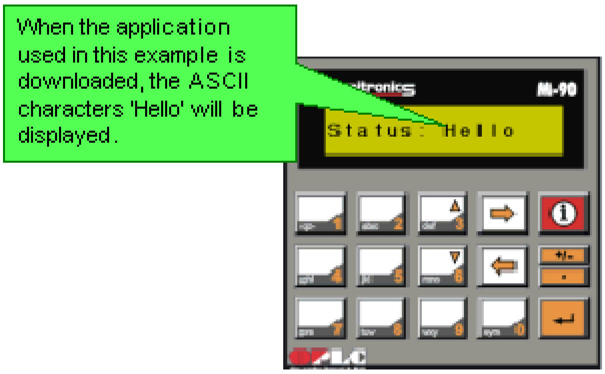
MI Value: 0, 1

3. The Ladder net below sets the Variable pointer and stores ASCII values into the MI vector.



This Store function causes MI 28, which is linked to the Linearized Variable, to point at MI 0, the first MI in the vector to be displayed.

The ASCII character values are stored into the rest of the MIs in the vector.

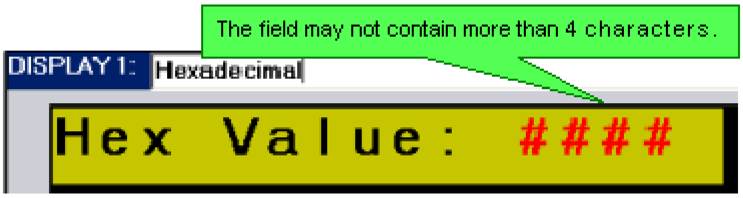


When the application used in this example is downloaded, the ASCII characters 'Hello' will be displayed.

Register Value in Hexadecimal

When the application shown in the example below is downloaded, the hexadecimal value of 63 will be displayed on the M90 screen.

1. Create a Variable Field in a Display, then attach a Variable. Note that if the field is too short, only the right-most characters are displayed. For example, the hex value 63(3F) cannot be shown in a field one character long.

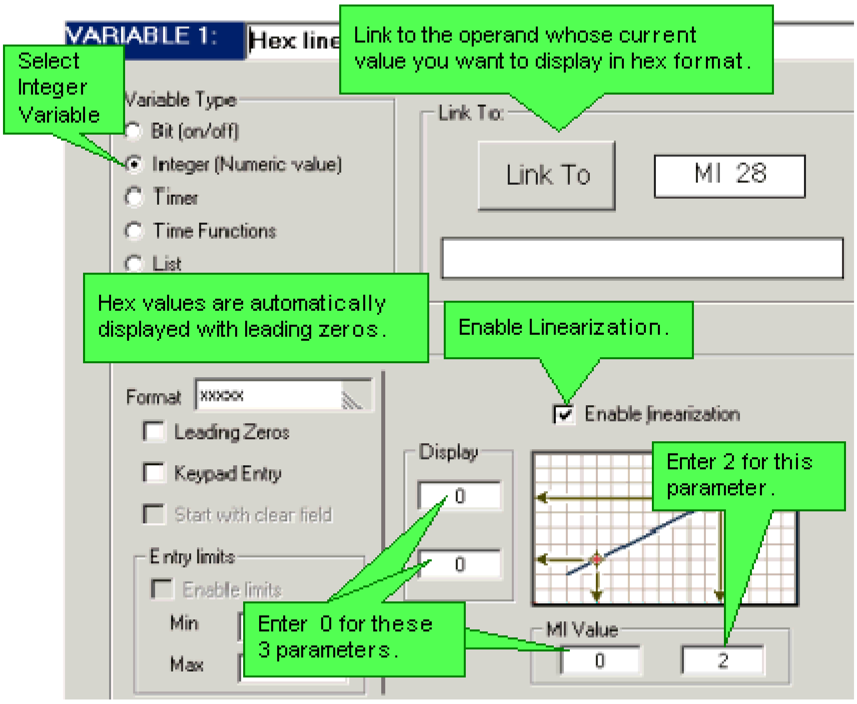


The field may not contain more than 4 characters.

DISPLAY 1: Hexadecimal

Hex Value: ####

2. Define the Variable as shown below.



VARIABLE 1: Hex line

Select Integer Variable

Link to the operand whose current value you want to display in hex format.

Variable Type
○ Bit (on/off)
● Integer (Numeric value)
○ Timer
○ Time Functions
○ List

Link To:
Link To
MI 28

Hex values are automatically displayed with leading zeros.

Enable Linearization.

Format xxxxx
☐ Leading Zeros
☐ Keypad Entry
☐ Start with clear field

Entry limits
☐ Enable limits
Min
Max

☑ Enable linearization

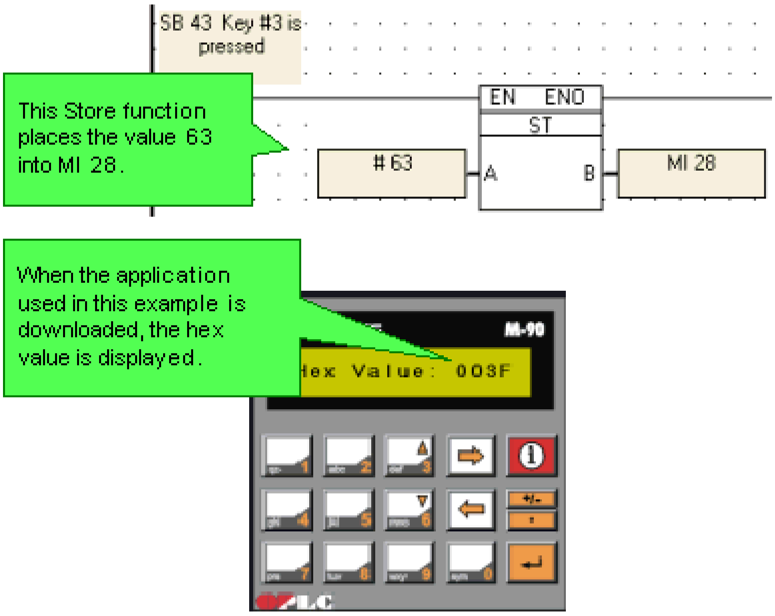
Display
0
0

Enter 2 for this parameter.

MI Value
0
2

Enter 0 for these 3 parameters.

3. The Ladder net below stores the value into the MI.


SB 43 Key #3 is pressed
This Store function places the value 63 into MI 28.
EN ENO
ST
# 63
A
B
MI 28
When the application used in this example is downloaded, the hex value is displayed.
M-90
Hex Value: 003F

Immediate: Read Inputs & HSC, Set/Reset Outputs

You can perform the following immediate actions, without regard to the program scan.

- Set SB 116 to immediately read the status of specific inputs and high-speed counter values. When SB 116 turns ON, the current input value written into linked SBs, current high-speed counter values are written into linked SIs.
- Set the appropriate SBs to immediately clear high-speed counter values.
- Set the appropriate SBs to immediately Set/Reset Outputs.

Note that:

- Values are stored in linked SBs and SIs according to your controller model.
- In the Ladder, inputs and high-speed counters retain the values updated at the beginning of the scan. Only the linked operands listed below are immediately updated. However, immediate changes in output status are immediately updated in the Ladder.

Use the table below to determine which actions, SBs, and SIs are relevant to your model controller.

M90 Model	Input #	Value stored in:	HSC #	Value stored in	HSC #	Immediate Clear	Output #	Set/ Reset via:
M90-T	I 6 I 7	SB 112 SB 113	HSC 0	SI 44	HSC 0	SB 117	None	
M90-T1 M90-T1-CAN	I 8 I 9 I 10 I 11	SB 110 SB 111 SB 112 SB 113	HSC 0	SI 44	HSC 0	SB 117	O 8 O 9 O 10 O 11	SB 120 SB 121 SB 122 SB 123
M90-19-B1A M90-R1 M90-R1-CAN M90-R2-CAN M90-TA2-CAN	I 8 I 9	SB 112 SB 113	HSC 0	SI 44	HSC 0	SB 117	None	

M91-19-TC2 **M91-19-UN2**	**I 0** **I 1** **I 2** **I 3**	SB 110 SB 111 SB 112 SB 113	HSC 0 HSC 1	SI 44 SI 45	HSC 0 HSC 1	SB 117 SB 118	O 0 O 1 O 10 O 11	SB 120 SB 121 SB 122 SB 123
M91-19-R1 **M91-19-R2** **M91-19-R2-CAN**	**I 0** **I 1** **I 2** **I 3** **I 4** **I 5**	SB 110 SB 111 SB 112 SB 113 SB 114 SB 115	HSC 0 HSC 1 HSC 2	SI 44 SI 45 SI 46	HSC 0 HSC 1 HSC 3	SB 117 SB 118 SB 119	O 0 O 1 O 2	SB 120 SB 121 SB 122

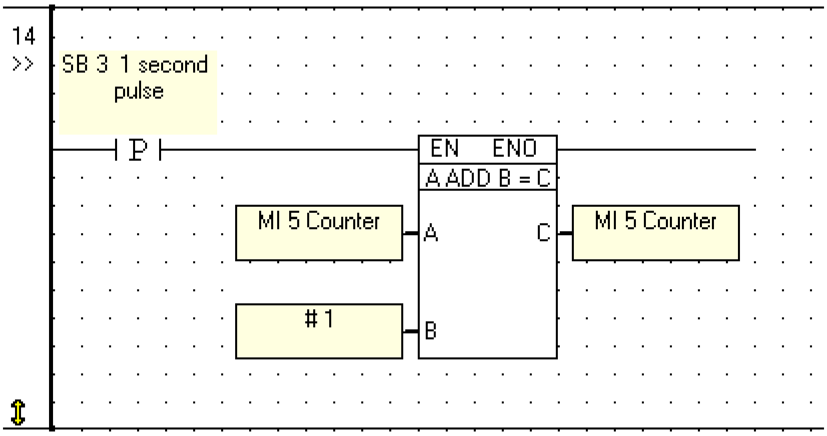
1 Second Pulse Oscillator

There is a built-in 1 second pulse oscillator that generates a 1 Hz pulse.

This pulse oscillator is embedded in **SB 3**. You can use this pulse oscillator as:

- Direct Contact
- Inverted Contact
- Positive Transition Contact
- Negative Transition Contact.

The following example creates a counter that progresses by one every one second.



10mS Counter

The value in SI 1 increments every 10mS. You can store a value into SI 1 at any time during your program, such as 0 to reset the counter.

Communication Utilities

Use this utility to enable your controller to receive data from external devices, such as bar-code readers, via an RS232 port. Since there is no Ladder element for this function; you perform it by storing values into SIs.

Note that the communication settings stored into these SIs only take effect at power-up.

SI	**Parameter**	**Value to Store**	**Notes**
141	STX (Start of Text)	0-255(ASCII) -1: No Start of Text (not recommended)	The STX parameter indicates where the data block begins. • Note that the ASCII character '/' (backslash) cannot be used to indicate the start of the

			data block.
142	ETX (End of Text)	0-255(ASCII) -1: ETX marked by Length -2: ETX marked by 'Silence'	The ETX parameter indicates where the data block ends. When the ETX is registered by the function, SB 60 turns ON. ● If you use an ASCII character (0-255), note that if this character occurs **after** the Length parameter defined in SI 143, SB 60 turns ON. ● Selecting -1 causes the function to use the **length** of a data block alone to determine its end. ● Selecting -2 causes the function to use the **duration** of silent time following the STX to determine the end of a data block.
143	ETX Length or Silent	Length: up to 128 Silent: up to 24000	● This defines both the length of text, or silence, that signal the end of text. ● Note that the duration of a silent 'counter' unit is approximately 2.509 mS. The 'silent' value should be lower than the M90 TimeOut value. ● When defined as length, SI 143 cannot exceed SI 144.
144	Maximum Length	Up to 128	● This is the maximum legal length for received text. ● When the maximum length is exceeded, the Receive Buffer is automatically cleared, and SB 60 is turned OFF, enabling new data to be received. ● This can be used to detect buffer overflow.
145	Start Address: Receive Buffer	MI Address	This MI contains the start address for the vector of registers that serves as the Receive Buffer.
60	Number of Bytes currently in Receive Buffer	Read only	SI 60 indicates how many bytes of data are currently in the Receive Buffer.
61	Number of Bytes in Receive Buffer when SB 60=1	Read only	SI 61 indicates how many bytes of data are in the Receive Buffer when SB 60 turns ON.
146	Copy Data: Format	0: copy each received byte 1: copy in groups of 4 received bytes.	● **0** causes each separate byte to be copied to a separate register including STX and ETX. ● **1** causes every **4** bytes to be copied to a single register, without the STX and ETX. This is used when the received data is in numeric format. For example 12345 would be copied to 2 consecutive MIs. The first MI would contain 1234, the second would contain 5.
140	Start receiving	300	In your application, use this to call the function after you have entered all of the other parameters. Note that when you run Test (Debug)

		Mode, the current value in SI 140 will **not** be displayed.
SB	**Description**	**Notes**
60	Data Successfully Received	Read only. Turns ON when the ETX condition is registered by the system.
61	Copy Data in Receive Buffer to MI Vector	Write only. • Turning this SB ON causes the buffer contents to be copied to the MI vector defined in SI 145. The data will be copied according to the format defined in SI 146. • If SI 146 is set to 0, this SB can be set at any time. If SI 146 is set to 1, this SB can be set after SB 60 turns ON.
62	Clear Receive Buffer, Clear SI 60, Clear SI 61, Reset SB 60	• This SB must be turned ON to enable a new message, or data block, to be received. • Turn this SB ON to enable data to be received **before** the maximum length, defined in SI 144, is exceeded.

Note that if no data is received for a period exceeding the M90 TimeOut, you will lose the data in the buffer.

To see how to use the Communications Utility, check the sample application **Read Card - Display Number Value.U90**. This may be found by accessing Sample U90 Projects from the Help menu.

This application demonstrates how to read a magnetic card number using an "IDTECH" card reader, then display that number on the M90's screen. The card reader transmits the number in ASCII characters in this format:

< %?[CR];xxxxx?[CR] > where **xxxxx** is the card number.

The ASCII character used to mark the Start Of Text (STX) is < **;** > (semicolon). End Of Text (ETX) is marked with the character < **?** > .

Since the card number is 5 digits long, the card number is copied to 2 separate MIs. The MIs are linked to 2 variables that are shown on the M90's screen in 2 separate Displays.

The parameters must be written into their respective operands using one scan condition. For this purpose, it is recommended to use SB 2 Power-up bit, as shown in the sample application.

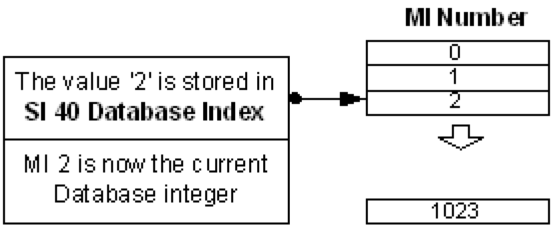
Access indirectly addressed registers: Using the Database

You can access and use integers 0 through 1023 within the M90 OPLC's memory as a database, via SI 40 and SI41.

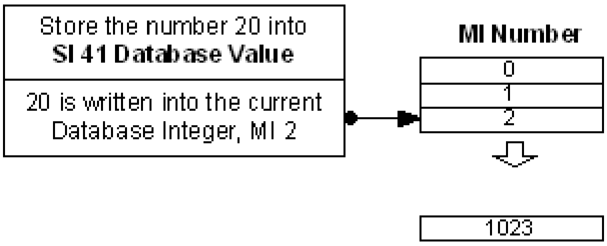
Note that when you run Test (Debug) Mode, the current value in SI 140 will **not** be displayed.

Writing Values

1. Use SI 40 Database Index to access a particular MI.
   For example, to access MI 2 you store the number 2 into SI 40.

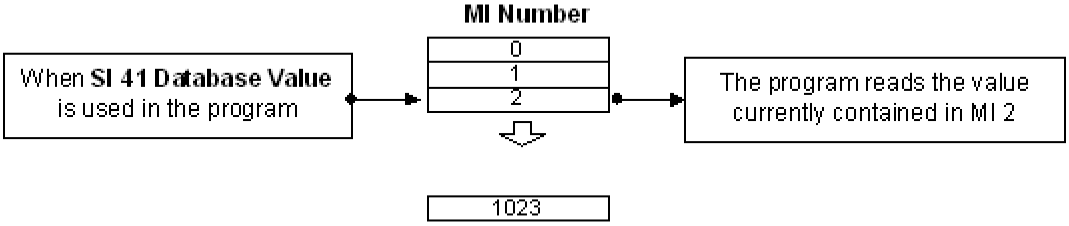


2. Use SI 41 Database Value to write a value into MI 2.
For example, you can store a number value into SI 41.



Reading Values

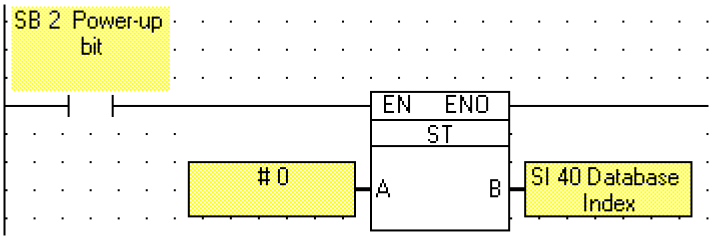
When you use SI 41 Database Value in your program, the program actually reads the MI that is referenced by SI 40 Database Index.


MI Number
When SI 41 Database Value
is used in the program
0
1
2
1023
The program reads the value
currently contained in MI 2

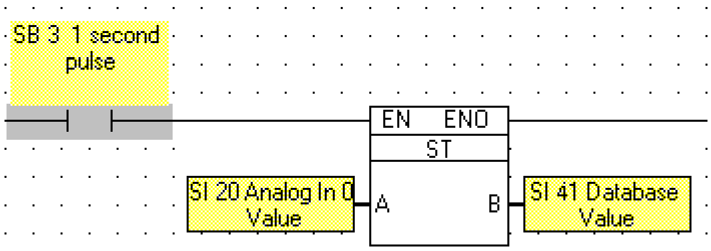
Examples

Example 1: Write

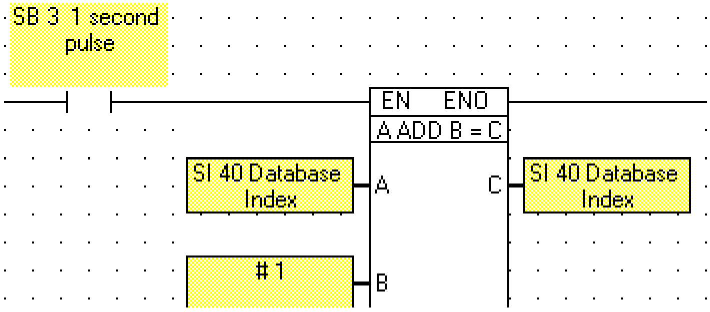
In the net below, 0 is stored in SI 40 when the M90 OPLC is powered up. This means that integer 0 is now the current 'database' integer.



In the net below, the analog value contained in SI 20 is stored in SI 41 every second. According to the net above, the current 'database' integer is 0. The analog value is therefore stored in integer 0.



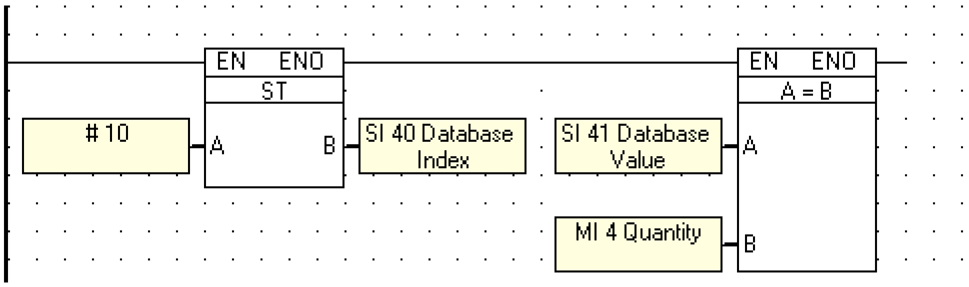
In the next net, the value in SI 40 is incremented by 1every second, changing the current database integer. This means that the first analog value will be stored in integer 0, the second analog value in integer 1, and so on.



Example 2: Read

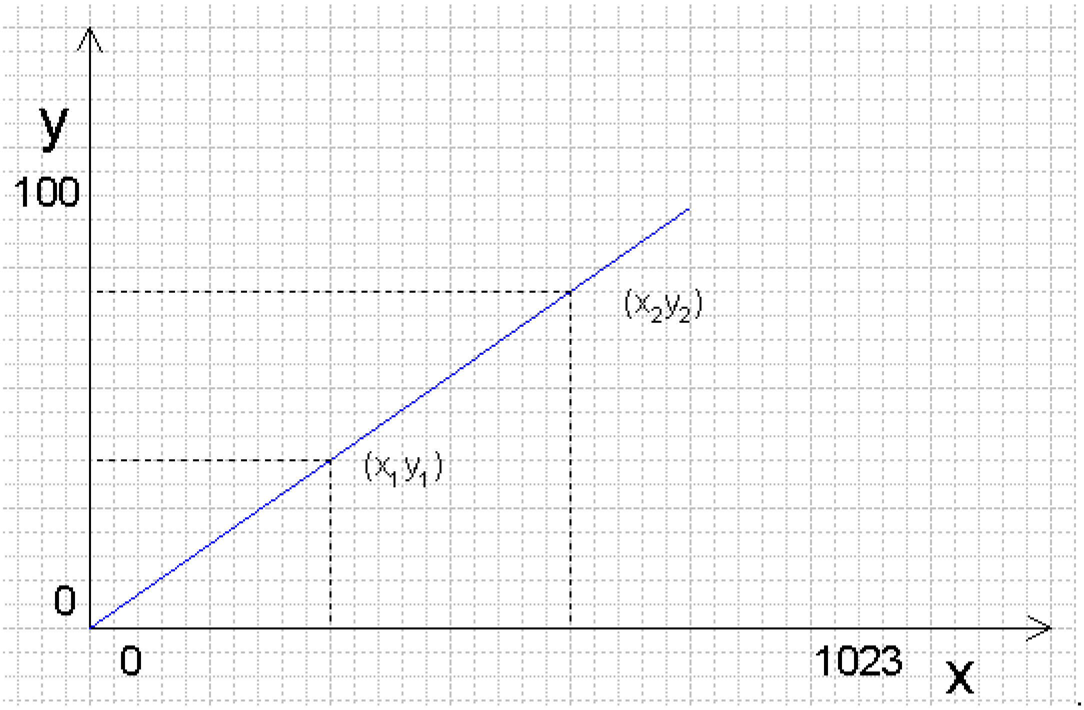
In the first part of the net below, 10 is stored into SI 40. Integer 10 is the 'database' integer. In the second part of the net, the value in SI 41 is compared to the value in integer 4.

The value in SI 41 is the value actually in integer 10—the current database integer.



Linearization

Linearization can be used to convert analog values from I/Os into decimal or other integer values. An analog value from a temperature probe, for example can be converted to degrees Celsius and displayed on the controller's display screen.



Linearize values for Display

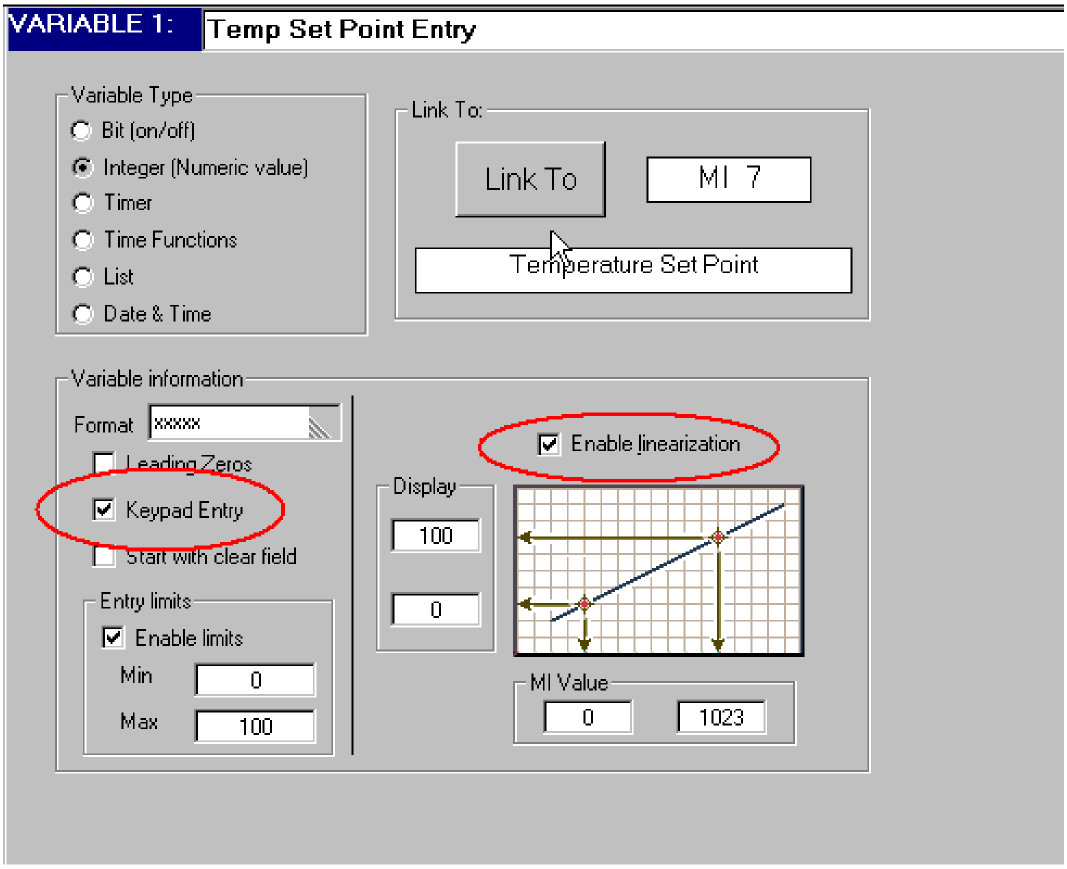
Note that the linearized value created in this way may be displayed-- **but** the value **cannot** be used anywhere else within the project for further calculations or operations.

You can enter an Analog value, such as temperature, via the M90 keypad, then convert that value into a Digital value for comparison with a digital value from a temperature probe by selecting **Enable Linearization** in the linked Variable.

This conversion process is Reverse Linearization.

To enable Analog to Digital conversion:

1. Create a Display for entering the analog value.
2. Create an Integer Variable.
3. Select **keypad entry** and **enable linearization**.
4. Enter the linearization values for the x and y axes.



VARIABLE 1: Temp Set Point Entry

Variable Type
○ Bit (on/off)
● Integer (Numeric value)
○ Timer
○ Time Functions
○ List
○ Date & Time

Link To:
Link To
MI 7
Temperature Set Point

Variable information
Format: xxxxx
□ Leading Zeros
☑ Keypad Entry
□ Start with clear field

Entry limits
☑ Enable limits
Min: 0
Max: 100

☑ Enable linearization

Display: 100, 0

MI Value: 0, 1023

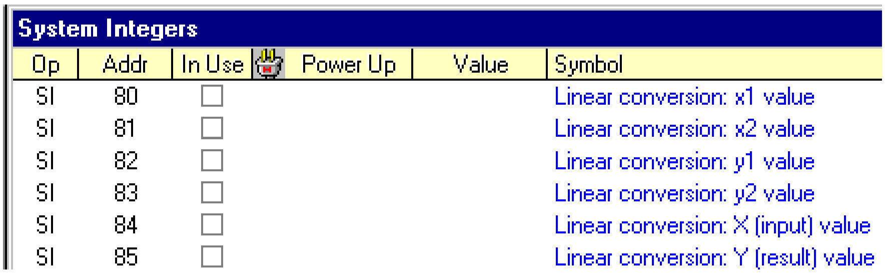
According to the above example:

- A temperature entry of 100^0 C will be converted to 1023 Digital value.
- A temperature entry of 50^0 C will be converted to 512 Digital value.

Linearize values in the Ladder

You can also linearize values in your Ladder and display them on the M90's LCD.

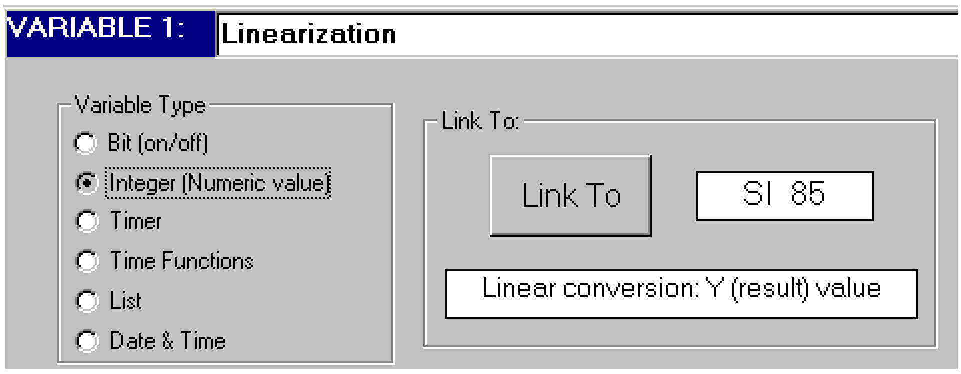
1. In your Ladder project, use SI 80 - 85 to set the (x,y) variable ranges. Use SB 80 to activate the **Linearization** function.



System Integers

Op	Addr	In Use	Power Up	Value	Symbol
SI	80	□			Linear conversion: x1 value
SI	81	□			Linear conversion: x2 value
SI	82	□			Linear conversion: y1 value
SI	83	□			Linear conversion: y2 value
SI	84	□			Linear conversion: X (input) value
SI	85	□			Linear conversion: Y (result) value

The linearization values created here can be displayed by linking SI 85 to a Display;the value **can** be used elsewhere within the project for further calculations or operations.



VARIABLE 1: Linearization

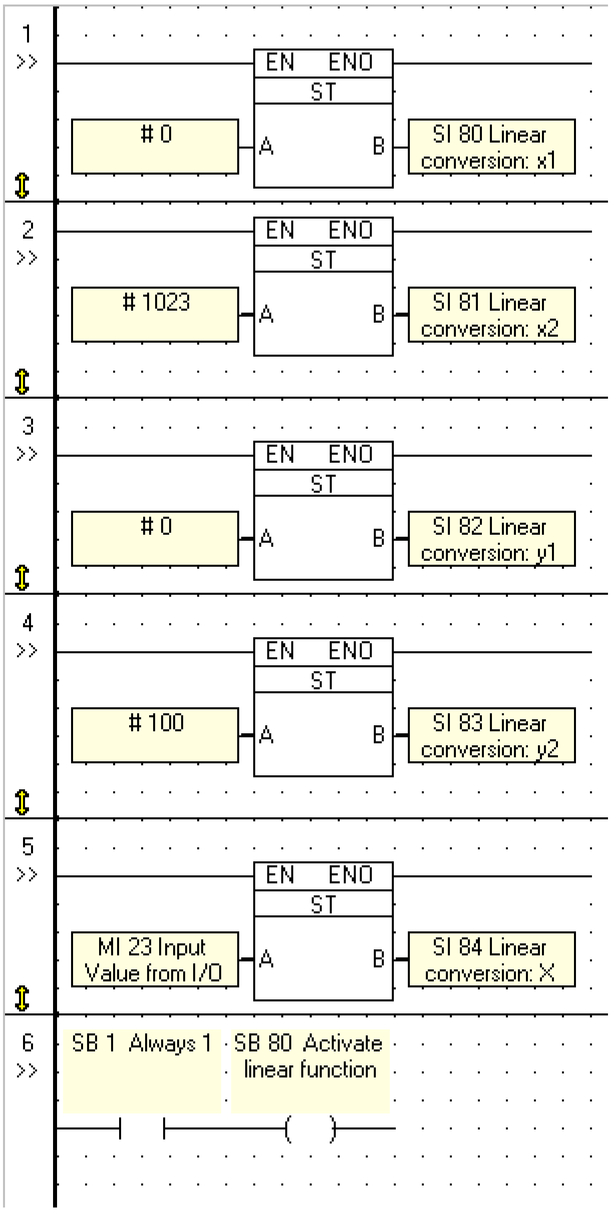
Variable Type
- ○ Bit (on/off)
- ● Integer (Numeric value)
- ○ Timer
- ○ Time Functions
- ○ List
- ○ Date & Time

Link To:

SI 85

Linear conversion: Y (result) value

Example: write the variable ranges into SI 80 - 83, then writing an analog input into SI 84:



1
>>
EN ENO
ST
# 0
A
B
SI 80 Linear conversion: x1
2
>>
EN ENO
ST
# 1023
A
B
SI 81 Linear conversion: x2
3
>>
EN ENO
ST
# 0
A
B
SI 82 Linear conversion: y1
4
>>
EN ENO
ST
# 100
A
B
SI 83 Linear conversion: y2
5
>>
EN ENO
ST
MI 23 Input Value from I/O
A
B
SI 84 Linear conversion: X
6
>>
SB 1 Always 1
SB 80 Activate linear function

FAQs

General

Can I work with more than one application open at a time?

No, you cannot work with more than **one** application open at a time. If you try to open a new or existing project, the project currently open project will close. If you have unsaved changes, you will be prompted to save them before the project closes.

How does the program know when a keypad entry is complete?

When a keypad entry is complete, there are special SBs that go to logic 1 for one system scan.

If there is more than one Variable on display, there is an HMI Var Keypad Entry Complete SB for each Variable.

The SBs are:

- SB 30 - HMI Keypad entries complete
- SB 31 - HMI Var 1 Keypad entry completed
- SB 32 - HMI Var 2 Keypad entry completed
- SB 33 - HMI Var 3 Keypad entry completed
- SB 34 - HMI Var 4 Keypad entry completed

You can use these special SBs in your Ladder project or Jump conditions to move from Display to Display when keypad entry is complete.

Update Real-Time-Clock (RTC)

You can update the RTC by storing values into the following SIs.

SI	Description	Values to Store
SI 32	Current Date - according to RTC	Store the day and month as 4 digits. For example, 0402 is February4th; 3012 is December 30th
SI 33	Current Year - according to RTC	Store the year. For example, 1961, or 2002.
SI 34	Current Day of Week - according to RTC	1 to 7, where 1= Sunday, 7 = Saturday

How many times can I use an Operand in a project?

There is no limit to the number of times you can use the same Operand and Address in your project.

Note that a specific Direct Coil or Inverted Coil Operand and Address should **only** be used **once** in a project.

Assigning a Unit ID number

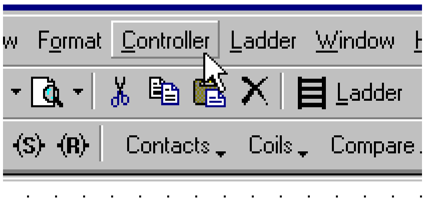
When you create an M90 network, you must assign a Unit ID number to each controller. A Unit ID number is unique. It **must** be used **only** once within a network.

You use this number for two purposes:

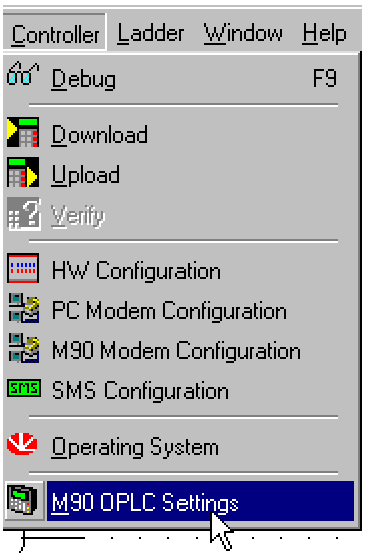
- To enable the M90 controllers to exchange data.
- To access a networked M90 via your PC.

To set a Unit ID number:

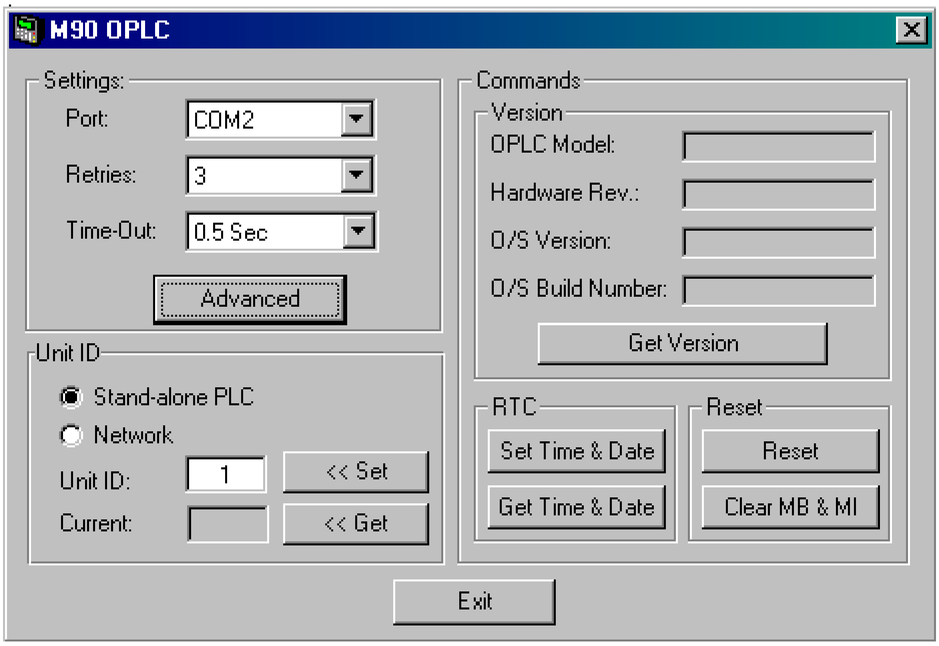
1. Click Controller on the Standard menu bar.


w Format Controller Ladder Window H
Ladder
(S) (R) Contacts Coils Compare

2. Select **M90 OPLC Settings** from the Controller menu.



3. The **M90 OPLC Settings** window opens.



Port: COM2

Retries: 3

Time-Out: 0.5 Sec

Unit ID

● Stand-alone PLC

○ Network

Unit ID: 1

Current:

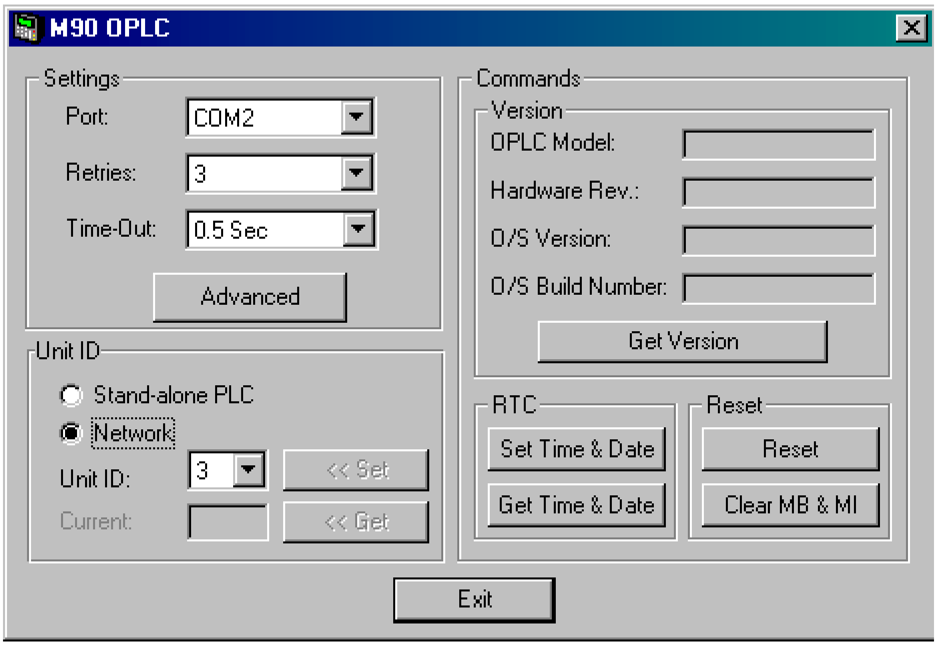
OPLC Model:

Hardware Rev.:

O/S Version:

O/S Build Number:

4. Enter the new ID number in the Unit ID window.



Port: COM2

Retries: 3

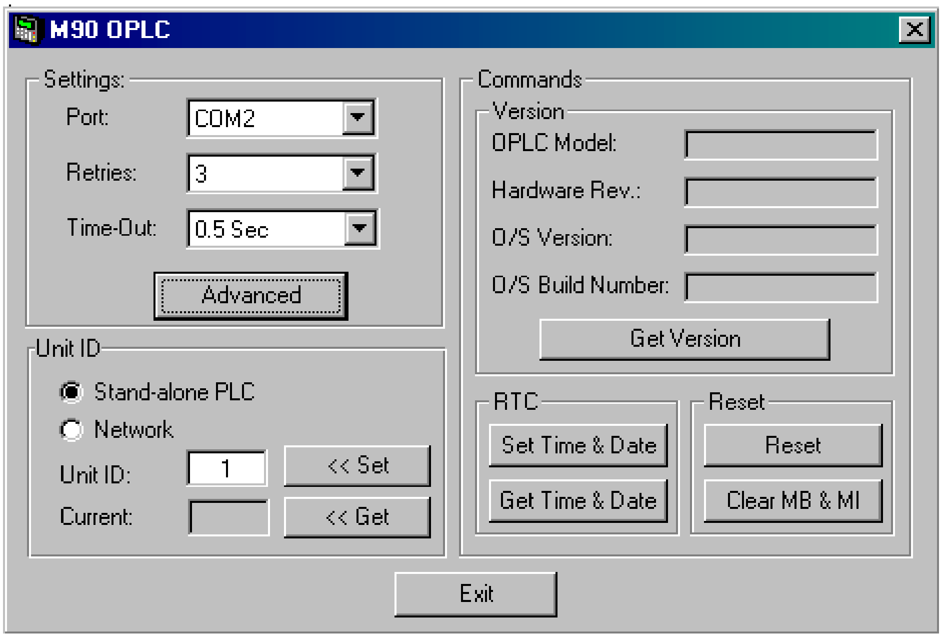
Time-Out: 0.5 Sec

○ Stand-alone PLC

● Network

Unit ID: 3

5. Click **<< Set** to enter the new IN number.



Port: COM2

Retries: 3

Time-Out: 0.5 Sec

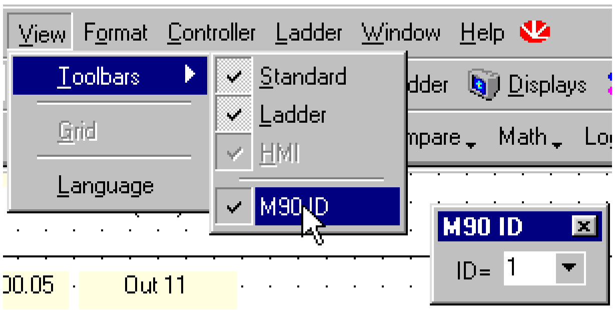
● Stand-alone PLC

○ Network

Unit ID: 1

Displaying the Unit ID Tool Bar

1. Display the Unit ID by selecting M90 ID from the controller.
2. The Unit ID tool bar opens as shown below.



To download via an M90 bridge to a networked M90, you must select the unique ID of the networked M90. When you enter '0' as the Unit ID number, you communicate directly with the M90 that you are using as a bridge to the network.

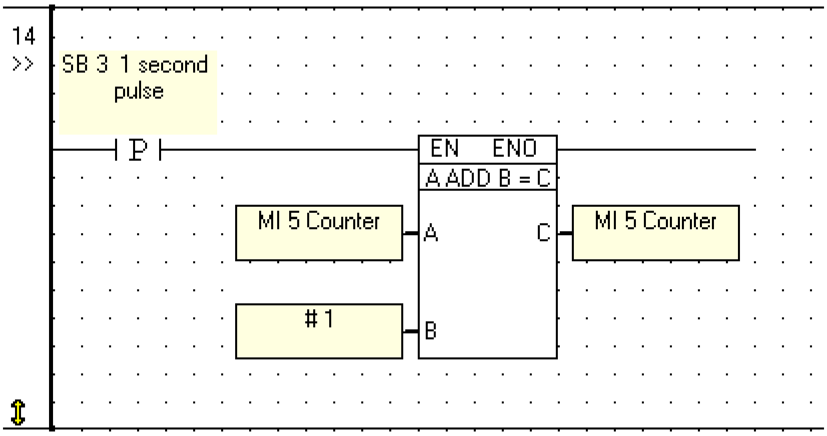
1 Second Pulse Oscillator

There is a built-in 1 second pulse oscillator that generates a 1 Hz pulse.

This pulse oscillator is embedded in **SB 3**. You can use this pulse oscillator as:

- Direct Contact
- Inverted Contact
- Positive Transition Contact
- Negative Transition Contact.

The following example creates a counter that progresses by one every one second.

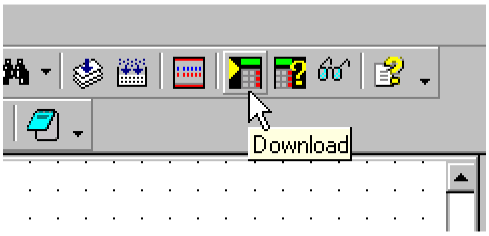


Downloading a Project

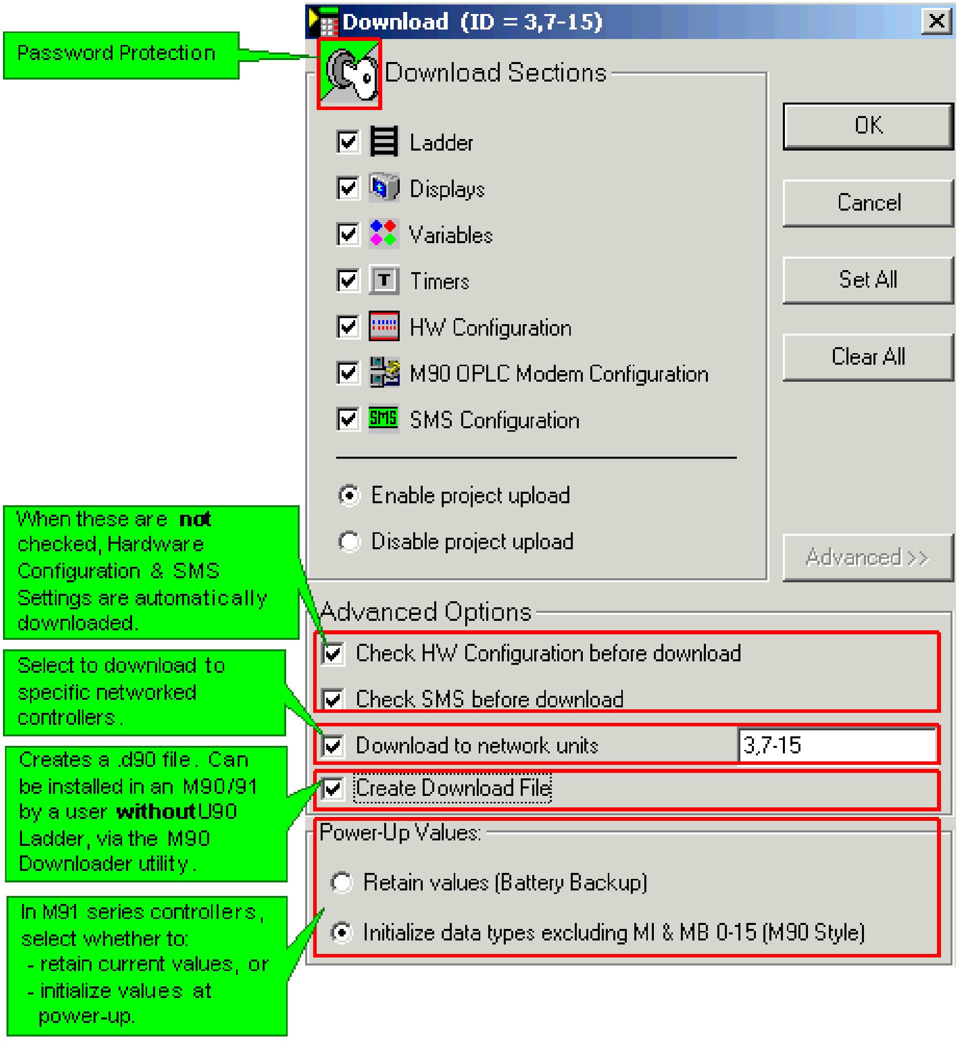
The Download process transfers your project from the PC to the controller.

To download a project to a controller:

1. Click the **Download** icon on the Standard toolbar.



2. The Download Window opens with Download Sections. Those sections which have yet to be downloaded to a controller will be selected. If you have made no changes in the project since the last download, you have to select the Download Sections manually. Click OK.



Password Protection

Download (ID = 3,7-15)

Download Sections

☑ Ladder

☑ Displays

☑ Variables

☑ Timers

☑ HW Configuration

☑ M90 OPLC Modem Configuration

☑ SMS Configuration

● Enable project upload

○ Disable project upload

Advanced Options

☑ Check HW Configuration before download

☑ Check SMS before download

☑ Download to network units 3,7-15

☑ Create Download File

Power-Up Values:

○ Retain values (Battery Backup)

● Initialize data types excluding MI & MB 0-15 (M90 Style)

When these are **not** checked, Hardware Configuration & SMS Settings are automatically downloaded.

Select to download to specific networked controllers.

Creates a .d90 file. Can be installed in an M90/91 by a user **without** U90 Ladder, via the M90 Downloader utility.

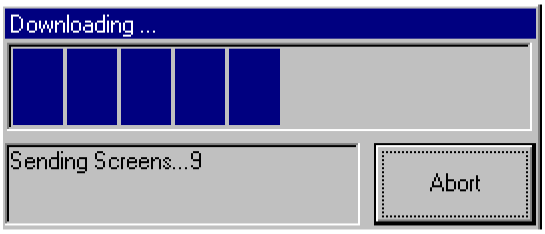
In M91 series controllers, select whether to:
- retain current values, or
- initialize values at power-up.

The key at the top tells you if the project is password protected. If so, the password will have to be supplied at upload.

Note Ladder Image and Project Symbols option. If you do not select this option, the Ladder program cannot be uploaded to a PC for editing. You only be able to view the uploaded program in STL. To enable the Ladder program to upload from the M90 into a PC, select this option.

Note the different Power-up value (Battery Backup) options.

3. The Downloading Progress window opens. This window closes when download is complete.



Downloading ...

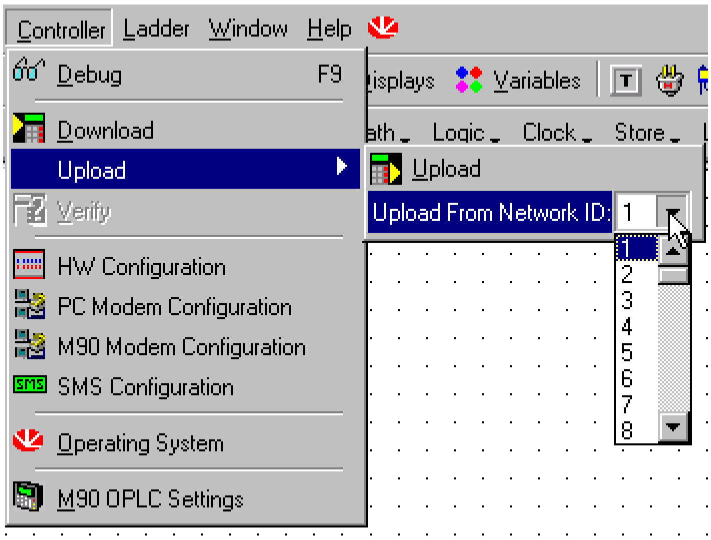
Sending Screens...9

Uploading a Project

1. Select Upload from the Controller menu.
2. Two new options are displayed: Upload, and Upload from Network ID.

3. Upload from:

   - a stand-alone M90 by clicking on the Upload button
   - from a specific M90 on a network by selecting the M90's ID number as shown below.



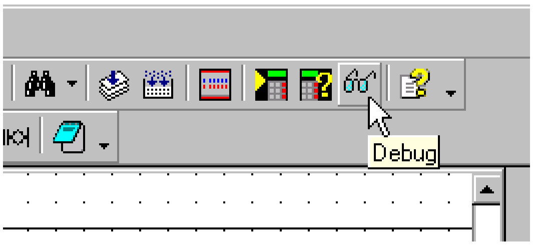
4. All sections of the project in the M90 will upload.

Note that if the program is protected by a password, you must supply this password in order to upload.

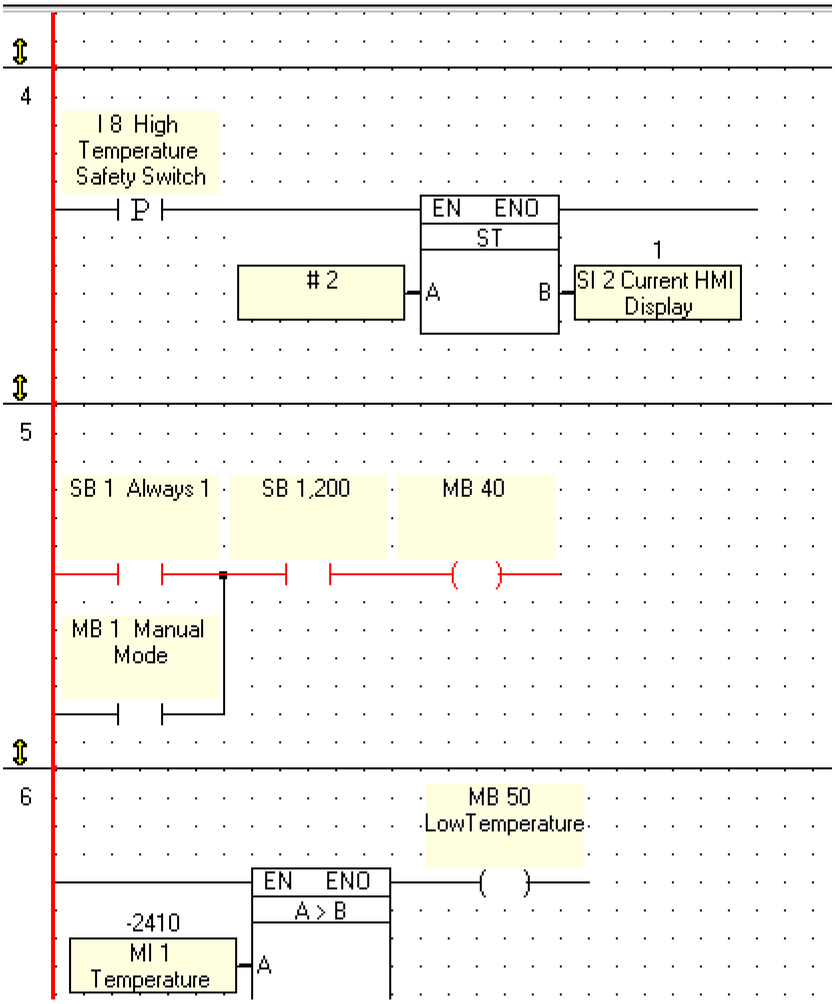
Testing your project (Debug mode)

To test a project:

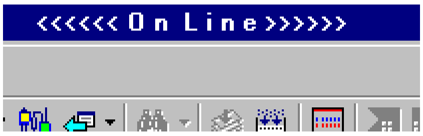
1. Connect the M90 to your PC using the communication cable provided with the software package.
2. Download your program into the M90 from your PC.
3. Click the **Test** icon on the Standard toolbar.



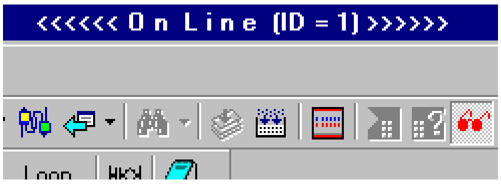
4. The left Ladder bar and any net with Logic flow will appear red. The current values of all MIs and SIs appear above the Operand Symbol.



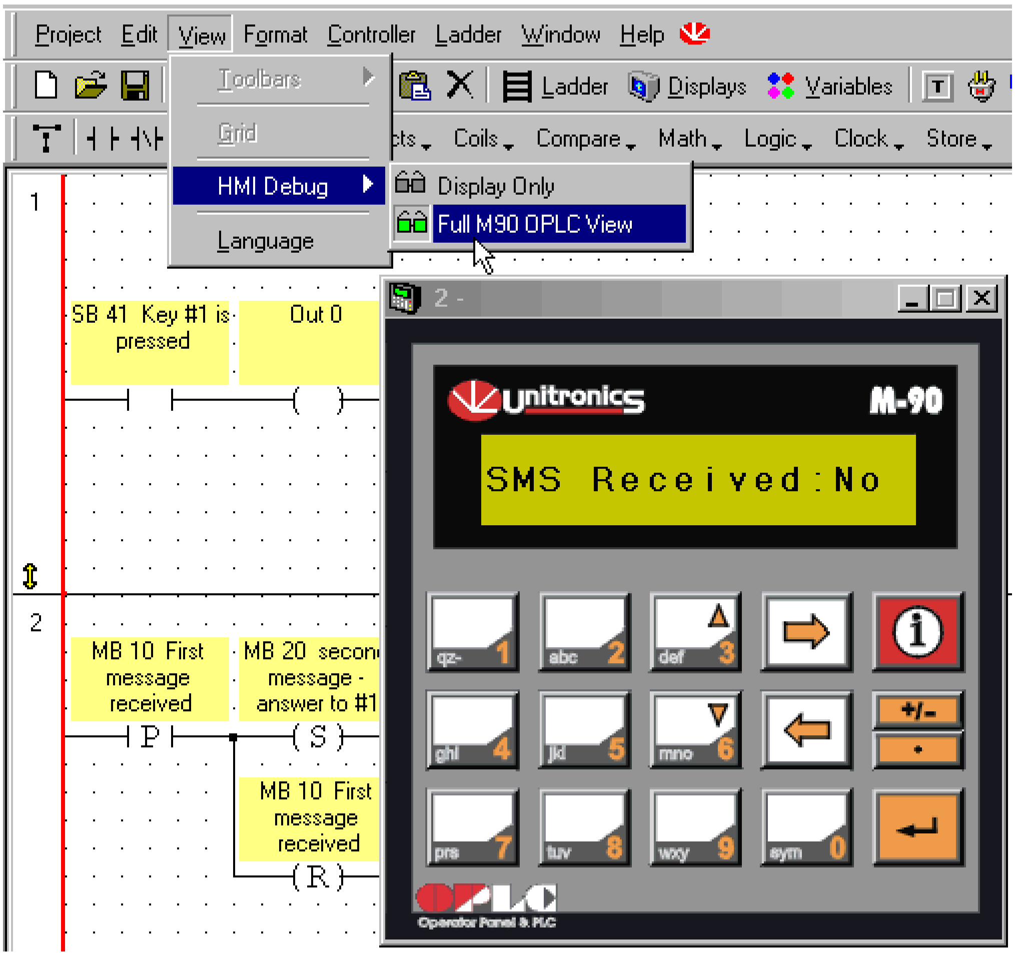
5. During **Test** mode the Title Bar notifies you that you are **On-Line**.


<<<<<< On Line >>>>>>

If you are working in a M90 network, the unit ID number appears as well.


<<<<<< On Line (ID = 1) >>>>>>

You can also view a fully functional, working representation of the M90 OPLC, by selecting Debug HMI from the View menu as shown below. You can choose to see only the current HMI display, or the complete M90, complete with keypad keys. You can test the keypad keys by clicking them, or by using the corresponding numeric keys on your PC's keyboard.


Project Edit View Format Controller Ladder Window Help
Toolbars
Grid
HMI Debug
Language
Display Only
Full M90 OPLC View
Ladder Displays Variables
Coils Compare Math Logic Clock Store
1
SB 41 Key #1 is pressed
Out 0
2
MB 10 First message received
P
MB 20 second message - answer to #1
S
MB 10 First message received
R
2 -
unitronics M-90
SMS Received:No
qz- 1
abc 2
def 3
ghi 4
jkl 5
mno 6
prs 7
tuv 8
wxy 9
sym 0
+/-
OPLC Operator Panel & PLC

Entering values via the M90 keypad

When you enter values from the M90 keypad, you move between entry value options using the **up** and **down** scroll arrow keys on the M90 keypad.

You select a value option with the **+/-** key on the M90 keypad. Each selected value appears with a flashing **+** sign. Pressing **+/-** on an already selected value will unselect that value.

You press **enter** when you have finished entering **all** desired values.

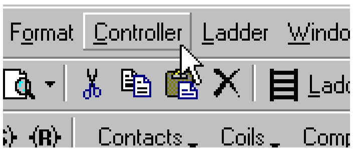
Upgrading the controller's Operating System (OS)

The Operating System runs the controller.

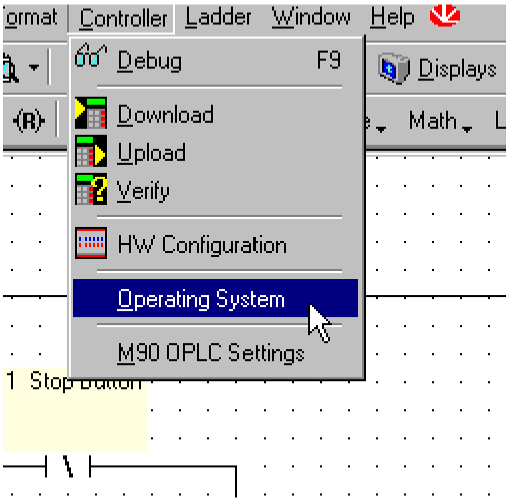
You will be asked to upgrade your OS when there have been manufacturing changes to the hardware and / or software.

To upgrade the Operating System:

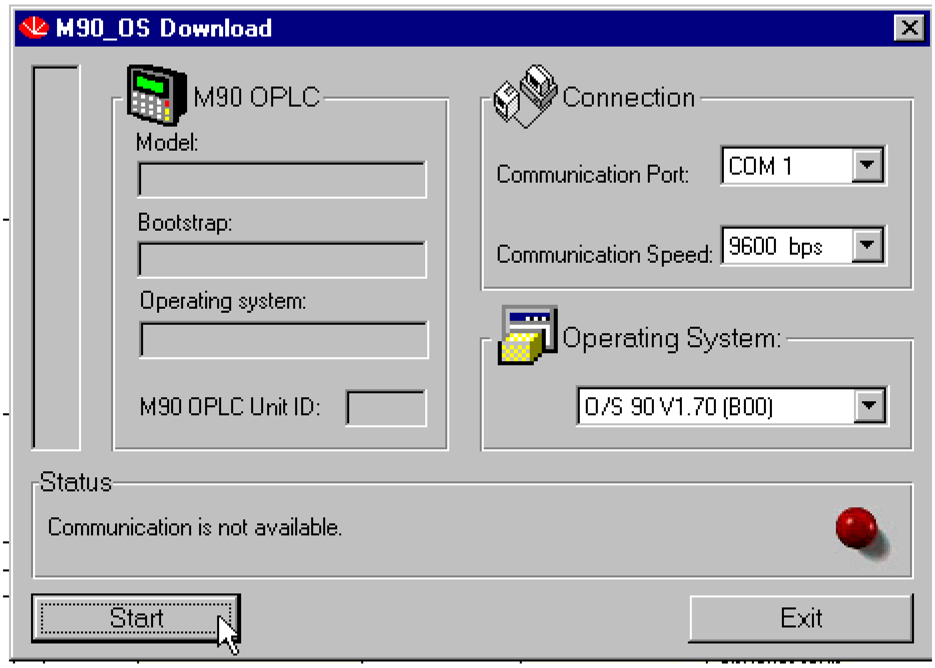
1. Click **Controller** on the Standard menu bar.



2. Select **Operating System** from the drop-down Controller menu.



3. The **M90_OS Download** dialog box opens. Check your communications settings and verify that your controller is connected to your PC. Click **Start**.


M90_OS Download

M90 OPLC

Model:

Bootstrap:

Operating system:

M90 OPLC Unit ID:

Connection

Communication Port: COM 1

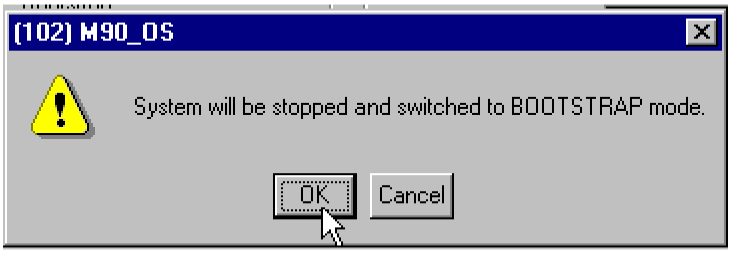
Communication Speed: 9600 bps

Operating System: O/S 90 V1.70 (B00)

Status

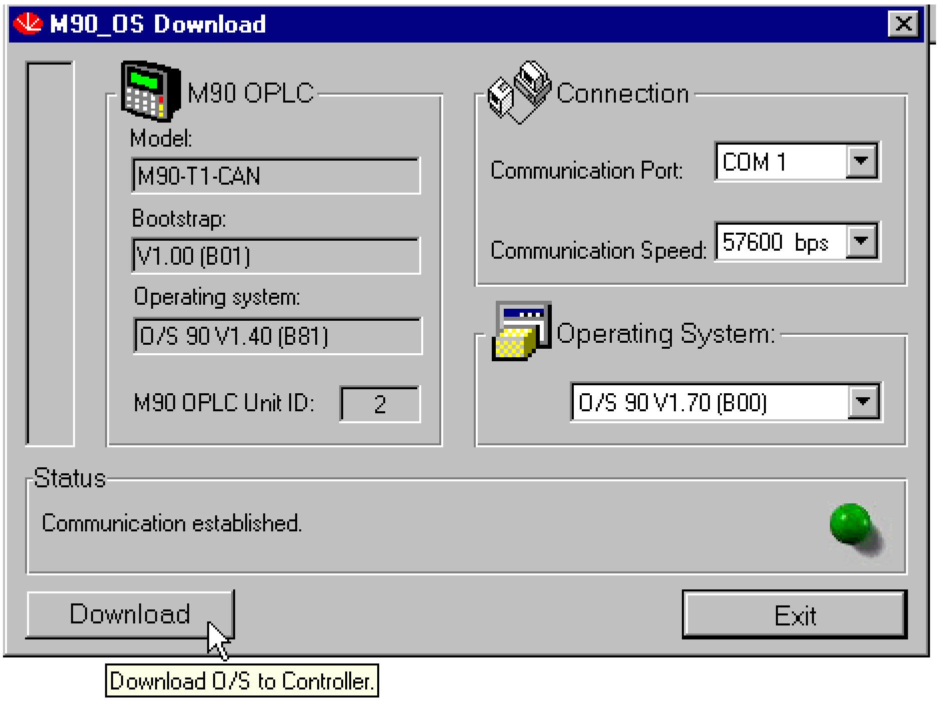
Communication is not available.

4. A message box prompts you to continue with the upgrade. Continuing will **stop all** controller operations.


(102) M90_OS

System will be stopped and switched to BOOTSTRAP mode.

5. Click **Download**.



M90_OS Download

M90 OPLC

Model: M90-T1-CAN

Bootstrap: V1.00 (B01)

Operating system: O/S 90 V1.40 (B81)

M90 OPLC Unit ID: 2

Connection

Communication Port: COM 1

Communication Speed: 57600 bps

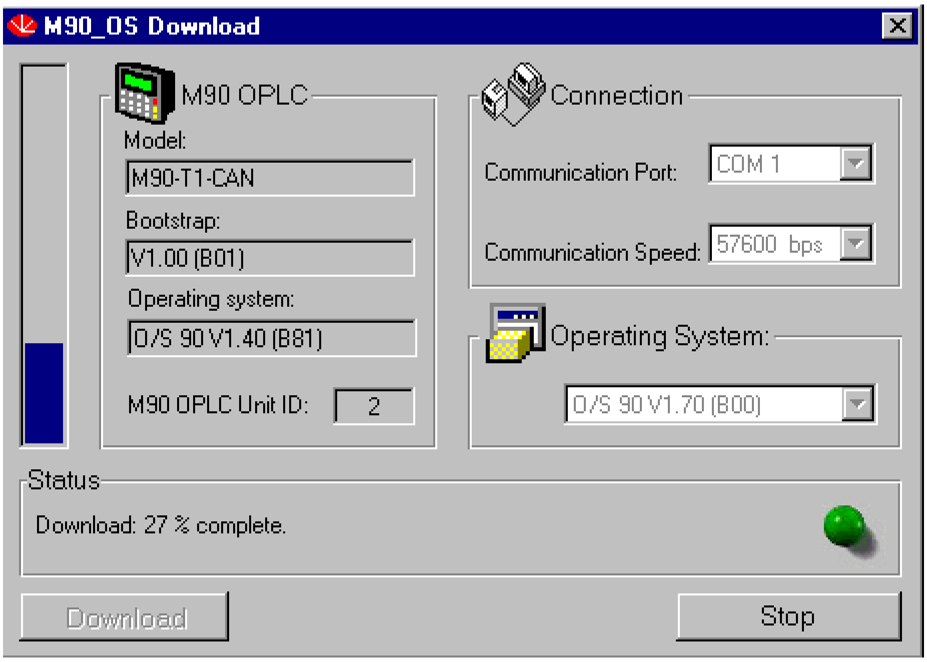
Operating System: O/S 90 V1.70 (B00)

Status

Communication established.

Download O/S to Controller.

6. The Status frame in the M90_OS Download dialog box shows the progress of the process.



M90_OS Download

M90 OPLC

Model: M90-T1-CAN

Bootstrap: V1.00 (B01)

Operating system: O/S 90 V1.40 (B81)

M90 OPLC Unit ID: 2

Connection

Communication Port: COM 1

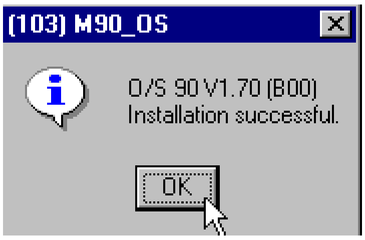
Communication Speed: 57600 bps

Operating System: O/S 90 V1.70 (B00)

Status

Download: 27 % complete.

7. An **Installation successful** message box appears at the end of the download.



(103) M90_OS

O/S 90 V1.70 (B00)
Installation successful.

Convert MB to MI, MI to MB

An M90 register is built of 16 bits.

Using the MB to MI function, you can convert 16 bits or more into a integer value. Conversely, you can convert an integer value into 16 bits or more using the MI to MB function.

Note that if the converted values exceed 16 bits, the function will write the value to consecutive registers. Any values in those registers will be overwritten.

To apply the functions, use the following System Integers (SI) and System Bits (SB)

SI	Description	SB	
SI170	Address of MI containing integer value	SB170	MB to MI
SI171	Start address of MB array (vector)	SB171	MI to MB
SI172	Amount of MBs		

You can use this function, for example to send an SMS when there is a change in the status of the M90's inputs:

1. Represent the status of the M90's inputs using MBs.
2. Convert these MBs into an MI
3. Perform a XOR operation on the result.

When there is a change in input status, the XOR operation will return a value different than 0, which may then be used to trigger the sending of an SMS.

Examples

Example 1:

1. Store the value 7 into SI 170, 10 into SI 171 and 9 into SI 172.
2. Set SB 170 to ON.

The program will calculate the binary value of a 9 bit array which starts with MB 10. The resulting value will be placed into MI 7.

Example 2:

1. Store the value 7 into SI 170, 10 into SI 171 and 9 into SI 172.
2. Set SB 171 to ON

The program will calculate the binary value of the value contained in MI 7. The result will be scattered on a 9 bit array which starts with MB 10.

Detecting short-circuited end devices

The M90 can detect short circuits in end devices (loads) that are connected to **transistor** outputs located on the M90 or on I/O expansion modules. Short circuits can also be detected in end-devices connected to analog outputs. Note that these features do not apply to the transistor outputs located on the M90-T1 and M90-T1-CAN, or the analog output located on the M90-TA2-CAN.

If a short-circuit is detected, SB 5 turns ON.

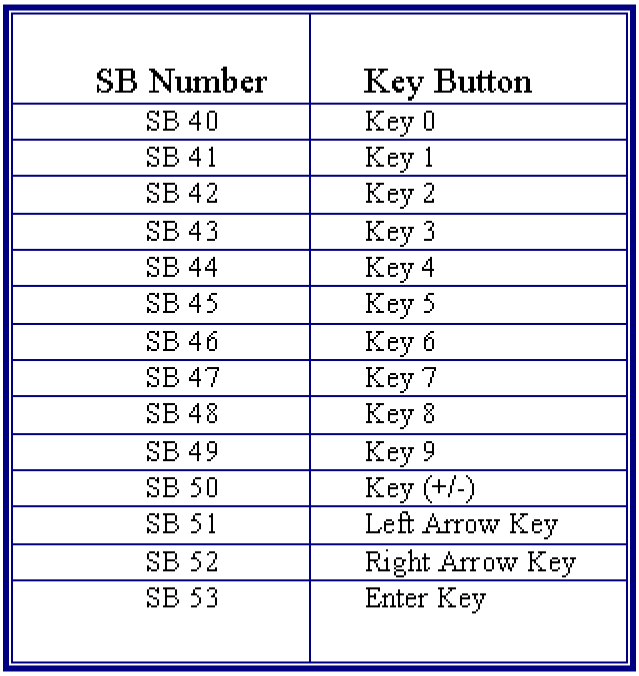
SI 5 contains a bitmap indicating on which device the affected output is located. When you include I/O expansion modules in your M90 hardware configuration, each module is assigned a place number, 0-7, according to its place in that configuration. In the bitmap, bits 0-7 correspond to these place numbers. Bit 8 is reserved for M90. A value of '1' indicates a short-circuited output.

In the bitmap below, short circuits have been detected in devices that are connected to expansion modules 1 and 3, and to the M90 itself.

Bit#	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Output location	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0

Keypad Keys: Linked to SBs 40-53

SB 40 - SB 53 are System Bits reserved for keypad buttons.



SB Number	Key Button
SB 40	Key 0
SB 41	Key 1
SB 42	Key 2
SB 43	Key 3
SB 44	Key 4
SB 45	Key 5
SB 46	Key 6
SB 47	Key 7
SB 48	Key 8
SB 49	Key 9
SB 50	Key (+/-)
SB 51	Left Arrow Key
SB 52	Right Arrow Key
SB 53	Enter Key

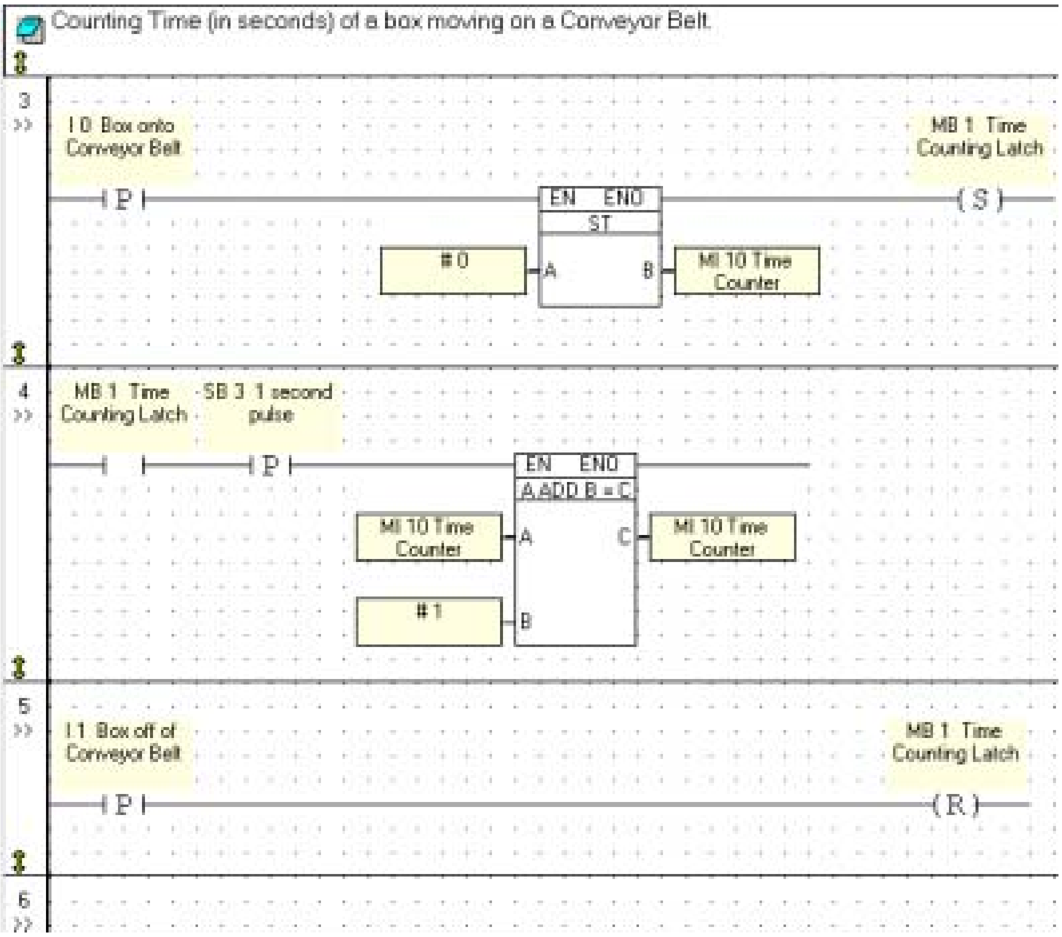
When you push a keypad button, that key button SB goes to logic 1. When you release the button, that key button SB goes to logic 0.

You can use SB 40 - SB 53

- In Jump to Display conditions.
- In Ladder diagram as a Contact.
- As a Bit value Operand.

Measuring time between events

To measure time between two events, you need to create a section in your project that will begin counting time at the start of the first event and continue counting until the second event stops the counter.



According to the above example, when a box is sensed by a sensor connected to Input 0 (I 0), the value 0 (zero) is written into MI 10 (counter) and MB 1 is latched at logic 1 and enables the counter.

Using a one second pulse (SB 3), the counter increments by one every pulse.

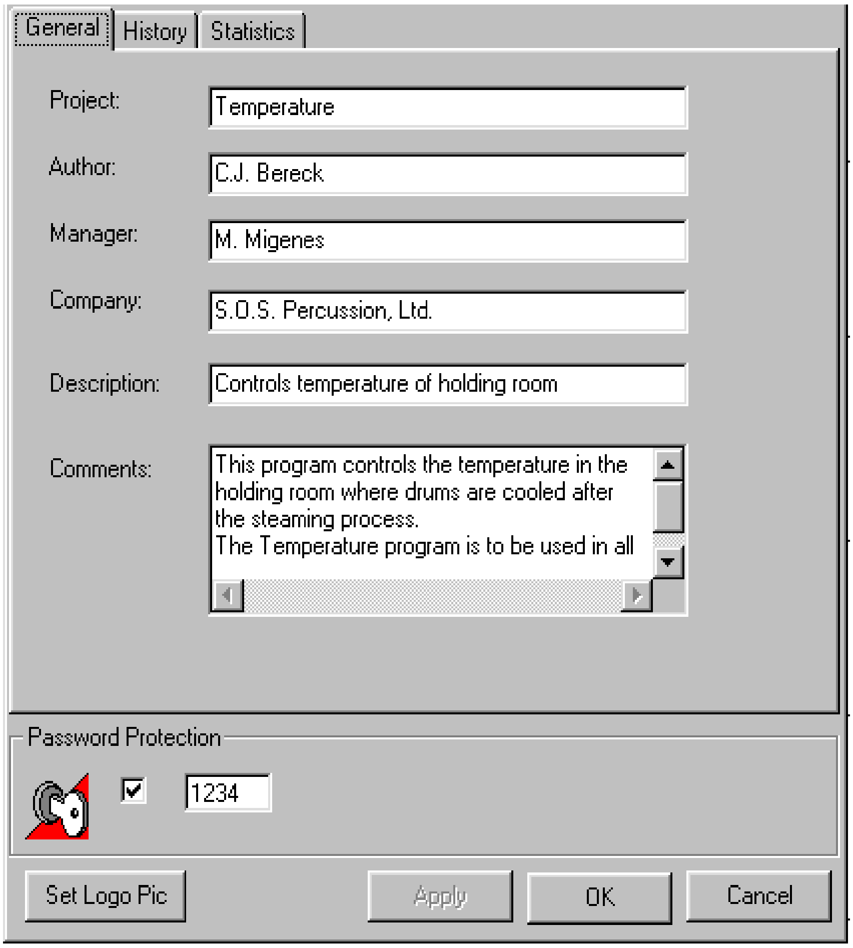
When a box is sensed by the second sensor connected to Input 1 (I 1), MB 1 is reset (unlatched) and the counting stops.

The integer value of MI 10 is the time value that passed between the two events. That is, the amount of time that the box was on the conveyor belt.

Including a logo

You can include a logo in your project. Then, when you print sections of your project, the logo will be printed at the top of the page. Logos can be in .bmp, .gif., jpg., or .jpeg format.

1. Display project Properties by selecting Properties from the Project menu.



Project: Temperature

Author: C.J. Bereck

Manager: M. Migenes

Company: S.O.S. Percussion, Ltd.

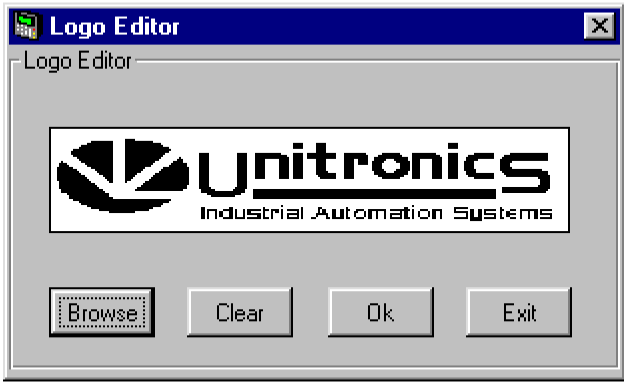
Description: Controls temperature of holding room

Comments: This program controls the temperature in the holding room where drums are cooled after the steaming process.
The Temperature program is to be used in all

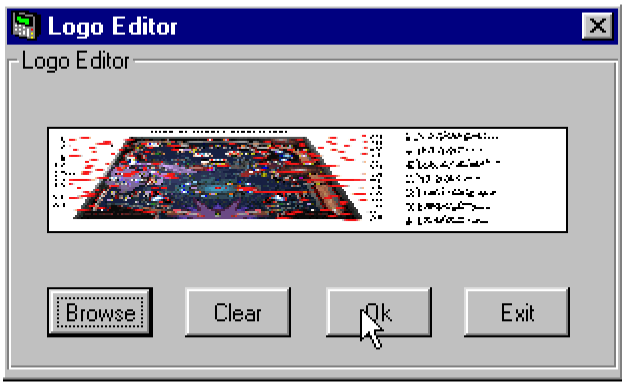
Password Protection

☑ 1234

2. Click Set Logo Pic. The Logo Editor box opens.

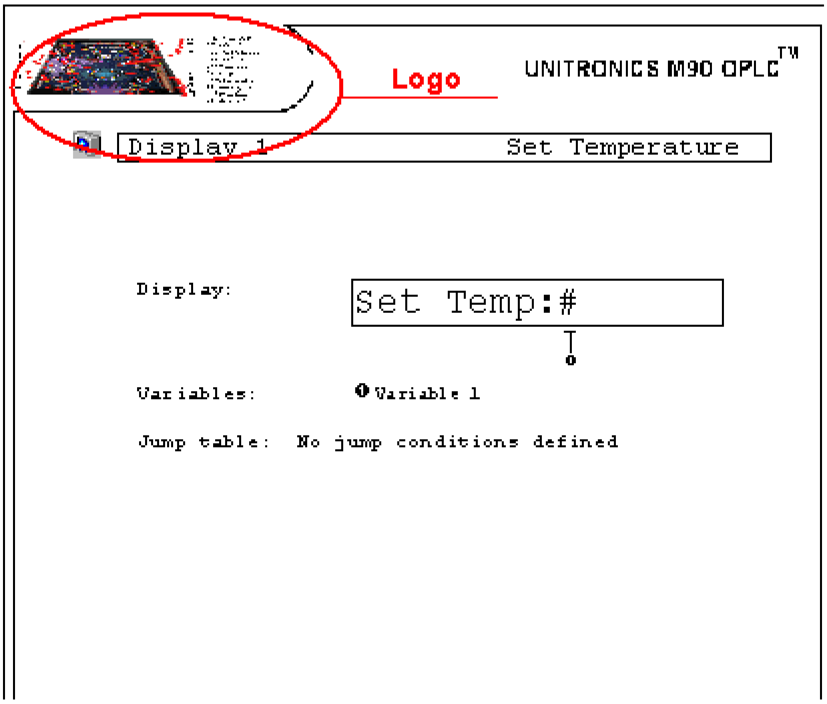


3. Locate your logo by clicking on Browse, navigating to the logo, and selecting it. The new logo now appears in the Logo Editor.



4. Click OK.

When you print a section from your project, the logo will appear as shown below.



Display 1 Set Temperature

Display: Set Temp:#

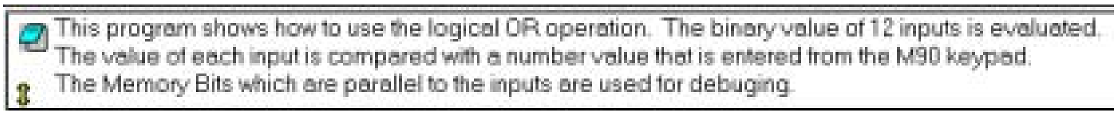
Variables: 0 Variable 1

Jump table: No jump conditions defined

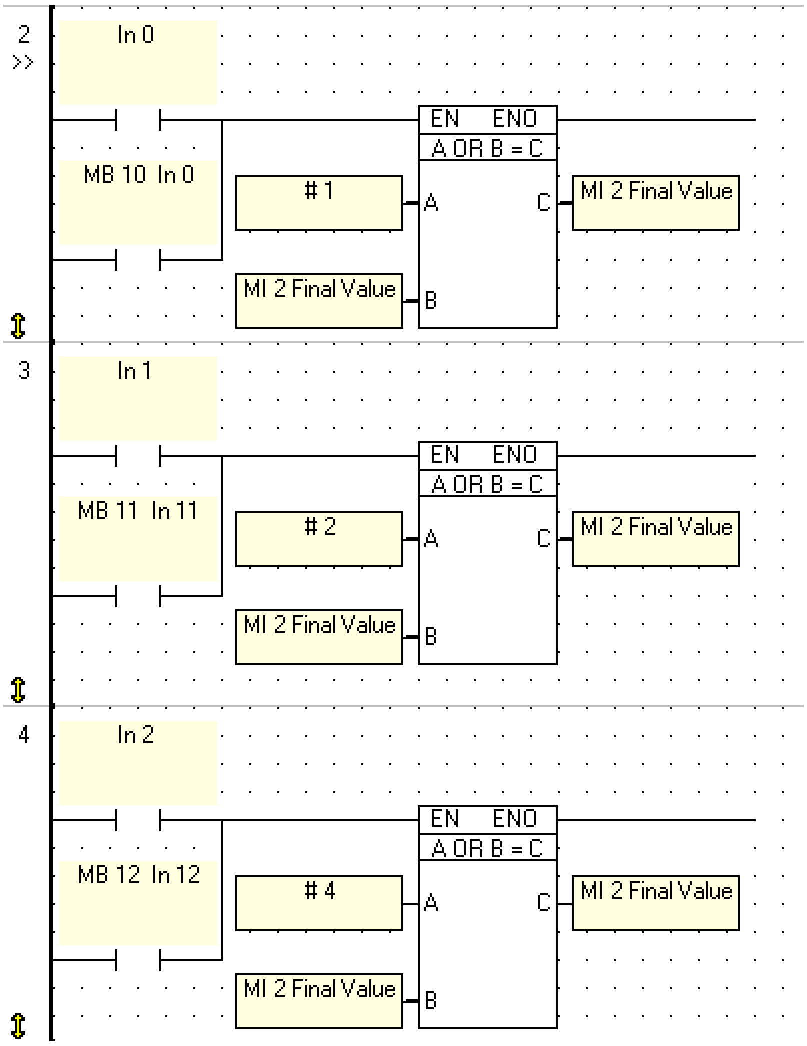
Binary Numbers

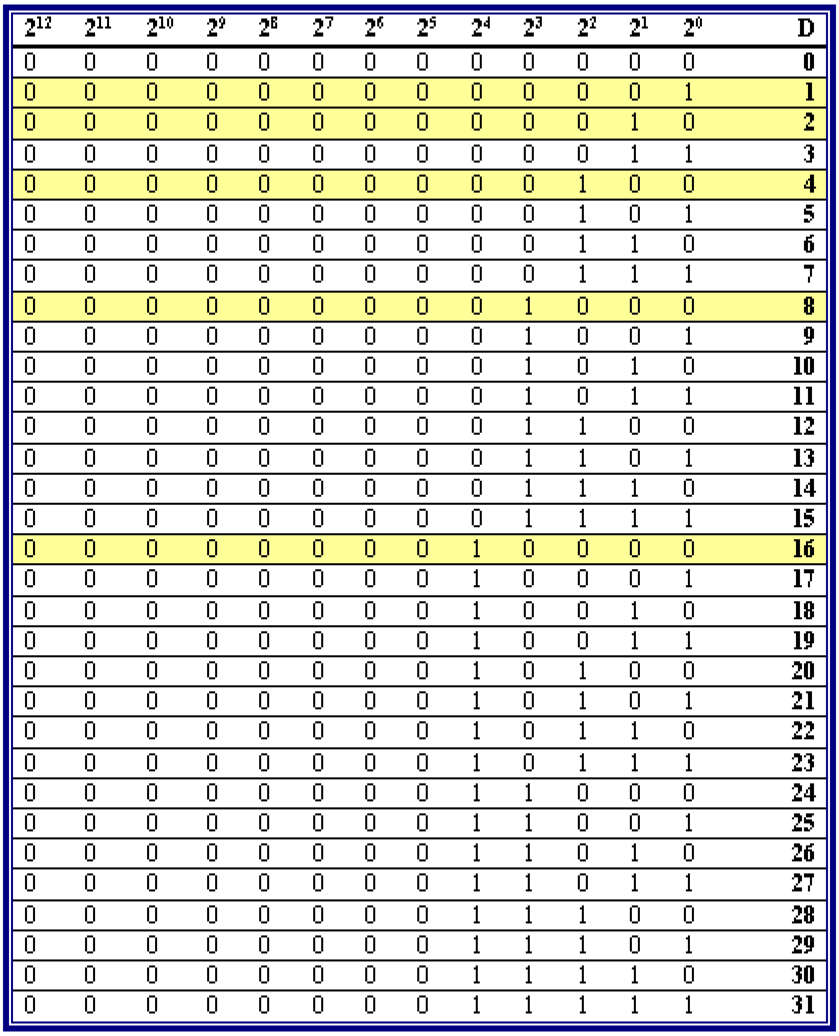
Memory Integers and System Integers are actually 16-bit binary numbers. You enter **decimal** numbers into Memory Integers and System Integers. The program converts these decimal numbers into binary numbers and performs the specified functions.

You may want to use a logic function to mask out bits or check for bit corruption. You will need to know what decimal number will convert to the appropriate binary number. The following charts will help you understand why the decimal numbers {0,1,2,4,8,16,32,64,128, etc} where chosen for use with logical OR to evaluate keypad input numbers in the following example.

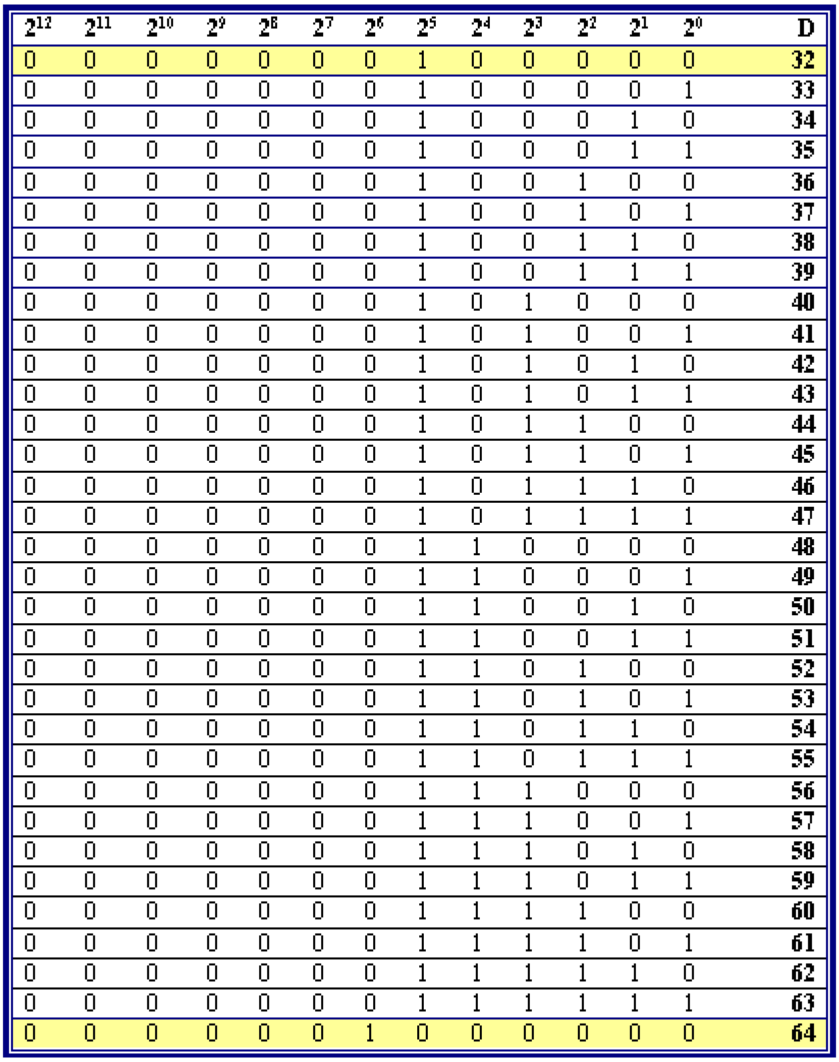


This program shows how to use the logical OR operation. The binary value of 12 inputs is evaluated.
The value of each input is compared with a number value that is entered from the M90 keypad.
The Memory Bits which are parallel to the inputs are used for debuging.


2
>>
In 0
MB 10 In 0
EN ENO
A OR B = C
# 1
A
C
MI 2 Final Value
MI 2 Final Value
B
3
In 1
MB 11 In 11
EN ENO
A OR B = C
# 2
A
C
MI 2 Final Value
MI 2 Final Value
B
4
In 2
MB 12 In 12
EN ENO
A OR B = C
# 4
A
C
MI 2 Final Value
MI 2 Final Value
B



2^{12}	2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	D
0	0	0	0	0	0	0	0	0	0	0	0	0	**0**
0	0	0	0	0	0	0	0	0	0	0	0	1	**1**
0	0	0	0	0	0	0	0	0	0	0	1	0	**2**
0	0	0	0	0	0	0	0	0	0	0	1	1	**3**
0	0	0	0	0	0	0	0	0	0	1	0	0	**4**
0	0	0	0	0	0	0	0	0	0	1	0	1	**5**
0	0	0	0	0	0	0	0	0	0	1	1	0	**6**
0	0	0	0	0	0	0	0	0	0	1	1	1	**7**
0	0	0	0	0	0	0	0	0	1	0	0	0	**8**
0	0	0	0	0	0	0	0	0	1	0	0	1	**9**
0	0	0	0	0	0	0	0	0	1	0	1	0	**10**
0	0	0	0	0	0	0	0	0	1	0	1	1	**11**
0	0	0	0	0	0	0	0	0	1	1	0	0	**12**
0	0	0	0	0	0	0	0	0	1	1	0	1	**13**
0	0	0	0	0	0	0	0	0	1	1	1	0	**14**
0	0	0	0	0	0	0	0	0	1	1	1	1	**15**
0	0	0	0	0	0	0	0	1	0	0	0	0	**16**
0	0	0	0	0	0	0	0	1	0	0	0	1	**17**
0	0	0	0	0	0	0	0	1	0	0	1	0	**18**
0	0	0	0	0	0	0	0	1	0	0	1	1	**19**
0	0	0	0	0	0	0	0	1	0	1	0	0	**20**
0	0	0	0	0	0	0	0	1	0	1	0	1	**21**
0	0	0	0	0	0	0	0	1	0	1	1	0	**22**
0	0	0	0	0	0	0	0	1	0	1	1	1	**23**
0	0	0	0	0	0	0	0	1	1	0	0	0	**24**
0	0	0	0	0	0	0	0	1	1	0	0	1	**25**
0	0	0	0	0	0	0	0	1	1	0	1	0	**26**
0	0	0	0	0	0	0	0	1	1	0	1	1	**27**
0	0	0	0	0	0	0	0	1	1	1	0	0	**28**
0	0	0	0	0	0	0	0	1	1	1	0	1	**29**
0	0	0	0	0	0	0	0	1	1	1	1	0	**30**
0	0	0	0	0	0	0	0	1	1	1	1	1	**31**



2^{12}	2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	D
0	0	0	0	0	0	0	1	0	0	0	0	0	32
0	0	0	0	0	0	0	1	0	0	0	0	1	33
0	0	0	0	0	0	0	1	0	0	0	1	0	34
0	0	0	0	0	0	0	1	0	0	0	1	1	35
0	0	0	0	0	0	0	1	0	0	1	0	0	36
0	0	0	0	0	0	0	1	0	0	1	0	1	37
0	0	0	0	0	0	0	1	0	0	1	1	0	38
0	0	0	0	0	0	0	1	0	0	1	1	1	39
0	0	0	0	0	0	0	1	0	1	0	0	0	40
0	0	0	0	0	0	0	1	0	1	0	0	1	41
0	0	0	0	0	0	0	1	0	1	0	1	0	42
0	0	0	0	0	0	0	1	0	1	0	1	1	43
0	0	0	0	0	0	0	1	0	1	1	0	0	44
0	0	0	0	0	0	0	1	0	1	1	0	1	45
0	0	0	0	0	0	0	1	0	1	1	1	0	46
0	0	0	0	0	0	0	1	0	1	1	1	1	47
0	0	0	0	0	0	0	1	1	0	0	0	0	48
0	0	0	0	0	0	0	1	1	0	0	0	1	49
0	0	0	0	0	0	0	1	1	0	0	1	0	50
0	0	0	0	0	0	0	1	1	0	0	1	1	51
0	0	0	0	0	0	0	1	1	0	1	0	0	52
0	0	0	0	0	0	0	1	1	0	1	0	1	53
0	0	0	0	0	0	0	1	1	0	1	1	0	54
0	0	0	0	0	0	0	1	1	0	1	1	1	55
0	0	0	0	0	0	0	1	1	1	0	0	0	56
0	0	0	0	0	0	0	1	1	1	0	0	1	57
0	0	0	0	0	0	0	1	1	1	0	1	0	58
0	0	0	0	0	0	0	1	1	1	0	1	1	59
0	0	0	0	0	0	0	1	1	1	1	0	0	60
0	0	0	0	0	0	0	1	1	1	1	0	1	61
0	0	0	0	0	0	0	1	1	1	1	1	0	62
0	0	0	0	0	0	0	1	1	1	1	1	1	63
0	0	0	0	0	0	1	0	0	0	0	0	0	64

What is a Unique Number?

Each M90 unit is assigned a unique number when it is manufactured. The unique number is contained in SI 38 and SI 39.

Hardware Configuration

Detecting short-circuited end devices

The M90 can detect short circuits in end devices (loads) that are connected to **transistor** outputs located on the M90 or on I/O expansion modules. Short circuits can also be detected in end-devices connected to analog outputs. Note that these features do not apply to the transistor outputs located on the M90-T1 and M90-T1-CAN, or the analog output located on the M90-TA2-CAN.

If a short-circuit is detected, SB 5 turns ON.

SI 5 contains a bitmap indicating on which device the affected output is located. When you include I/O expansion modules in your M90 hardware configuration, each module is assigned a place number, 0-7, according to its place in that configuration. In the bitmap, bits 0-7 correspond to these place numbers. Bit 8 is reserved for M90. A value of '1' indicates a short-circuited output.

In the bitmap below, short circuits have been detected in devices that are connected to expansion modules 1 and 3, and to the M90 itself.

Bit#	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Output location	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0

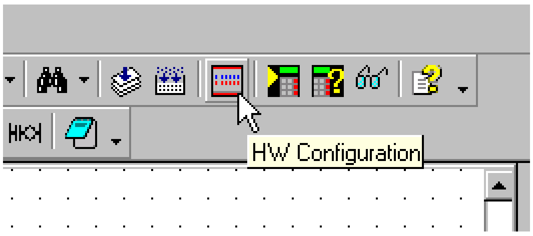
Configuring I/O Expansion Modules

Certain M90 models can be hooked up to I/O Expansion Modules.

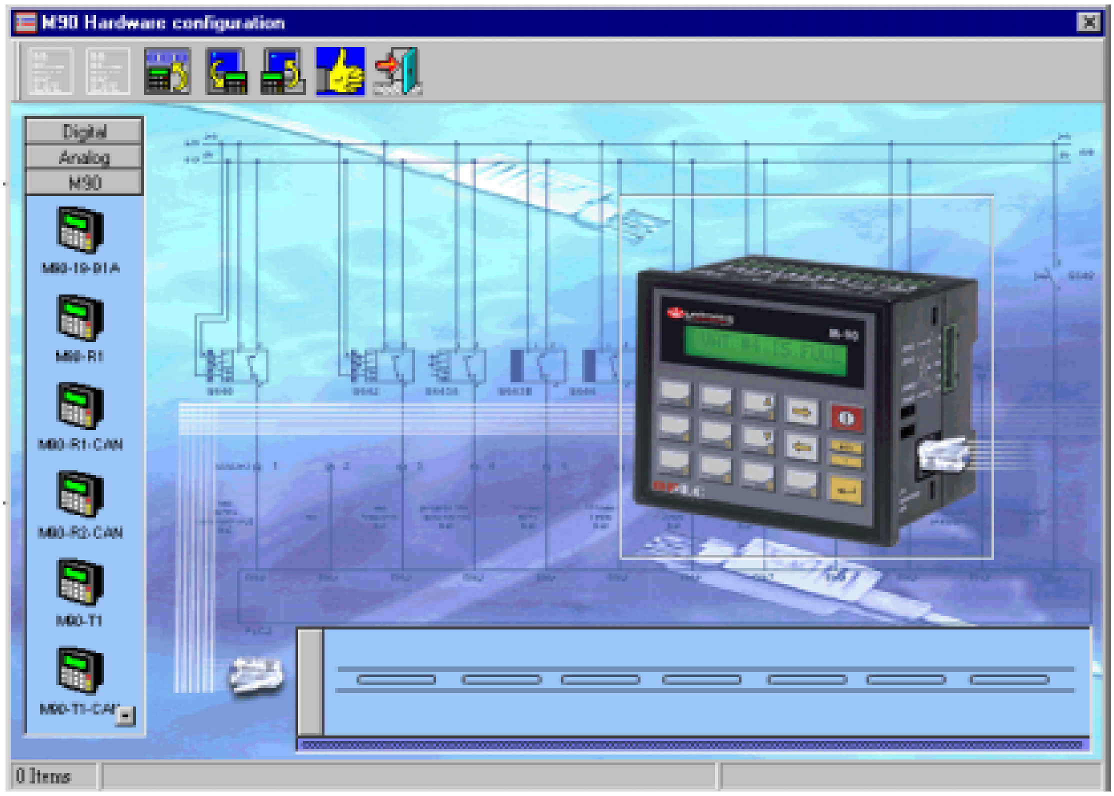
You must configure the M90 according to the I/O Expansion Modules you are connecting.

Adding I/O Expansion Modules to your Hardware Configuration

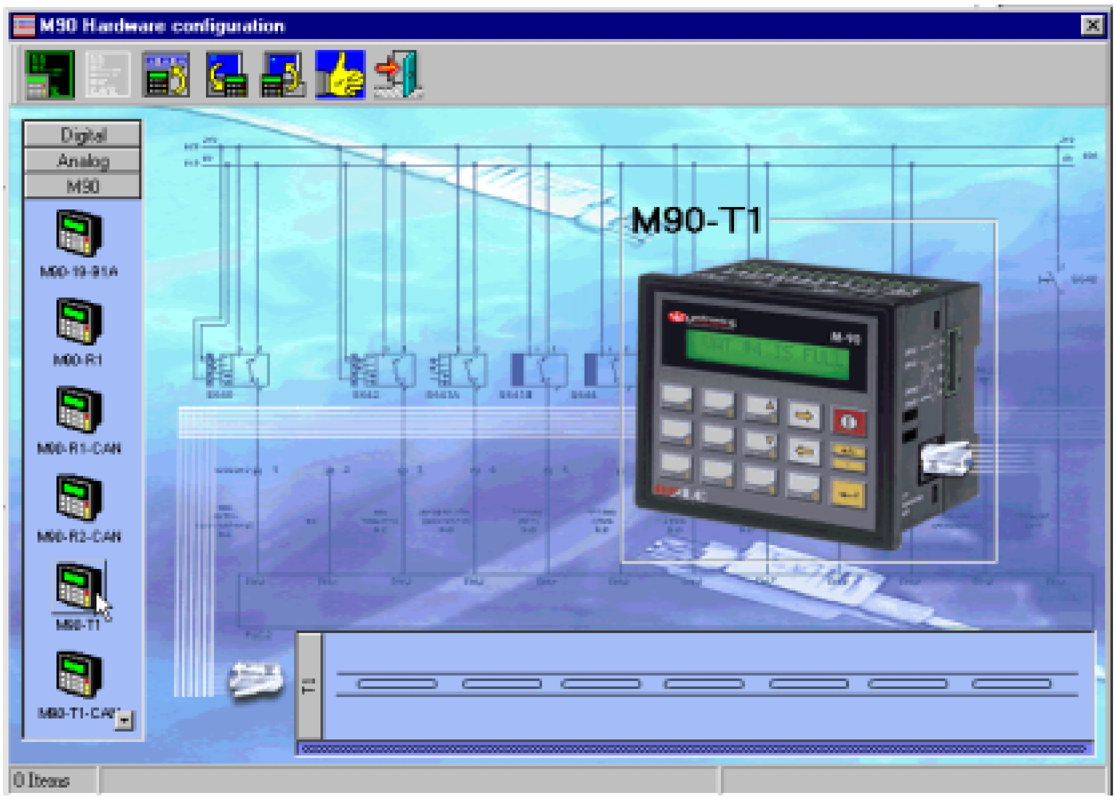
1. Click on the **Hardware Configuration** icon on the Standard toolbar.



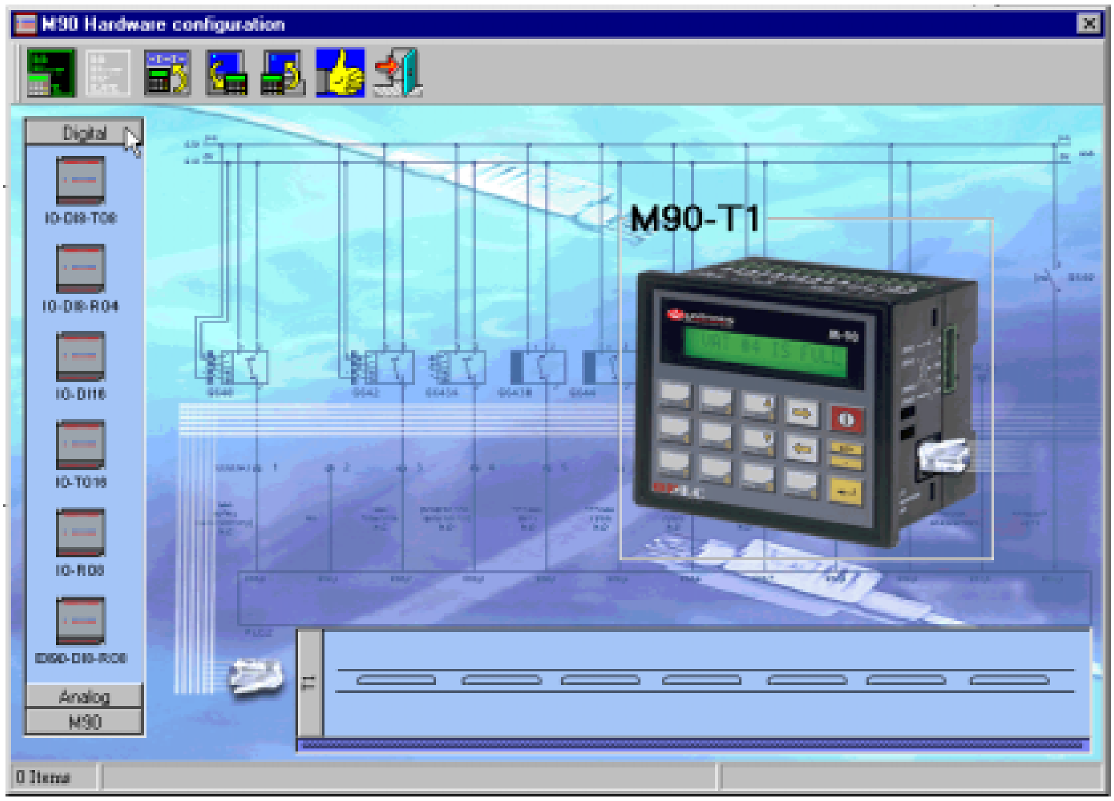
2. The **M90 Hardware configuration** window opens.



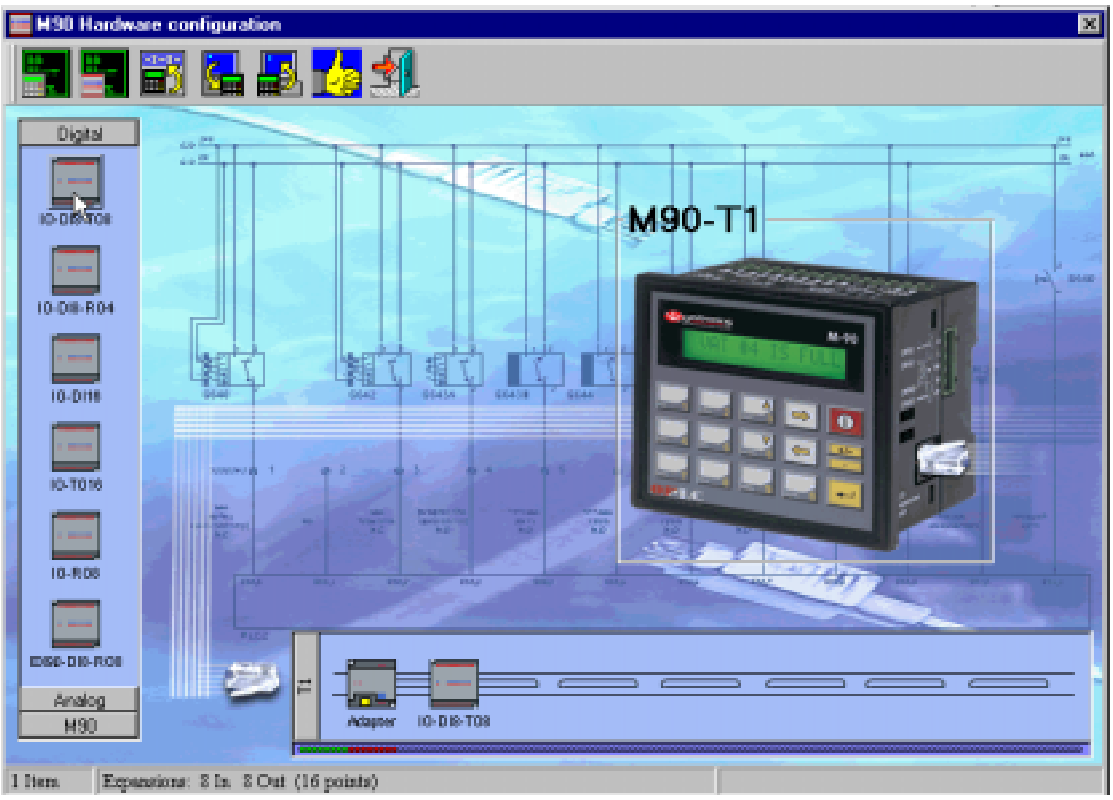
3. Select the M90 model for your project application from the M90 icon menu.



4. The selected model name appears above the M90 controller. Open the Digital or Analog menu according to the module you are connecting.

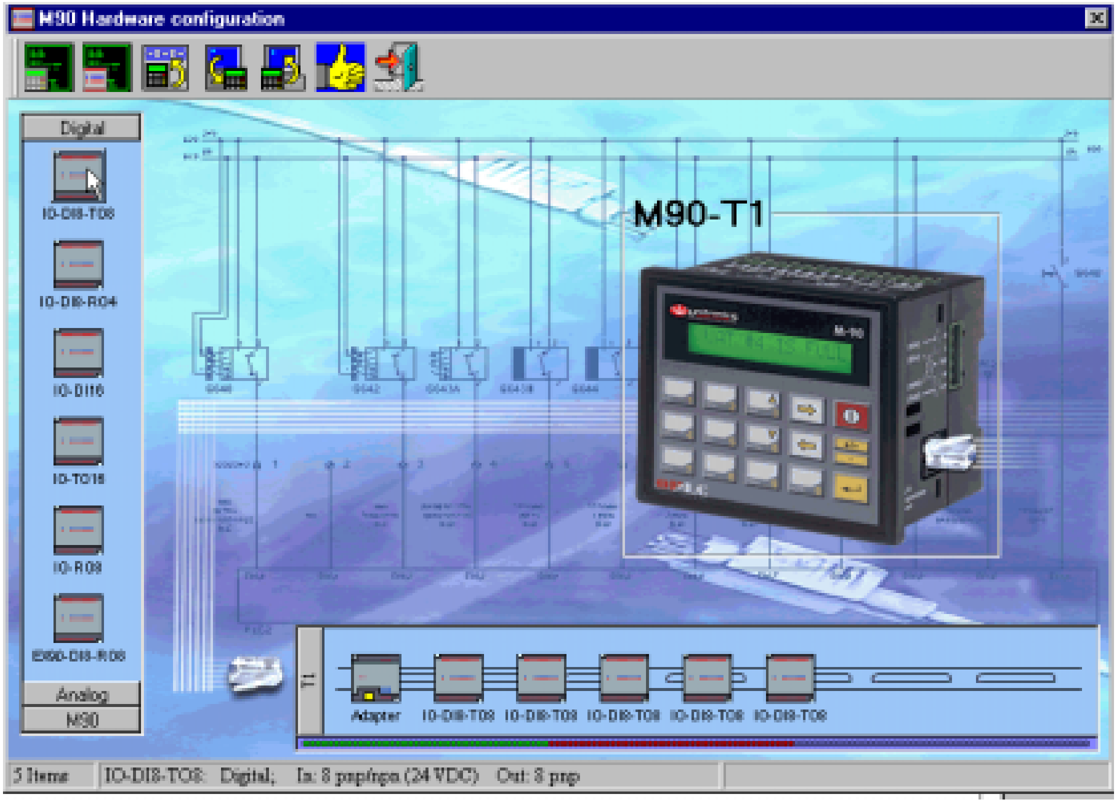


5. Double-click on the appropriate I/O module. The selected module(s) will appear on the Module Expansion bar.



1 Item | Expansions: 8 In 8 Out (16 points)

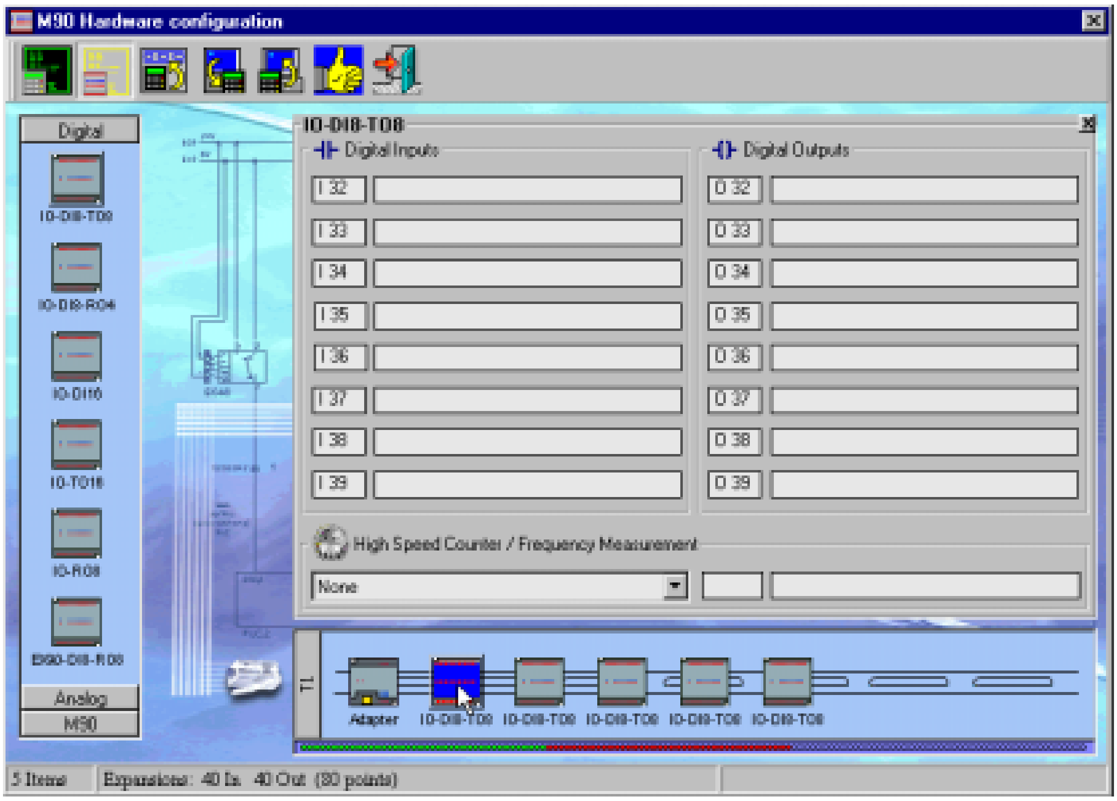
6. Continue adding I/O Expansion Modules according to your expansion configuration.



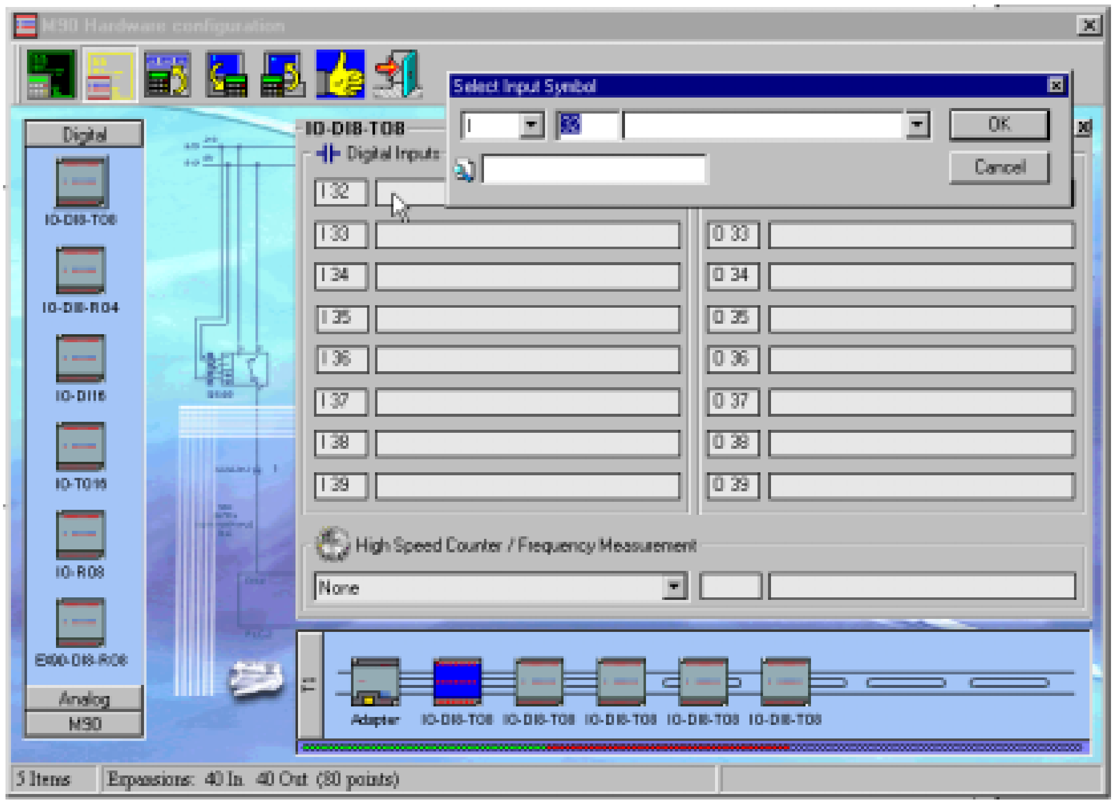
5 Items | IO-DI8-TO8: Digital; In: 8 pnp/npn (24 VDC) Out: 8 pnp

Configuring I/Os: Linking Operands

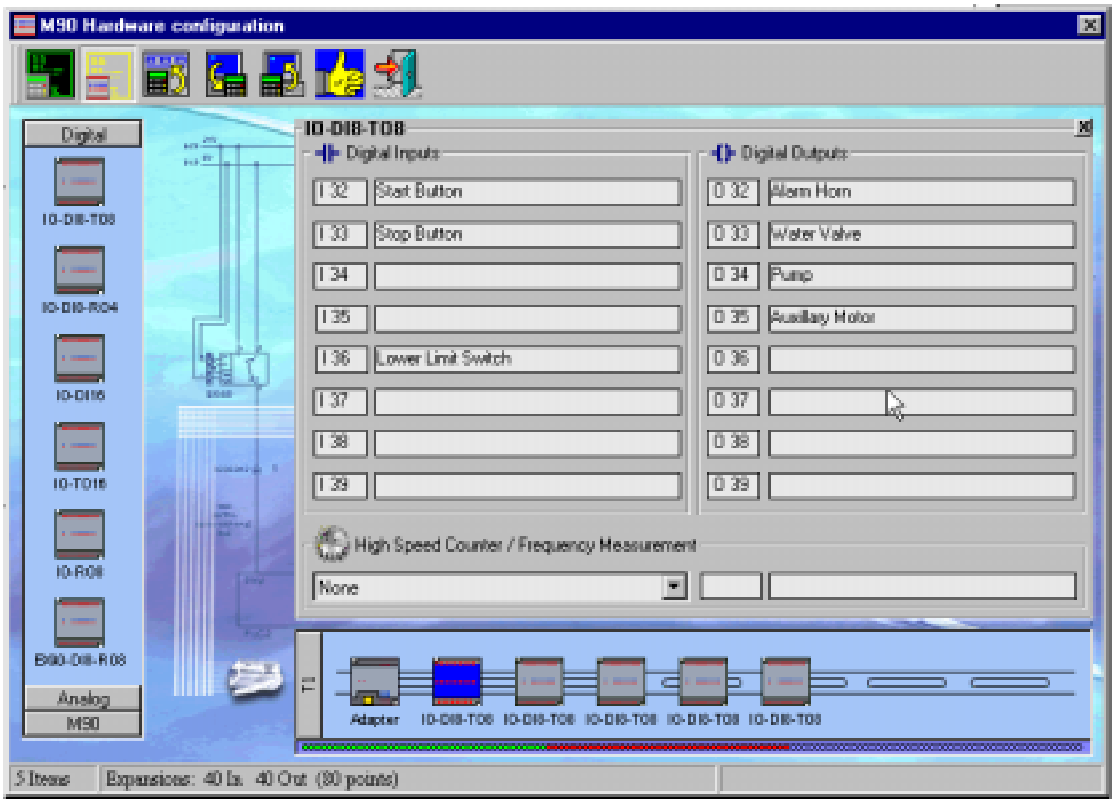
1. Double-click on an I/O expansion icon in the Model Expansion bar. An I/O Details window opens.



2. Click on the appropriate Inputs / Outputs to enter the desired Addresses and Symbols.



3. The Addresses and Symbols appear in the I/O Details window.



M90 Hardware configuration

IO-DI8-TO8

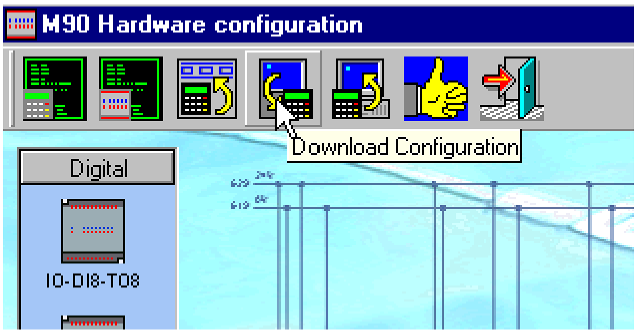
Digital Inputs		Digital Outputs	
I 32	Start Button	O 32	Alarm Horn
I 33	Stop Button	O 33	Water Valve
I 34		O 34	Pump
I 35		O 35	Auxiliary Motor
I 36	Lower Limit Switch	O 36	
I 37		O 37	
I 38		O 38	
I 39		O 39	

High Speed Counter / Frequency Measurement: None

5 Items | Expansions: 40 In 40 Out (80 points)

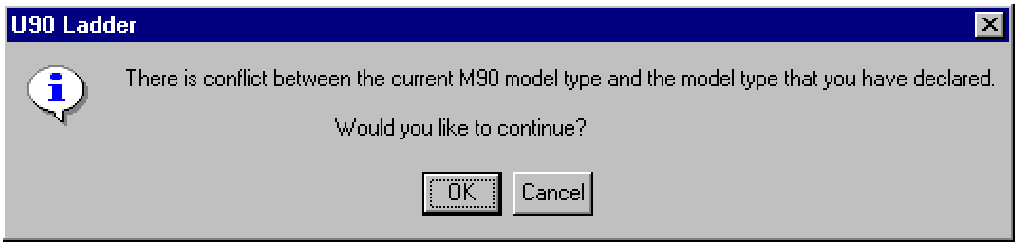
Downloading Hardware Configuration properties

1. Click the Download Configuration icon.



M90 Hardware configuration

2. If there is a conflict between the current M90 hardware information and the project configuration, you will be prompted to choose how to proceed.

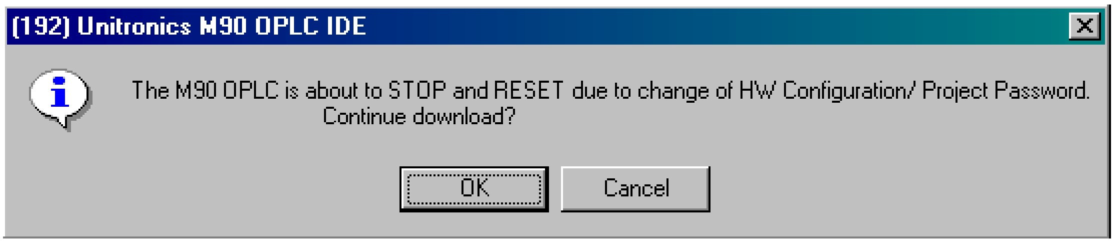


U90 Ladder

There is conflict between the current M90 model type and the model type that you have declared.

Would you like to continue?

3. If you decide to continue with the Download, the M90 OPLC will be stopped and reset during the Download procedure. Click OK. The Download process is activated.



(192) Unitronics M90 OPLC IDE

The M90 OPLC is about to STOP and RESET due to change of HW Configuration/ Project Password.
Continue download?

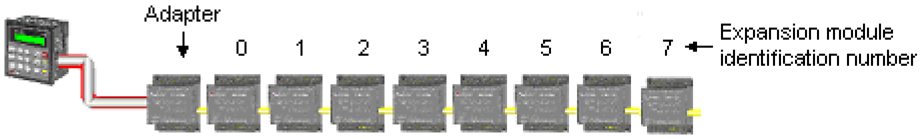
The Hardware configuration is now updated.

Note: If your application does not require that you use all of the I/Os on a particular I/O Expansion Module, do not select the unused I/Os when you configure the module. Selecting unused I/Os may add to the M90's scan time.

Addressing: I/O Expansion Modules

Inputs and outputs located on I/O expansion modules that are connected into an M90 OPLC are assigned addresses that comprise a letter and a number. The letter indicates whether the I/O is an input (I) or an output (O). The number indicates the I/O's location in the system. This number relates to both the expansion module's position in the system, and to the position of the I/O on that module.

Expansion modules are numbered from 0-7 as shown in the figure below.



The formula below is used to assign addresses for I/O modules used in conjunction with the M90 OPLC.

X is the number representing a specific module's location (0-7). Y is the number of the input or output on that specific module (0-15).

The number that represents the I/O's location is equal to: 32 + x • 16 + y

Example

- Input #3, located on expansion module #2 in the system, will be addressed as I 67, 67 = 32 + 2 • 16 + 3
- Output #4, located on expansion module #3 in the system, will be addressed as O 84, 84 = 32 + 3 • 16 + 4.

EX90-DI8-RO8 is a stand-alone I/O module. Even if it is the only module in the configuration, the EX90-DI8-RO8 is always assigned the number 7. Its I/Os are addressed accordingly.

Example

- Input #5, located on an EX90-DI8-RO8 connected to an M90 OPLC will be addressed as I 149, 149 = 32 + 7 • 16 + 5

High-Speed Counters (HSC), Shaft Encoders, Frequency Measurer

The M90 series offers high-speed counter functions of the following types:

- Shaft encoder, at resolutions x2 and x4.
- High-speed counter.
- High-speed counter + reset,
- Frequency measurement, at 100, 500, and 1000 msec.

Some of the sample programs installed together with U90 Ladder include high-speed counters of different types.

HSC Types & Functions

High-speed counter functions are built into the M90 hardware. This is why you do not 'build' a high-speed counter within your Ladder program. Instead, you define it as part of the M90 OPLC's hardware configuration by:

1. Selecting the counter type as shown below
2. Linking it to an MI that contains the counter value.

Note that the counter value is an integer with a range of -32768 to +32767. After the counter reaches the maximum value of +3,2767 it will continue to count in the negative range.

The last on-board input on an M90 is the actual counter, and is capable of counting 5,000 pulses per second. Note that the M90 high-speed input is a pnp-type input, requiring a nominal voltage of 24V, a minimum of 15V.

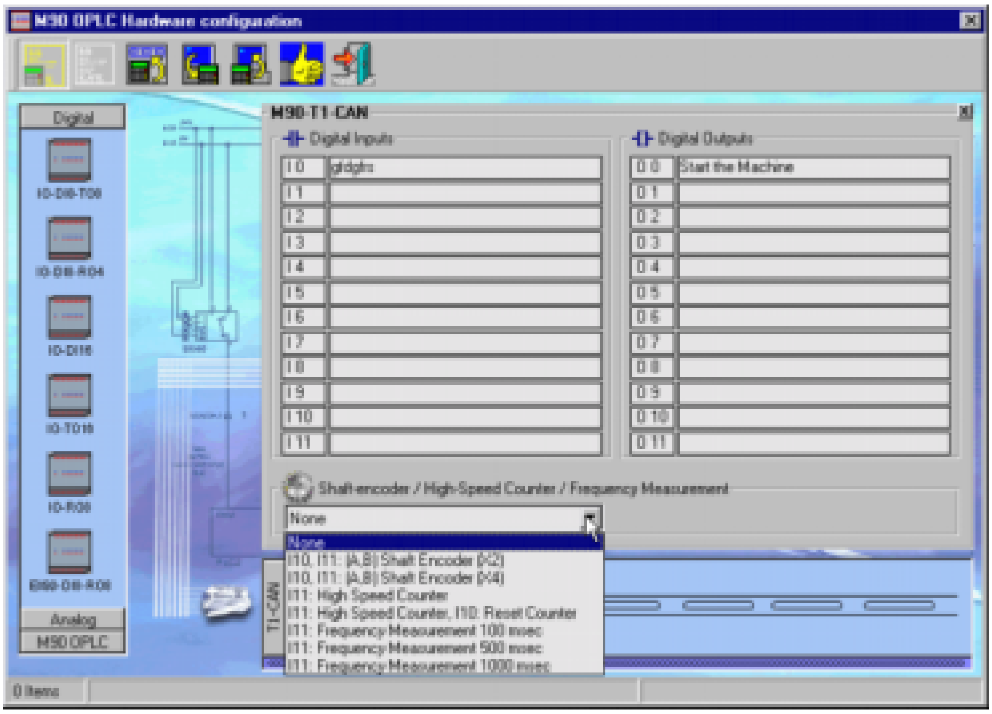
The next-to-last input also serves a purpose in certain high-speed counter functions:

- Shaft encoder function: the next-to-last input serves to indicate the direction of the encoder.
- High-speed counter + reset function: the next-to-last input serves to reset the counter.

When the next-to-last input is used in a high-speed counter function, it is normally OFF. It remains OFF until it receives a signal; the input then turns ON, stopping and resetting the high-speed counter. The high-speed counter begins counting pulses only after the counter reset turns OFF. Note that SB 10 High Speed Counter Reset Enable must be ON; it is ON by default.

Configuring a High-speed counter

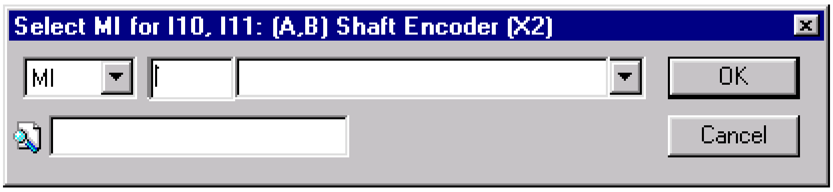
1. Select Hardware Configuration from the Controller menu. The Hardware Configuration window opens.


M90 OPLC Hardware configuration
Digital
IO-DI8-TO8
IO-DI8-RO4
IO-DI16
IO-TO16
IO-RO8
EX90-DI8-RO8
Analog
M90 OPLC
M90-T1-CAN
Digital Inputs
Digital Outputs
Shaft-encoder / High-Speed Counter / Frequency Measurement
None
I10, I11: [A,B] Shaft Encoder (X2)
I10, I11: [A,B] Shaft Encoder (X4)
I11: High Speed Counter
I11: High Speed Counter, I10: Reset Counter
I11: Frequency Measurement 100 msec
I11: Frequency Measurement 500 msec
I11: Frequency Measurement 1000 msec
T1-CAN
0 Items

Digital Inputs		Digital Outputs	
I 0	gtdghs	O 0	Start the Machine
I 1		O 1	
I 2		O 2	
I 3		O 3	
I 4		O 4	
I 5		O 5	
I 6		O 6	
I 7		O 7	
I 8		O 8	
I 9		O 9	
I 10		O 10	
I 11		O 11	

Shaft-encoder / High-Speed Counter / Frequency Measurement: None

2. Click on the icon representing your controller model. The appropriate hardware model window opens.
3. Select a high-speed counter type by clicking the drop-down arrow to display the options, then clicking one.
4. The Select Operand Address box opens. Select an MI to contain the counter value, and then click OK.



This MI contains the counter value which is current at the last program scan. Use this MI in your program like any other MI. You can reset the counter by placing a 0 value into this MI via the Store function. Note that in order to reset the counter, SB 10 High Speed Counter Reset Enable must be turned ON; SB 10 is ON by default.

Shaft Encoder

Selecting the shaft encoder function enables the counter to count both up (-3, -2, -1, 0, 1, 2, 3, …) and down (3, 2, 1, 0, -1, -2,-3 …). Note that the input requires you to use pnp-type shaft encoders.

High-speed Counter

If you select the high-speed counter function that does not include Reset, note that you must reset it within your Ladder program. This type of counter only counts up.

If you select the high-speed counter function with reset, the counter is capable of counting up within the positive range, 0-32767. This function uses the next-to-last input as a counter reset. Since the reset is done via the hardware, the reset is immediate and independent of the program scan.

Frequency Measurement

This counts the number of pulses over the selected period of time (sample rate): 100 msec, 500 msec, or 1000 msec (1 second), expressing the result in Hertz. For example, 155 pulses counted over 100 msec is equal to 1550Hz; 155 pulses counted over 500 msec is equal to 310Hz.

Compare Functions and Counter Values

It is probable that a counter value will not be read at the exact moment that a Compare function in your program is being carried out. This can cause an Equal (=) function to miss the desired counter value; if the counter does not reach the value required by the Equal function at the moment the function is carried out, the Equal function cannot register that the value has been reached. To avoid this problem, use functions Greater Than Or Equal To (≥) and Lesser Than Or Equal To (≤).

High-Speed Output: PWM

M90

M90 OS versions 2.00 (B01) and later enable you to use the last on-board output of M90 models T1 and T1-CAN in either:

- High Speed Output (HSO) mode
- Normal output mode.

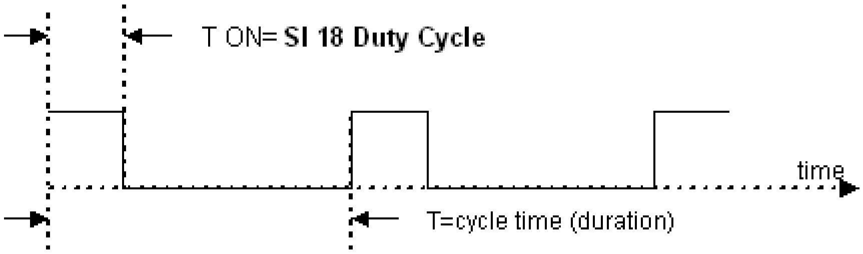
Using HSO mode gives you the ability to use an output as a PWM (Pulse Width Modulation) output. You can also use an output in HSO mode together with stepper motor controllers.

To use HSO mode:

1. Use System Integer SI 16 HSO Mode to change the operating mode of Output 11 from Normal mode to HSO mode: 0=Normal Mode, 1: HSO Mode.
   This should be part of your program's Power-up tasks.
2. Set the output frequency (F) by storing a value into SI 17 HSO Frequency.
   Note that F=1/T, where T is the duration time of a complete cycle.
   You can store a value of 0, or a value from 3-1500Hz; other frequency values are not supported.
3. Set the duty cycle—the ratio of the "on" period of a cycle to the total cycle period—by storing a value into SI 18 Duty Cycle. This value may be from 0-1000, and is expressed as a percentage.

   If, for example, the constant 750 is stored into SI 18, the duty cycle is equal to 75.0%. This means that the pulse will hold a positive state during 75.0% of the total cycle.
4. Use SB 16 HSO RUN to control the output; when SB 16 is ON, Output 11 operates.

In the figure below, SI 18 is equal to 250. This results in the duty cycle being 25% of the total cycle time.

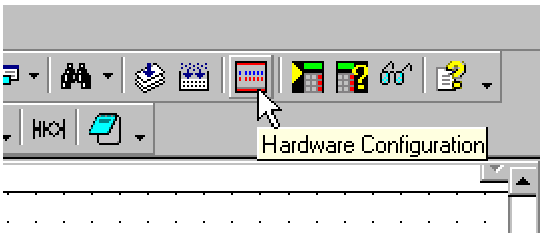


Note that:

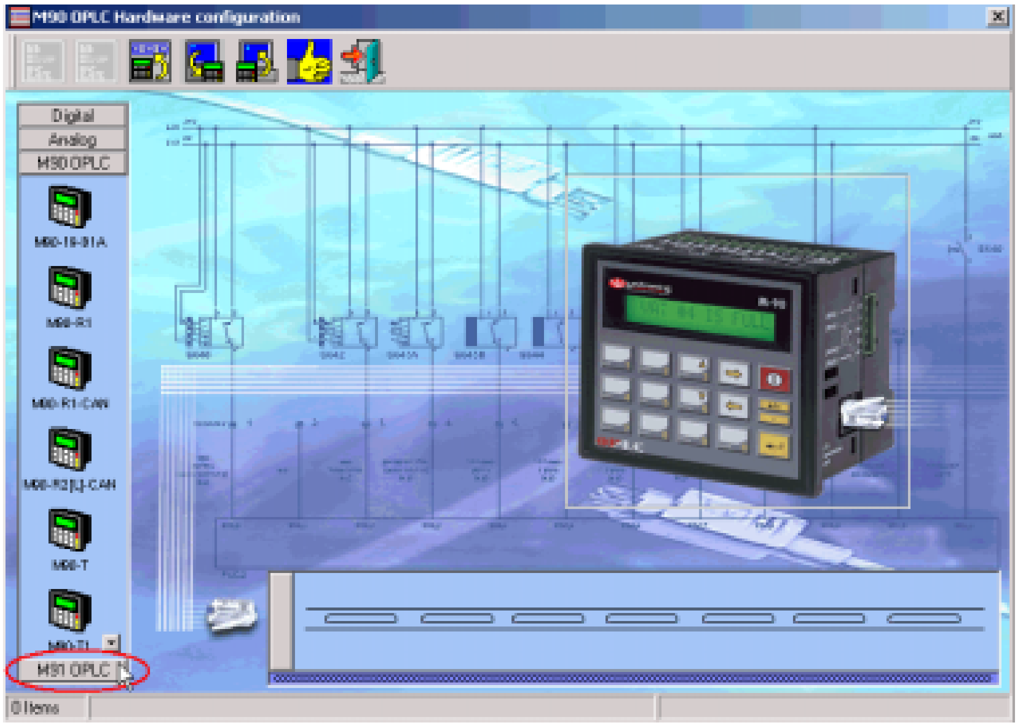
- If you store out-of-range values into SI 17 and SI 18, their values remain unchanged—they retain the last legal values stored.
- Note 2. All parameters except SI 16 may be changed during run-time.

M91

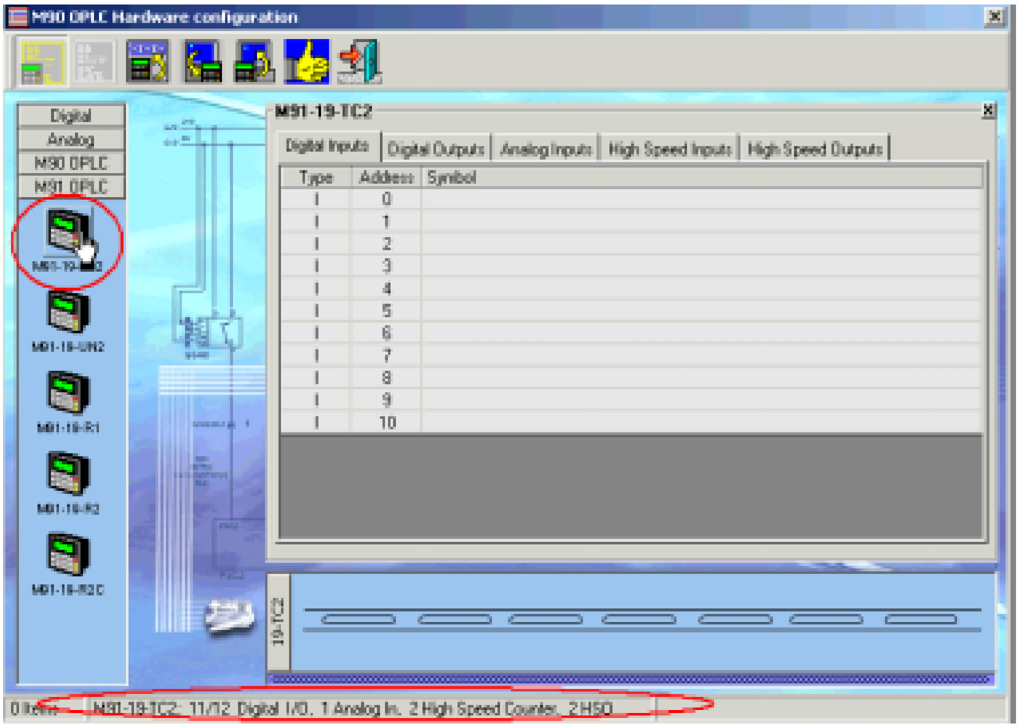
1. Click Hardware Configuration on the Standard toolbar.



2. The M90 Hardware Configuration window opens.
3. Click on the M91 bar.



4. Select the appropriate M91 model; the model's I/O options are displayed.

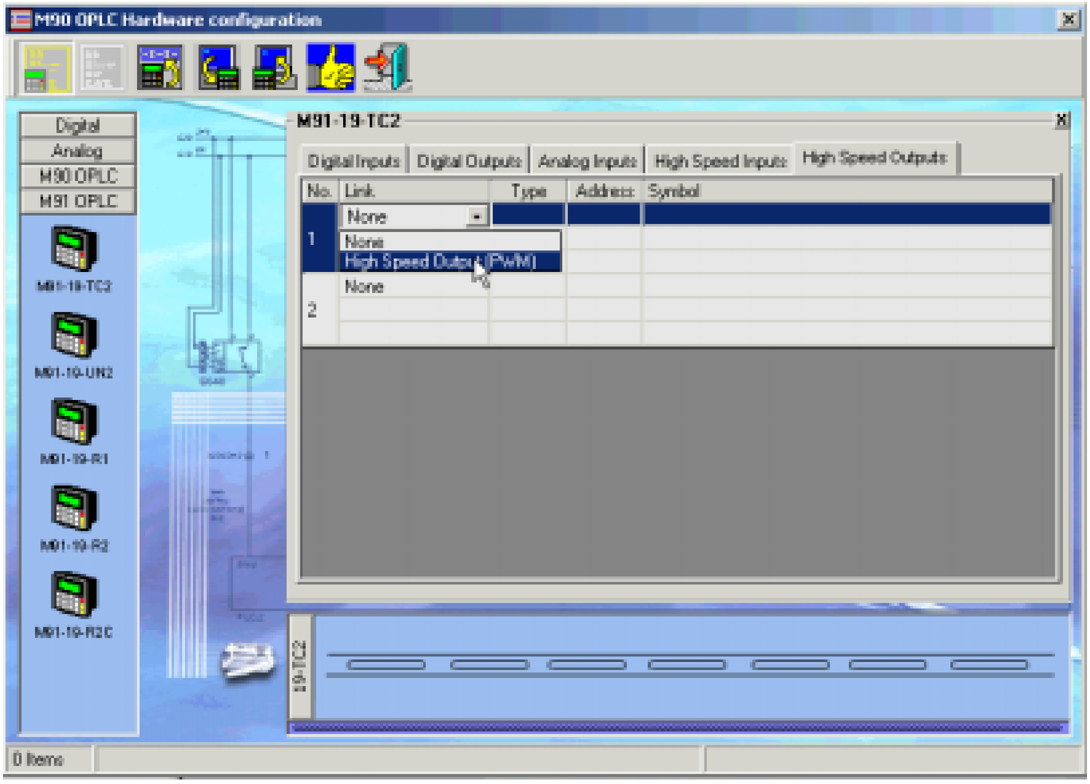


M91-19-TC2

Type	Address	Symbol
I	0	
I	1	
I	2	
I	3	
I	4	
I	5	
I	6	
I	7	
I	8	
I	9	
I	10	

M91-19-TC2: 11/12 Digital I/O, 1 Analog In, 2 High Speed Counter, 2 HSO

3. Click on the High Speed Outputs tab, then select High Speed Output (PWM).



M90 OPLC Hardware configuration

Digital
Analog
M90 OPLC
M91 OPLC
M91-19-TC2
M91-19-UN2
M91-19-R1
M91-19-R2
M91-19-R2C

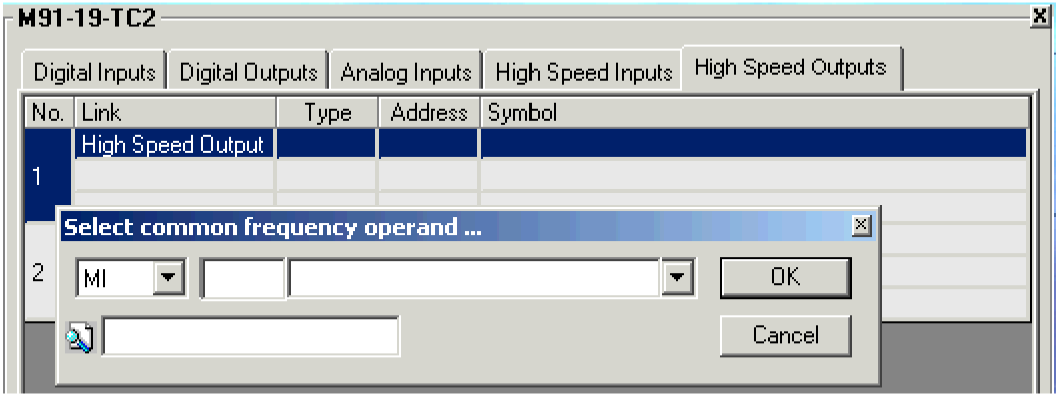
M91-19-TC2

Digital Inputs | Digital Outputs | Analog Inputs | High Speed Inputs | High Speed Outputs

No.	Link	Type	Address	Symbol
1	None			
	None High Speed Output (PWM)			
	None			
2				

0 Items

4. The Select Operand and Address box will open 3 times, enabling you to link MIs for Common Frequency & Duty Cycle, and MB for Enable Output.



M91-19-TC2

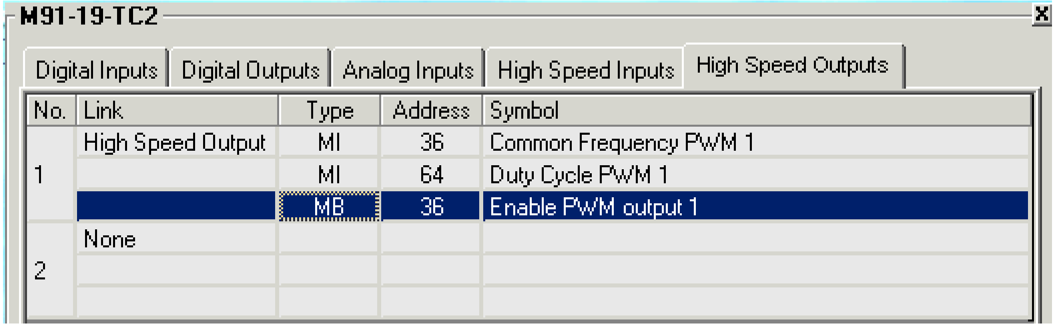
Digital Inputs | Digital Outputs | Analog Inputs | High Speed Inputs | High Speed Outputs

No.	Link	Type	Address	Symbol
1	High Speed Output			
2				

Select common frequency operand ...

MI

7. The PWM output is now part of the configuration.



M91-19-TC2

Digital Inputs | Digital Outputs | Analog Inputs | High Speed Inputs | High Speed Outputs

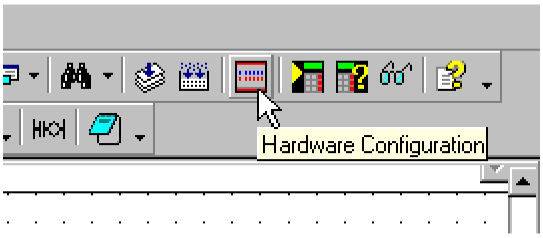
No.	Link	Type	Address	Symbol
1	High Speed Output	MI	36	Common Frequency PWM 1
		MI	64	Duty Cycle PWM 1
		MB	36	Enable PWM output 1
2	None			

Configuring an Analog Input

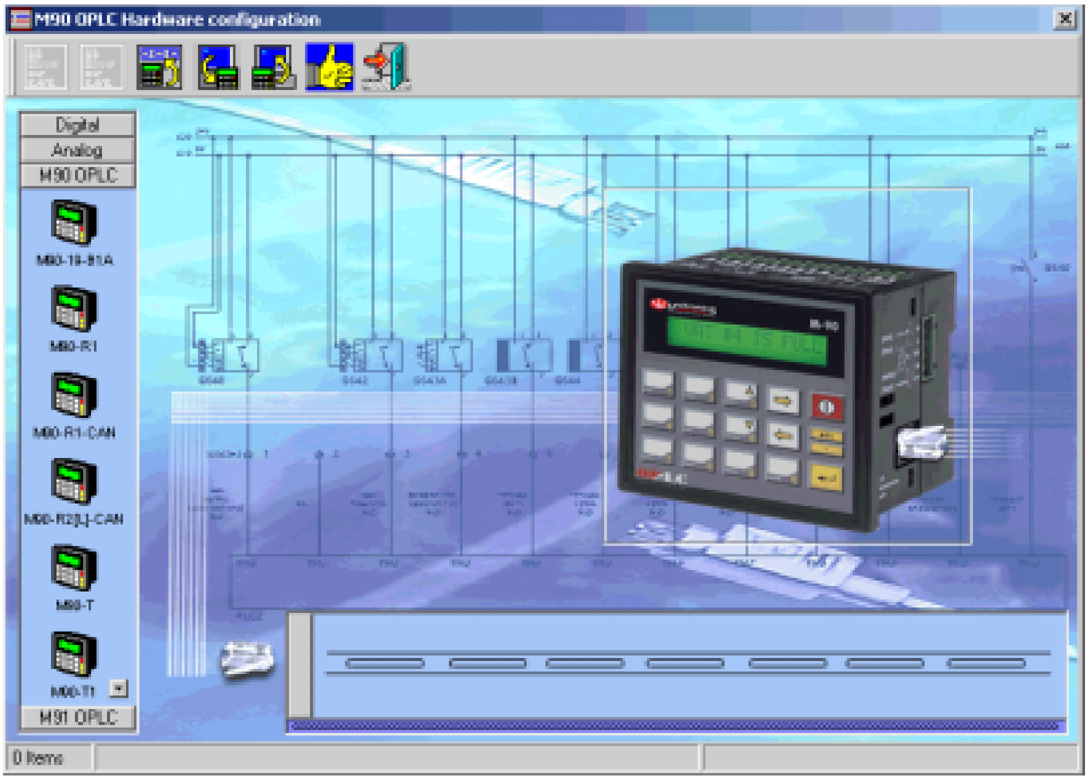
M90

To attach an Analog Input to an MI:

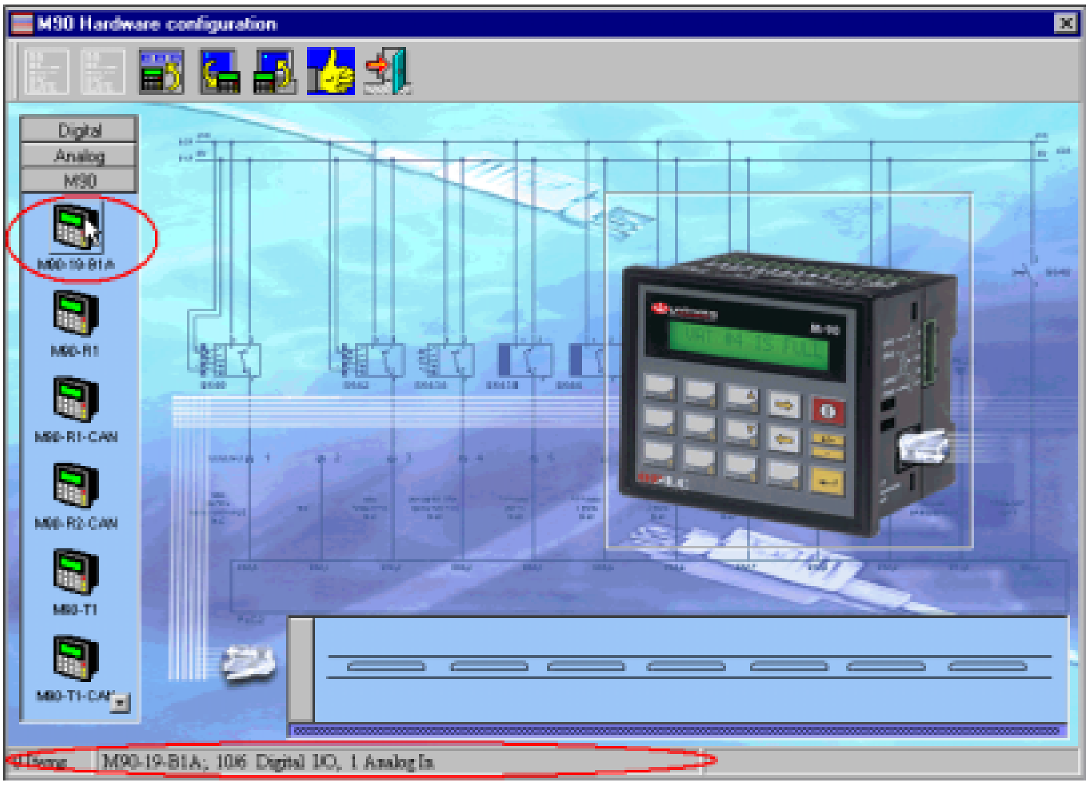
1. Click Hardware Configuration on the Standard toolbar.



2. The M90 Hardware Configuration window opens.



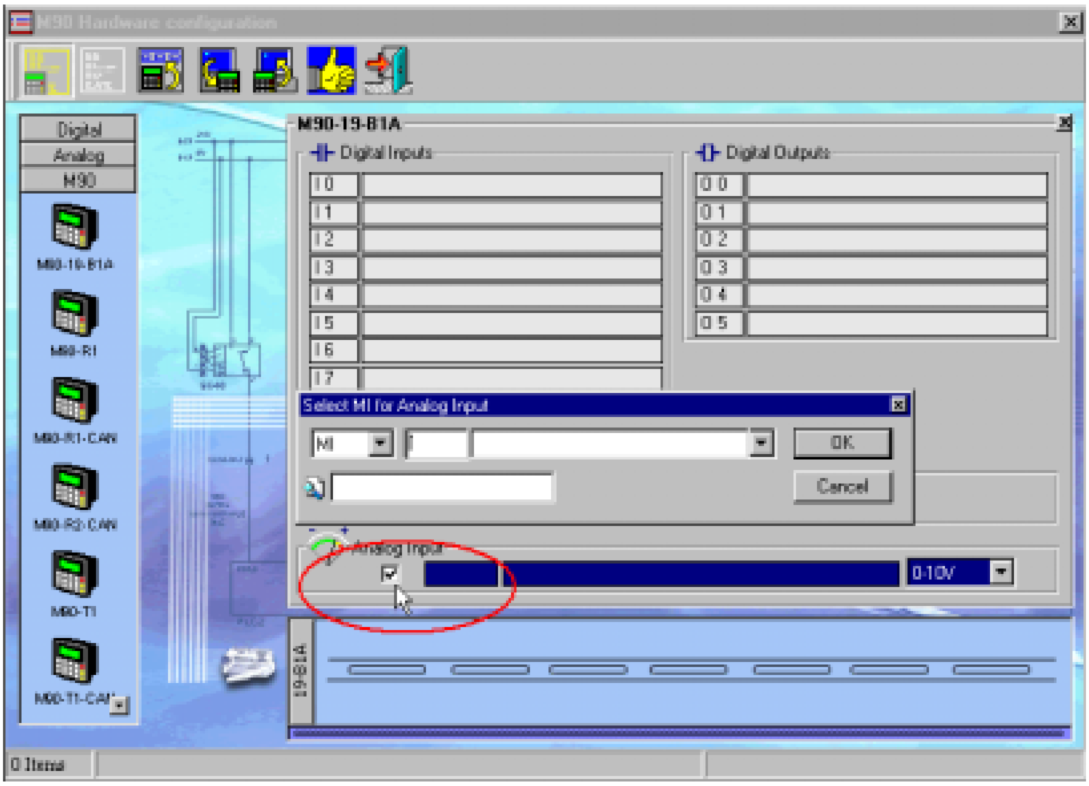
3. Click the appropriate M90 model.



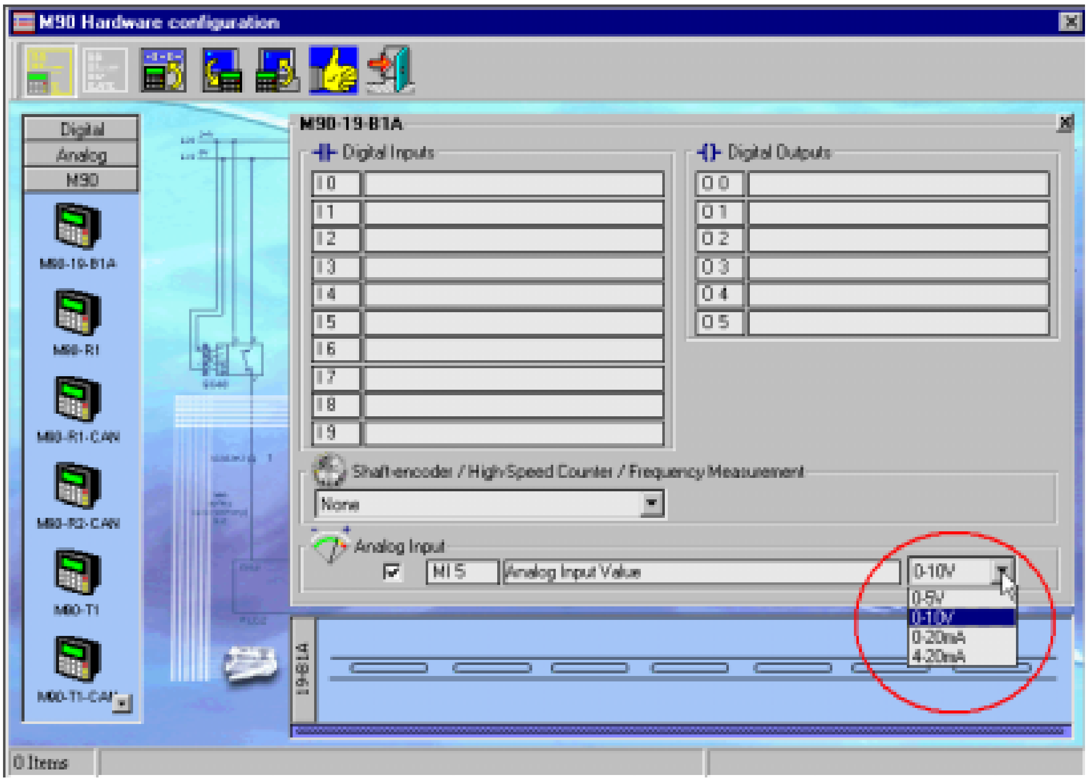
4 Items M90-19-B1A; 10/6 Digital I/O, 1 Analog In

4. The I/O options for that model are displayed.

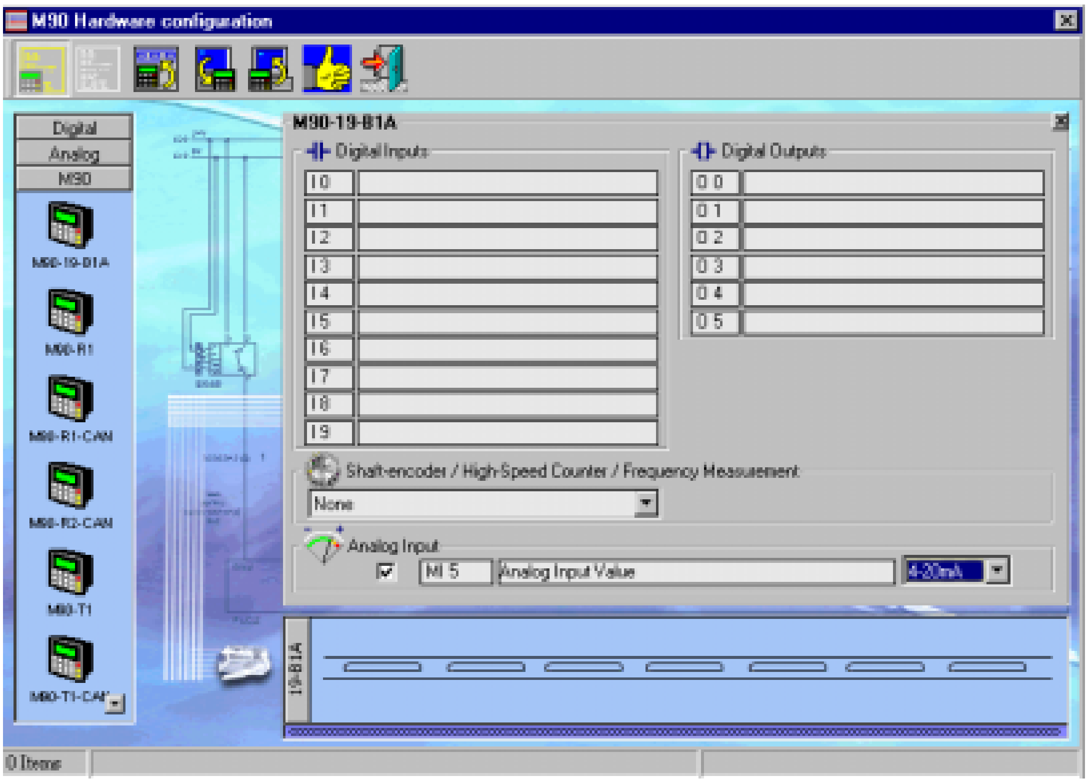
5. Check the Analog Input check box. The Select MI for Analog Input window opens.



6. Enter the desired Address and Symbol of the MI Operand. Select the Analog Input type from the drop-down menu.



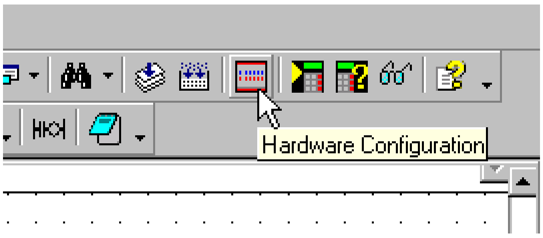
7. The M90 Hardware Configuration window now appears with the new Analog Input configuration.



M91

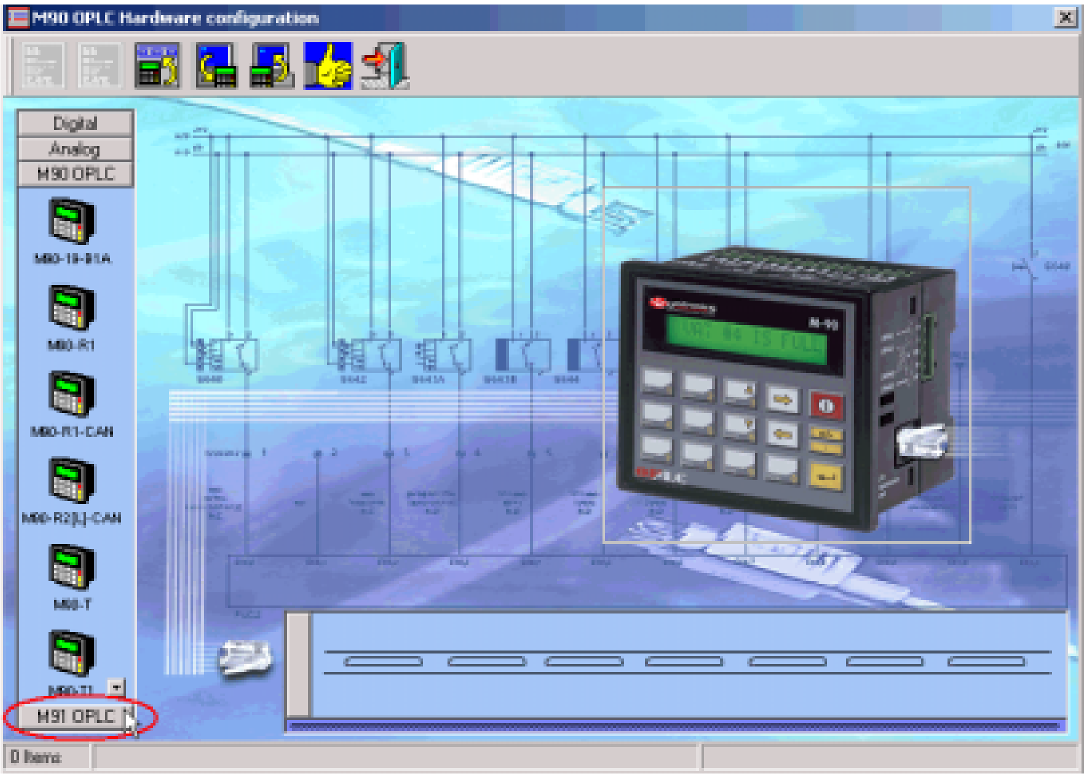
To attach an Analog Input to an MI:

1. Click Hardware Configuration on the Standard toolbar.



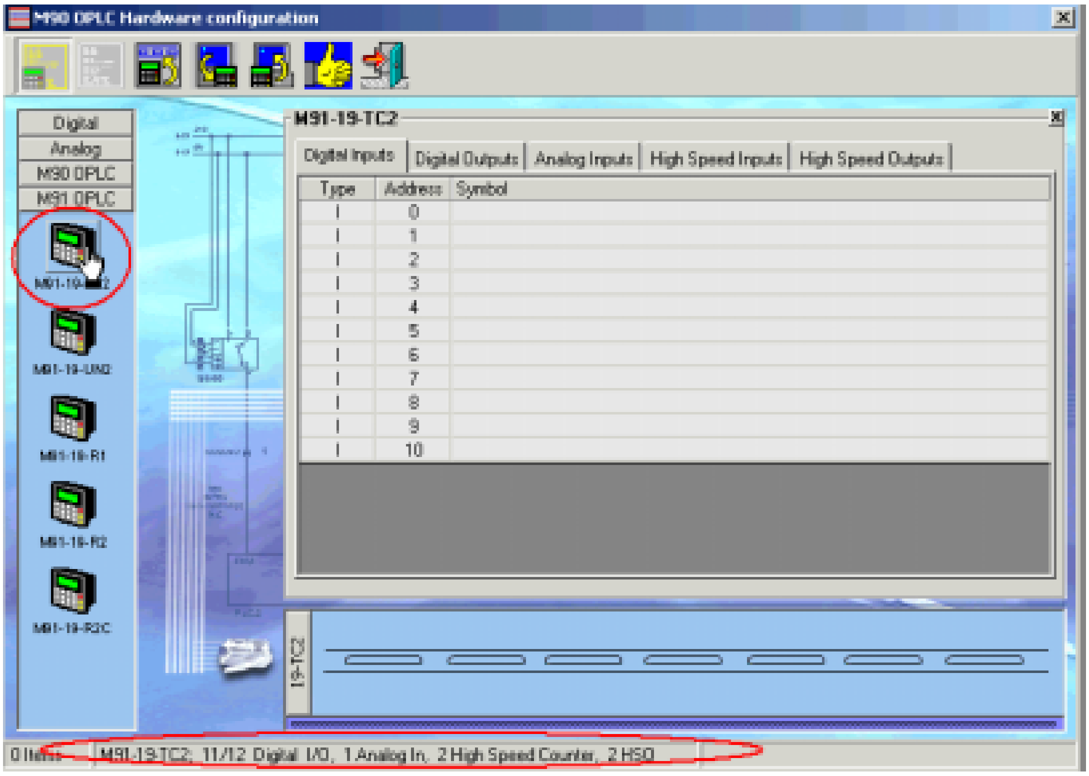
2. The M90 Hardware Configuration window opens.

3. Click on the M91 bar.



3. Select the appropriate M91 model; the model's I/O options are displayed.

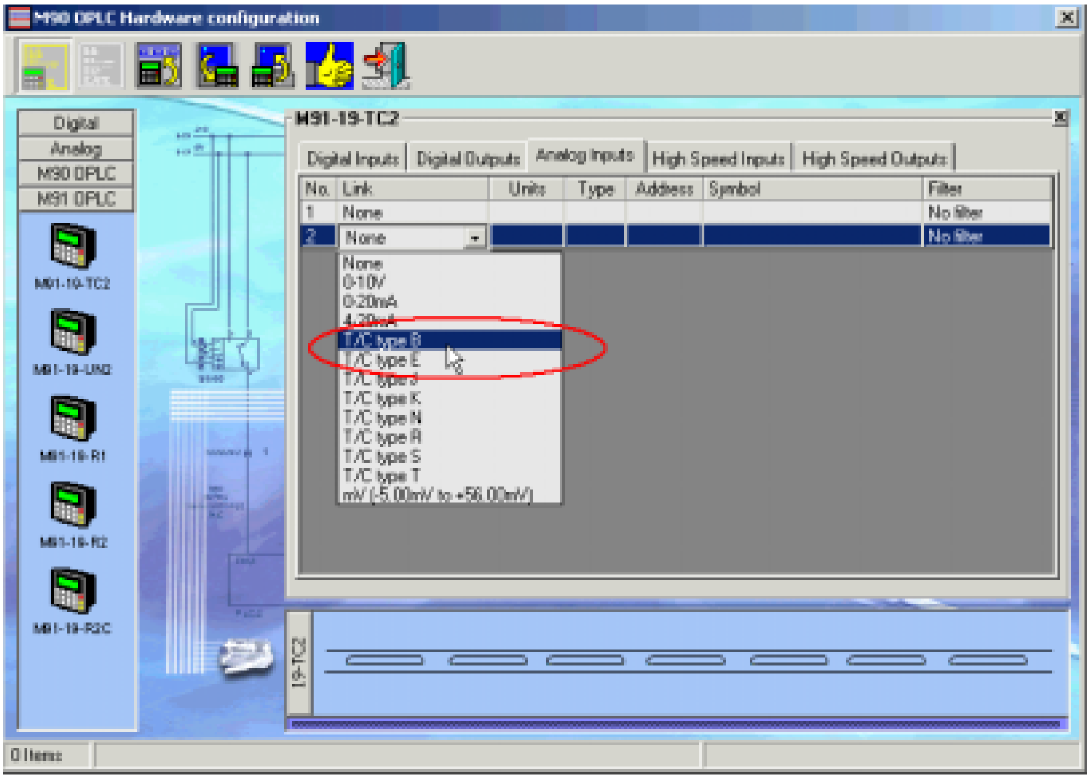
4. Click on the Analog Inputs tab.



Type	Address	Symbol
I	0	
I	1	
I	2	
I	3	
I	4	
I	5	
I	6	
I	7	
I	8	
I	9	
I	10	

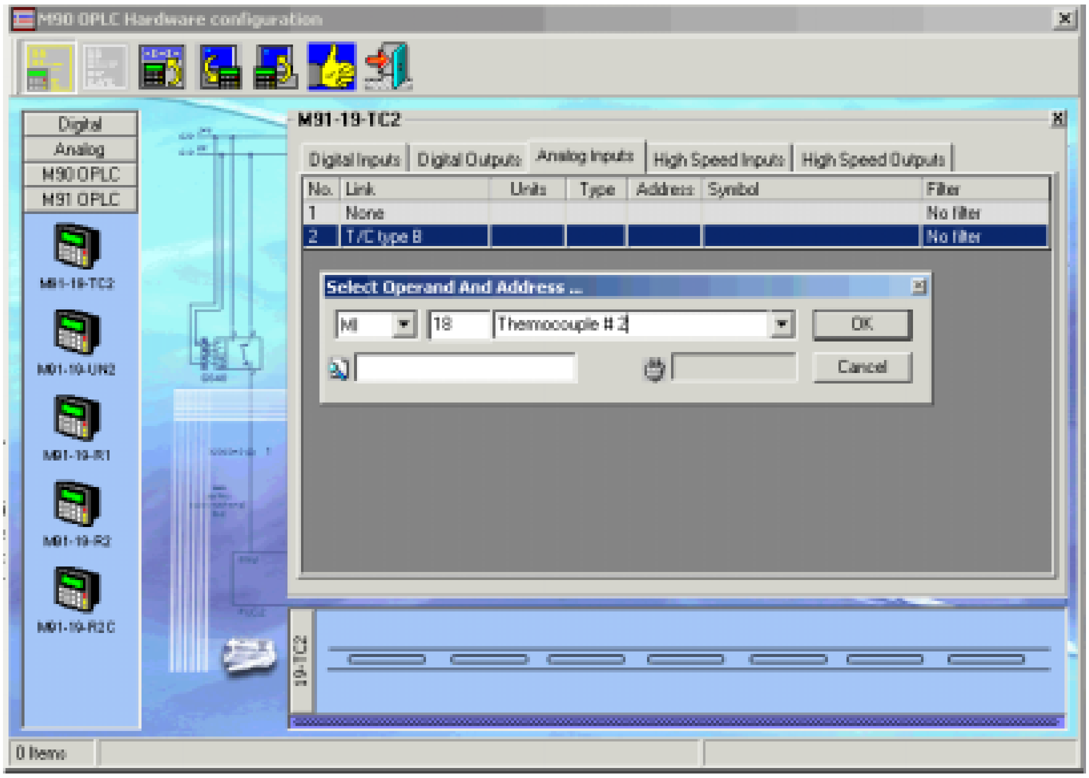
0 Items M91-19-TC2: 11/12 Digital I/O, 1 Analog In, 2 High Speed Counter, 2 HSO

4. Click the Link field, then select the desired type of input. The Select MI for Analog Input window opens.



No.	Link	Units	Type	Address	Symbol	Filter
1	None					No filter
2	None					No filter

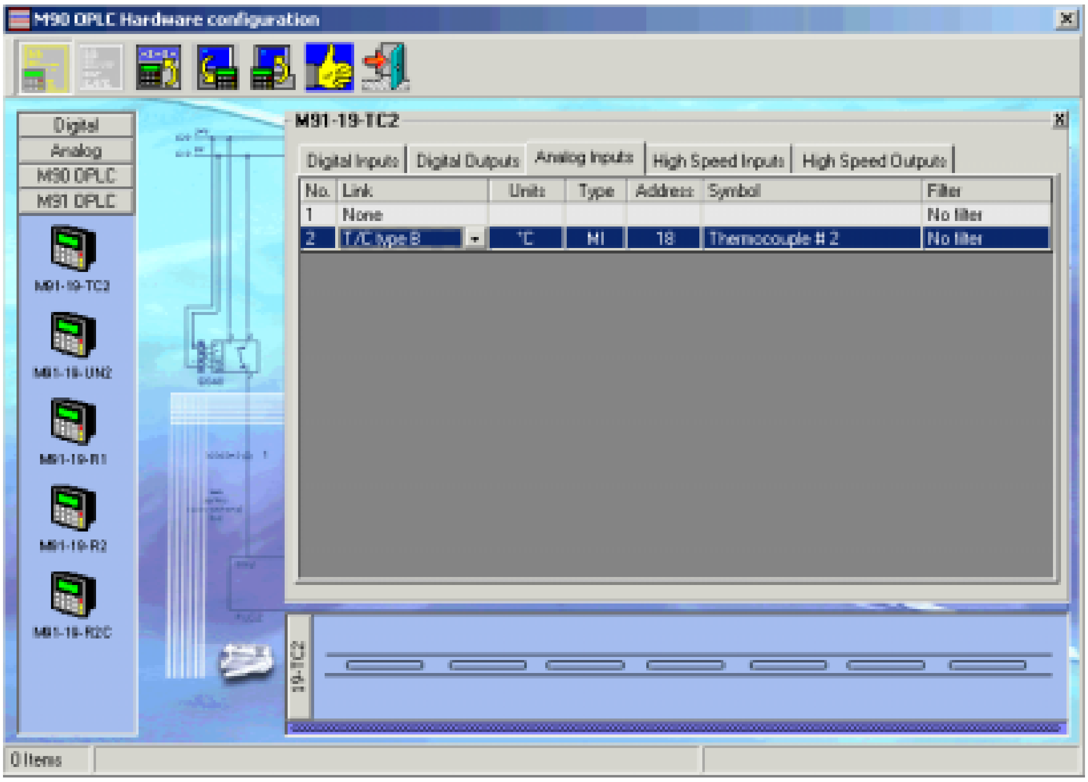
6. Enter the desired Address and Symbol of the MI Operand.



No.	Link	Units	Type	Address	Symbol	Filter
1	None					No filter
2	T/C type B					No filter

MI 18 Thermocouple # 2

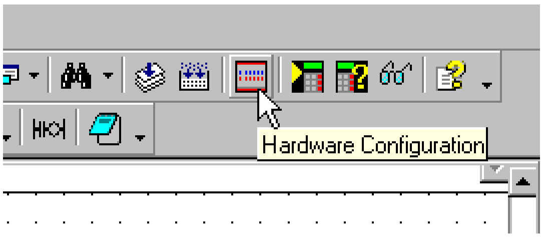
7. The Analog Input is now part of the configuration.



No.	Link	Units	Type	Address	Symbol	Filter
1	None					No filter
2	T/C type B	°C	MI	18	Thermocouple # 2	No filter

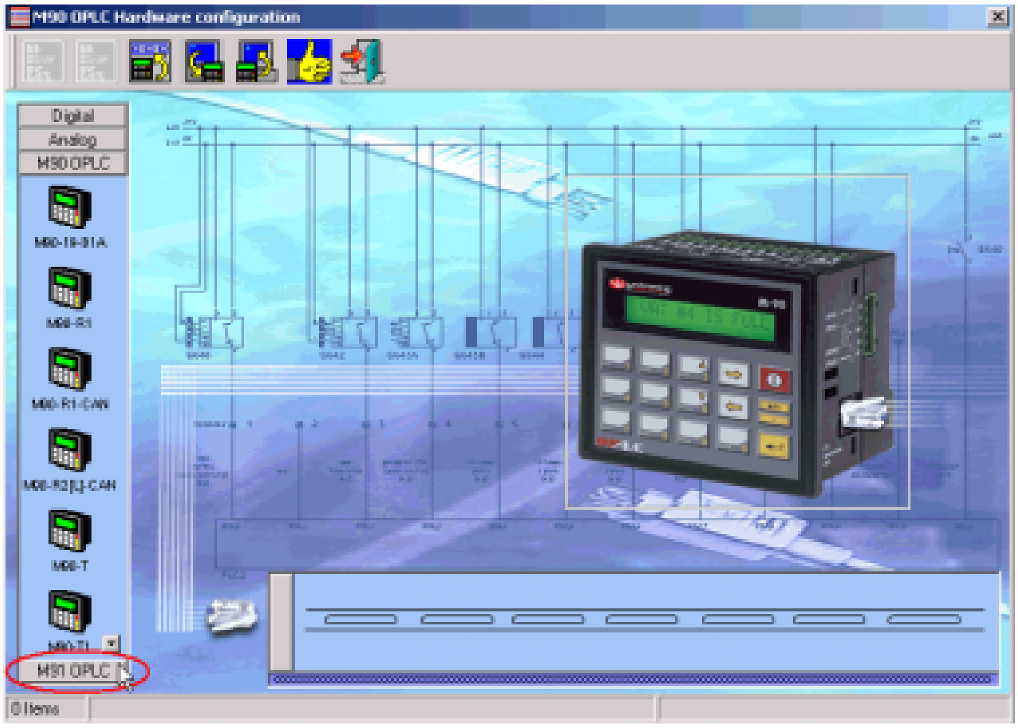
Configuring a Thermocouple: M91 OPLC series

1. Click ***Hardware Configuration*** on the Standard toolbar.

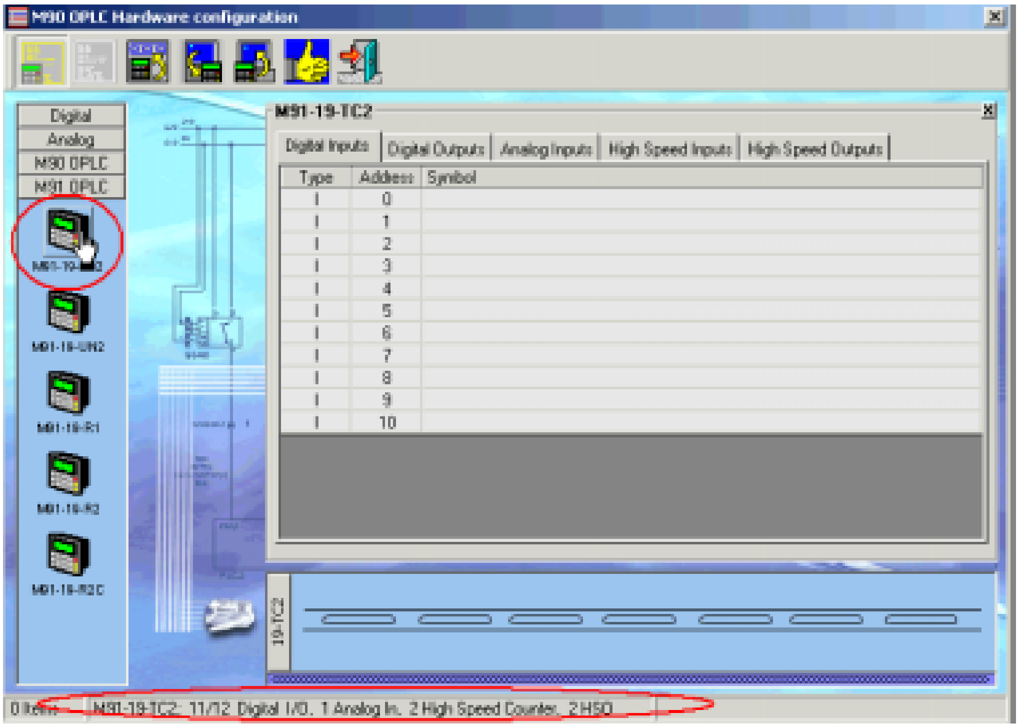


2. The **M90 Hardware Configuration** window opens.

3. Click on the *M91 bar*.



4. Select the appropriate M91 model; the *model's I/O options* are displayed.



M91-19-TC2

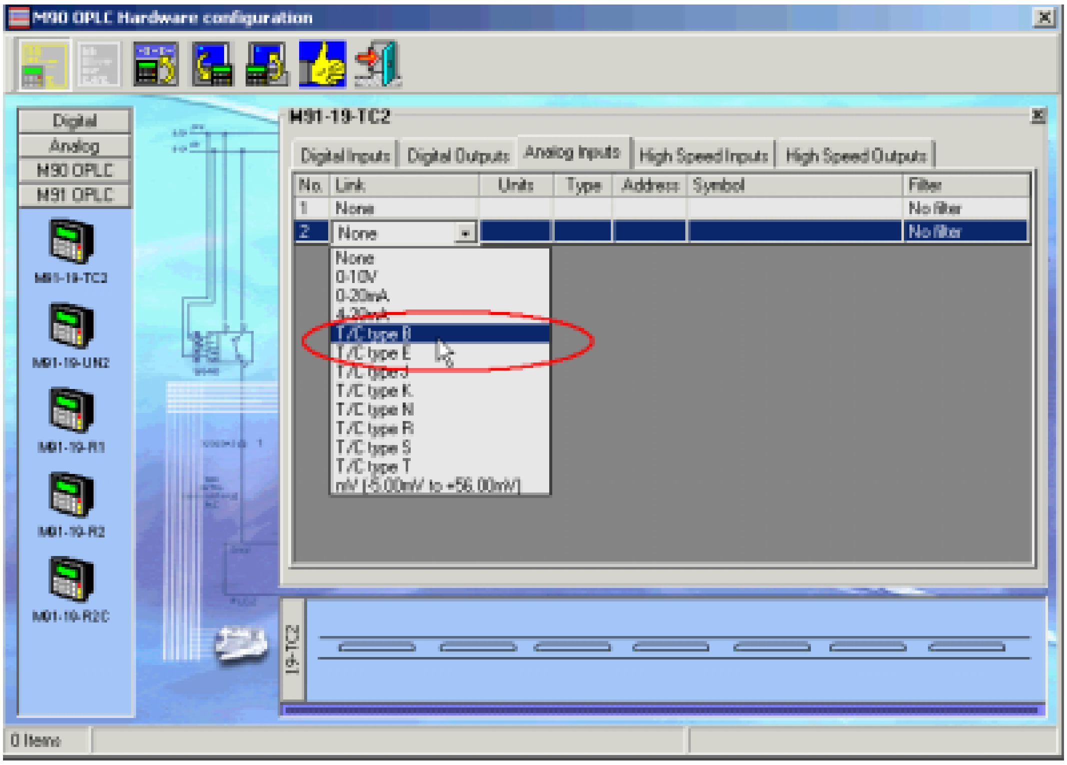
Digital Inputs | Digital Outputs | Analog Inputs | High Speed Inputs | High Speed Outputs

Type	Address	Symbol
I	0	
I	1	
I	2	
I	3	
I	4	
I	5	
I	6	
I	7	
I	8	
I	9	
I	10	

M91-19-TC2: 11/12 Digital I/O, 1 Analog In, 2 High Speed Counter, 2 HSO

5. Click on the Analog Inputs tab.

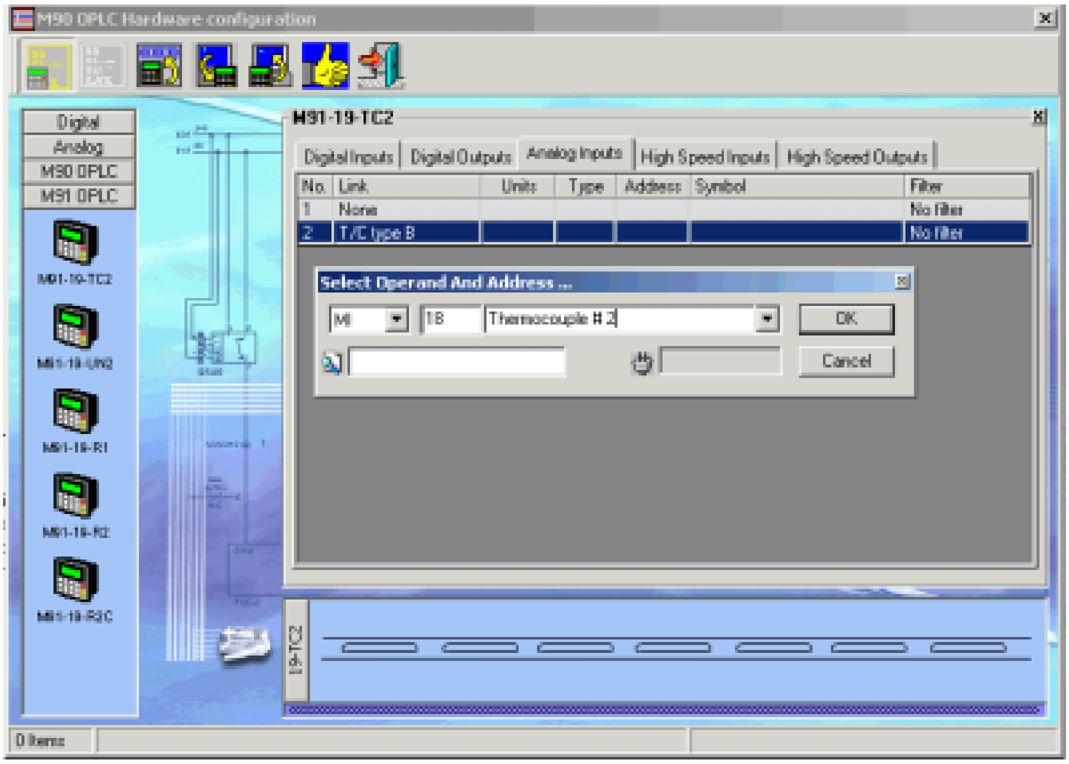
6. Click the Link field, then select the desired type of input. The **Select Operand and Address** box opens.



No.	Link	Units	Type	Address	Symbol	Filter
1	None					No filter
2	None					No filter

None
0-10V
0-20mA
4-20mA
T/C type B
T/C type E
T/C type J
T/C type K
T/C type N
T/C type R
T/C type S
T/C type T
mV [-5.00mV to +56.00mV]

6. Enter the desired Address and Symbol of the MI Operand.

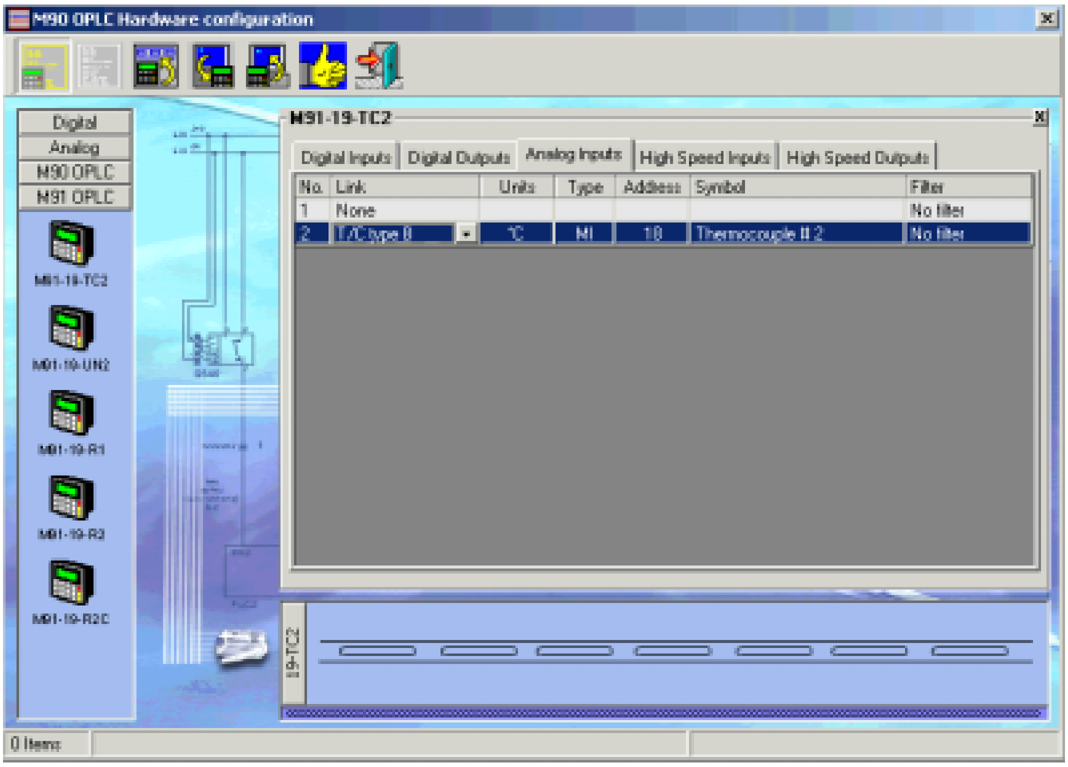


No.	Link	Units	Type	Address	Symbol	Filter
1	None					No filter
2	T/C type B					No filter

Select Operand And Address ...

MI | 18 | Thermocouple # 2

7. The thermocouple is now part of the configuration.



No.	Link	Units	Type	Address	Symbol	Filter
1	None					No filter
2	T/C type B	°C	MI	18	Thermocouple # 2	No filter

HMI

Variables

You insert Variables into a Display to:

- Show varying values and text on the controller screen.
- Enter values into the controller.

Use the Variable Editor to link variables to the operands that contain the data you want to use in your program. You can use variables in your HMI program to display text that varies according to current conditions or events. Variable integers also can receive data input from the M90's keypad keys, such as an employee ID number, or a set point for process control.

Displaying Variable Values in a Display

To display data from an HMI variable within an M90 display, you must:

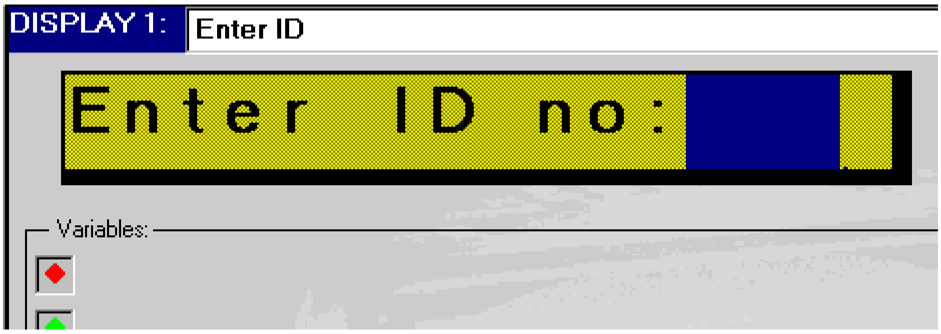
- Create a field within the display that is long enough to hold the variable data.
- Attach a variable to the field.

To Create a Field

1. Click your cursor in the display. This is the starting point of the field.
2. To create the field, either:
3. Drag the cursor across the display. The field you create is automatically highlighted in blue.

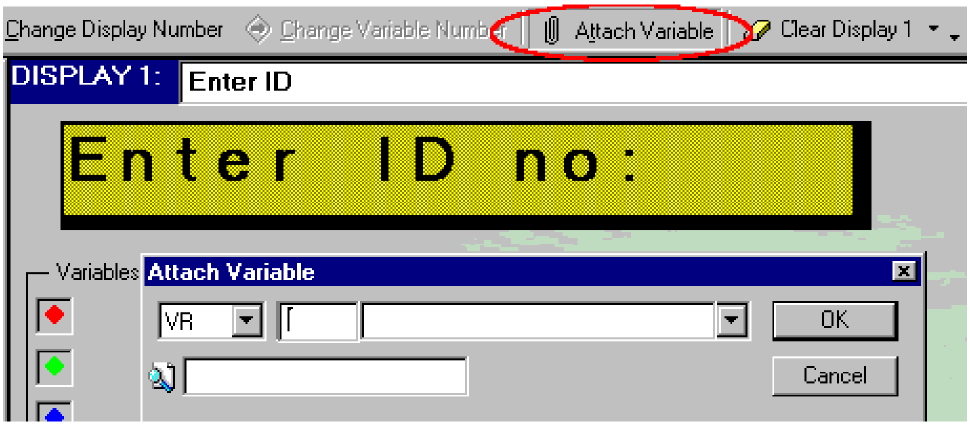
   OR
4. Hold the SHIFT key down, and press the right-pointing arrow key. Each time you press the arrow key, a space is automatically highlighted in blue.

In the figure below, the display contains a field two spaces long.

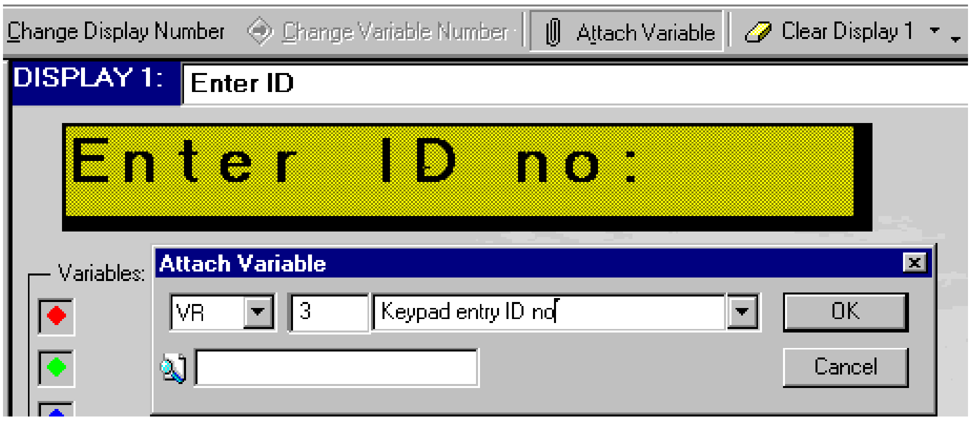


To Attach a Variable

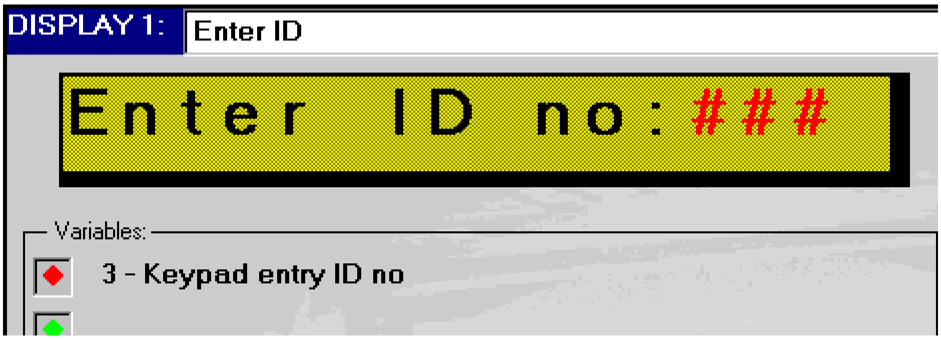
1. Click Attach Variable on the HMI toolbar. T he Attach Variable dialog box opens as shown below.



2. Enter the number of the desired variable as shown below and press OK. If you do not enter a variable number, the program assigns a default variable.



3. The variable-linked spaces now appear as red pound signs, and the variable itself appears in the Variable pane of this Display as shown below.


DISPLAY 1: Enter ID

Enter ID no:###

Variables:

3 - Keypad entry ID no

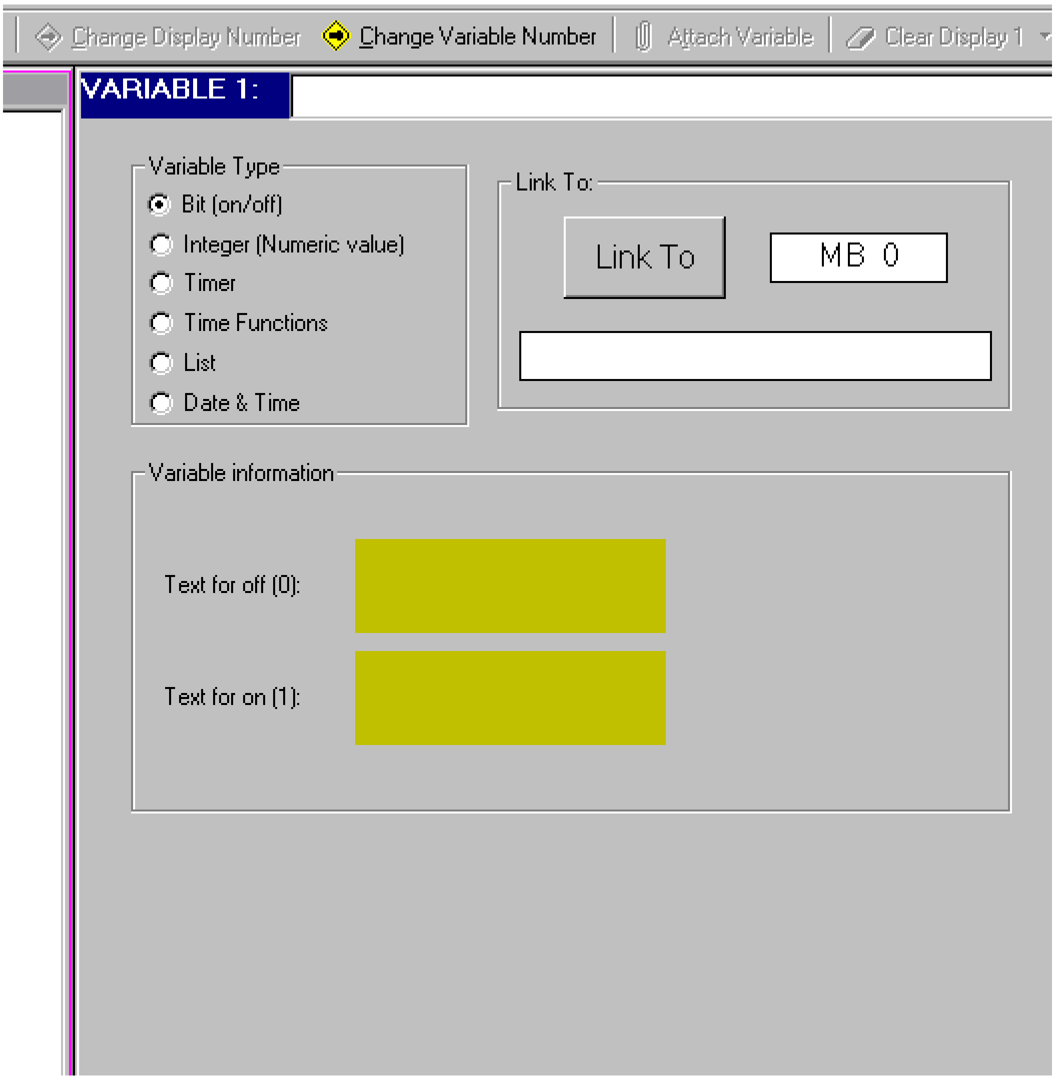
Use the Variable Editor to:

- Set variable types and properties.
- Create up to 120 list variables to display fixed text messages.
- Enable data entry via the M90 keypad.

Up to fifty variables may be included in your application. The different types of variables are listed below.

Variable Type	**Linked to**	**Display Options:**
Bit	MB	Create a text display for ON and OFF.
Integer	MI	Choose integer display format; enable linearization and keypad entry.
Timer	T	Display either elapsed time or remaining time and allow timer modification via the M90 keypad.
Time Functions	MI	Display and modify Time function from hour up to year.
List	MI	Create up to 120 additional fixed text messages for different values of an MI / SI.
Date & Time	RTC	Set the display format (from Hours/Minutes to Month/Day/Year) and enable keypad entry.

Variable Editor view:



VARIABLE 1:

Variable Type

● Bit (on/off)
○ Integer (Numeric value)
○ Timer
○ Time Functions
○ List
○ Date & Time

Link To: MB 0

Variable information

Text for off (0):

Text for on (1):

What is an HMI?

HMI stands for Human Machine Interface. This is the interface between the operator and the controller.

The M90 HMI is the controller operating panel. The panel comprises a 15 key numeric keypad and a 16 character LCD Display screen.

The keypad is used to input data into the application, such as Timer values.

The M90's Display screen can show operator messages, variable information from the program and system information.

HMI messages are created in the Display Editor.

Variable information fields are created in the Variable Editor.

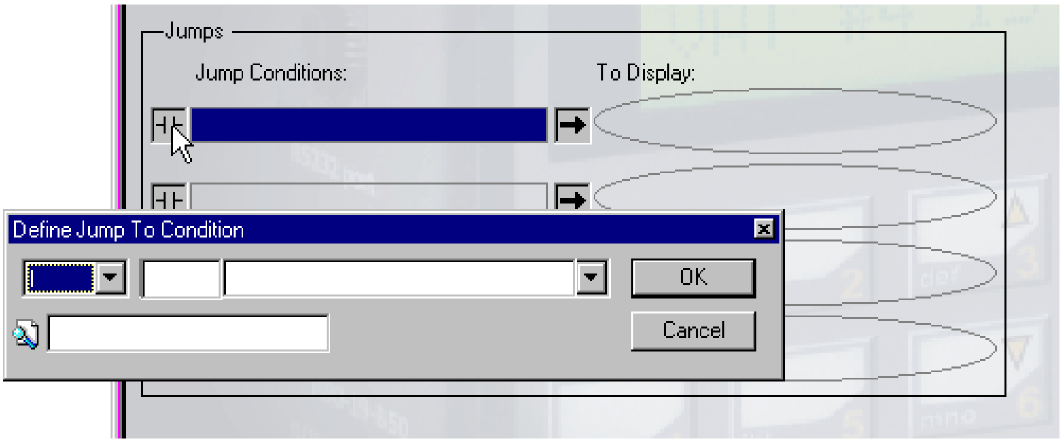
HMI applications are featured in several sample applications, such as the applications ' Display Jumps from Ladder', 'Names from List Var', 'Password', 'Special characters on List', 'Display of Events', and '5 Vars on Display'. These applications may be found by selecting Sample U90 Projects from the Help Menu.

Jump to Display: scrolling between Displays

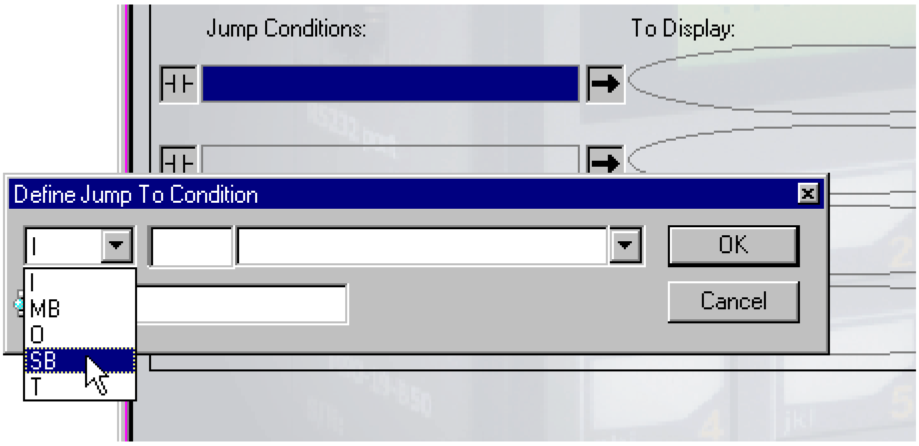
Display Jumps allow you to move between Displays via the M90 keypad or any bit positive transition. You can create up to 4 Jumps for each Display in the Display Editor. If you want to create more than 4 Jumps for a Display, you must create the logic conditions in the Ladder Editor.

To create a jump:

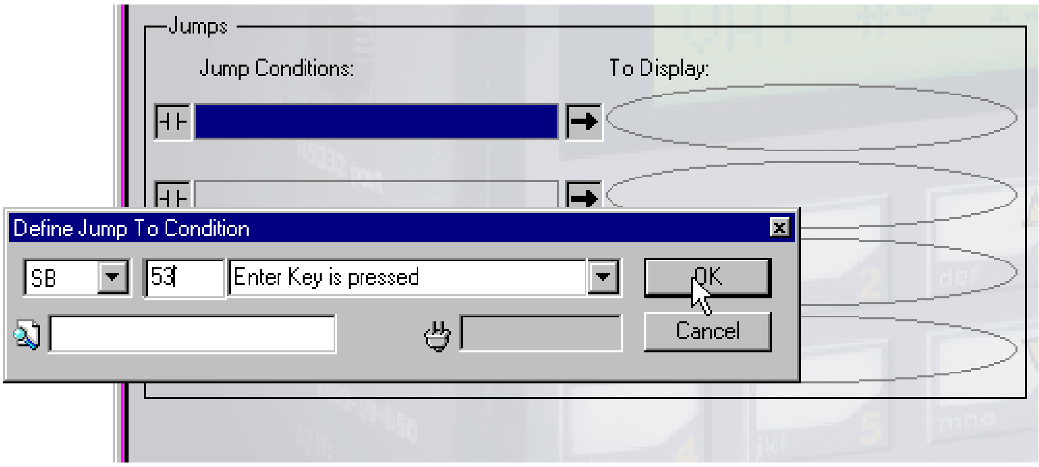
1. Click on a Jump Condition and the Define Jump to Condition dialog box opens.



2. Select a Jump Operand from the drop-down menu.

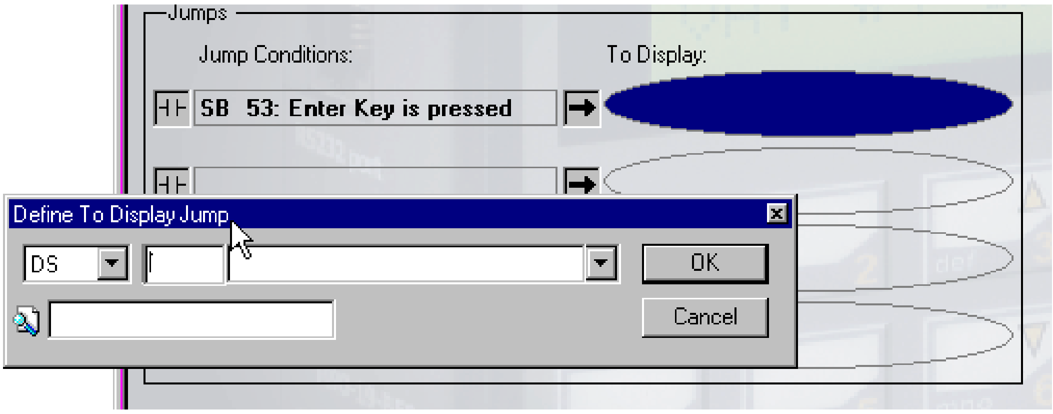


3. Enter the desired Address and symbol for the Jump Operand. Click OK.



SB 53 Enter Key is pressed

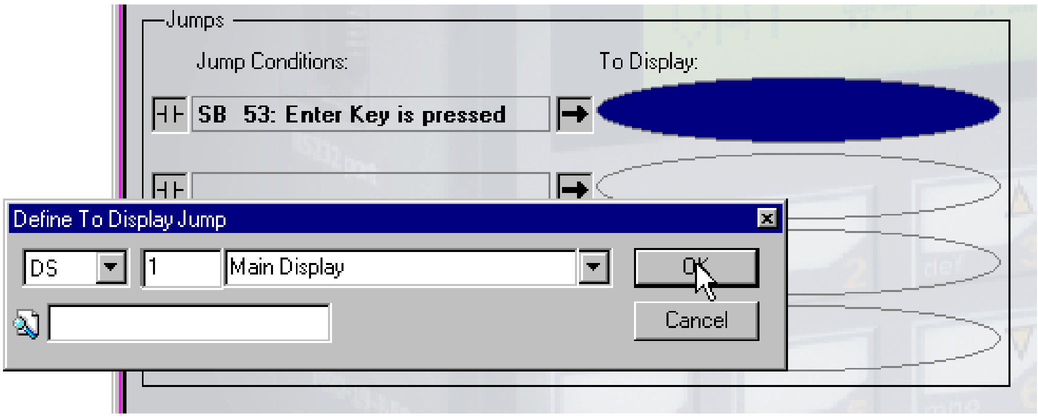
4. The Define To Display Jump dialog box opens.



SB 53: Enter Key is pressed

DS

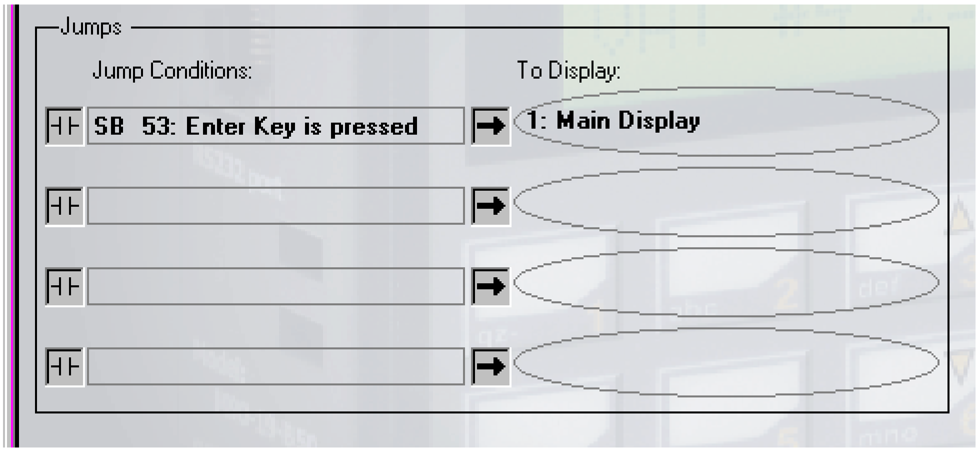
5. Enter the Display number to which you want to jump. Click OK.



SB 53: Enter Key is pressed

DS 1 Main Display

6. The result will be:



SB 53: Enter Key is pressed → 1: Main Display

Note that Display Jump conditions based on MBs can **only** be linked to MB 0-127; jumps may not be linked to MB 128 -255.

Note ♦ When an HMI keypad entry variable is active, and the Enter key is pressed on the controller keypad, SB 30 HMI Keypad Entries Complete turns ON. This can be used as a Jump condition.

In addition , note that a Display may contain a total of 4 variables. Each one has an SB:

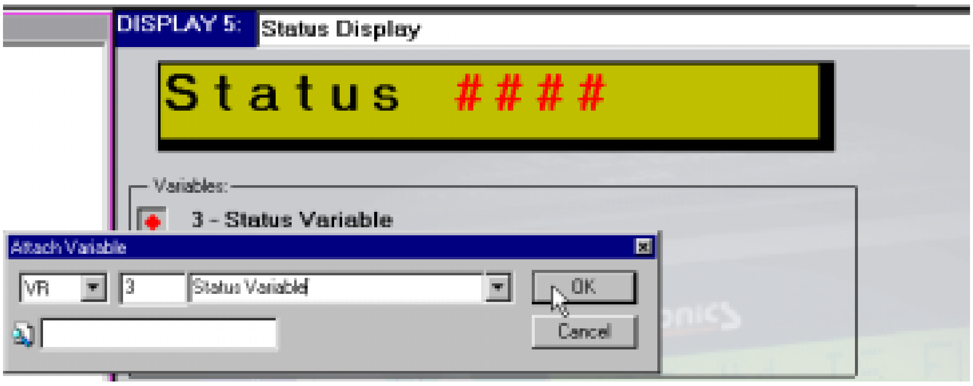
- SB 31 HMI Var 1 Keypad entry completed
- SB 32 HMI Var 2 Keypad entry completed
- SB 33 HMI Var 3 Keypad entry completed
- SB 34 HMI Var 4 Keypad entry completed

The condition of these SBs may be used as Jump Conditions, or to drive calculations in your program.

Displaying text according to the value of a MB or SB

To display a text according to the value of a MB or SB:

1. Create a Display and variable field.



DISPLAY 5: Status Display

Status ####

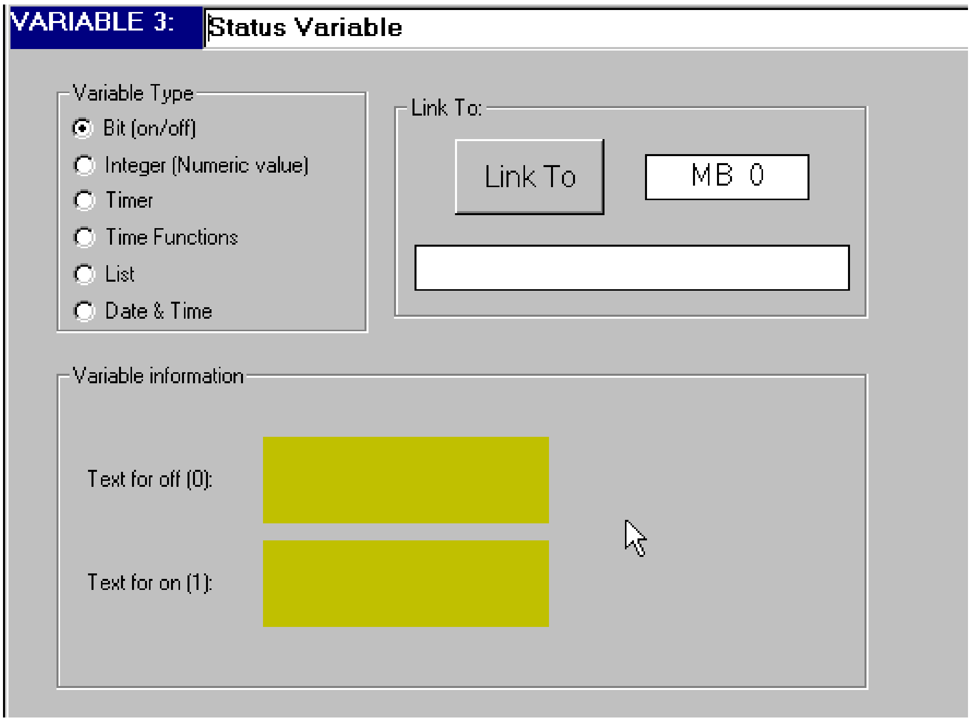
Variables:

3 - Status Variable

Attach Variable

VR | 3 | Status Variable

2. Create a **Bit** type variable attached to the field .



VARIABLE 3: Status Variable

Variable Type

● Bit (on/off)
○ Integer (Numeric value)
○ Timer
○ Time Functions
○ List
○ Date & Time

Link To:

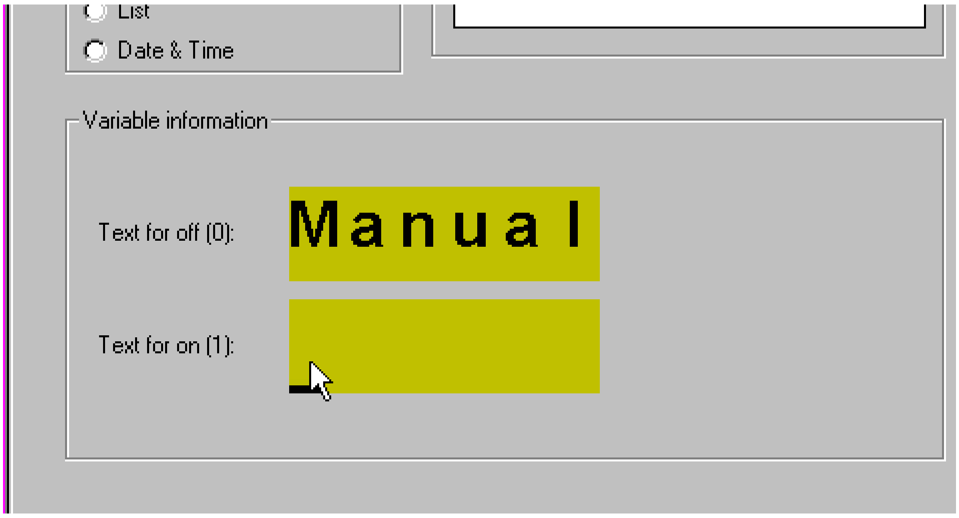
Link To: MB 0

Variable information

Text for off (0):

Text for on (1):

3. Enter a text Display for the "0" value of the MB / SB.

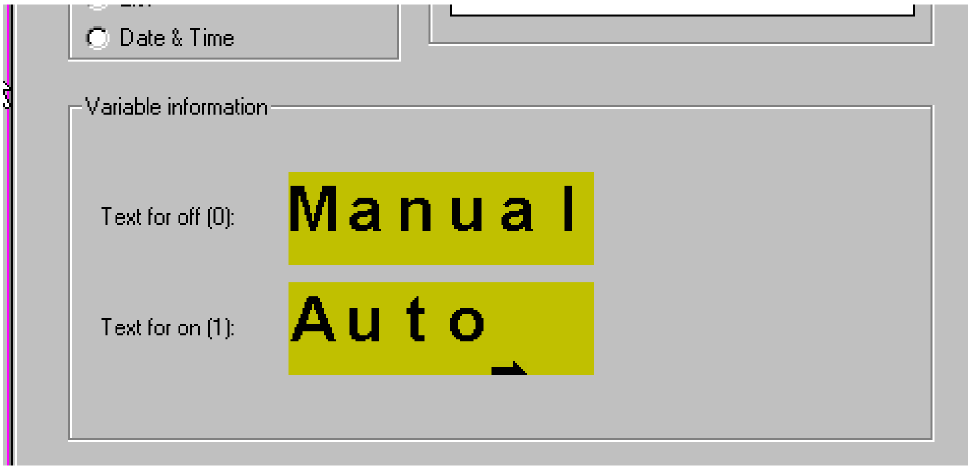

○ List
○ Date & Time

Variable information

Text for off (0): Manual

Text for on (1):

4. Enter a text Display for the "1" value of the MB / SB.


○ Date & Time

Variable information

Text for off (0): Manual

Text for on (1): Auto

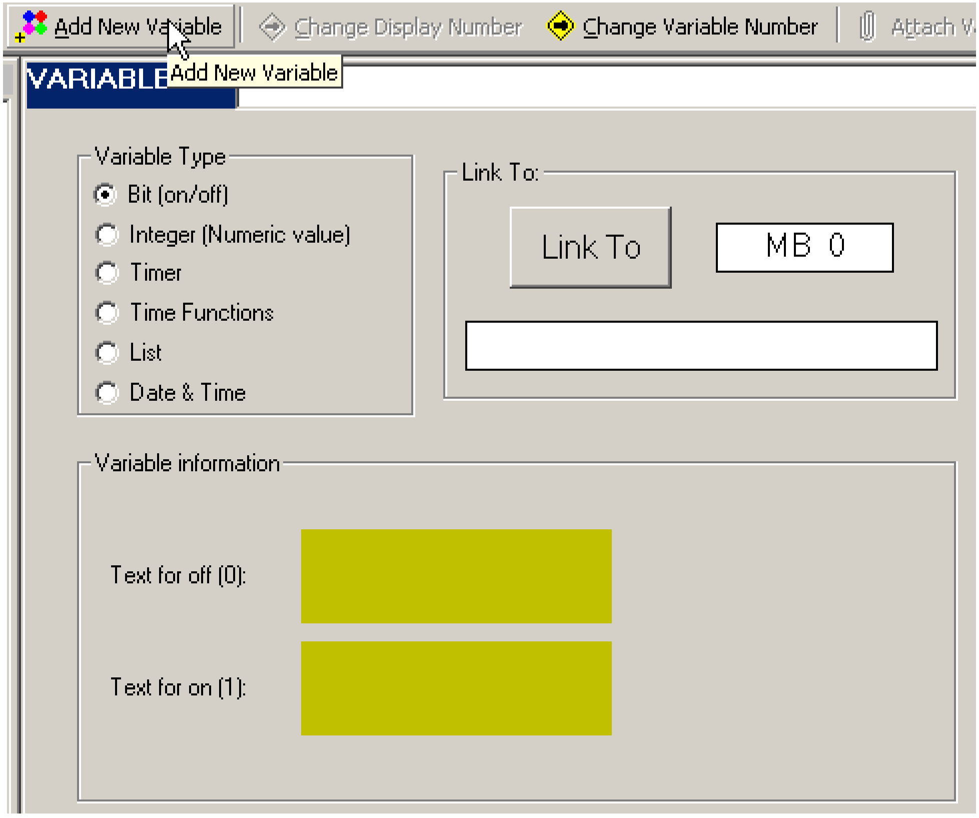
The text will be displayed according to the value of the MB / SB. Note that the Display field must be large enough for the defined text.

For the above example, the Display field must be 6 characters.

List Variable: Display text according to a changing MI value

To display different texts for different values of the same MI:

1. Create a new Variable.



Variable Type
● Bit (on/off)
○ Integer (Numeric value)
○ Timer
○ Time Functions
○ List
○ Date & Time

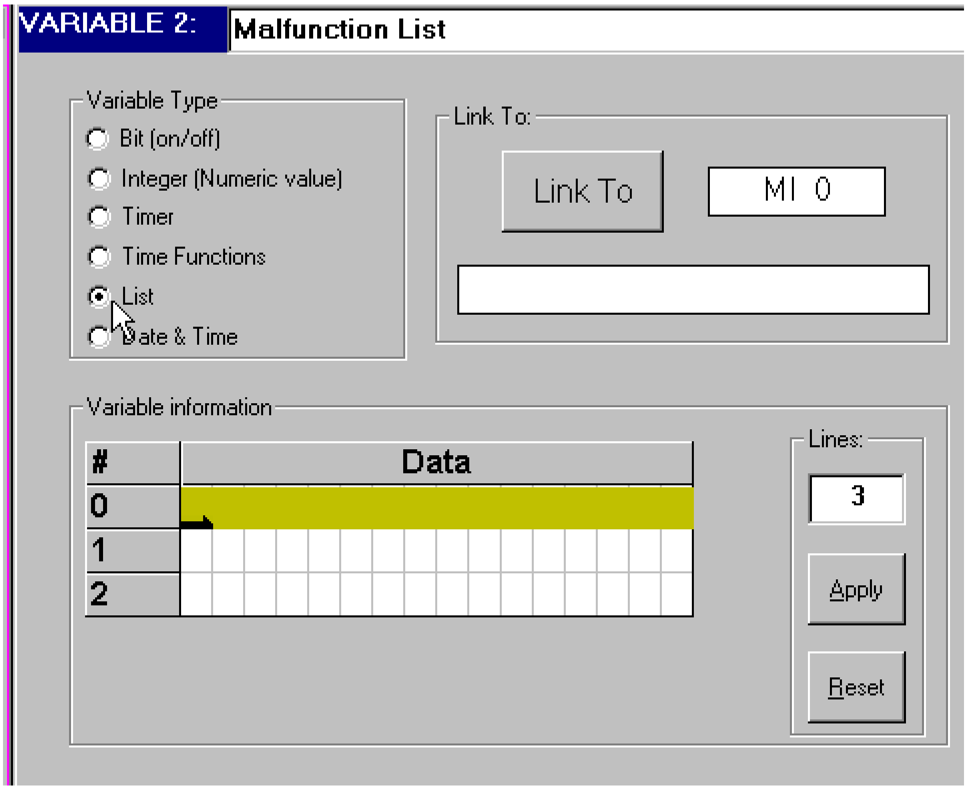
Link To: MB 0

Variable information

Text for off (0):

Text for on (1):

2. Select **List** Variable type.



VARIABLE 2: Malfunction List

Variable Type
○ Bit (on/off)
○ Integer (Numeric value)
○ Timer
○ Time Functions
● List
○ Date & Time

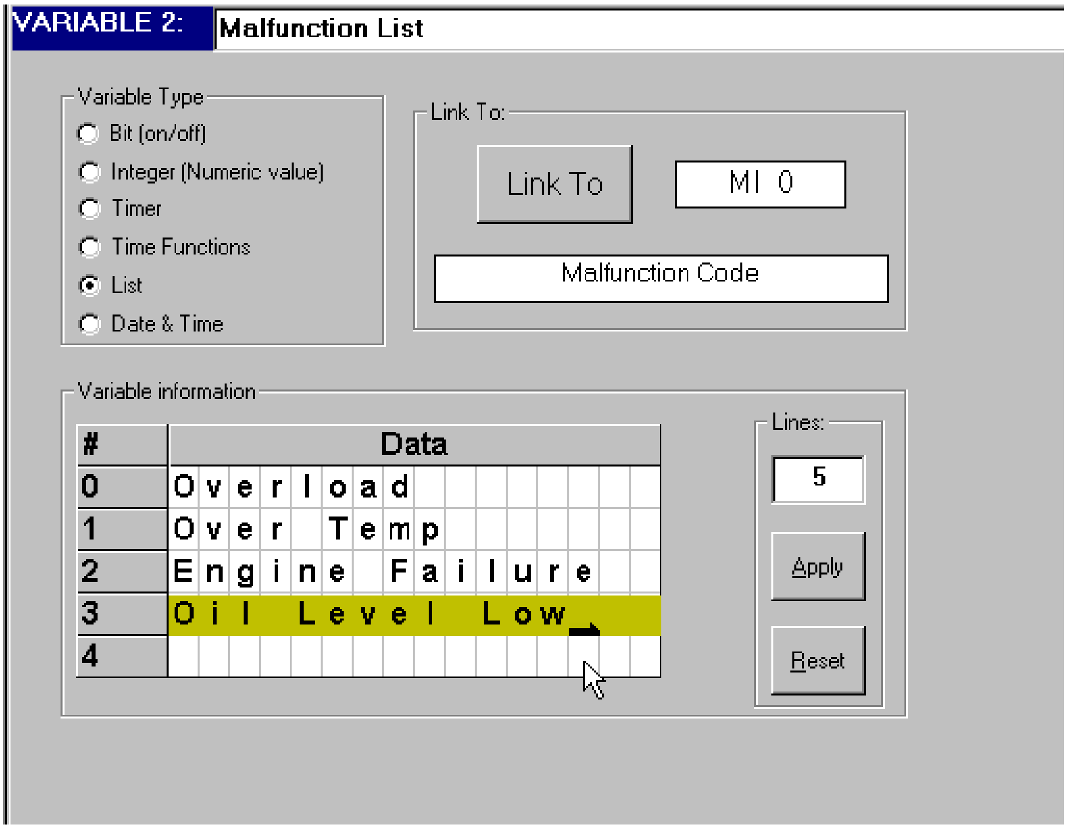
Link To: MI 0

Variable information

#	Data
0	
1	
2	

Lines: 3

3. Enter the desired text for each possible value of the linked MI.



VARIABLE 2: Malfunction List

Variable Type

○ Bit (on/off)
○ Integer (Numeric value)
○ Timer
○ Time Functions
● List
○ Date & Time

Link To:

Link To | MI 0

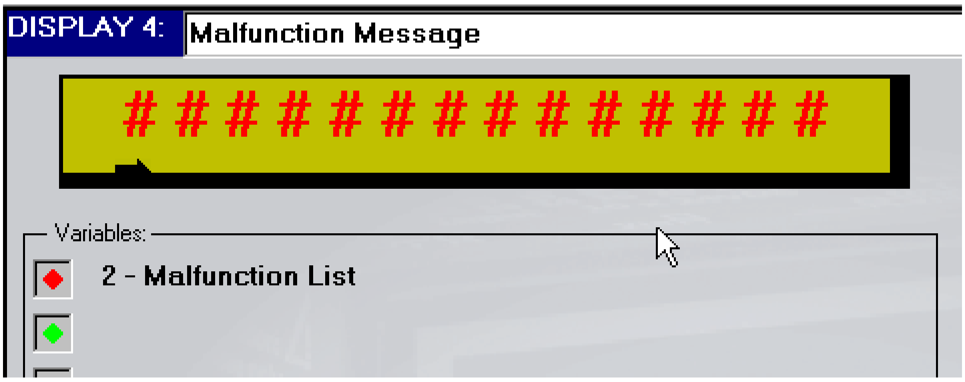
Malfunction Code

Variable information

#	Data
0	Overload
1	Over Temp
2	Engine Failure
3	Oil Level Low
4	

Lines: 5

4. Attach the Variable to a Display field.



DISPLAY 4: Malfunction Message

# # # # # # # # # # # # # #

Variables:

2 - Malfunction List

The text on the Display will be determined by the value written into MI 0 in the Ladder.

Example:

If MI 0 = 2, then the message will be **Engine Failure**.

Display Integer values as ASCII or Hexadecimal

You can:

- Display the values in an MI vector as ASCII characters.
- Display a register value in hexadecimal format.

To do this, attach a numeric Variable to a Display. The variable uses linearization to display the value(s) in the desired format.

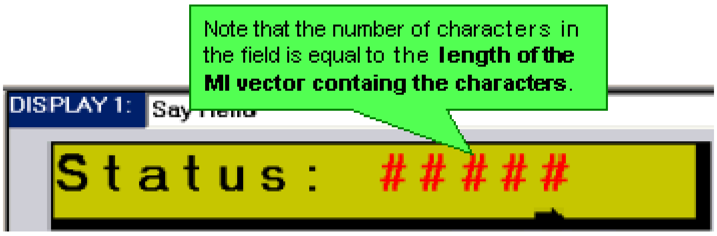
Note that non-supported ASCII characters will be shown as <space> characters.

ASCII -Hexadecimal character table

Vector as ASCII

When the application shown in the example below is downloaded, the ASCII characters 'Hello' will be displayed on the M90 screen when Key #3 is pressed.

1. Create a Variable Field in a Display, then attach a Variable.

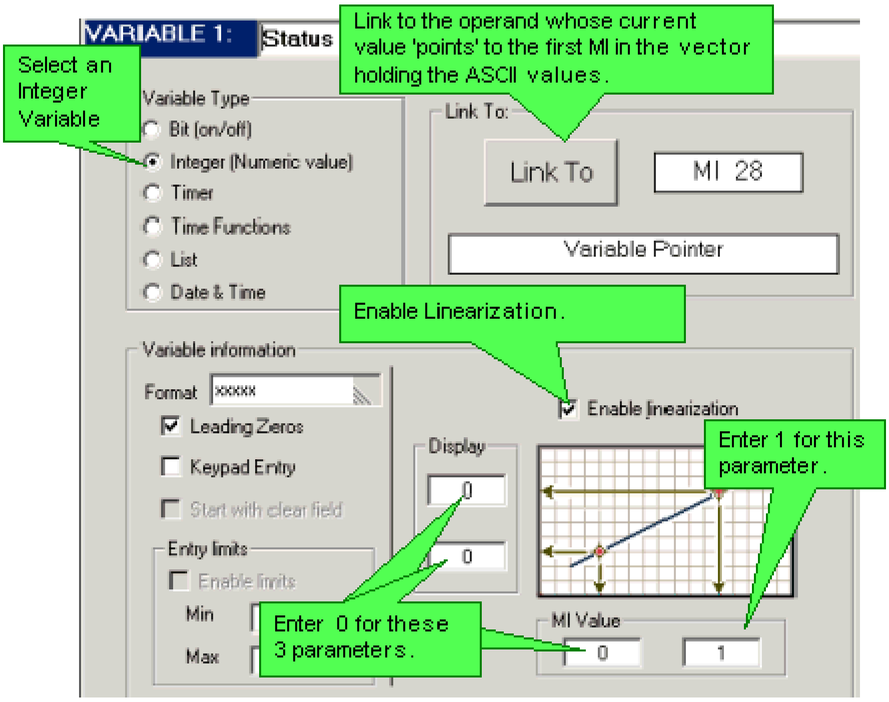


Note that the number of characters in the field is equal to the **length of the MI vector containing the characters**.

DISPLAY 1: Say Hello

Status: #####

2. Define the Variable as shown below.



VARIABLE 1: Status

Select an Integer Variable

Link to the operand whose current value 'points' to the first MI in the vector holding the ASCII values.

Enable Linearization.

Enter 1 for this parameter.

Enter 0 for these 3 parameters.

Variable Type: ● Integer (Numeric value)

Link To: MI 28

Variable Pointer

Format: xxxxx

☑ Leading Zeros

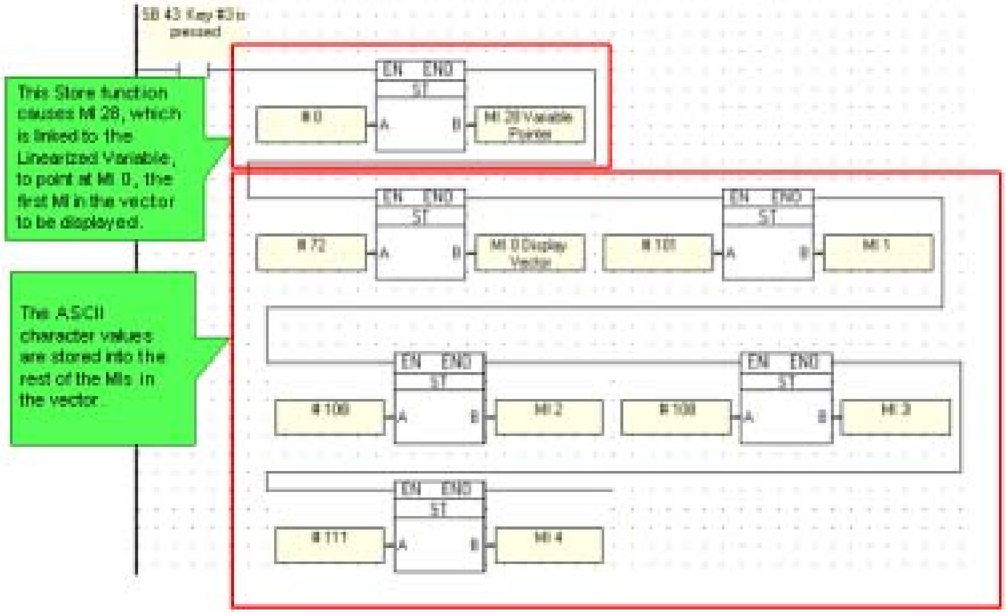
□ Keypad Entry

☑ Enable linearization

Display: 0, 0

MI Value: 0, 1

3. The Ladder net below sets the Variable pointer and stores ASCII values into the MI vector.



This Store function causes MI 28, which is linked to the Linearized Variable, to point at MI 0, the first MI in the vector to be displayed.

The ASCII character values are stored into the rest of the MIs in the vector.

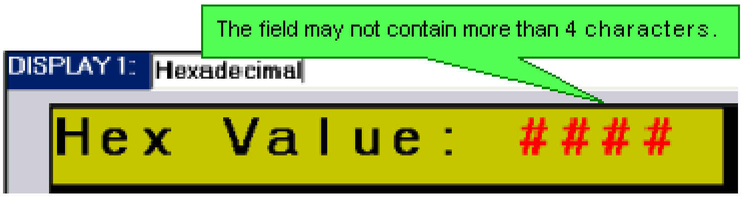
M-90
Status: Hello

When the application used in this example is downloaded, the ASCII characters 'Hello' will be displayed.

Register Value in Hexadecimal

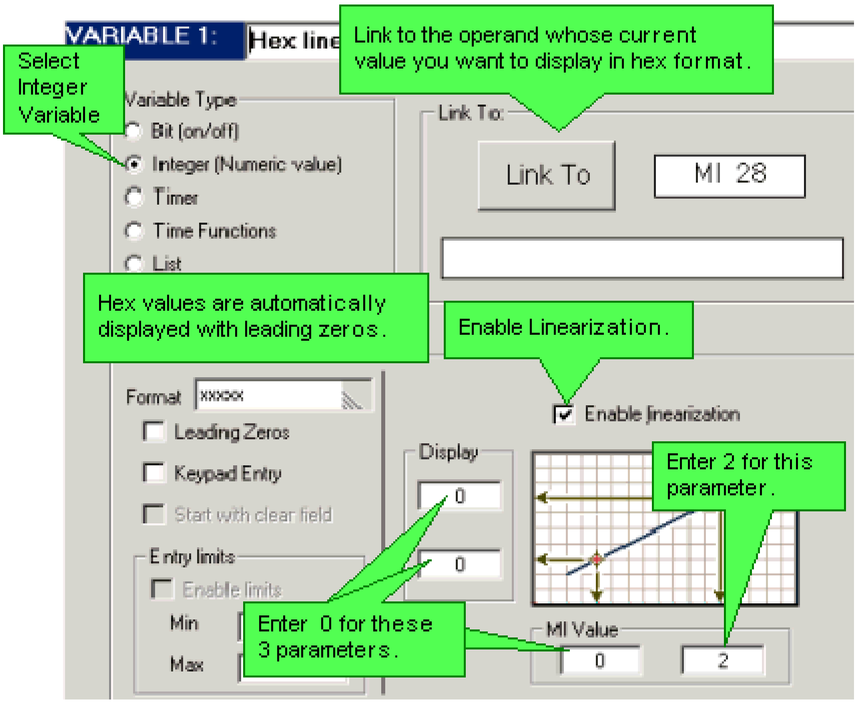
When the application shown in the example below is downloaded, the hexadecimal value of 63 will be displayed on the M90 screen.

1. Create a Variable Field in a Display, then attach a Variable. Note that if the field is too short, only the right-most characters are displayed. For example, the hex value 63(3F) cannot be shown in a field one character long.



The field may not contain more than 4 characters.

2. Define the Variable as shown below.



VARIABLE 1: Hex line

Select Integer Variable

Link to the operand whose current value you want to display in hex format.

Variable Type
- ○ Bit (on/off)
- ● Integer (Numeric value)
- ○ Timer
- ○ Time Functions
- ○ List

Link To: Link To MI 28

Hex values are automatically displayed with leading zeros.

Enable Linearization.

Format xxxxx
- □ Leading Zeros
- □ Keypad Entry
- □ Start with clear field

Entry limits
- □ Enable limits

Min

Max

☑ Enable linearization

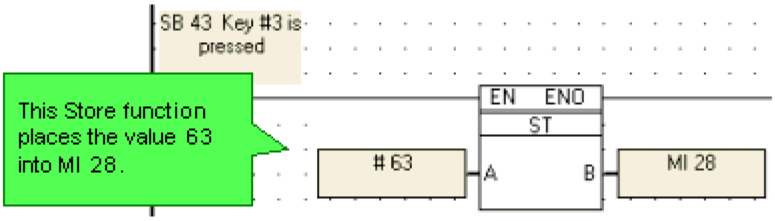
Display 0 0

Enter 2 for this parameter.

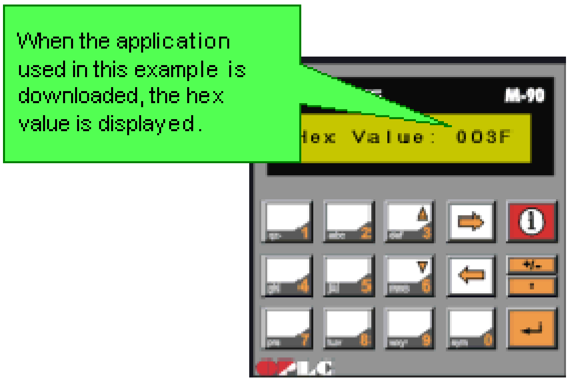
Enter 0 for these 3 parameters.

MI Value 0 2

3. The Ladder net below stores the value into the MI.



This Store function places the value 63 into MI 28.



When the application used in this example is downloaded, the hex value is displayed.

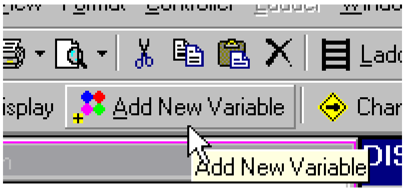
Showing an MI value on the controller's LCD

To display an MI value on the controller display:

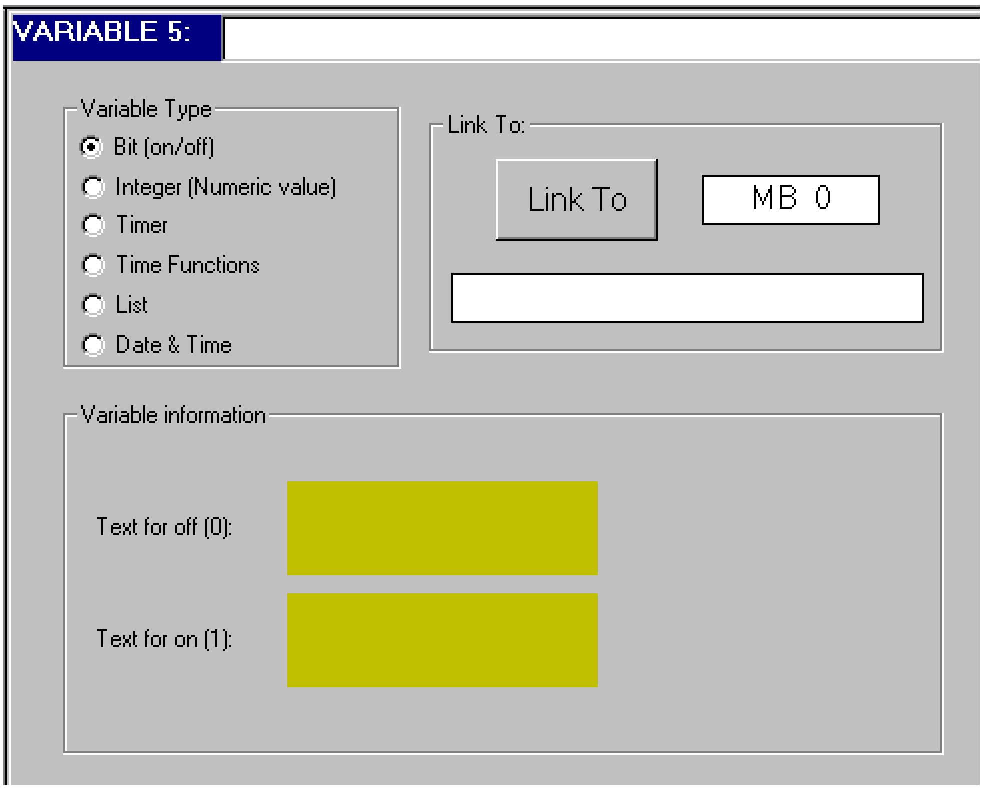
1. *Create a Variable*

   To create a new Variable:

   1. Click the Add New Variable icon on the HMI toolbar.



2. A new Variable opens in the Variable Editor.



VARIABLE 5:

Variable Type

● Bit (on/off)
○ Integer (Numeric value)
○ Timer
○ Time Functions
○ List
○ Date & Time

Link To:

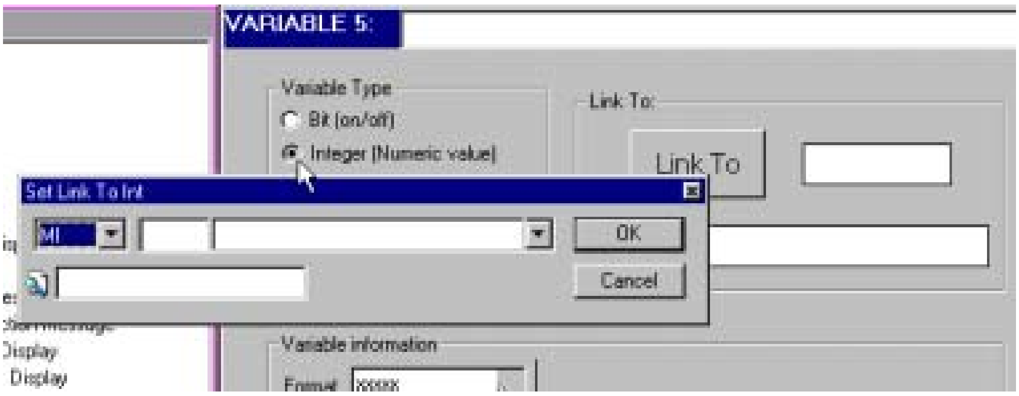
Link To: MB 0

Variable information

Text for off (0):

Text for on (1):

3. Select the desired Variable Type.



VARIABLE 5:

Variable Type

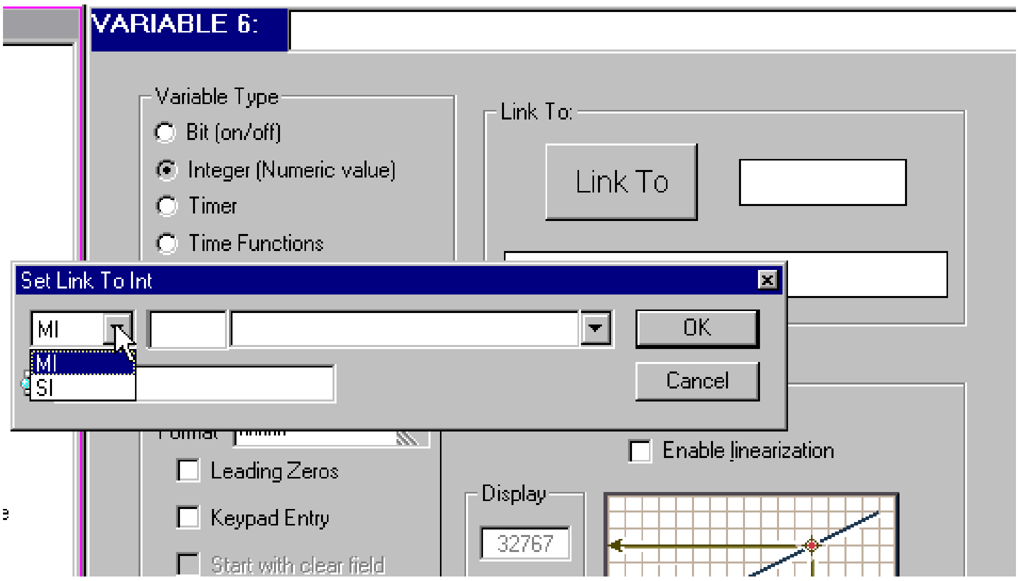
○ Bit (on/off)
● Integer (Numeric value)

Link To:

Set Link To Int

MI

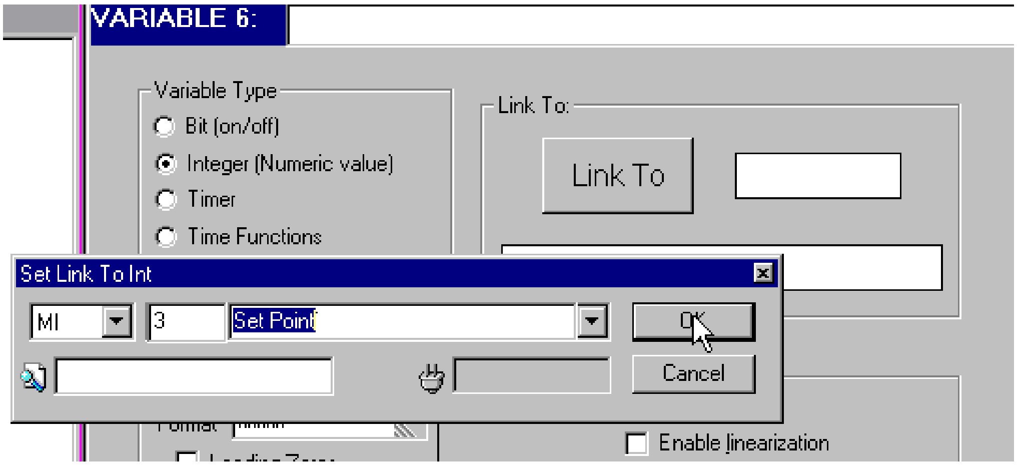
4. Select the Operand type.


VARIABLE 6:

Variable Type
○ Bit (on/off)
● Integer (Numeric value)
○ Timer
○ Time Functions

Set Link To Int
MI
MI
SI

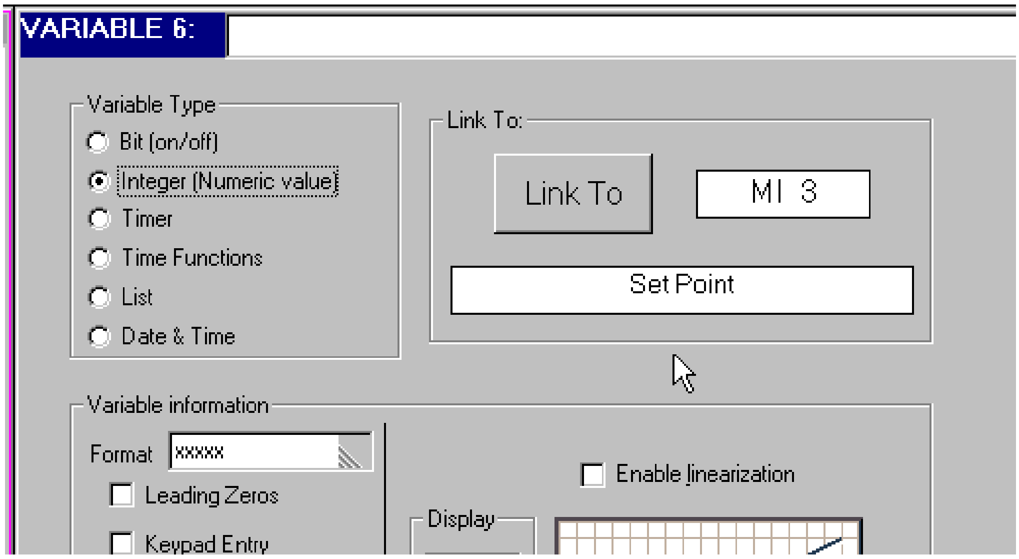
5. Enter the Operand Address and Symbol.


VARIABLE 6:

Variable Type
○ Bit (on/off)
● Integer (Numeric value)
○ Timer
○ Time Functions

Set Link To Int
MI 3 Set Point

6. The new Variable appears with the appropriate link in the Variable Editor.


VARIABLE 6:

Variable Type
○ Bit (on/off)
● Integer (Numeric value)
○ Timer
○ Time Functions
○ List
○ Date & Time

Link To:
Link To MI 3
Set Point

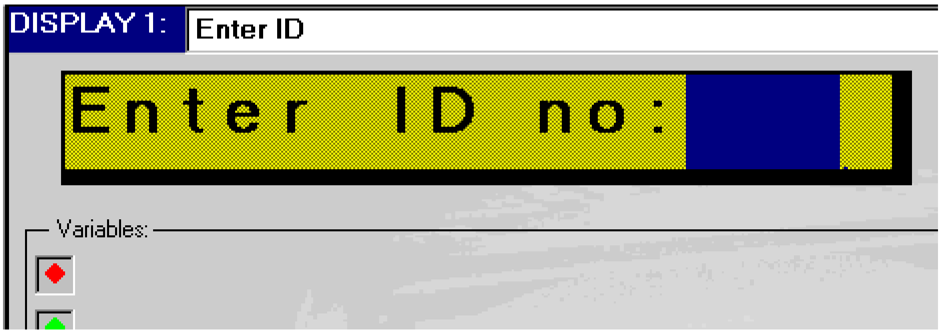
Variable information
Format xxxxx
□ Leading Zeros
□ Keypad Entry
□ Enable linearization

2. *Create a Variable Field in a Display* and attach it to the Variable.

To Create a Field

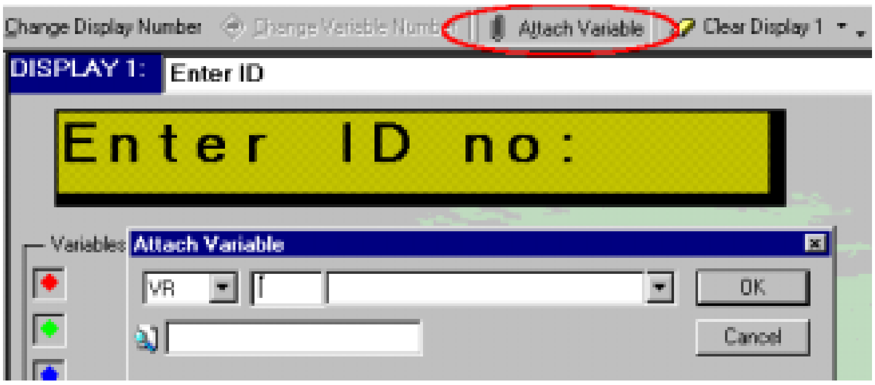
1. Click your cursor in the display. This is the starting point of the field.
2. To create the field, either:
3. Drag the cursor across the display. The field you create is automatically highlighted in blue.
   OR
4. Hold the SHIFT key down, and press the right-pointing arrow key. Each time you press the arrow key, a space is automatically highlighted in blue.

In the figure below, the display contains a field two spaces long.

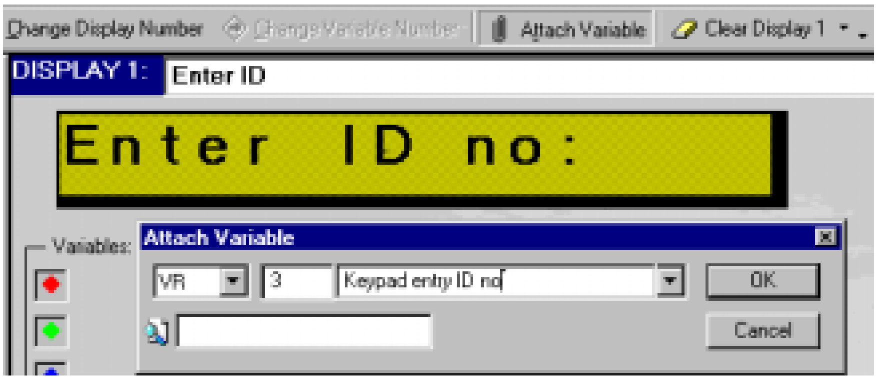


To Attach a Variable

5. Click Attach Variable on the HMI toolbar. The Attach Variable dialog box opens as shown below.

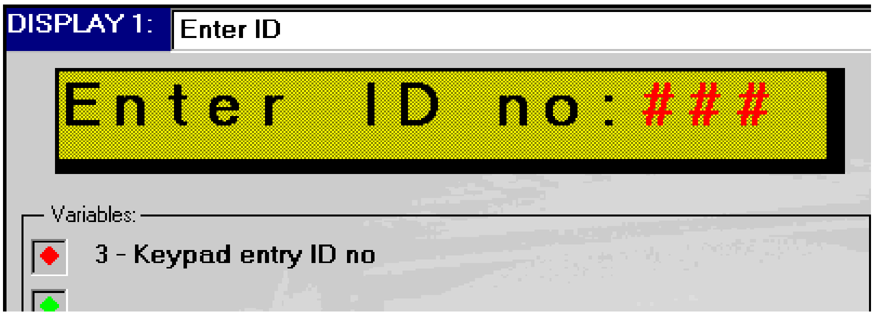


6. Enter the number of the desired variable as shown below and press OK. If you do not enter a variable number, the program assigns a default variable.



VR 3 Keypad entry ID no

7. The variable-linked spaces now appear as red pound signs, and the variable itself appears in the Variable pane of this Display as shown below.



DISPLAY 1: Enter ID

Enter ID no: ###

Variables:

3 - Keypad entry ID no

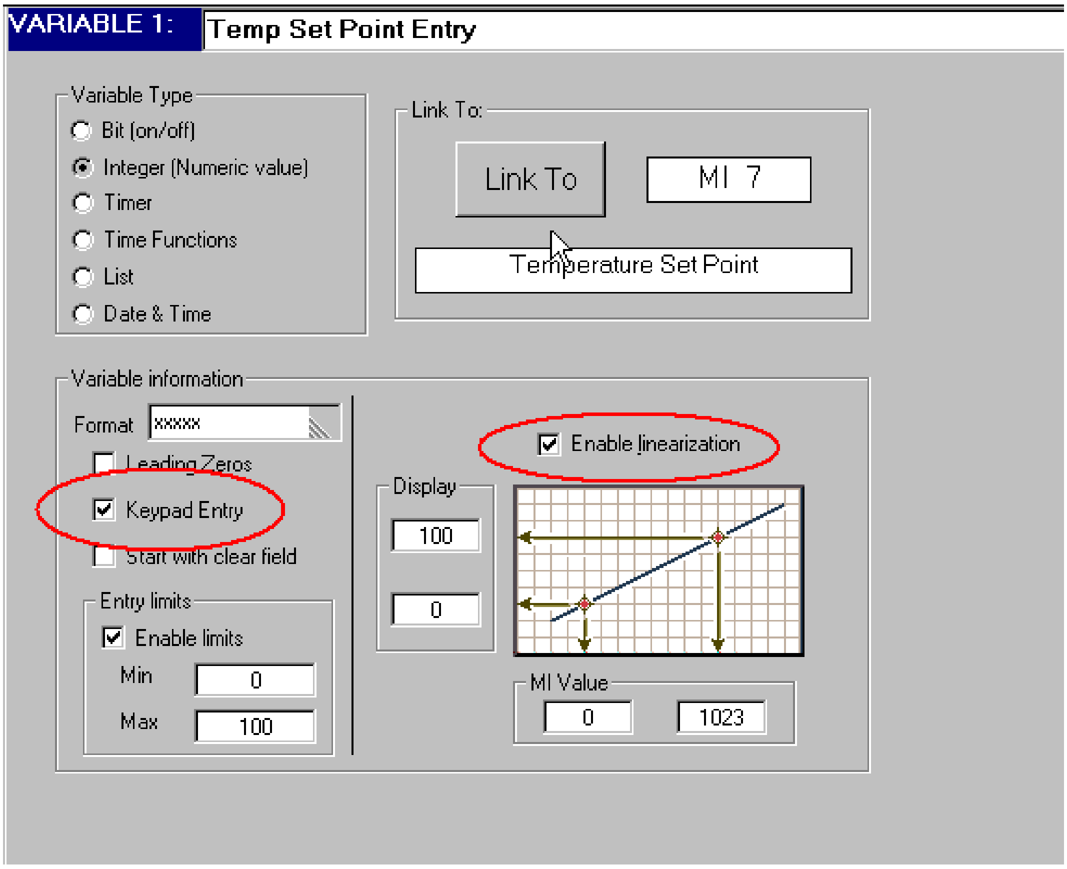
Converting Display values: Linearization

If you want to enter an Analog value, such as temperature, via the M90 keypad and convert that value into a Digital value for comparison with a digital value from a temperature probe, you use the **Enable Linearization** feature in the linked Variable.

This conversion process is Reverse Linearization.

To enable Analog to Digital conversion:

1. Create a Display for entering the analog value.
2. Create an Integer Variable.
3. Select **keypad entry** and **enable linearization**.
4. Enter the linearization values for the x and y axes.



VARIABLE 1: Temp Set Point Entry

Variable Type
- ○ Bit (on/off)
- ● Integer (Numeric value)
- ○ Timer
- ○ Time Functions
- ○ List
- ○ Date & Time

Link To: MI 7 — Temperature Set Point

Variable information

Format: xxxxx

- □ Leading Zeros
- ☑ Keypad Entry
- □ Start with clear field

Entry limits
- ☑ Enable limits
- Min: 0
- Max: 100

☑ Enable linearization

Display: 100, 0

MI Value: 0, 1023

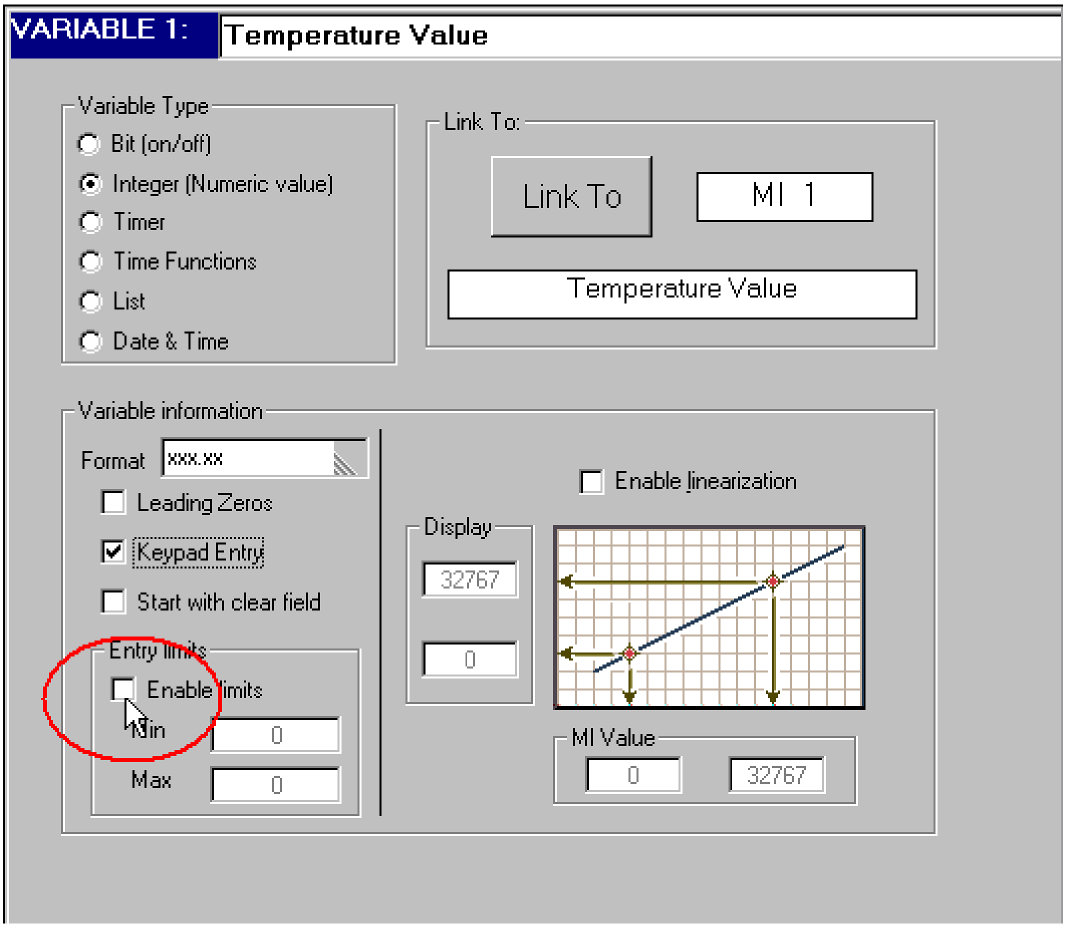
According to the above example:

- A temperature entry of 100^0 C will be converted to 1023 Digital value.
- A temperature entry of 50^0 C will be converted to 512 Digital value.

Limiting an MI keypad entry value

To limit an MI keypad entry value:

1. While creating an Integer Value, select **Keypad Entry** and **Enable Limits** in the Variable information window.



VARIABLE 1: Temperature Value

Variable Type

○ Bit (on/off)
● Integer (Numeric value)
○ Timer
○ Time Functions
○ List
○ Date & Time

Link To:

Link To MI 1

Temperature Value

Variable information

Format xxx.xx

□ Leading Zeros
☑ Keypad Entry
□ Start with clear field

Entry limits

□ Enable limits

Min 0

Max 0

□ Enable linearization

Display

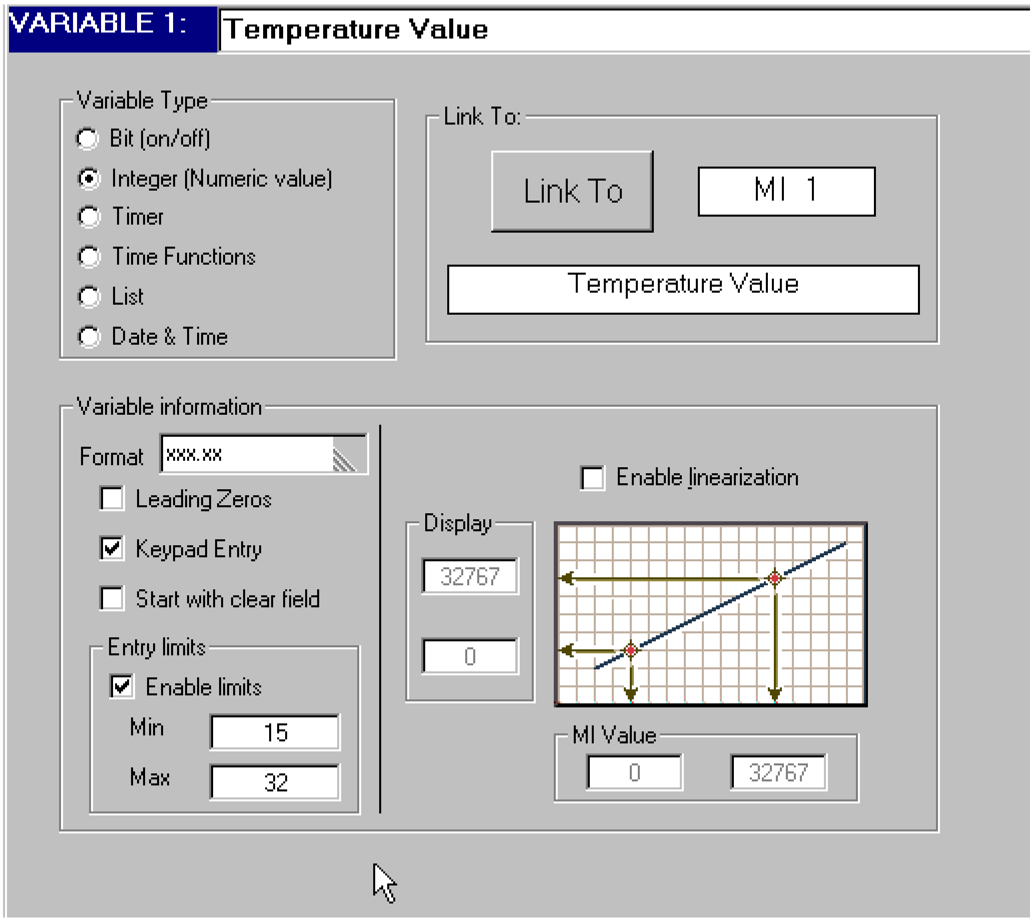
32767

0

MI Value

0 32767

2. Enter the **Minimum** and **Maximum** variable limits.



VARIABLE 1: Temperature Value

Variable Type
○ Bit (on/off)
● Integer (Numeric value)
○ Timer
○ Time Functions
○ List
○ Date & Time

Link To: MI 1
Temperature Value

Variable information
Format xxx.xx
☐ Leading Zeros
☑ Keypad Entry
☐ Start with clear field

Entry limits
☑ Enable limits
Min 15
Max 32

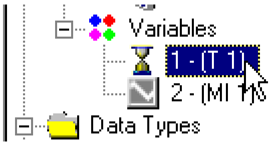
☐ Enable linearization

Display 32767 / 0

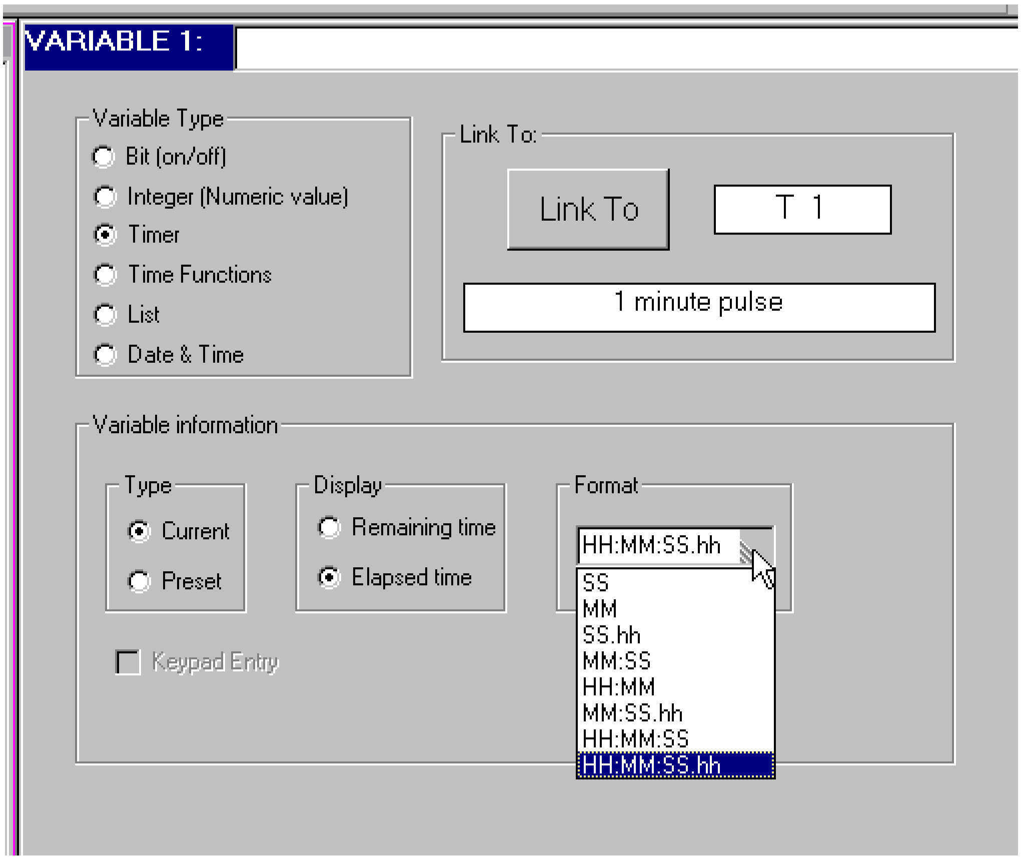
MI Value 0 / 32767

Selecting a Timer Display format

1. From the Navigator Window, create or choose an existing Timer Variable.



2. Open the Timer format drop-down menu in the Variable Editor.



VARIABLE 1:

Variable Type
○ Bit (on/off)
○ Integer (Numeric value)
● Timer
○ Time Functions
○ List
○ Date & Time

Link To:
Link To
T 1
1 minute pulse

Variable information

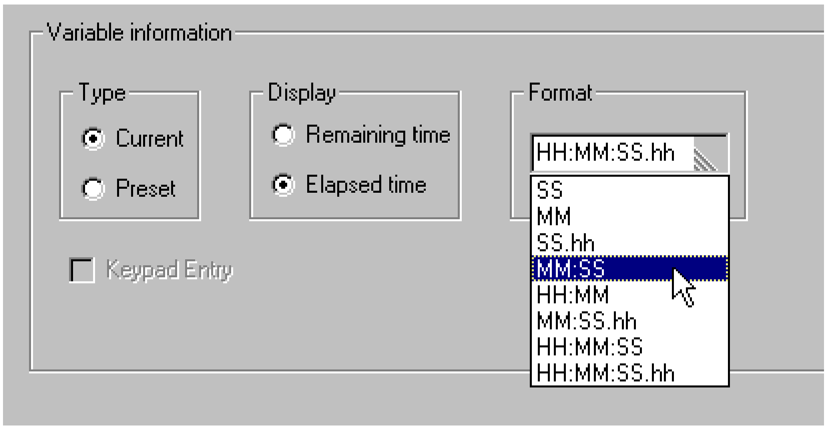
Type
● Current
○ Preset

Display
○ Remaining time
● Elapsed time

Format
HH:MM:SS.hh
SS
MM
SS.hh
MM:SS
HH:MM
MM:SS.hh
HH:MM:SS
HH:MM:SS.hh

□ Keypad Entry

3. Select the Timer format from the drop-down menu in the Variable Editor.



Variable information

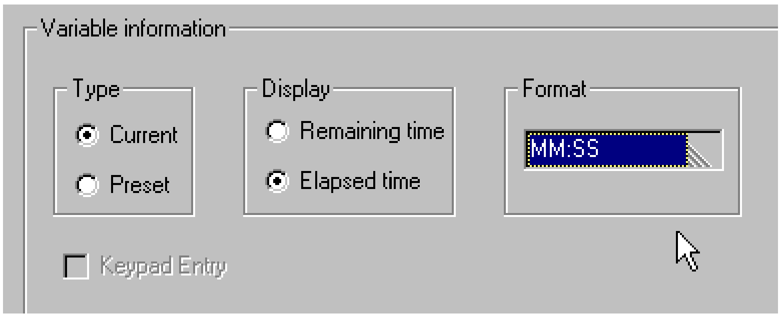
Type
● Current
○ Preset

Display
○ Remaining time
● Elapsed time

Format
HH:MM:SS.hh
SS
MM
SS.hh
MM:SS
HH:MM
MM:SS.hh
HH:MM:SS
HH:MM:SS.hh

□ Keypad Entry

4. The selected format is displayed in the Format window.



Variable information

Type
● Current
○ Preset

Display
○ Remaining time
● Elapsed time

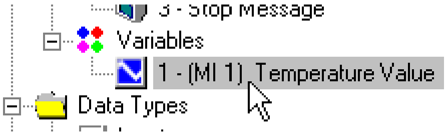
Format
MM:SS

□ Keypad Entry

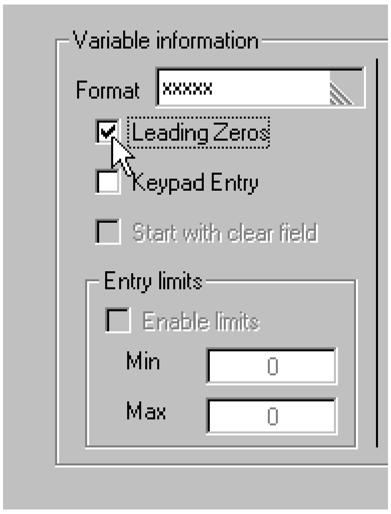
Displaying an MI value with a leading zero

To display an MI with a Leading Zero:

1. Select the desired Variable from the Navigator Window.



2. Select Leading Zeros from the Variable Information check box.



Variable information

Format: xxxxx

☑ Leading Zeros

□ Keypad Entry

□ Start with clear field

Entry limits

□ Enable limits

Min: 0

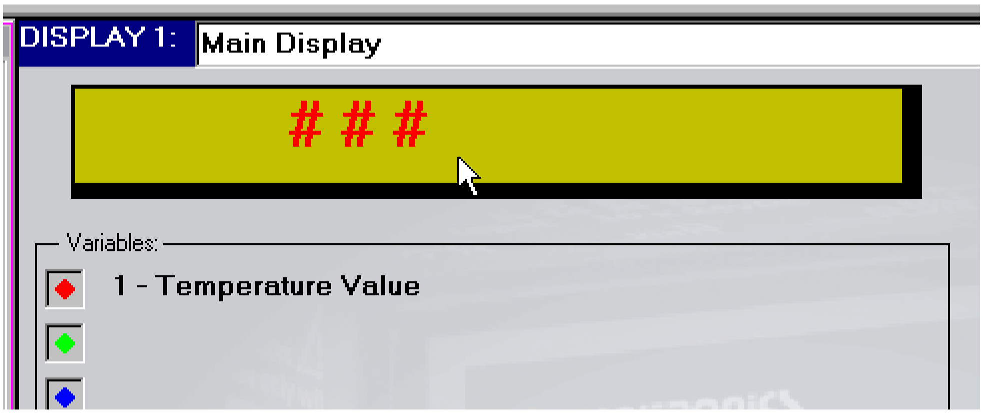
Max: 0

Displaying Special Symbols on the LCD

There are a number of Special Symbols such as arrows and degree signs, that may be displayed on the M90' LCD.

To enter a Special Symbol into a Display:

1. Choose the position in the Display field .



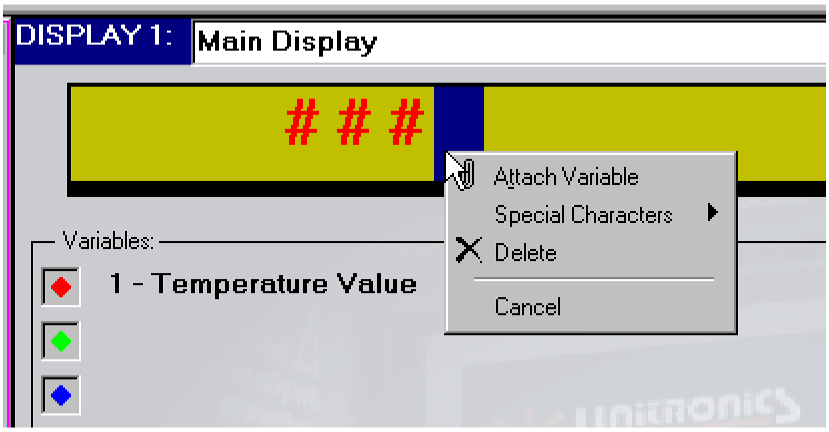
DISPLAY 1: Main Display

###

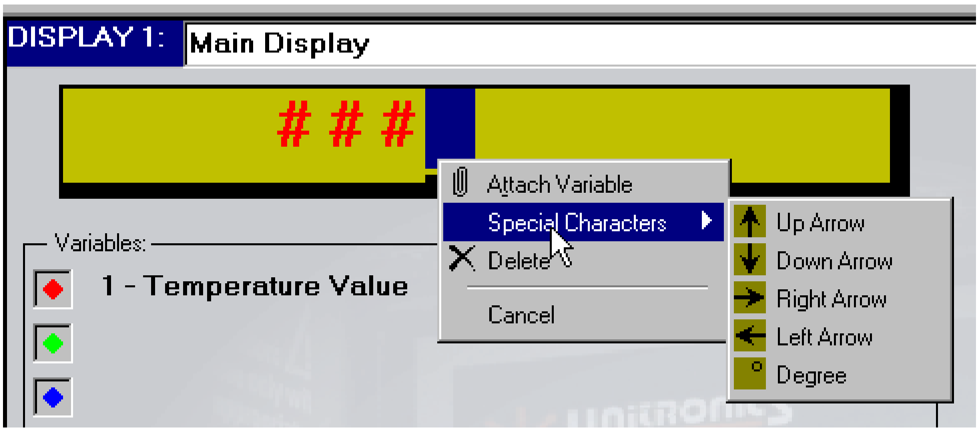
Variables:

1 - Temperature Value

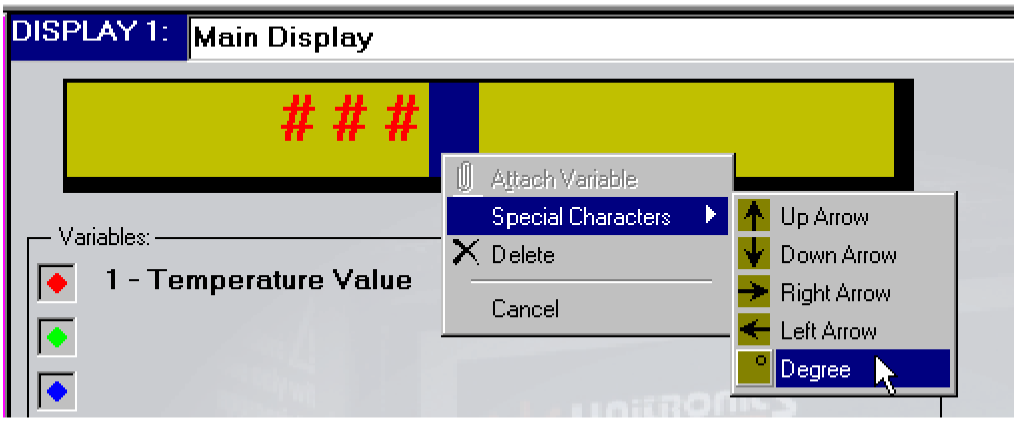
2. Right click to open the Variable modification menu.



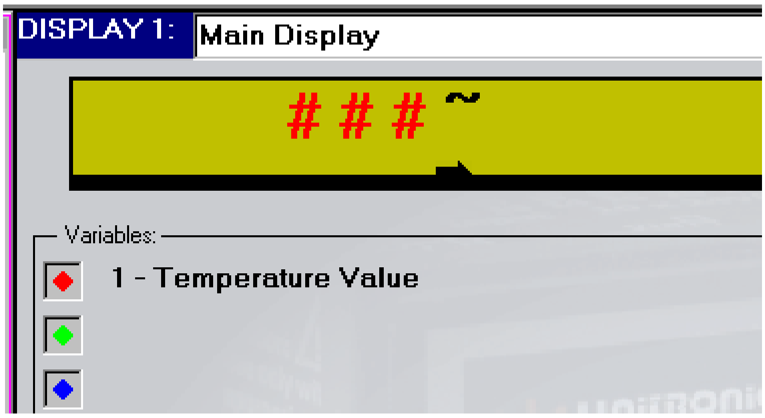
3. Select **Special Characters** from the menu. The Special Characters menu opens.



4. Select the Special Character you wish to add.



5. A ~ symbol will appear in the Display screen to show you that a Special Symbol was inserted. The selected symbol will appear on the controller.



Ladder

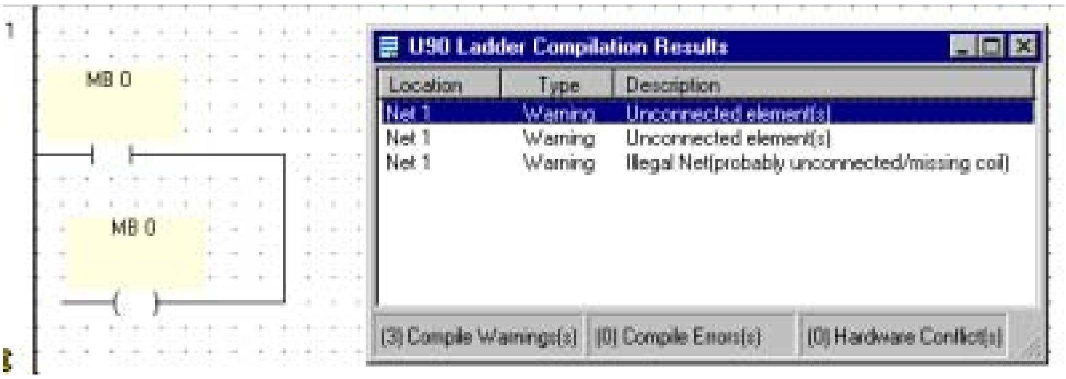
Ladder Net

A U90 Ladder net is the smallest division of a ladder diagram in Unitronics' U90 Ladder software.

Your first ladder element on the left must be connected to the left side of the ladder in each net. You do **not** need to connect the last element on the right to the right side of the ladder in each net.

You should place only one ladder rung on a Ladder net.

Power flows through the ladder elements in a net from left to right. If you build a net that would result in reverse power flow (right to left) the following error message occurs:

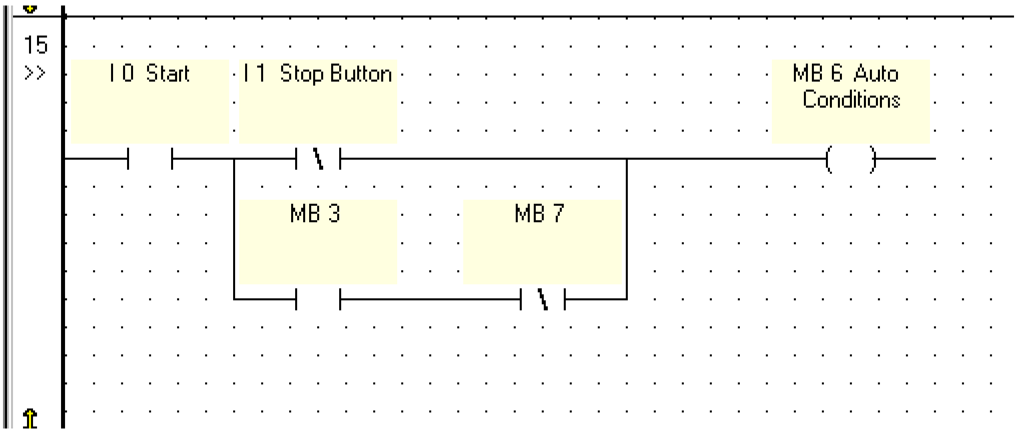


Location	Type	Description
Net 1	Warning	Unconnected element(s)
Net 1	Warning	Unconnected element(s)
Net 1	Warning	Illegal Net(probably unconnected/missing coil)

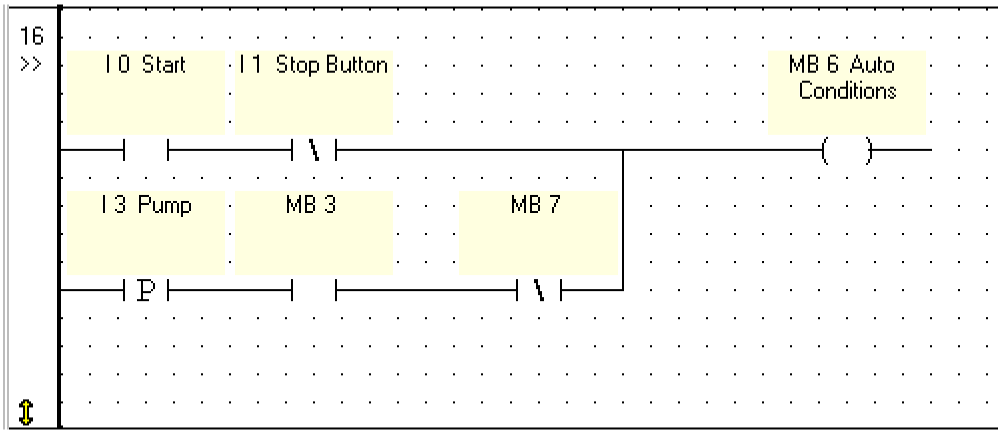
Placing more than one rung in a net may cause compiler problems in your project.

Examples:

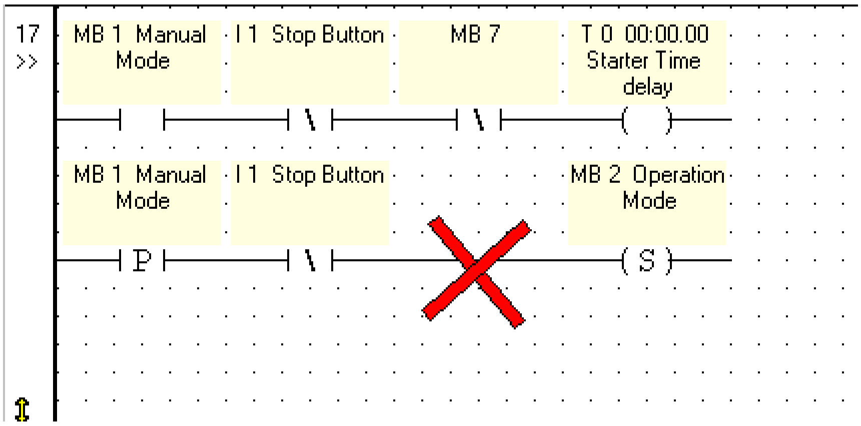
This net is constructed properly.



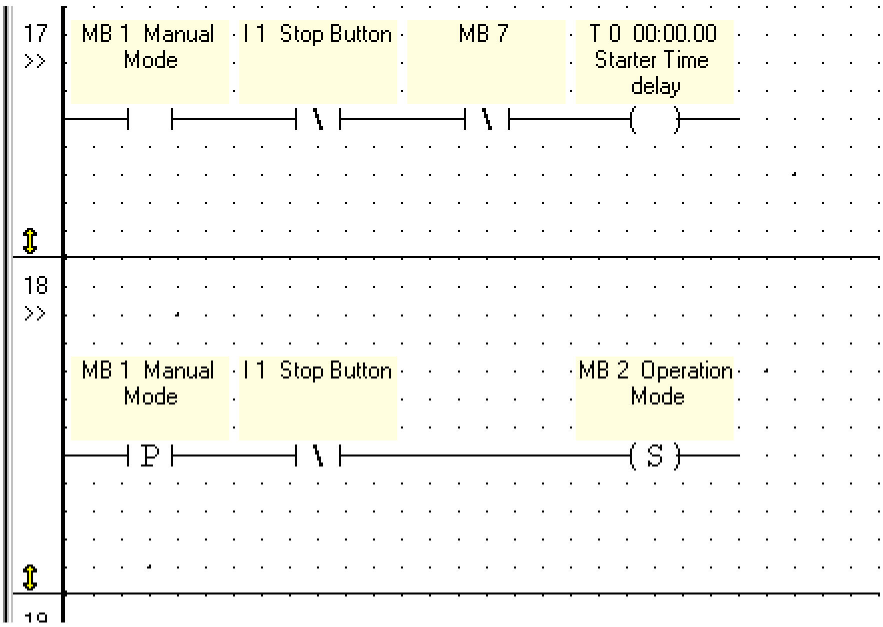
This net is constructed properly.



This net is improperly constructed and contains two rungs.



The rungs in the net below should be placed in two nets as shown below.:



Operands

An element's Operand is the form in which information is stored and operated on in the U90 Ladder program.

Operand lists are organized in categories, according to operand type:

- Input: **I** (according to model and expansion)
- Output: **O** (according to model and expansion)
- Memory Bit: **MB** (0 - 255)
- Memory Integer: **MI** (0 - 255)
- System Bit: **SB** (0 - 255)
- System Integer: **SI** (0 - 255)
- Timer:**T** (0 - 63)

Every Operand has an Address and a Symbol.

Symbols appear together with the operand every time the operand and address are used in the program. There are two types of symbols: preset and user-created.

- Preset symbols are descriptions that are connected to System Bits and System Integers.
- User-created symbols are descriptions that are written by the user for a specific project application. The user assigns a particular description to a particular operand.

U90 Ladder Elements

Contacts	Icon
Direct Contact (NO)	-| |-
Inverted Contact (NC)	-|\\\|-
Positive Transition (Rise)	-|P|-
Negative Transition (Fall)	-|N|-

Coils	Icon
Direct Coil	( )
Inverted (negated) Coil	(\)
Set Coil	(S)
Reset Coil	(R)

Compare Functions	Icon
Greater Than	>
Greater/Equal	>=
Equal	=
Not Equal	<>
Less/Equal	<=
Less Than	<

Math Functions	Icon
Add	+
Subtract	-
Multiply	*
Divide	/

Logic Functions	
AND	
OR	
XOR	

Clock Functions	Icon
Time	🕒
Day Of Week	7
Day Of Month	31
Month	📅
Year	📅

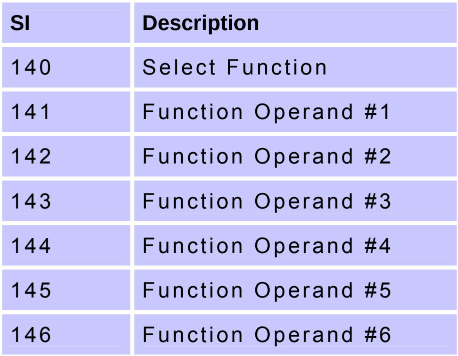
Functions

The following types of Function Blocks can be used in your program:

- Compare Functions
- Logic Functions
- Math Functions
- Store functions
- Clock Functions
- Loops: Jump to Label

Functions without Ladder elements

VisiLogic contains functions that are not represented by Ladder Elements. You can perform these functions by storing values into the System Integers listed here.



SI	Description
140	Select Function
141	Function Operand #1
142	Function Operand #2
143	Function Operand #3
144	Function Operand #4
145	Function Operand #5
146	Function Operand #6

To select the function type, first store the number of the function in SI 140, then use SI 141 to 146 to contain the data to be used in the function.

Note that when you run Test (Debug) Mode, the current value in SI 140 will **not** be displayed.

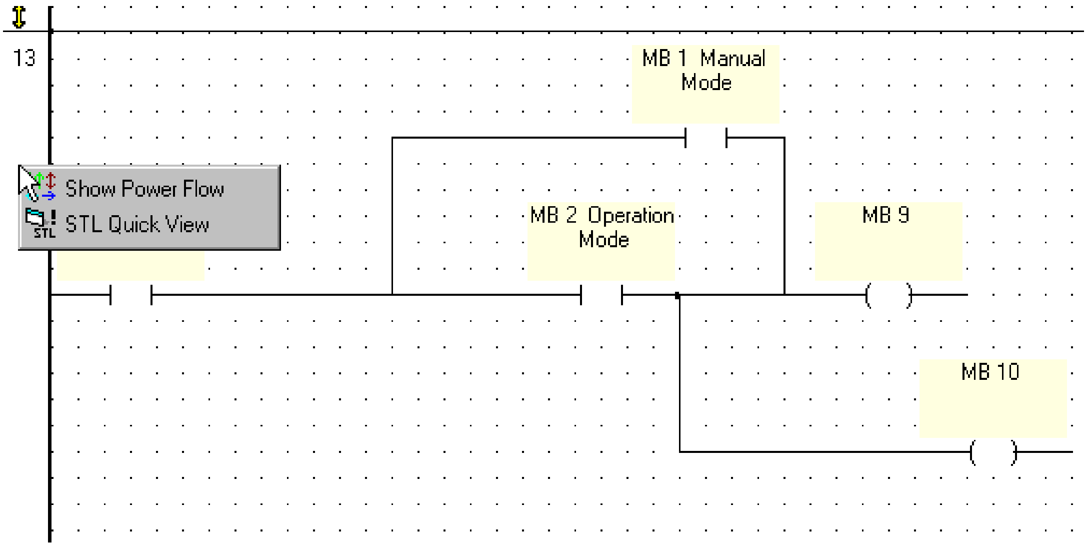
- Communication Utilities
- Interrupt
- Access indirectly addressed registers: Using the Database
- Load Indirect
- Load Timer Preset/Current Value
- Store Timer's Preset/Current Value
- SMS phone number: via MI Pointer
- Shift Register
- Copy Vector
- Copy MI to Output vector, Input vector to MI
- Fill Vector
- Convert MB to MI, MI to MB
- Linearization
- Find Mean, Maximum, and Minimum Values
- A*B/C
- Square Root

What is STL?

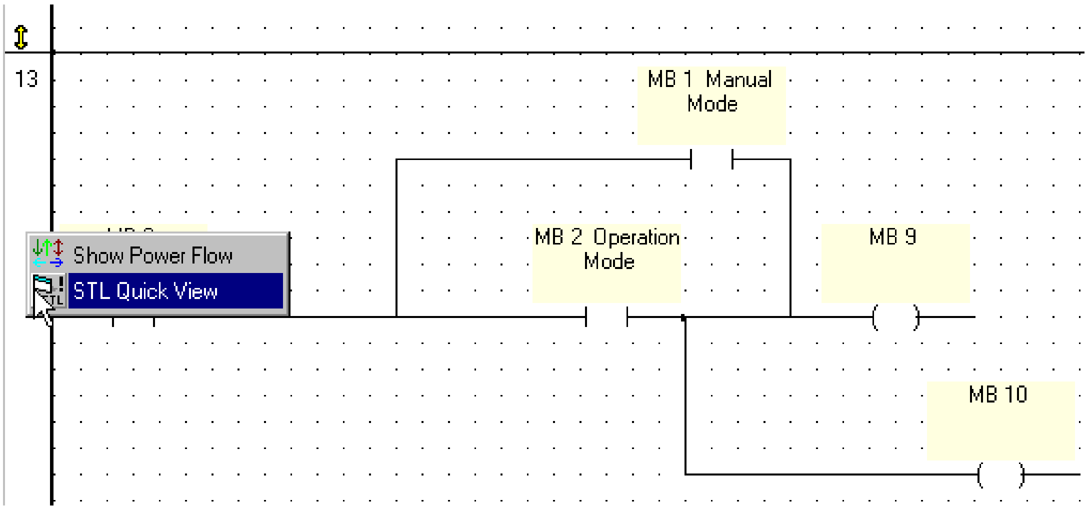
STL is a statement list that is created after you compile your project.

You can view the STL compiling result of net through the STL window.

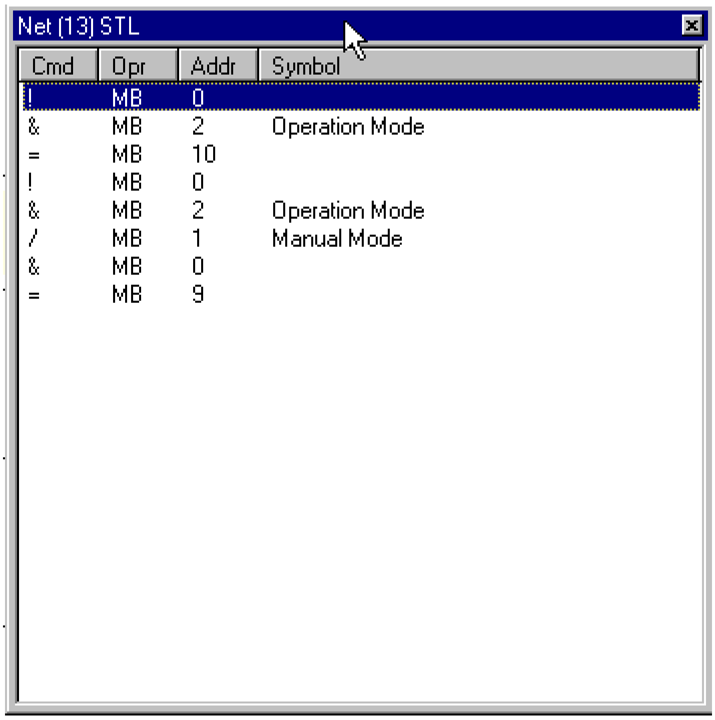
1. Right-click on the left net bar. The Compiling Result menu appears.



2. Select **STL Quick View**.



3. The **Net STL** window opens. The Net number appears in parentheses.



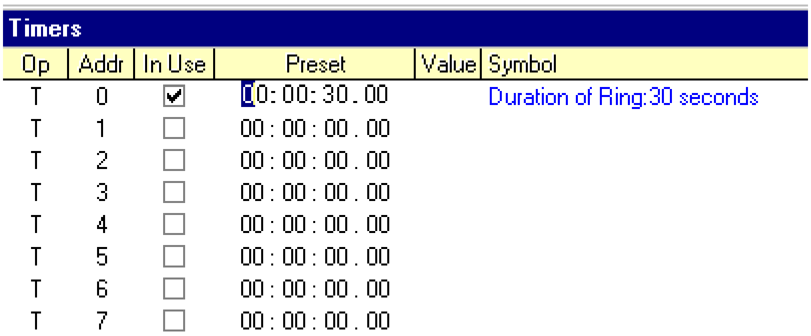
Net (13) STL

Cmd	Opr	Addr	Symbol
!	MB	0	
&	MB	2	Operation Mode
=	MB	10	
!	MB	0	
&	MB	2	Operation Mode
/	MB	1	Manual Mode
&	MB	0	
=	MB	9	

Timers (T)

U90 Ladder offers 64 On Delay Timers. Timers have a preset value, a current value, and a bit value. Timers always count **down** from the Preset Value.

Click on the Timers folder in the Program Navigation pane to display the complete list of Timers. Scroll down to view the complete list.



Timers

Op	Addr	In Use	Preset	Value	Symbol
T	0	☑	00: 00: 30. 00		Duration of Ring:30 seconds
T	1	□	00 : 00 : 00 . 00		
T	2	□	00 : 00 : 00 . 00		
T	3	□	00 : 00 : 00 . 00		
T	4	□	00 : 00 : 00 . 00		
T	5	□	00 : 00 : 00 . 00		
T	6	□	00 : 00 : 00 . 00		
T	7	□	00 : 00 : 00 . 00		

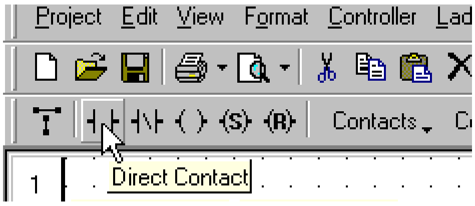
To place a Timer in your program, place a direct coil in a net, and select T.

Note that a Timer value can be displayed in a Display as a current or elapsed value.

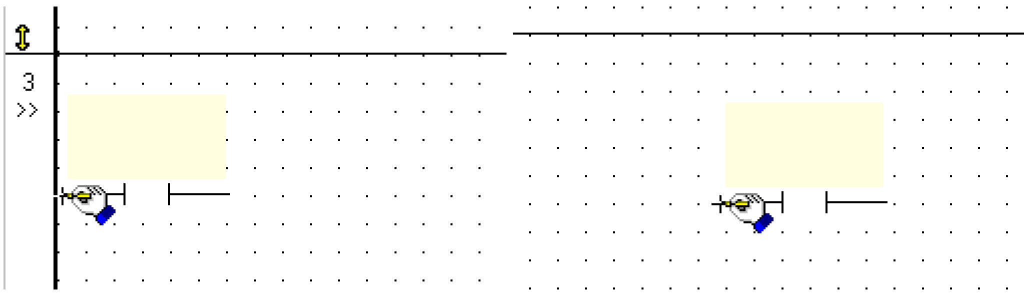
Placing Contacts & Coils

To place a Contact / Coil on a net:

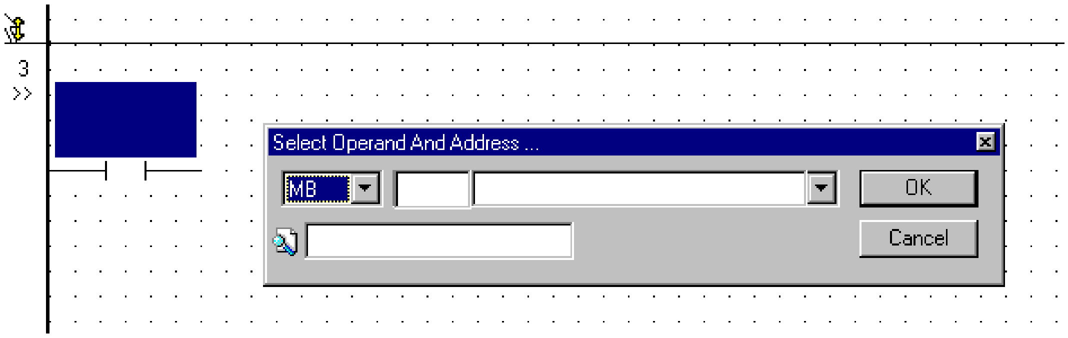
1. Click once to select the desired contact / coil.



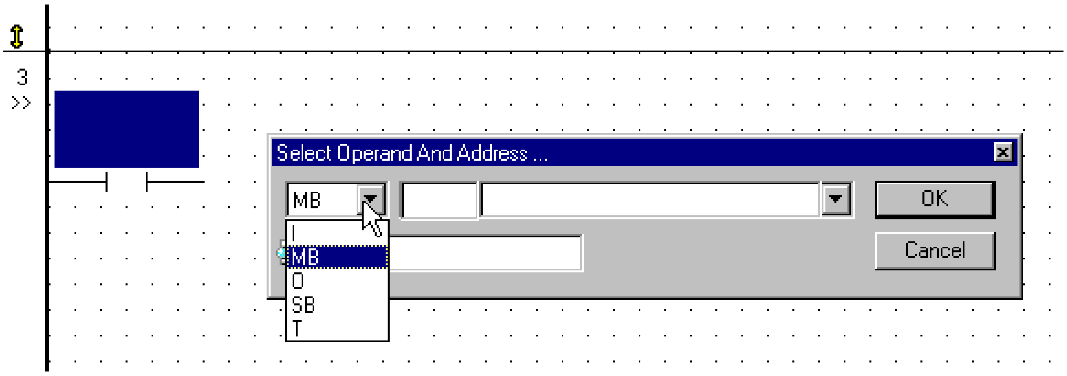
2. Move the element to the desired net position.



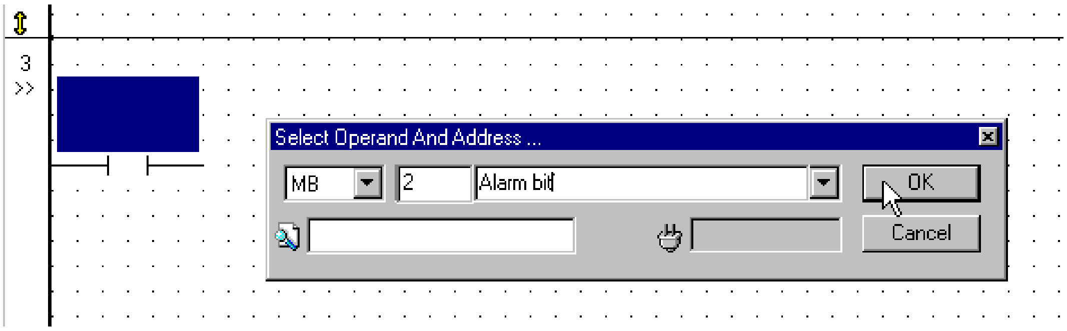
3. Click to place the element. The Operand and Address dialog box opens.



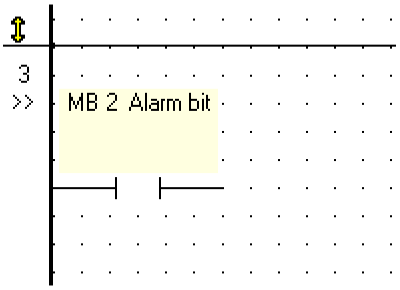
4. Select the Operand type from the drop - down menu.



5. Enter the Operand Address and symbol. Click OK.



6. The element appears on the net with the selected Operand Address and symbol



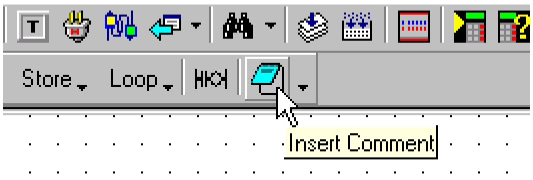
Comments Tool

You can insert comments into the Ladder Editor to label different parts of your program. Comments can be written in Notepad and added later to the project using **Cut** and **Paste** functions.

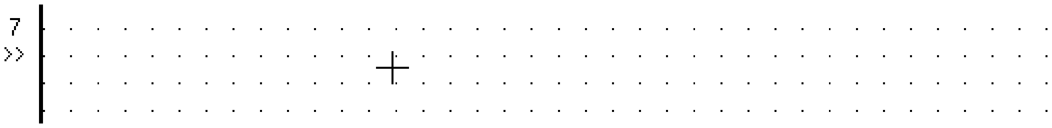
These Comments are 'internal' comments for the programmer(s). The Comments are not downloaded to or displayed on the controller.

To insert comments:

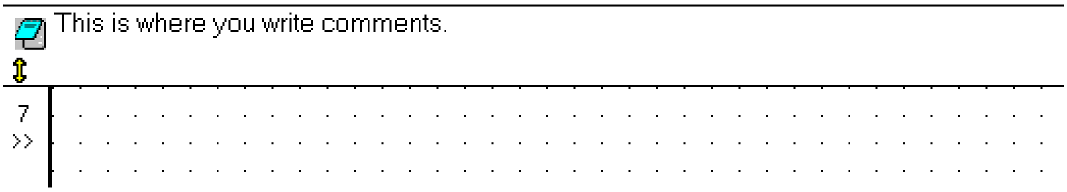
1. On the Ladder toolbar, click **Insert Comment** icon .



2. Move your cursor to the net in which you wish to insert a comment and click.



3. The Comment will appear above the net.
4. Type in your comments.



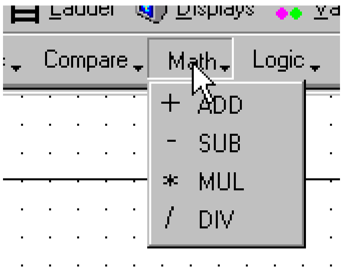
The length and content of your comments will have no effect on your project. They are not downloaded to the controller and do not affect the memory or word size of a project.

Placing a Function Block

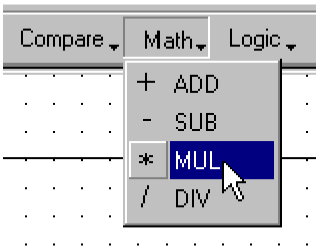
To place a Compare / Math / Logic function block on a net:

1. Click on the menu containing the desired type of function block,
   OR
   Right-click on a net to display the toolbar, then click on the desired menu;

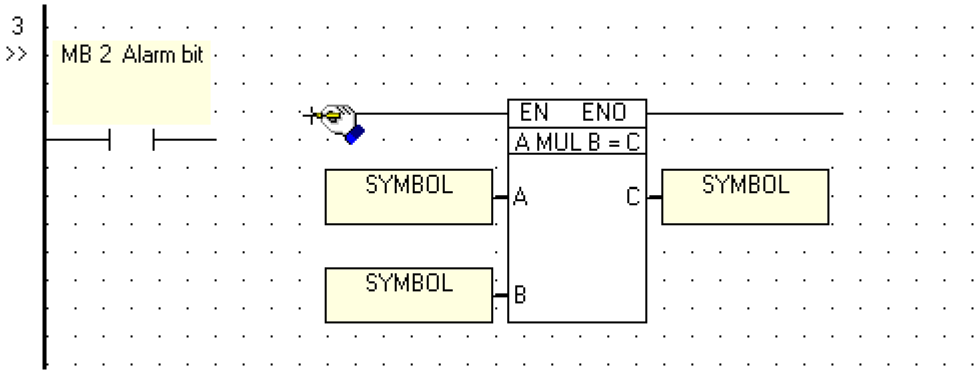
   the menu opens.



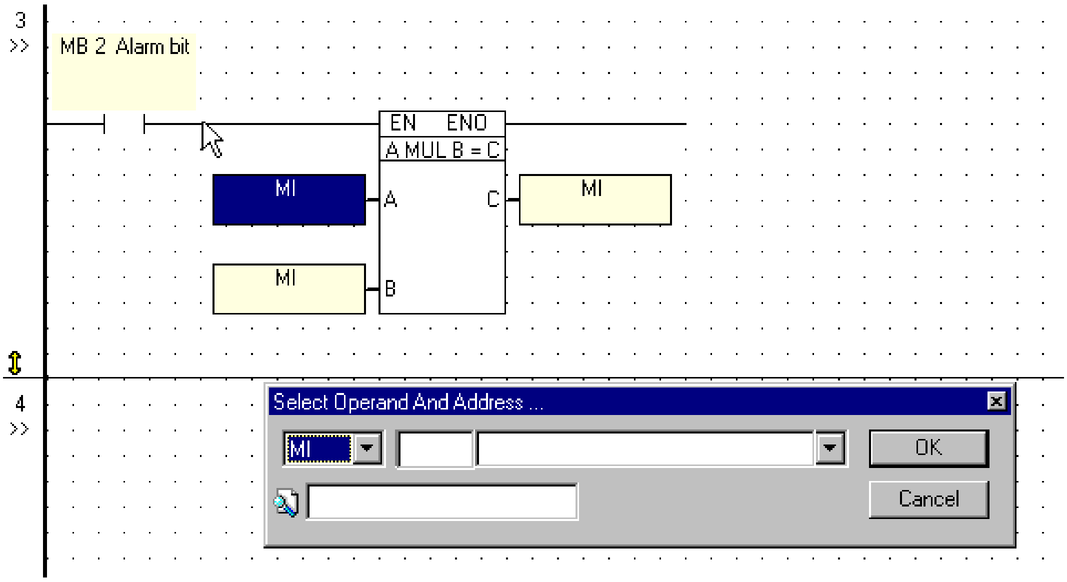
2. Select the desired operation.



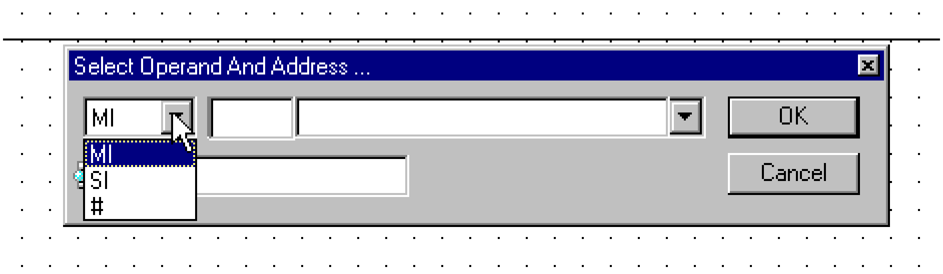
3. Move the function block to the desired net position.



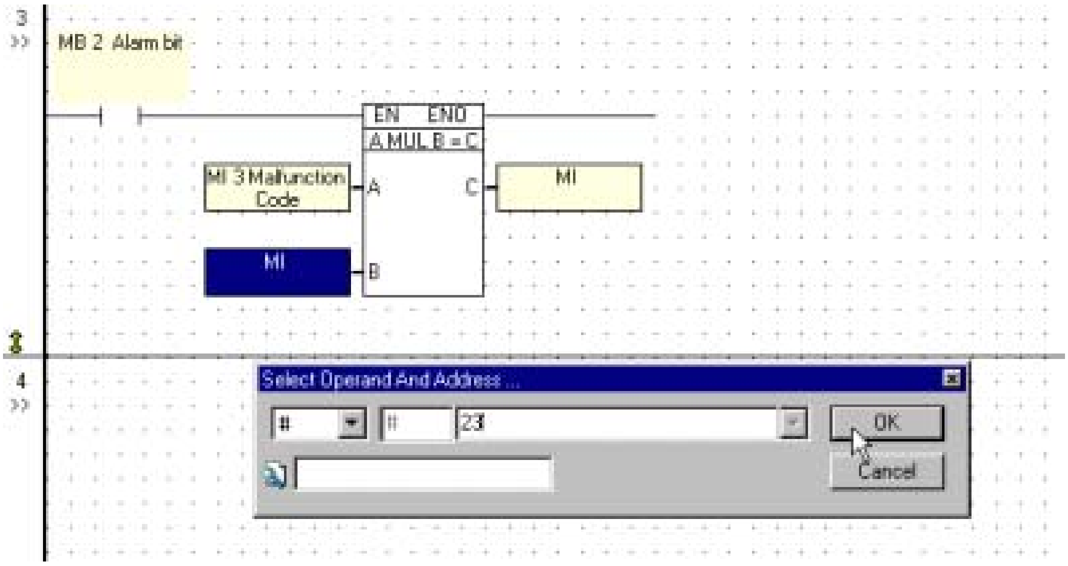
4. Click to place the element. The Operand Address and symbol dialog box opens.



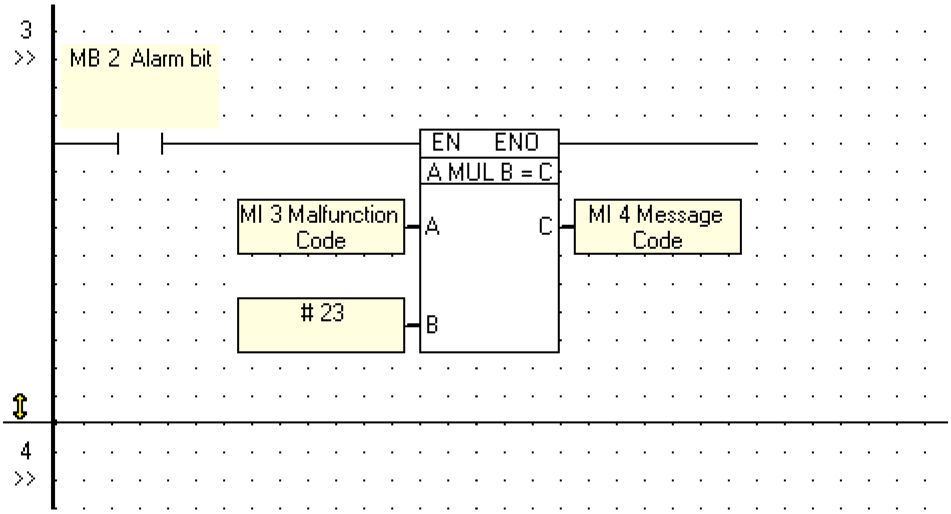
5. Select the desired Operand type.



6. Enter the Operand Address and symbol or constant value for each block variable. Click OK



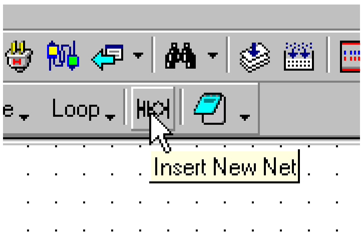
7. The function block appears on the net with the selected block variable values and symbols.



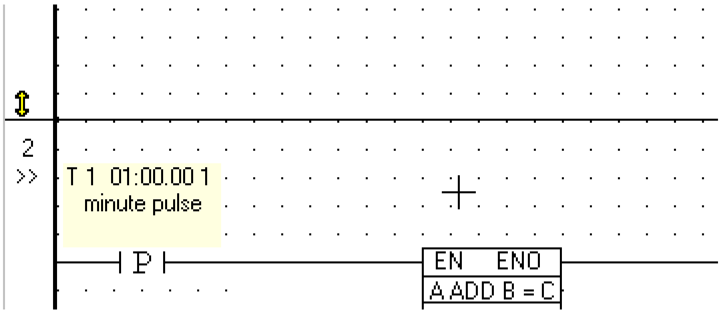
Inserting a new net between two existing nets

To insert a new net between 2 existing nets:

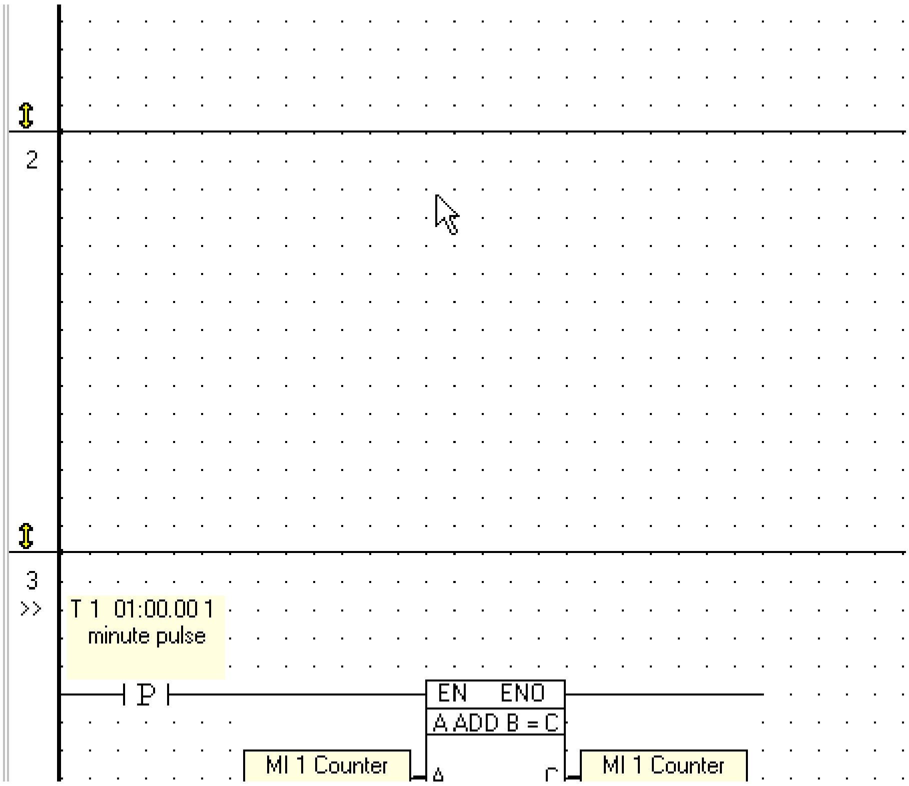
1. On the Ladder Toolbar, click **Insert New Net** .



2. Place your cursor in the spot where you want to insert the new net. Note that the net will be added above the net in which the cursor is located.



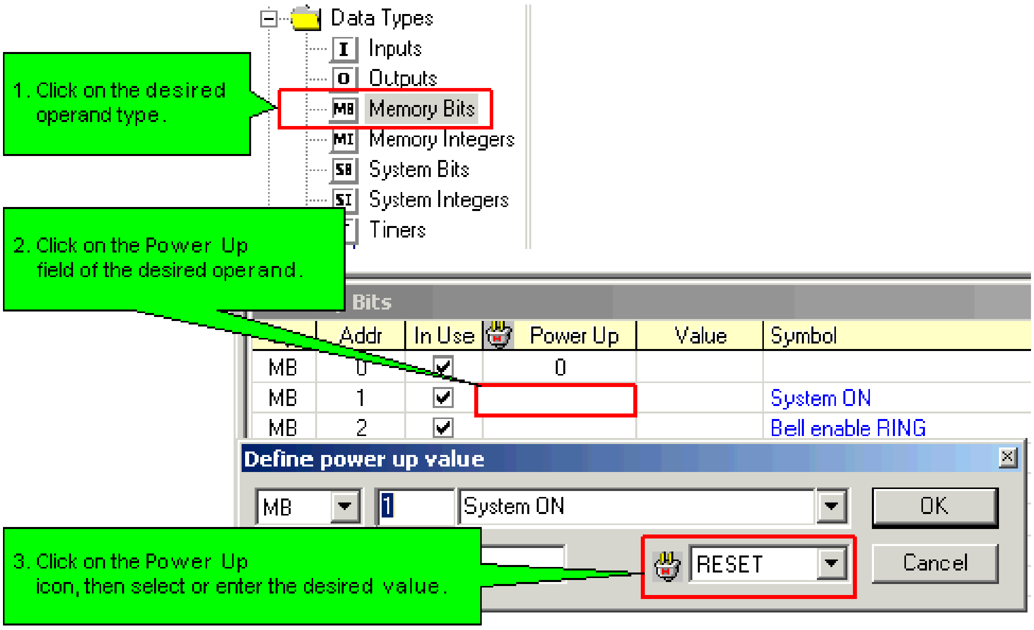
3. Click once. The new net is inserted.



Power-up

You can assign Power Up values to most Data Types. These values are written into the operand by the program when the controller is turned on. Outputs, MBs, SBs can be set or reset; integer values can be written into MIs and SIs.

You can assign Power Up values when you place an element into a net, or by opening a Data Type list as shown below.



1. Click on the desired operand type.

2. Click on the Power Up field of the desired operand.

3. Click on the Power Up icon, then select or enter the desired value.

Communication Utilities

Use this utility to enable your controller to receive data from external devices, such as bar-code readers, via an RS232 port. Since there is no Ladder element for this function; you perform it by storing values into SIs.

Note that the communication settings stored into these SIs only take effect at power-up.

SI	Parameter	Value to Store	Notes
141	STX (Start of Text)	0-255(ASCII) -1: No Start of Text (not recommended)	The STX parameter indicates where the data block begins. • Note that the ASCII character '/' (backslash) cannot be used to indicate the start of the data block.
142	ETX (End of Text)	0-255(ASCII) -1: ETX marked by Length -2: ETX marked by 'Silence'	The ETX parameter indicates where the data block ends. When the ETX is registered by the function, SB 60 turns ON. • If you use an ASCII character (0-255), note that if this character occurs **after** the Length parameter defined in SI 143, SB 60 turns ON. • Selecting -1 causes the function to use the **length** of a data block alone to determine its end. • Selecting -2 causes the function to use the **duration** of silent time following the STX to determine the end of a data block.
143	ETX Length or Silent	Length: up to 128 Silent: up to 24000	• This defines both the length of text, or silence, that signal the end of text. • Note that the duration of a silent 'counter' unit is approximately 2.509 mS. The 'silent' value should be lower than the M90 TimeOut value.

			• When defined as length, SI 143 cannot exceed SI 144.
144	Maximum Length	Up to 128	• This is the maximum legal length for received text. • When the maximum length is exceeded, the Receive Buffer is automatically cleared, and SB 60 is turned OFF, enabling new data to be received. • This can be used to detect buffer overflow.
145	Start Address: Receive Buffer	MI Address	This MI contains the start address for the vector of registers that serves as the Receive Buffer.
60	Number of Bytes currently in Receive Buffer	Read only	SI 60 indicates how many bytes of data are currently in the Receive Buffer.
61	Number of Bytes in Receive Buffer when SB 60=1	Read only	SI 61 indicates how many bytes of data are in the Receive Buffer when SB 60 turns ON.
146	Copy Data: Format	0: copy each received byte 1: copy in groups of 4 received bytes.	• **0** causes each separate byte to be copied to a separate register including STX and ETX. • **1** causes every **4** bytes to be copied to a single register, without the STX and ETX. This is used when the received data is in numeric format. For example 12345 would be copied to 2 consecutive MIs. The first MI would contain 1234, the second would contain 5.
140	Start receiving	300	In your application, use this to call the function after you have entered all of the other parameters. Note that when you run Test (Debug) Mode, the current value in SI 140 will **not** be displayed.

SB	**Description**	**Notes**
60	Data Successfully Received	Read only. Turns ON when the ETX condition is registered by the system.
61	Copy Data in Receive Buffer to MI Vector	Write only. • Turning this SB ON causes the buffer contents to be copied to the MI vector defined in SI 145. The data will be copied according to the format defined in SI 146. • If SI 146 is set to 0, this SB can be set at any time. If SI 146 is set to 1, this SB can be set after SB 60 turns ON.
62	Clear Receive Buffer, Clear SI 60, Clear SI 61, Reset SB 60	• This SB must be turned ON to enable a new message, or data block, to be received. • Turn this SB ON to enable data to be received **before** the maximum length, defined in SI 144, is exceeded.

Note that if no data is received for a period exceeding the M90 TimeOut, you will lose the data in the buffer.

To see how to use the Communications Utility, check the sample application **Read Card - Display Number Value.U90**. This may be found by accessing Sample U90 Projects from the Help menu.

This application demonstrates how to read a magnetic card number using an "IDTECH" card reader, then display that number on the M90's screen. The card reader transmits the number in ASCII characters in this format:

< %?[CR];xxxxx?[CR] > where **xxxxx** is the card number.

The ASCII character used to mark the Start Of Text (STX) is < **;** > (semicolon). End Of Text (ETX) is marked with the character < **?** > .

Since the card number is 5 digits long, the card number is copied to 2 separate MIs. The MIs are linked to 2 variables that are shown on the M90's screen in 2 separate Displays.

The parameters must be written into their respective operands using one scan condition. For this purpose, it is recommended to use SB 2 Power-up bit, as shown in the sample application.

Clock Functions

You perform clock and calendar functions in the U90 Ladder with Clock function blocks. Function blocks are provided for:

- Time
- Day of the Week
- Day of the Month
- Month
- Year

You activate these functions through the **Clock** drop-down menu of the Ladder toolbar.

The U90 Ladder provides 2 methods for executing Clock functions:

- Direct
- Indirect

You set the value of **Direct Clock** functions when you write your project.

The **user** sets the value of an **Indirect Clock** function from the M90 via the keypad.

Clock functions are featured in several sample applications, such as the applications ' School Bell Direct', 'Database Log', and 'Print & Time'. These applications may be found by selecting Sample U90 Projects from the Help Menu.

Interrupt

This function is time-based. You call an interrupt routine by storing 500 into SI 140. The interrupt function causes:

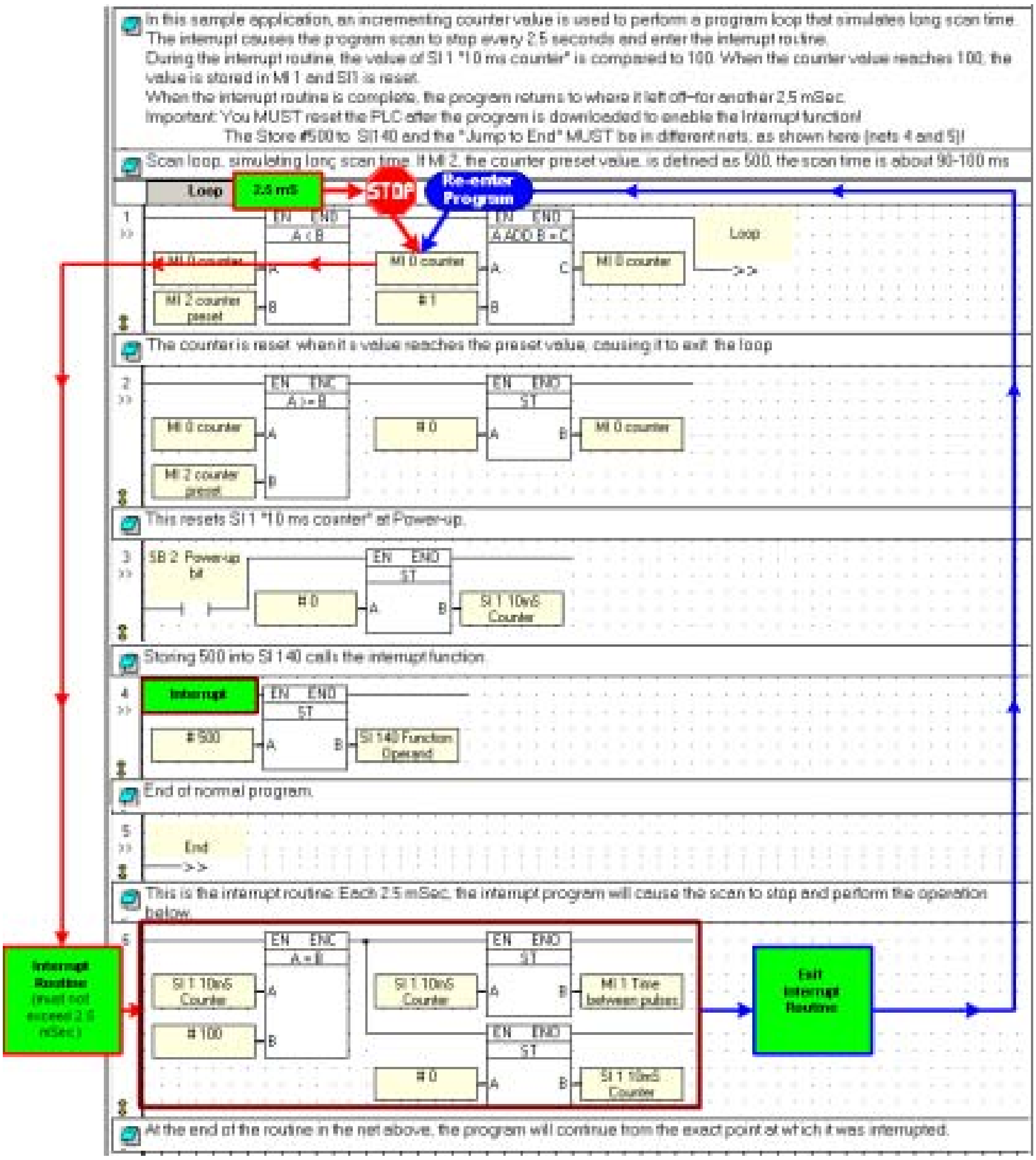
- The program scan to pause every 2.509 mSec. The interrupt causes the program to stop immediately without regard to the program scan, even if it occurs in the middle of a net.
- A jump to the net which follows the interrupt. The nets following the interrupt comprise the interrupt routine. Note that the interrupt routine should be as short as possible, and must not exceed approximately 0.5 mSec.
- When the interrupt routine is finished, the program continues from where it left off.

Note that the nets containing the Interrupt routine must be the last ones in the program. The format must be as shown in the example below:

- Store 500 into SI140 to call the function
- Jump to End
- The nets containing the actual interrupt routine.

Note that when you run Test (Debug) Mode, the current value in SI 140 will **not** be displayed.

Example



In this sample application, an incrementing counter value is used to perform a program loop that simulates long scan time. The interrupt causes the program scan to stop every 2.5 seconds and enter the interrupt routine.
During the interrupt routine, the value of SI 1 "10 ms counter" is compared to 100. When the counter value reaches 100, the value is stored in MI 1 and SI1 is reset.
When the interrupt routine is complete, the program returns to where it left off--for another 2.5 mSec.
Important: You MUST reset the PLC after the program is downloaded to enable the Interrupt function!
The Store #500 to SI140 and the "Jump to End" MUST be in different nets, as shown here (nets 4 and 5)!

Scan loop, simulating long scan time. If MI 2, the counter preset value, is defined as 500, the scan time is about 90-100 ms

The counter is reset when it s value reaches the preset value, causing it to exit the loop

This resets SI 1 "10 ms counter" at Power-up.

Storing 500 into SI 140 calls the interrupt function

End of normal program.

This is the interrupt routine. Each 2.5 mSec, the interrupt program will cause the scan to stop and perform the operation below.

At the end of the routine in the net above, the program will continue from the exact point at which it was interrupted.

Immediate: Read Inputs & HSC, Set/Reset Outputs

You can perform the following immediate actions, without regard to the program scan.

- Set SB 116 to immediately read the status of specific inputs and high-speed counter values. When SB 116 turns ON, the current input value written into linked SBs, current high-speed counter values are written into linked SIs.
- Set the appropriate SBs to immediately clear high-speed counter values.
- Set the appropriate SBs to immediately Set/Reset Outputs.

Note that:

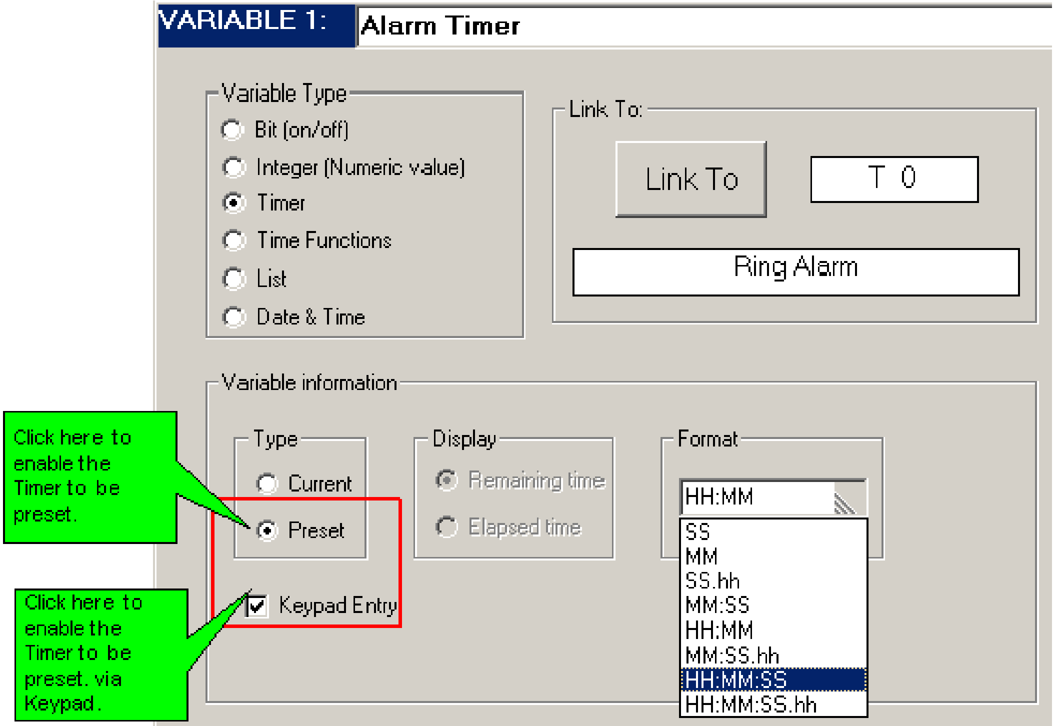
- Values are stored in linked SBs and SIs according to your controller model.
- In the Ladder, inputs and high-speed counters retain the values updated at the beginning of the scan. Only the linked operands listed below are immediately updated. However, immediate changes in output status are immediately updated in the Ladder.

Use the table below to determine which actions, SBs, and SIs are relevant to your model controller.

M90 Model	Input #	Value stored in:	HSC #	Value stored in	HSC #	Immediate Clear	Output #	Set/ Reset via:
M90-T	I 6 I 7	SB 112 SB 113	HSC 0	SI 44	HSC 0	SB 117	None	
M90-T1 M90-T1-CAN	I 8 I 9 I 10 I 11	SB 110 SB 111 SB 112 SB 113	HSC 0	SI 44	HSC 0	SB 117	O 8 O 9 O 10 O 11	SB 120 SB 121 SB 122 SB 123
M90-19-B1A M90-R1 M90-R1-CAN M90-R2-CAN M90-TA2-CAN	I 8 I 9	SB 112 SB 113	HSC 0	SI 44	HSC 0	SB 117	None	
M91-19-TC2 M91-19-UN2	I 0 I 1 I 2 I 3	SB 110 SB 111 SB 112 SB 113	HSC 0 HSC 1	SI 44 SI 45	HSC 0 HSC 1	SB 117 SB 118	O 0 O 1 O 10 O 11	SB 120 SB 121 SB 122 SB 123
M91-19-R1 M91-19-R2 M91-19-R2-CAN	I 0 I 1 I 2 I 3 I 4 I 5	SB 110 SB 111 SB 112 SB 113 SB 114 SB 115	HSC 0 HSC 1 HSC 2	SI 44 SI 45 SI 46	HSC 0 HSC 1 HSC 3	SB 117 SB 118 SB 119	O 0 O 1 O 2	SB 120 SB 121 SB 122

Presetting Timers via Keypad

You can choose to set a timer via the M90 keypad.



VARIABLE 1: Alarm Timer

Variable Type
○ Bit (on/off)
○ Integer (Numeric value)
● Timer
○ Time Functions
○ List
○ Date & Time

Link To:
Link To T 0
Ring Alarm

Variable information
Type: ○ Current ● Preset
☑ Keypad Entry
Display: ● Remaining time ○ Elapsed time
Format: HH:MM
SS
MM
SS.hh
MM:SS
HH:MM
MM:SS.hh
HH:MM:SS
HH:MM:SS.hh

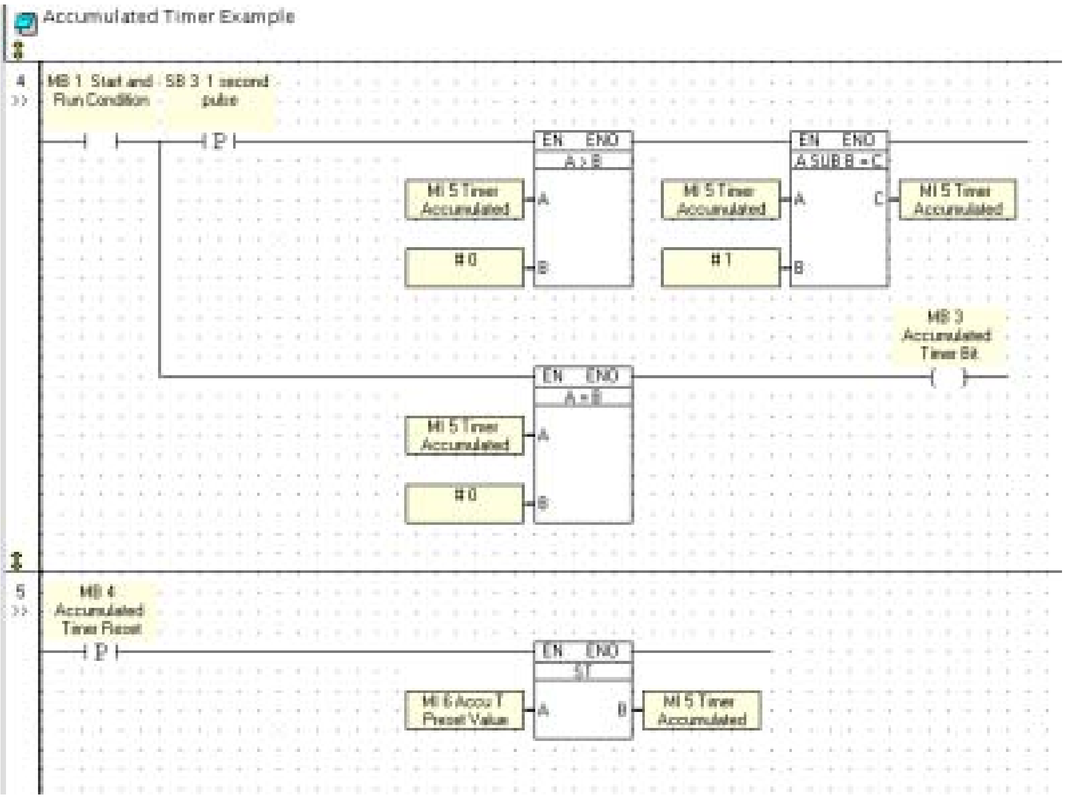
Click here to enable the Timer to be preset.

Click here to enable the Timer to be preset. via Keypad.

Counting accumulated time

The M90 built-in Timers return to their preset time when the Timers' Start and Run condition goes to logic 0 (OFF). This feature prevents you from accumulating Timer times.

If you want to measure accumulated time, you must build the Ladder logic to do this.



According to the above example:

- MI 5 - Accumulated Timer value.
- MI 6 - Accumulated Timer preset value.
- MB 4 - Reset bit.
- MB 1 - Start and Run Condition bit

Use SB 3 - 1 second pulse to advance the time count.

Find Mean, Maximum, and Minimum Values

This function enables you to take a vector of registers and find the:

- Mean of all the values in the vector,
- Minimum value in the vector,
- Maximum value in the vector.

You can base the function on a vector of MI registers or Database registers by selecting the appropriate function.

Note that since there is no Ladder element for this function; you perform it by storing values into:

- SI 141 to determine the start of the vector,
- SI 142 to determine the length of the vector,
- SI 140 to select the type of function. Storing the function number calls the function. In your application, call the function **after** you have entered all of the other parameters.

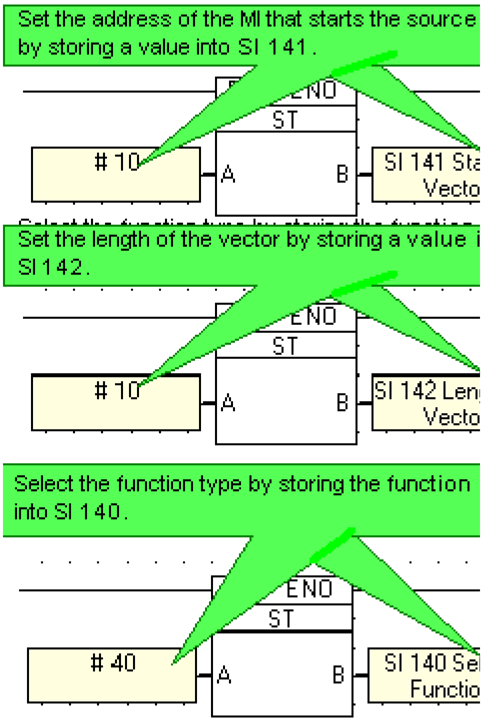
The results will be placed in:

- SI 143: Mean
- SI 144: Minimum

- SI 145: Maximum

Note that if a remainder value results from the division operation used to calculate the Mean, that remainder value will be place in SI 4, Divide Remainder.

To use this function:



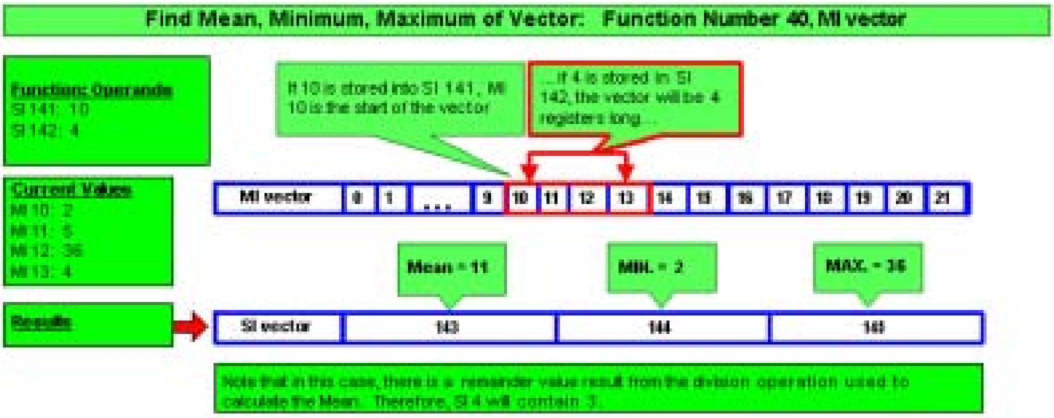
Set the address of the MI that starts the source by storing a value into SI 141.

Set the length of the vector by storing a value i SI 142.

Select the function type by storing the function into SI 140.

Function Number (SI 140)	Description
40	Find Mean, Minimum, Maximum in MI vector
41	Find Mean, Minimum, Maximum in DB vector

Note that when you run Test (Debug) Mode, the current value in SI 140 will **not** be displayed.



Find Mean, Minimum, Maximum of Vector: Function Number 40, MI vector

Function: Operands
SI 141: 10
SI 142: 4

Current Values
MI 10: 2
MI 11: 5
MI 12: 36
MI 13: 4

Results

If 10 is stored into SI 141, MI 10 is the start of the vector

...if 4 is stored in SI 142, the vector will be 4 registers long...

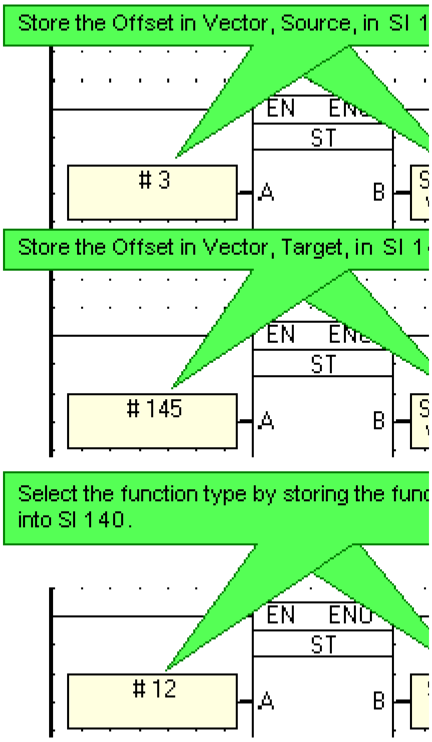
Note that in this case, there is a remainder value result from the division operation used to calculate the Mean. Therefore, SI 4 will contain 3.

Load Indirect

Load Indirect allows you to take a value contained in a **source** operand and load that value into a **target** operand using indirect addressing. Note that since there is no Ladder element for this function; you perform it by storing values into:

- SI 141 to determine the data source,
- SI 142 to determine the load target,
- SI 140 to select the type of function. Storing the function number calls the function. In your application, call the function **after** you have entered all of the other parameters.

To use Load Indirect:



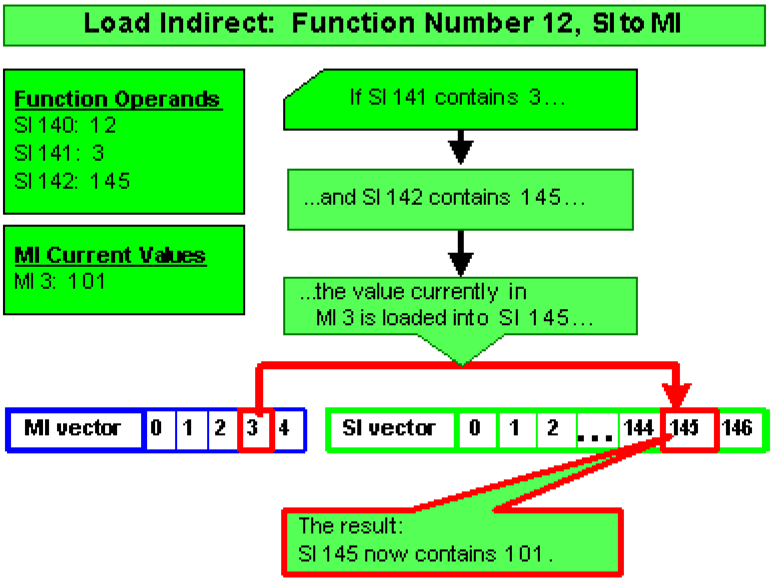
Store the Offset in Vector, Source, in SI 1

Store the Offset in Vector, Target, in SI 1

Select the function type by storing the func into SI 140.

Function Number (SI 140)	Offset in Vector, Source (SI 141)	Offset in Vector, Target (SI 142)
10	MI	MI
11	SI	MI
12	MI	S
13	SI	S

Note that when you run Test (Debug) Mode, the current value in SI 140 will **not** be displayed.



Load Indirect: Function Number 12, SI to MI

Function Operands
SI 140: 12
SI 141: 3
SI 142: 145

MI Current Values
MI 3: 101

If SI 141 contains 3...

...and SI 142 contains 145...

...the value currently in MI 3 is loaded into SI 145...

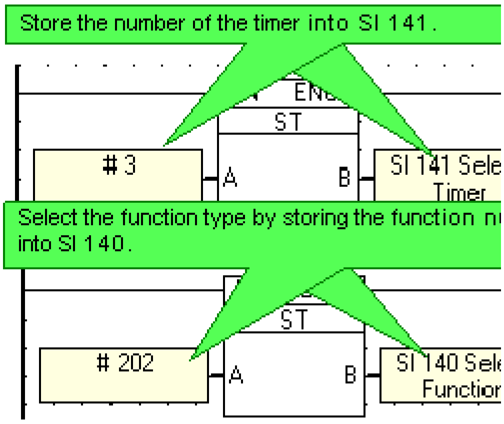
The result:
SI 145 now contains 101.

Load Timer Preset/Current Value

This function allows you to take a preset or current timer value and load it into another operand. Note that since there is no Ladder element for this function; you perform it by storing values into:

- SI 141 to select the timer; 0-63,
- SI 140 to select the type of function. Storing the function number calls the function. In your application, call the function **after** you have entered all of the other parameters.

To use this function:

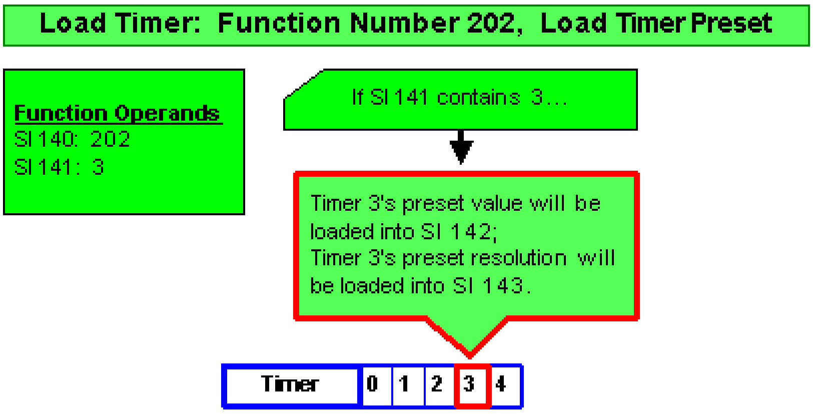

Store the number of the timer into SI 141.

Select the function type by storing the function nu into SI 140.

Function Number (SI 140)	Description
202	Load Timer Preset
203	Load Timer Current

Note that when you run Test (Debug) Mode, the current value in SI 140 will **not** be displayed.

Timer Resolution (stored into SI 143)	
Value	**Resolution**
1	10mS (0.01S)
10	100mS (001S)
100	1000mS (1S)
1000	10000mS (10S)


Load Timer: Function Number 202, Load Timer Preset

Function Operands
SI 140: 202
SI 141: 3

If SI 141 contains 3...

Timer 3's preset value will be loaded into SI 142; Timer 3's preset resolution will be loaded into SI 143.

Store Timer's Preset/Current Value

This function allows you to take a value and store it into a timer to change the preset or current timer value. Since there is no Ladder element for this function; you perform it by storing values into :

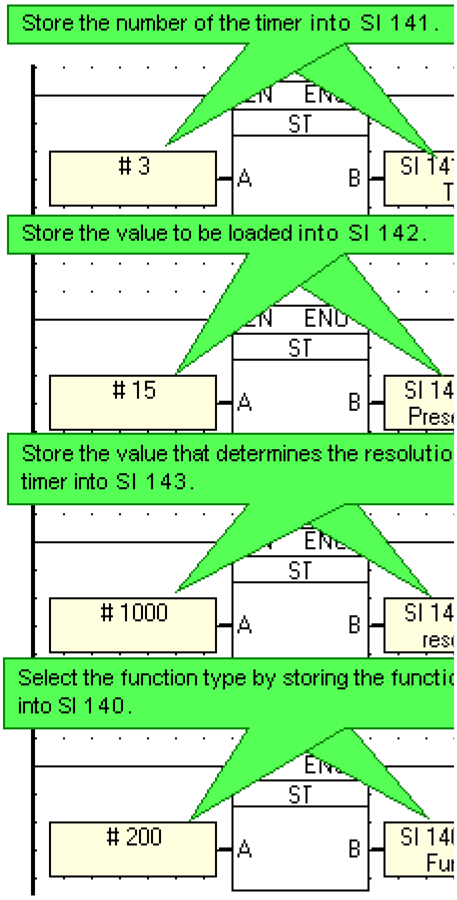
- SI 141 to select the timer; 0-63,
- SI 142 to determine the timer value,
- SI 143 to select the timer's resolution (timer units, or 'ticks'),
- SI 140 to select the type of function. Storing the function number calls the function. In your application, call the function **after** you have entered all of the other parameters.

Take into account that:

- Since you cannot change the resolution of a timer when the application is running, SI 143 is not used in a Store Timer's Current Value function.
- A timer's current value can be changed at any time, including when the timer is active. The new value can be either greater or smaller than the previous value; storing 0 into a timer's current value stops it immediately.
- A change of Timer Preset value without changing the resolution will take effect when the timer restarts.
- Changing the resolution of the timer's preset value does not affect the current resolution; it is therefore recommended that the resolution not be changed while the timer is active.

To use this function:

-



Store the number of the timer into SI 141.

Store the value to be loaded into SI 142.

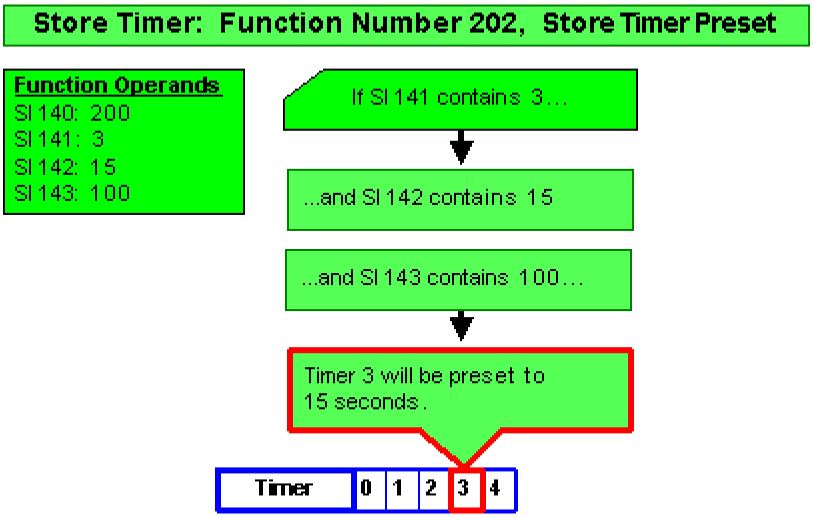
Store the value that determines the resolutio timer into SI 143.

Select the function type by storing the functic into SI 140.

Function Number (SI 140)	Description
200	Store Timer Preset
201	Store Timer Current

Note that when you run Test (Debug) Mode, the current value in SI 140 will **not** be displayed.

Timer Resolution (stored into SI 143)	
Value	**Resolution**
0	Maintain Timer Resolution
1	10mS (0.01S)
10	100mS (001S)
100	1000mS (1S)
1000	10000mS (10S)



Store Timer: Function Number 202, Store Timer Preset

Function Operands
SI 140: 200
SI 141: 3
SI 142: 15
SI 143: 100

If SI 141 contains 3...

...and SI 142 contains 15

...and SI 143 contains 100...

Timer 3 will be preset to 15 seconds.

Note that the timer value is 14 bits.

Shift Register

You can use the following SIs and SBs to perform Shift Left and Shift Right Functions.

SI	Symbol	Description
87	Shift Value	This register contains the number to be shifted.
88	Shift By	This register contains the number of bits to be shifted (Default is 1 bit).

SB	Symbol
87	Shift Left
88	Shift Right

Example : Shift Left

To shift the number 64 **left** by 1 bit:

1. Use a Store function to write the number 64 into SI 87.
2. Use a Store function to write the number 1 into SI 88.
3. Turn SB 87 ON.

Once the function is performed SI 87 will contain 128.

In binary:
Start value: 0000000001000000 = 64
After Shift Left : 0000000010000000 =128

Example : Shift Right

To shift the number 64 **right** by 1 bit:

1. Use a Store function to write the number 64 into SI 87.
2. Use a Store function to write the number 1 into SI 88.
3. Turn SB 88 ON.

Once the function is performed SI 87 will contain 32.

In binary:
Start value: 0000000001000000 = 64
After Shift Right: 0000000000100000 =32

Square Root

This function enables you to find the square root of a number.

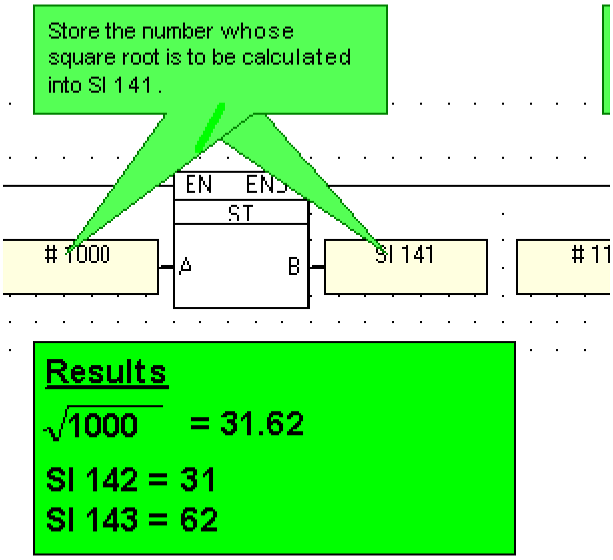
Since there is no Ladder element for this function; you perform it by storing the number whose square root is to be calculated into SI 141.

Store 110 into SI 140 to call the function. In your application, call the function **after** you have entered all of the other parameters.

The results will be placed in:

- SI 142. This contains the whole number result.
- SI 143. If the result is not a whole number, this contains up to 2 digits to the left of the decimal point.

To use this function:



Store the number whose square root is to be calculated into SI 141 .

Results

$\sqrt{1000} = 31.62$

SI 142 = 31

SI 143 = 62

Function Number (SI 140)	Description
110	Calculate square root

Note that when you run Test (Debug) Mode, the current value in SI 140 will **not** be displayed.

Copy MI to Output vector, Input vector to MI

Using this function, you can:

- Copy a vector of Inputs (**I**) to a register.
- Copy a register value to a vector of Outputs (**O**).

Note that an M90 register contains 16 bits. If the converted values exceed 16 bits, the function will write the value to consecutive registers. Any values in those registers will be overwritten. When a register value is copied to outputs, the function will store the register value in consecutive outputs.

Input to Register

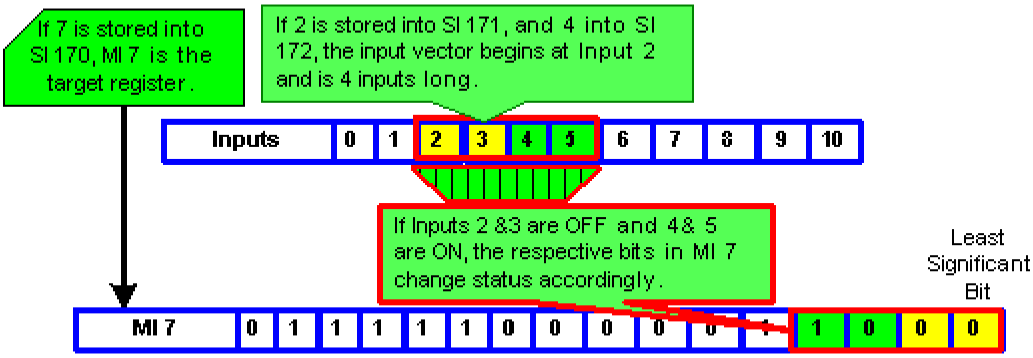
SI	Description	SB	Function
SI170	Address of MI containing integer value	SB172	I to MI
SI171	Start address of bit array (vector)	SB173	MI to O
SI172	Amount of bits		

Example: Input to MI, SB 172

1. Store the value 7 into SI 170, 2 into SI 171 and 4 into SI 172.
2. Set SB 172 to ON.

The program takes the status of I2 to I5, and changes the status of the respective bits in MI 7.

Bits in the target register that are outside of the defined range are not affected.


If 7 is stored into SI 170, MI 7 is the target register.
If 2 is stored into SI 171, and 4 into SI 172, the input vector begins at Input 2 and is 4 inputs long.
Inputs
0
1
2
3
4
5
6
7
8
9
10
If Inputs 2 &3 are OFF and 4& 5 are ON, the respective bits in MI 7 change status accordingly.
Least Significant Bit
MI 7
0
1
1
1
1
1
0
0
0
0
0
1
1
0
0
0

Example: MI to Output, SB 173

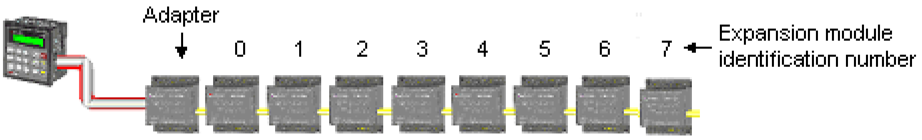
1. Store the value 7 into SI 170, 3 into SI 171 and 7into SI 172.
2. Set SB 173 to ON.

The program will take the binary value of the MI 7, and change the status of the respective outputs in the defined vector, O3 to O7.

Addressing: I/O Expansion Modules

Inputs and outputs located on I/O expansion modules that are connected into an M90 OPLC are assigned addresses that comprise a letter and a number. The letter indicates whether the I/O is an input (I) or an output (O). The number indicates the I/O's location in the system. This number relates to both the expansion module's position in the system, and to the position of the I/O on that module.

Expansion modules are numbered from 0-7 as shown in the figure below.



The formula below is used to assign addresses for I/O modules used in conjunction with the M90 OPLC.

X is the number representing a specific module's location (0-7). Y is the number of the input or output on that specific module (0-15).

The number that represents the I/O's location is equal to: 32 + x • 16 + y

Example

- Input #3, located on expansion module #2 in the system, will be addressed as I 67, 67 = 32 + 2 • 16 + 3
- Output #4, located on expansion module #3 in the system, will be addressed as O 84, 84 = 32 + 3 • 16 + 4.

EX90-DI8-RO8 is a stand-alone I/O module. Even if it is the only module in the configuration, the EX90-DI8-RO8 is always assigned the number 7. Its I/Os are addressed accordingly.

Example

- Input #5, located on an EX90-DI8-RO8 connected to an M90 OPLC will be addressed as I 149, 149 = 32 + 7 • 16 + 5

Convert MB to MI, MI to MB

An M90 register is built of 16 bits.

Using the MB to MI function, you can convert 16 bits or more into a integer value. Conversely, you can convert an integer value into 16 bits or more using the MI to MB function.

Note that if the converted values exceed 16 bits, the function will write the value to consecutive registers. Any values in those registers will be overwritten.

To apply the functions, use the following System Integers (SI) and System Bits (SB)

SI	Description	SB	
SI170	Address of MI containing integer value	SB170	MB to MI
SI171	Start address of MB array (vector)	SB171	MI to MB
SI172	Amount of MBs		

You can use this function, for example to send an SMS when there is a change in the status of the M90's inputs:

1. Represent the status of the M90's inputs using MBs.
2. Convert these MBs into an MI
3. Perform a XOR operation on the result.

When there is a change in input status, the XOR operation will return a value different than 0, which may then be used to trigger the sending of an SMS.

Examples

Example 1:

1. Store the value 7 into SI 170, 10 into SI 171 and 9 into SI 172.
2. Set SB 170 to ON.

The program will calculate the binary value of a 9 bit array which starts with MB 10. The resulting value will be placed into MI 7.

Example 2:

1. Store the value 7 into SI 170, 10 into SI 171 and 9 into SI 172.
2. Set SB 171 to ON

The program will calculate the binary value of the value contained in MI 7. The result will be scattered on a 9 bit array which starts with MB 10.

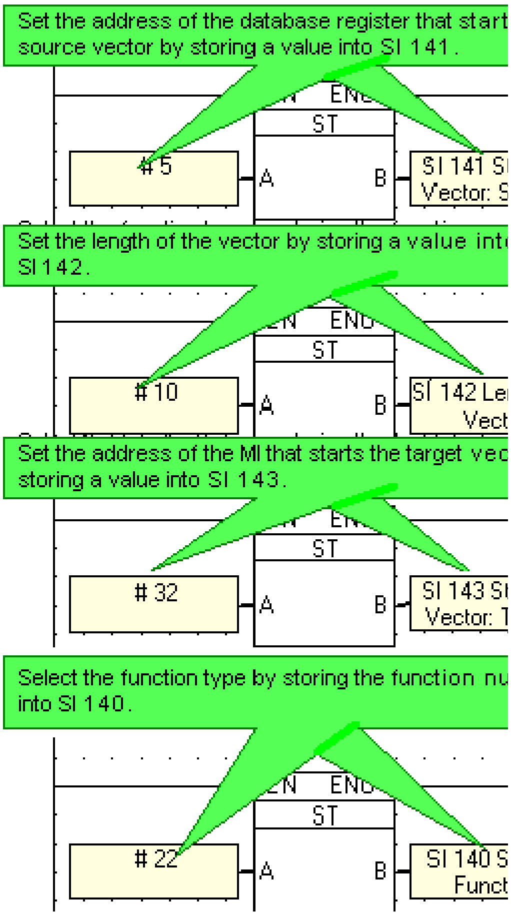
Copy Vector

Vector Copy enables you to set a range of operands, copy the values of each operand within that range **(source)**, then write those values into a corresponding range of operands of the same length **(target)**. You can copy from/to a vector of MI registers or Database registers by selecting the appropriate function.

Note that since there is no Ladder element for this function; you perform it by storing values into:

- SI 141 to determine the source vector,
- SI 142 to determine the length of the vector,
- SI 143 to determine the target vector,
- SI 140 to select the type of function. Storing the function number calls the function. In your application, call the function **after** you have entered all of the other parameters.

To use Copy Vector:



Set the address of the database register that start source vector by storing a value into SI 141.

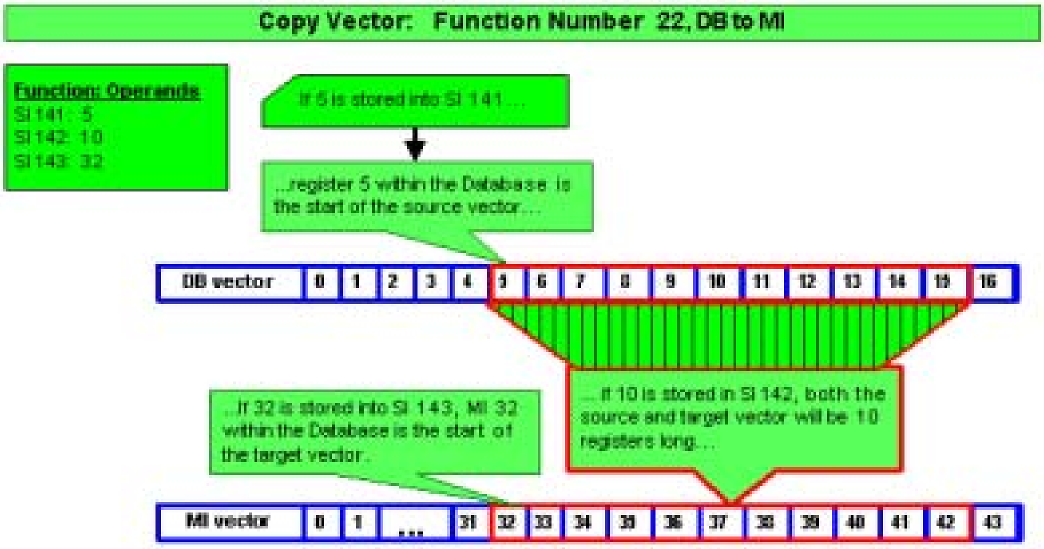
Set the length of the vector by storing a value int SI 142.

Set the address of the MI that starts the target vec storing a value into SI 143.

Select the function type by storing the function nu into SI 140.

Function Number (SI 140)	Source Vector, (SI 141)	Target Vector, (SI 142)
20	MI	MI
21	MI	DB
22	DB	MI
23	DB	DB

Note that when you run Test (Debug) Mode, the current value in SI 140 will **not** be displayed.



Copy Vector: Function Number 22, DB to MI

Function: Operands
SI 141: 5
SI 142: 10
SI 143: 32

If 5 is stored into SI 141...

...register 5 within the Database is the start of the source vector...

...If 32 is stored into SI 143, MI 32 within the Database is the start of the target vector.

...If 10 is stored in SI 142, both the source and target vector will be 10 registers long...

Fill Vector

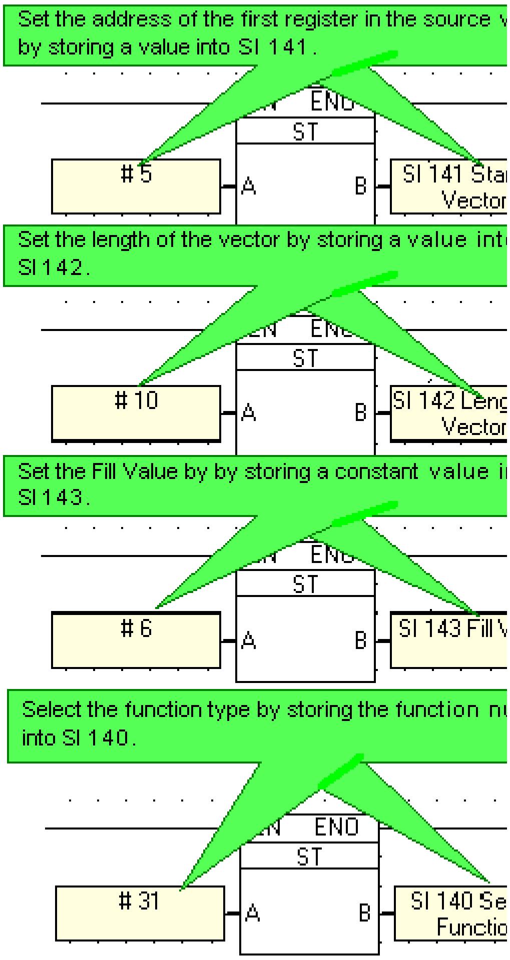
Fill Direct enables you to set a range of registers. The function copies a value from a desired operand or constant value **(source)**, then writes that value into every operand within the range **(target vector)**.

You can fill a vector of MI registers or Database registers by selecting the appropriate function.

Note that since there is no Ladder element for this function; you perform it by storing values into:

- SI 141 to determine the start of the target vector,
- SI 142 to determine the length of the target vector,
- SI 143 to select the Fill Value; the register whose value will be written into each register within the target vector,
- SI 140 to select the type of function. Storing the function number calls the function. In your application, call the function **after** you have entered all of the other parameters.

To use Fill Vector:


Set the address of the first register in the source v by storing a value into SI 141.

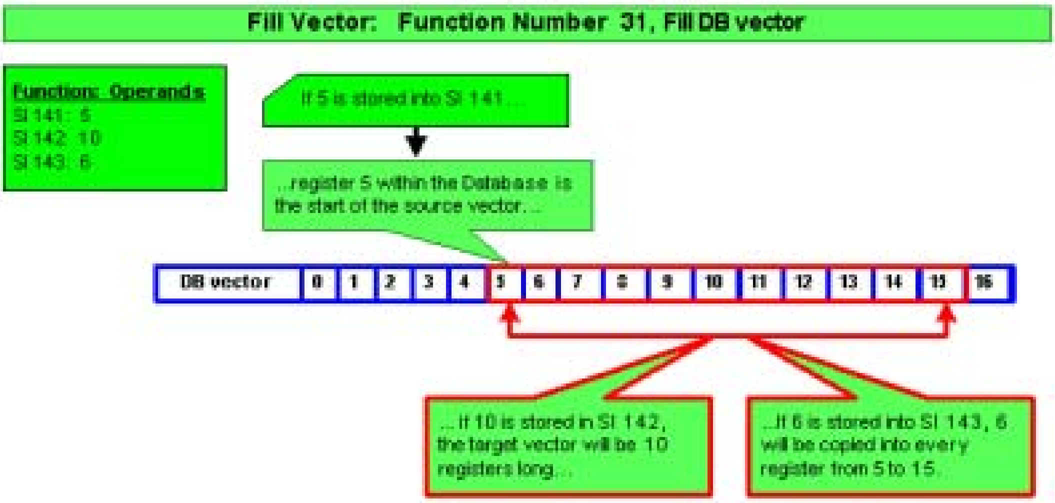
Set the length of the vector by storing a value int SI 142.

Set the Fill Value by by storing a constant value i SI 143.

Select the function type by storing the function nu into SI 140.

Function Number (SI 140)	Description
30	Fill MI Vector
31	Fill DB Vector

Note that when you run Test (Debug) Mode, the current value in SI 140 will **not** be displayed.


Fill Vector: Function Number 31, Fill DB vector

Functions: Operands
SI 141: 5
SI 142: 10
SI 143: 6

If 5 is stored into SI 141...

...register 5 within the Database is the start of the source vector...

... If 10 is stored in SI 142, the target vector will be 10 registers long...

...If 6 is stored into SI 143, 6 will be copied into every register from 5 to 15.

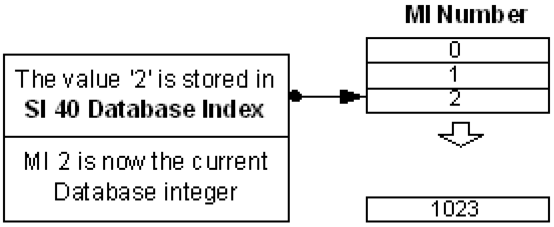
Access indirectly addressed registers: Using the Database

You can access and use integers 0 through 1023 within the M90 OPLC's memory as a database, via SI 40 and SI41.

Note that when you run Test (Debug) Mode, the current value in SI 140 will **not** be displayed.

Writing Values

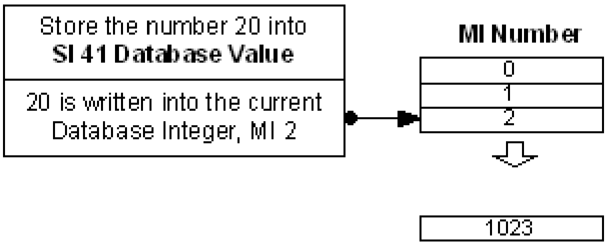
1. Use SI 40 Database Index to access a particular MI.
   For example, to access MI 2 you store the number 2 into SI 40.



The value '2' is stored in **SI 40 Database Index**

MI 2 is now the current Database integer

2. Use SI 41 Database Value to write a value into MI 2.
For example, you can store a number value into SI 41.

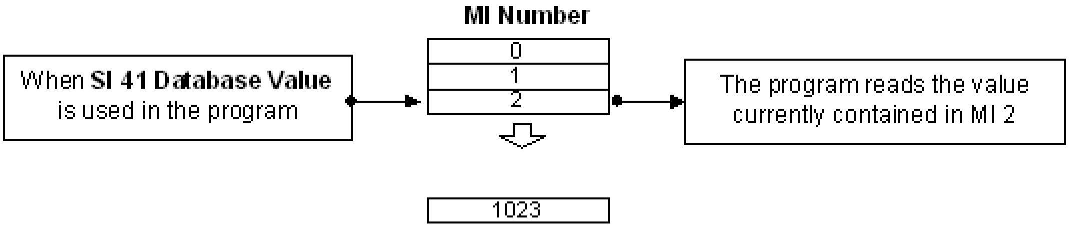


Store the number 20 into **SI 41 Database Value**

20 is written into the current Database Integer, MI 2

Reading Values

When you use SI 41 Database Value in your program, the program actually reads the MI that is referenced by SI 40 Database Index.



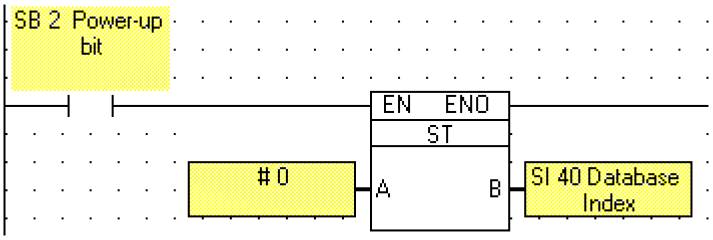
When **SI 41 Database Value** is used in the program

The program reads the value currently contained in MI 2

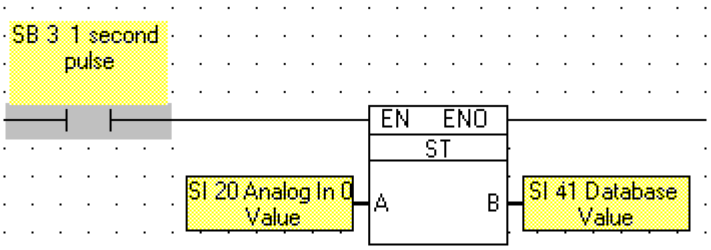
Examples

Example 1: Write

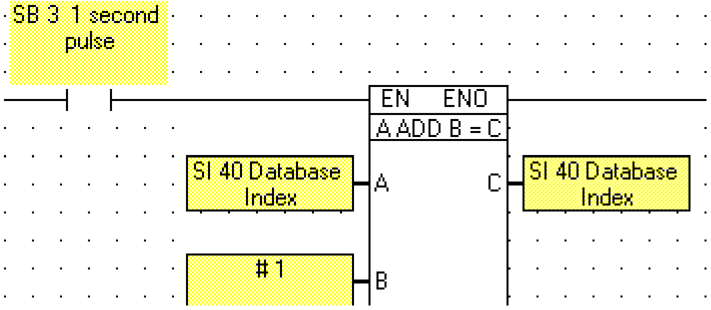
In the net below, 0 is stored in SI 40 when the M90 OPLC is powered up. This means that integer 0 is now the current 'database' integer.



In the net below, the analog value contained in SI 20 is stored in SI 41 every second. According to the net above, the current 'database' integer is 0. The analog value is therefore stored in integer 0.



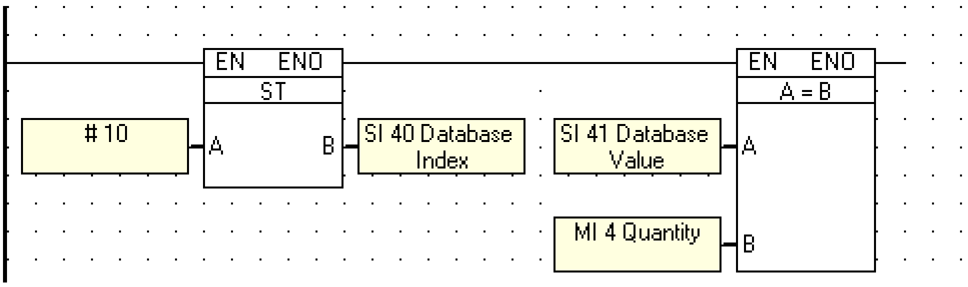
In the next net, the value in SI 40 is incremented by 1every second, changing the current database integer. This means that the first analog value will be stored in integer 0, the second analog value in integer 1, and so on.



Example 2: Read

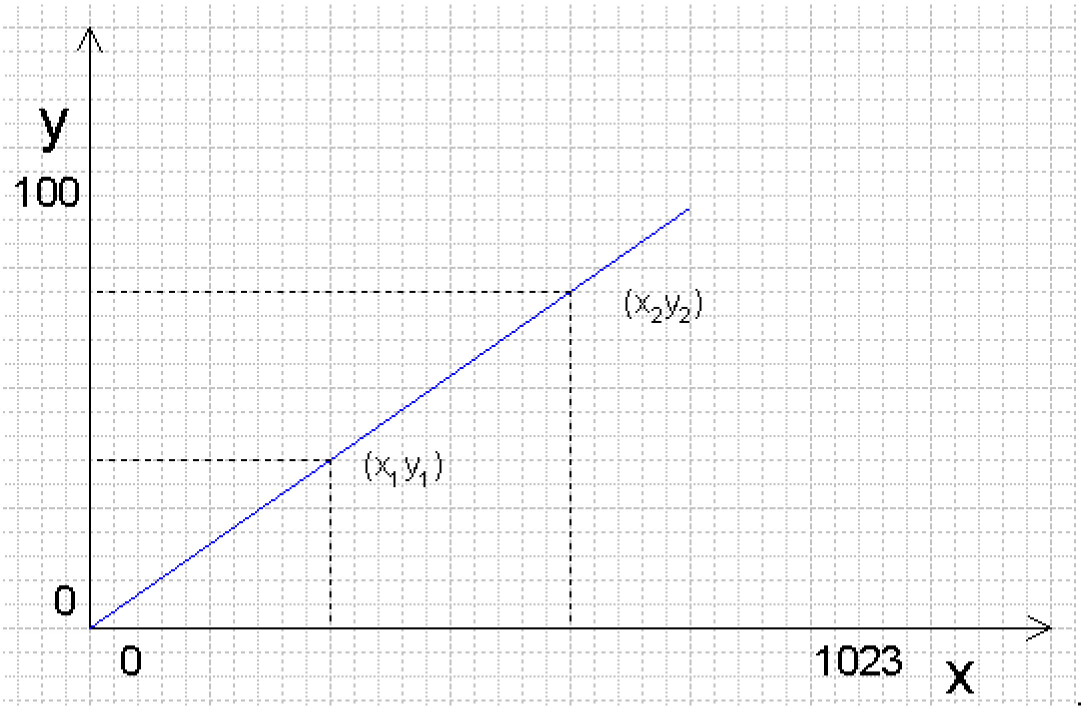
In the first part of the net below, 10 is stored into SI 40. Integer 10 is the 'database' integer. In the second part of the net, the value in SI 41 is compared to the value in integer 4.

The value in SI 41 is the value actually in integer 10—the current database integer.



Linearization

Linearization can be used to convert analog values from I/Os into decimal or other integer values. An analog value from a temperature probe, for example can be converted to degrees Celsius and displayed on the controller's display screen.



Linearize values for Display

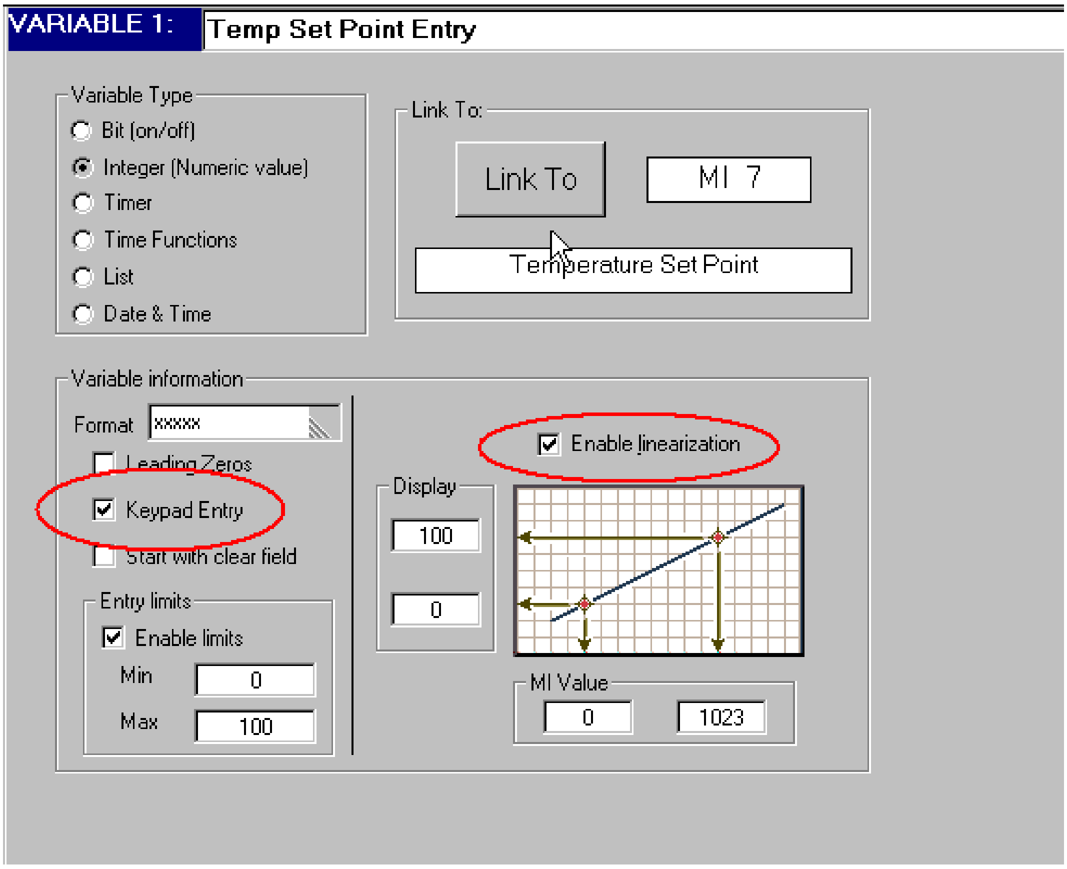
Note that the linearized value created in this way may be displayed-- but the value cannot be used anywhere else within the project for further calculations or operations.

You can enter an Analog value, such as temperature, via the M90 keypad, then convert that value into a Digital value for comparison with a digital value from a temperature probe by selecting Enable Linearization in the linked Variable.

This conversion process is Reverse Linearization.

To enable Analog to Digital conversion:

1. Create a Display for entering the analog value.
2. Create an Integer Variable.
3. Select keypad entry and enable linearization.
4. Enter the linearization values for the x and y axes.



VARIABLE 1: Temp Set Point Entry

Variable Type
- ○ Bit (on/off)
- ● Integer (Numeric value)
- ○ Timer
- ○ Time Functions
- ○ List
- ○ Date & Time

Link To:
Link To | MI 7
Temperature Set Point

Variable information
Format: xxxxx
- □ Leading Zeros
- ☑ Keypad Entry
- □ Start with clear field

Entry limits
- ☑ Enable limits
- Min: 0
- Max: 100

☑ Enable linearization

Display: 100, 0

MI Value: 0, 1023

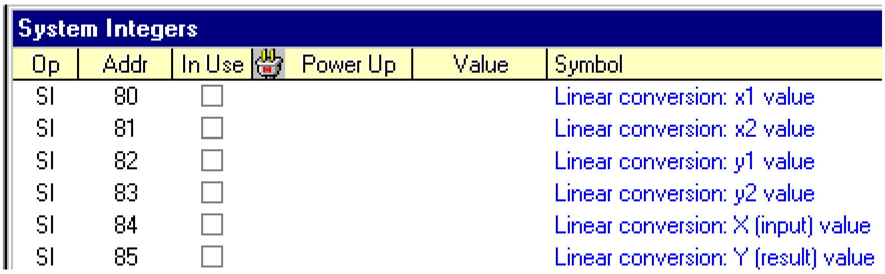
According to the above example:

- A temperature entry of 1000 C will be converted to 1023 Digital value.
- A temperature entry of 500 C will be converted to 512 Digital value.

Linearize values in the Ladder

You can also linearize values in your Ladder and display them on the M90's LCD.

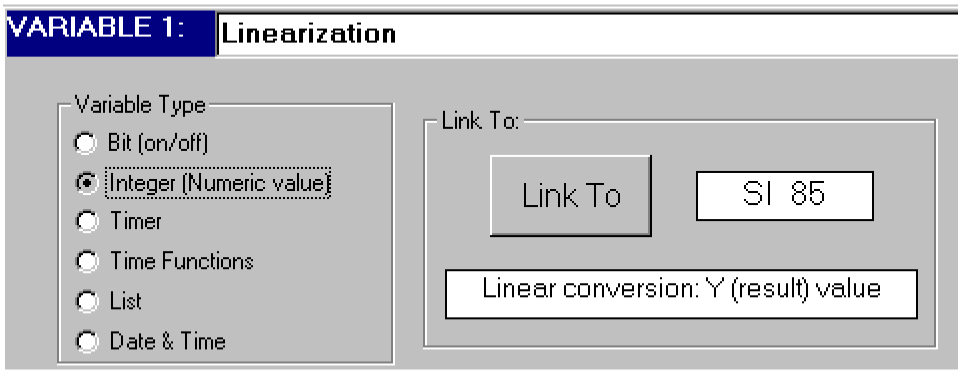
1. In your Ladder project, use SI 80 - 85 to set the (x,y) variable ranges. Use SB 80 to activate the Linearization function.



System Integers

Op	Addr	In Use	Power Up	Value	Symbol
SI	80	□			Linear conversion: x1 value
SI	81	□			Linear conversion: x2 value
SI	82	□			Linear conversion: y1 value
SI	83	□			Linear conversion: y2 value
SI	84	□			Linear conversion: X (input) value
SI	85	□			Linear conversion: Y (result) value

The linearization values created here can be displayed by linking SI 85 to a Display;the value can be used elsewhere within the project for further calculations or operations.



VARIABLE 1: Linearization

Variable Type

○ Bit (on/off)

● Integer (Numeric value)

○ Timer

○ Time Functions

○ List

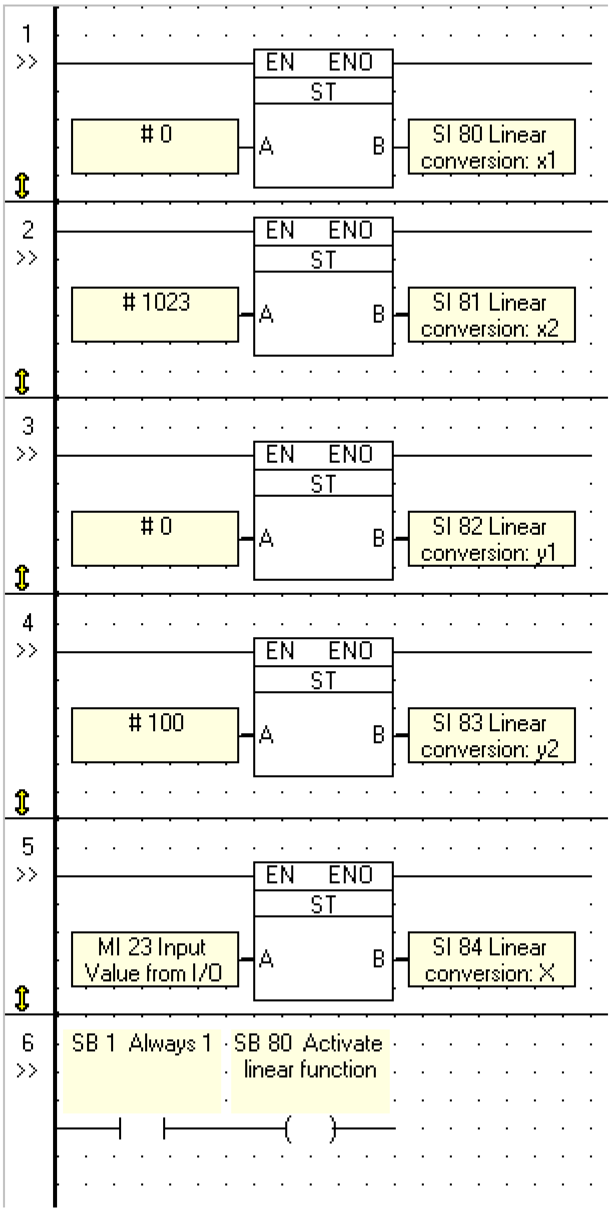
○ Date & Time

Link To:

SI 85

Linear conversion: Y (result) value

Example: write the variable ranges into SI 80 - 83, then writing an analog input into SI 84:

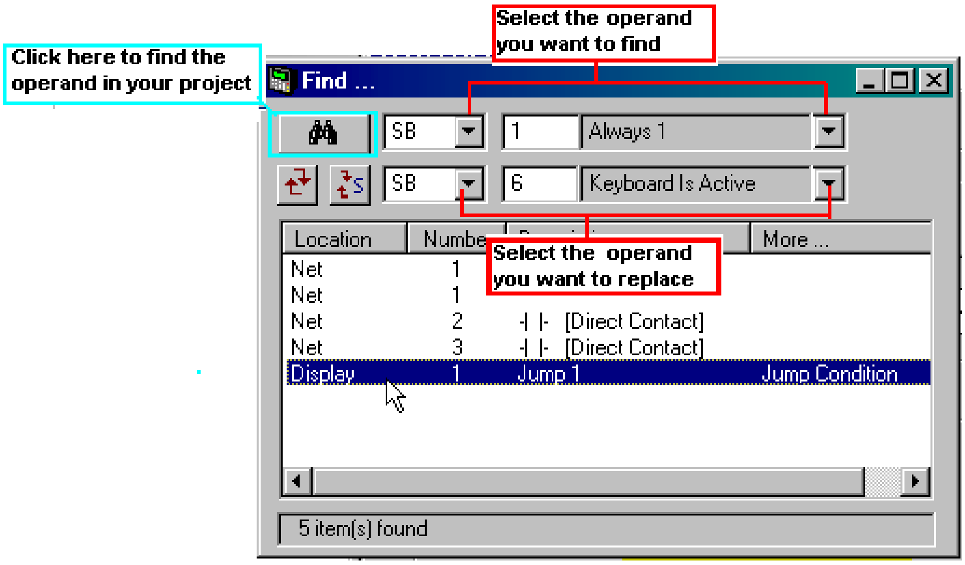

1
EN ENO
ST
# 0
A B
SI 80 Linear conversion: x1
2
EN ENO
ST
# 1023
A B
SI 81 Linear conversion: x2
3
EN ENO
ST
# 0
A B
SI 82 Linear conversion: y1
4
EN ENO
ST
# 100
A B
SI 83 Linear conversion: y2
5
EN ENO
ST
MI 23 Input Value from I/O
A B
SI 84 Linear conversion: X
6
SB 1 Always 1
SB 80 Activate linear function

Find and Replace Elements

To use Find and Replace:

1. Open the **Find** function by clicking on the Find button on the U90 Ladder toolbar.
2. The Find function opens.

3. Select the name and address of the operand you wish to find.
4. Click the Find button shown below; a list appears showing every time that operand is used in the project.
5. Select the name and address of the operand you wish to replace as shown below.



Click here to find the operand in your project

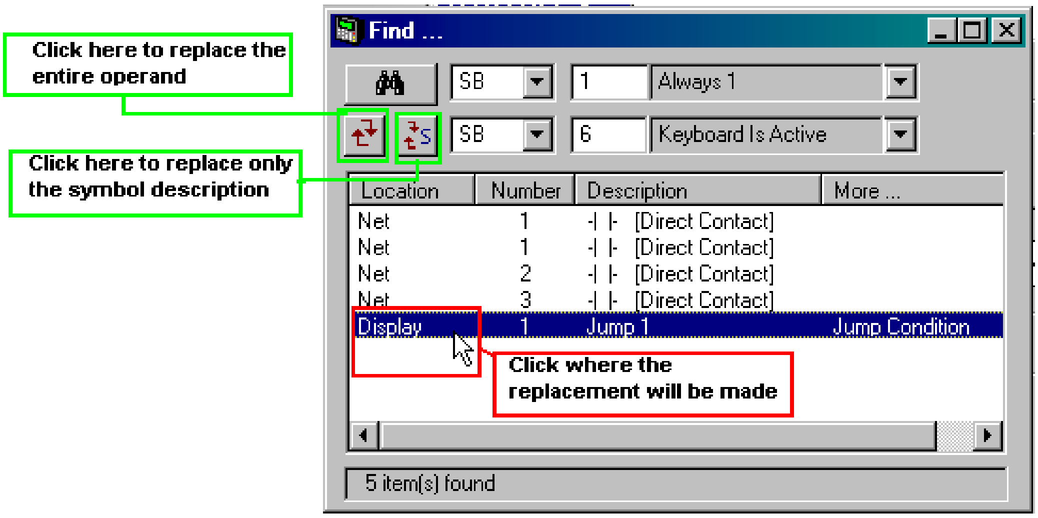
Select the operand you want to find

Select the operand you want to replace

Location	Number	Description	More ...
Net	1		
Net	1		
Net	2	-| |- [Direct Contact]	
Net	3	-| |- [Direct Contact]	
Display	1	Jump 1	Jump Condition

5 item(s) found

6. Select the location of the operand or description you wish to replace by clicking it within the list.
7. Replace operands or their descriptions by clicking the buttons shown below.



Click here to replace the entire operand

Click here to replace only the symbol description

Click where the replacement will be made

Location	Number	Description	More ...
Net	1	-| |- [Direct Contact]	
Net	1	-| |- [Direct Contact]	
Net	2	-| |- [Direct Contact]	
Net	3	-| |- [Direct Contact]	
Display	1	Jump 1	Jump Condition

5 item(s) found

Building a Counter

If you want to use a counter in your application, you build it using:

- Math function
- Compare function
- Store function

Use a Positive / Negative Transition contact on the event operand to activate the counter.

Example:

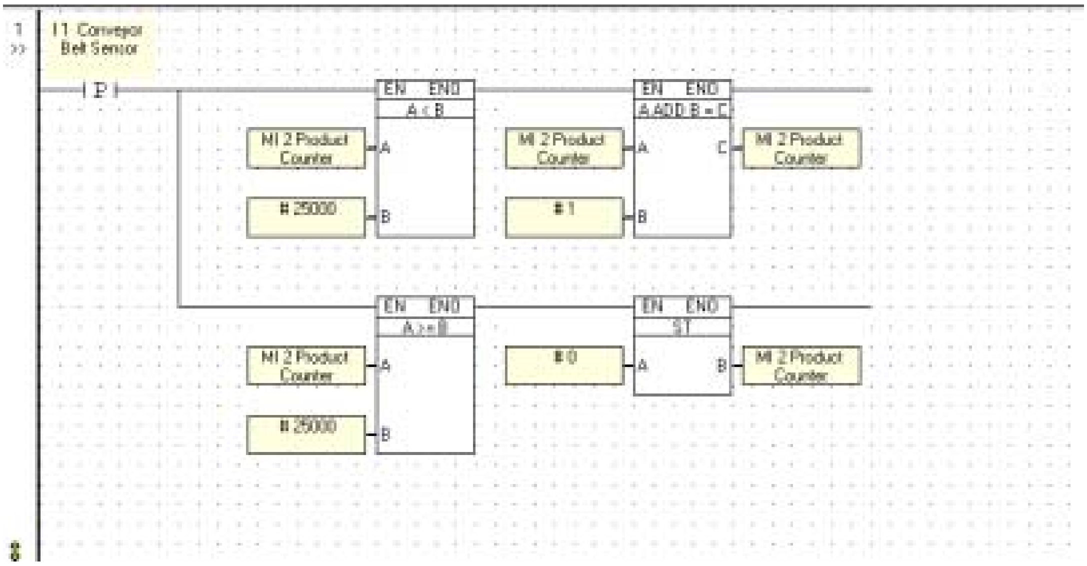
You want to count the gross number of a product traveling across a conveyor belt. There is a sensor (e.g. photocell, limit switch or proximity switch) at a specific point across the conveyor belt which senses the product as it passes.

The sensor is connected to an M90 Input. The Positive Transition from this Input will advance the counter by one.

When the counter value reaches the maximum defined value, the counter will reset to 0.

Counter Ladder example:

- Input 1 is the sensor
- MI 2 is the Counter
- The maximum defined value is 25000.



Keep in mind when building your counter that adding a number to 32767 will return a negative number.

Counters are featured in several sample applications, such as the applications ' Time Interval- SI 1', 'Outputs-activate in sequence', and 'Logging analog values'. These applications may be found by selecting Sample U90 Projects from the Help Menu.

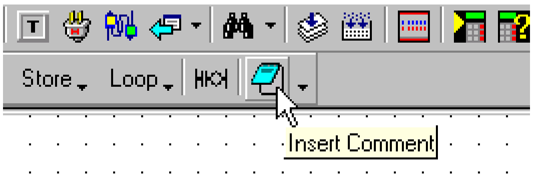
Comments Tool

You can insert comments into the Ladder Editor to label different parts of your program. Comments can be written in Notepad and added later to the project using **Cut** and **Paste** functions.

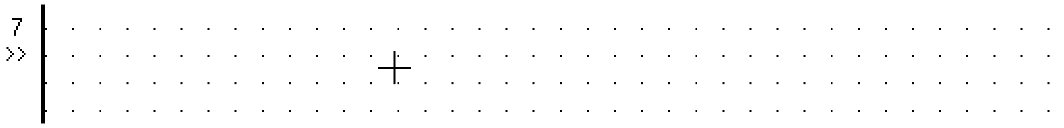
These Comments are 'internal' comments for the programmer(s). The Comments are not downloaded to or displayed on the controller.

To insert comments:

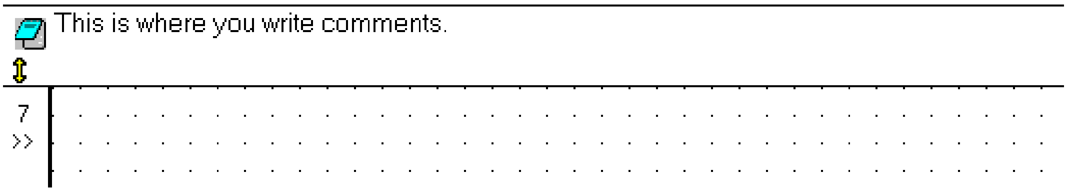
1. On the Ladder toolbar, click **Insert Comment** icon .



2. Move your cursor to the net in which you wish to insert a comment and click.



3. The Comment will appear above the net.
4. Type in your comments.


This is where you write comments.

The length and content of your comments will have no effect on your project. They are not downloaded to the controller and do not affect the memory or word size of a project.

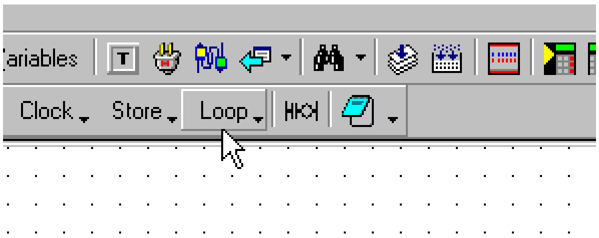
Loops: Jump to Label

Loops in a Ladder project cause the program to jump over certain net(s), according to specific logic conditions.

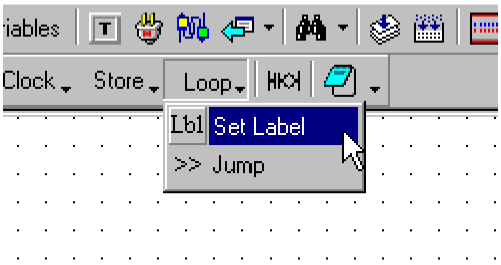
A Loop contains a Jump element and a Label. When the Jump condition(s) is true, the project jumps to the associated Label.

To create a Loop in your project:

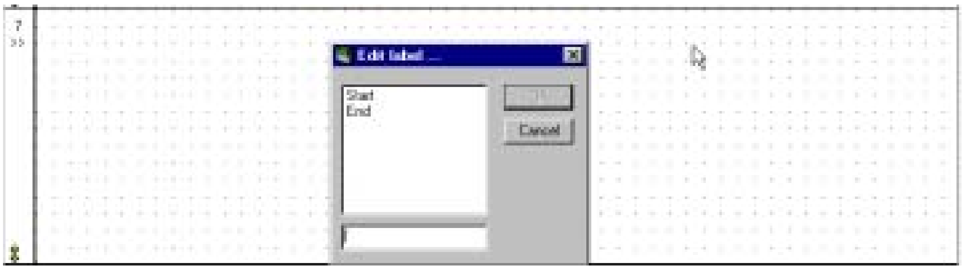
1. Click **Loop** on the Ladder toolbar.



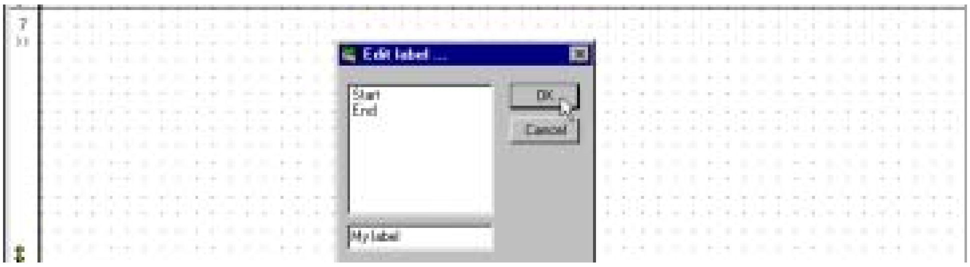
2. Select **Set Label** from the **Loop** menu. Place the cursor in the desired net and click.



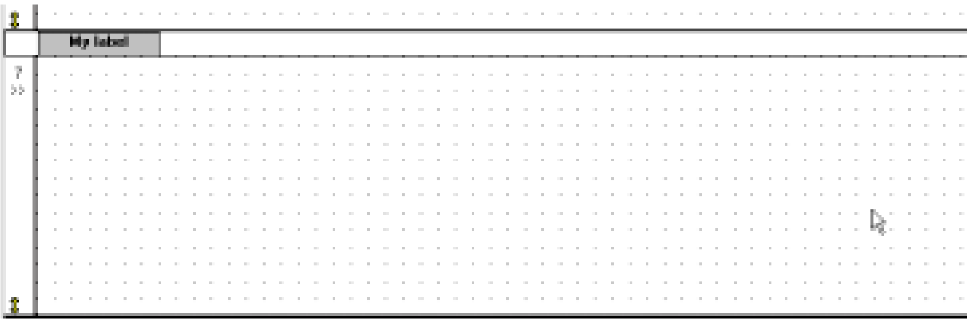
3. The **Edit Label** box opens.



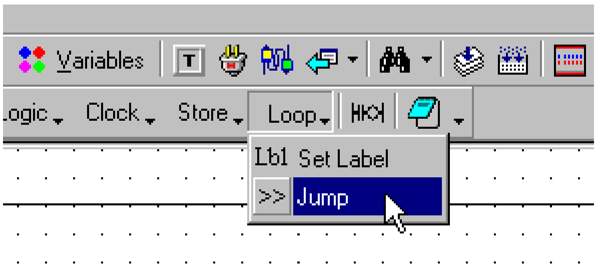
4. Enter a Label name of up to **eight** characters.



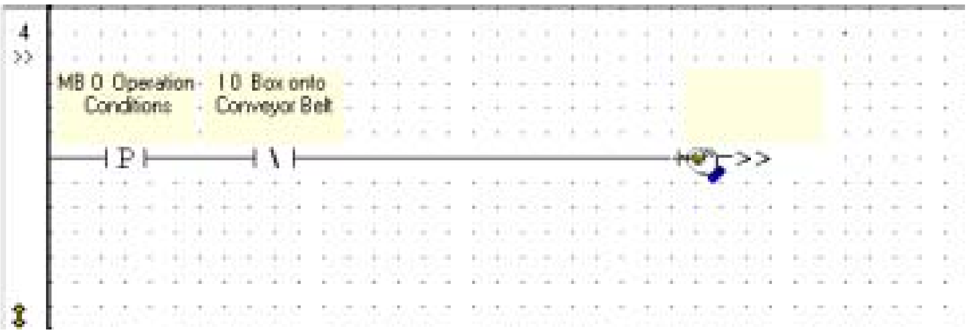
5. The Label appears above the net.



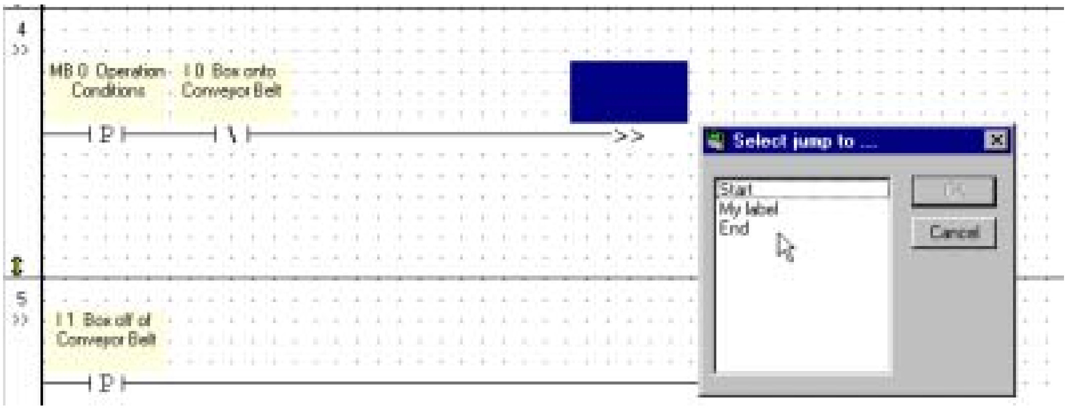
6. Select **Jump** from the Loop menu.



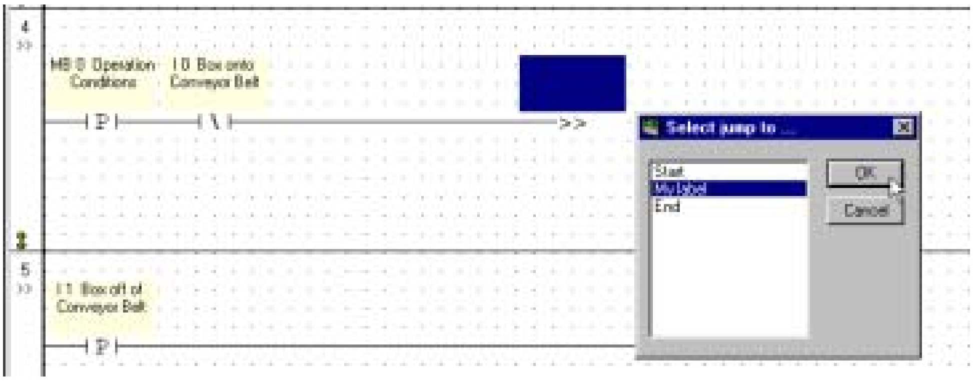
7. Place the Jump in the desired place on the desired net.



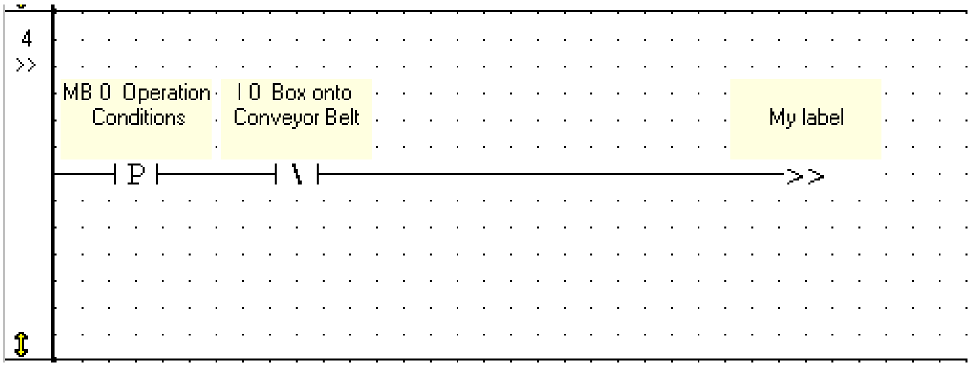
8. **Select Jump to...** window appears.


4
>>
MB 0 Operation Conditions
I 0 Box onto Conveyor Belt
Select jump to ...
Start
My label
End
OK
Cancel
5
>>
I 1 Box off of Conveyor Belt

9. Select the desired Label name to which you want to jump. Click OK.


4
>>
MB 0 Operation Conditions
I 0 Box onto Conveyor Belt
Select jump to ...
Start
My label
End
OK
Cancel
5
>>
I 1 Box off of Conveyor Belt

10. The Jump element appears with the selected Label name on the net.



According to the above example, if Ladder logic is true for net 4, the program will jump over nets 5 and 6 and continue from net 7.

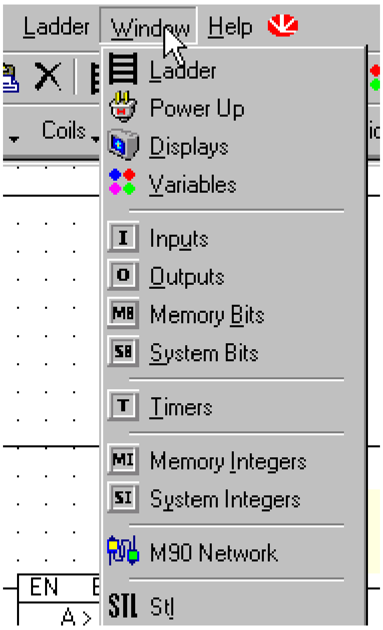
Important note: You must take care when creating Loops not to create an endless Loop. While you can place Labels before a Jump condition and you can refer to a Label more than once, repeated referrals to a Label above a Jump element can create an endless loop which will cause the controller to stop with an error message "PROGRAM LOOP."

Loop functions are featured in the sample application, such as the applications ' Shortening scan time-jump'. This application may be found by selecting Sample U90 Projects from the Help Menu.

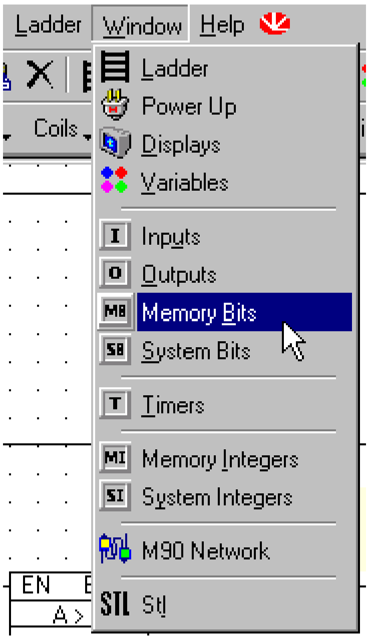
Operands in use

To check what Operands are being used in a project:

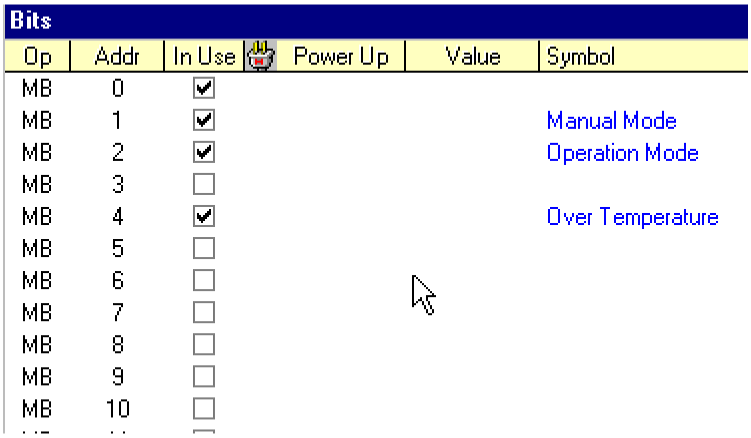
1. Open the Window Menu on the Main menu bar.



2. Select the Operand type you wish to check.



3. The Operand List window opens. The Operands in use are marked with a check mark in the **In Use** box.



Bits

Op	Addr	In Use	Power Up	Value	Symbol
MB	0	☑			
MB	1	☑			Manual Mode
MB	2	☑			Operation Mode
MB	3	□			
MB	4	☑			Over Temperature
MB	5	□			
MB	6	□			
MB	7	□			
MB	8	□			
MB	9	□			
MB	10	□			

SMS

Sending SMS messages from a GSM cell phone

To send SMS messages from your cell phone you must:

- Write and download SMS messages to the M90 as described in Creating SMS messages.
- Write an SMS message in your cell phone.
- Send the message to the M90's GSM modem

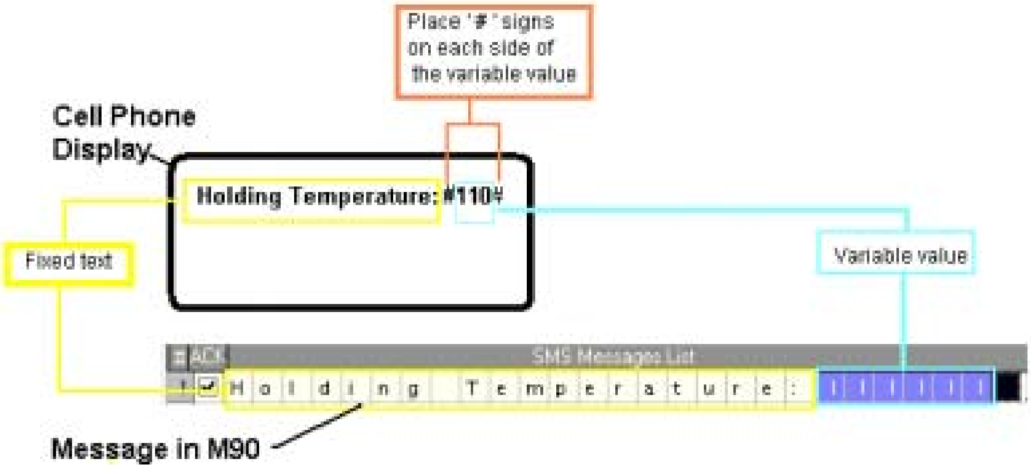
Note that you can only send messages that have already been set in the M90. In addition, if an M90 is configured with the Limited to Authorized Phone Numbers option, you will not be able to send it SMS messages if your number is not in the list .

Writing SMS messages in your cell phone

You write an SMS message using your cell phone keypad. Make sure that:

- The fixed text in your cell phone is identical to the M90's SMS message **in every detail**: spaces, characters--and note that characters are case-sensitive.
- You bracket variable values with number signs (#) as shown below. These signs '#' do not count as spaces.
- The variable field in the M90 is big enough to hold the value.

The figure below shows the same SMS message: as it appears on a cell phone display, and as it appears in the M90's SMS Messages List.



When you send this message from your cell phone, the value 110 will be written into Variable 1 in the M90.

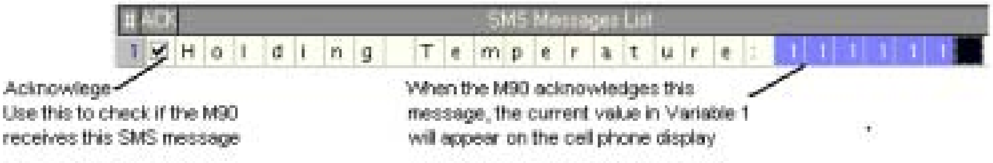
Sending the message to the M90

1. Enter the number of the M90's GSM modem exactly as you would enter any GSM cell phone number, then send the message.

Checking that the M90 has received the SMS message

You can check if the M90 received your message by using the Acknowledge feature:

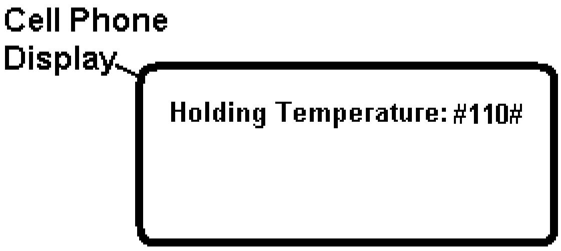
1. Select the **ACK** box as shown below.



Acknowlege
Use this to check if the M90 receives this SMS message

When the M90 acknowledges this message, the current value in Variable 1 will appear on the cell phone display

2. Use your cell phone to send the message **"Holding Temperature:#110#"** to the M90.

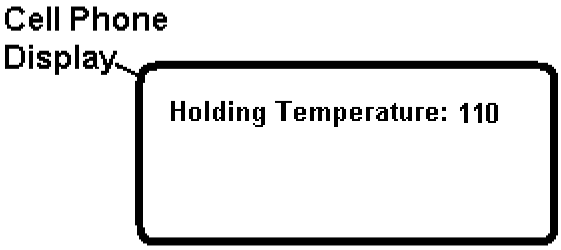


Holding Temperature: #110#

3. The M90 receives this SMS message.

4. The M90 immediately returns the message to your cell phone, together with the current variable value.

5. You can now view this SMS message on your cell phone display, together with changes in the variable value.



Holding Temperature: 110

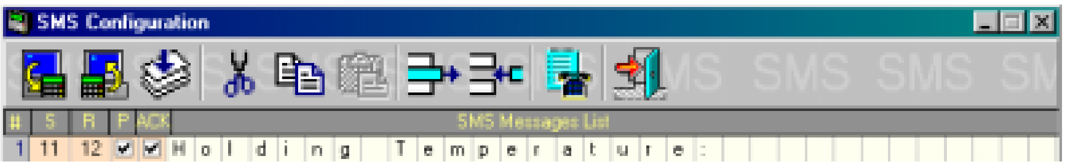
Variable Types

Although SMS messaging supports Integer and List variables, note that you cannot send List variables via cell phone.

Using SMS messages in your application

To cause the M90 to send an SMS message, you use the Send MB which is linked to that message. In the figures below, the Send MB is 11. When MB 11 is turned ON in your application, this message will be sent. The Send MB is turned OFF automatically after the message has been sent.

The Receive MB is 12. When this message is received by the M90, MB 12 will turn ON. You must turn the Receive MB OFF in your application in order to register the next time this message is received.



SMS Configuration

#	S	R	P	ACK	SMS Messages List
1	11	12	☑	☑	Holding Temperature:

M90 Receives the SMS message

When the M90 receives this SMS message, MB 12 will turn ON .

M90 Sends the SMS message

When you turn MB 11 on in your application, the M90 will send this message to all of the selected phone numbers.

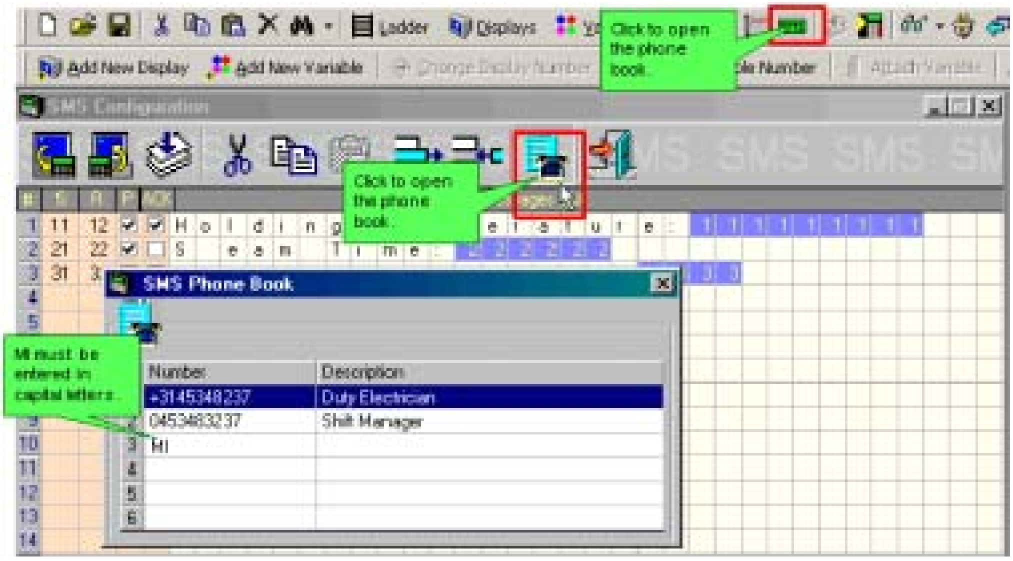
SMS Phone Number: via MI Pointer

Use this utility to use an MI vector as one of the phone numbers in the SMS phone book. This allows you to:

- Enable a number to be dialed via the M90's keypad.
- Exceed the 6 number limit of the SMS phone book.

Note that since there is no Ladder element for this function; you perform it by:

- Storing the start address of the MI vector needed to contain the phone number into SI 141,
- Entering the characters MI, in capital letters, in the *SMS phone book*,



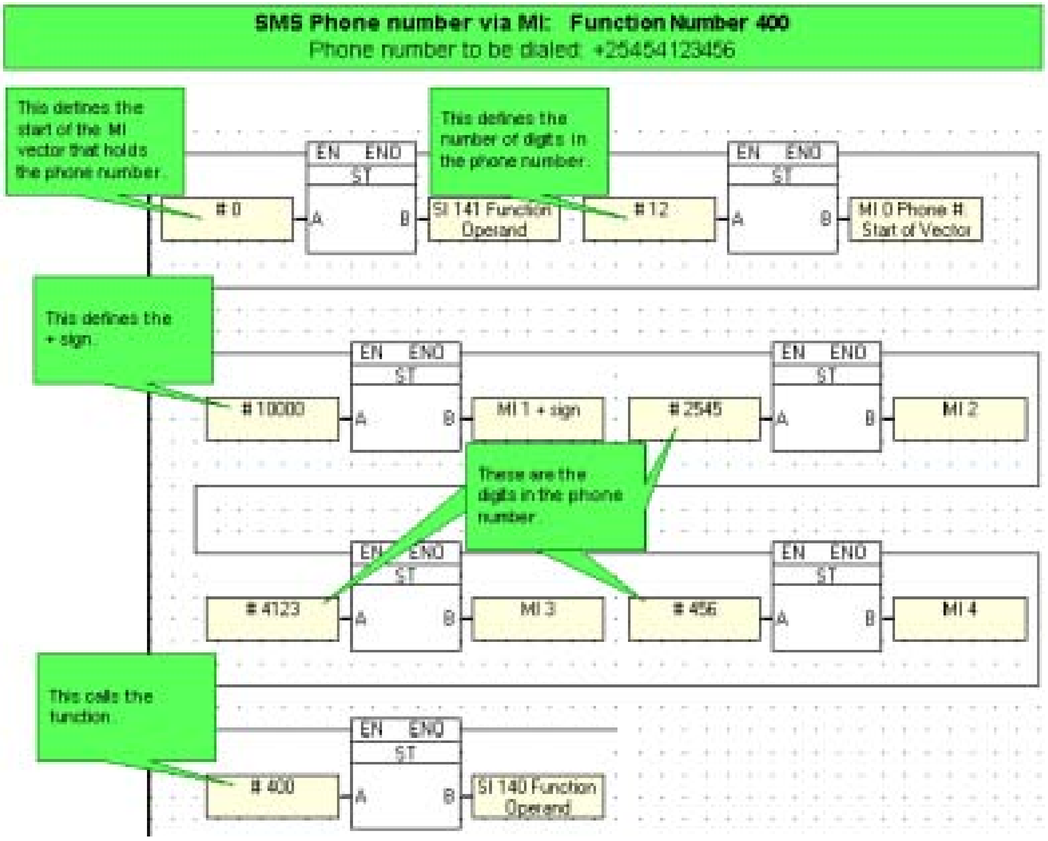
Click to open the phone book.

Click to open the phone book.

MI must be entered in capital letters.

	Number	Description
1	+3145348237	Duty Electrician
2	0453483237	Shift Manager
3	MI	
4		
5		
6		

- Using the index number of that line to call the number, which enables the number in the MI vector to be called,
- Storing 400 into SI 140 to select the function. Storing the function number calls the function. In your application, call the function **after** you have entered all of the other parameters. Note that when you run Test (Debug) Mode, the current value in SI 140 will **not** be displayed.


SMS Phone number via MI: Function Number 400
Phone number to be dialed: +25454123456

This defines the start of the MI vector that holds the phone number.

This defines the number of digits in the phone number.

This defines the + sign.

These are the digits in the phone number

This calls the function

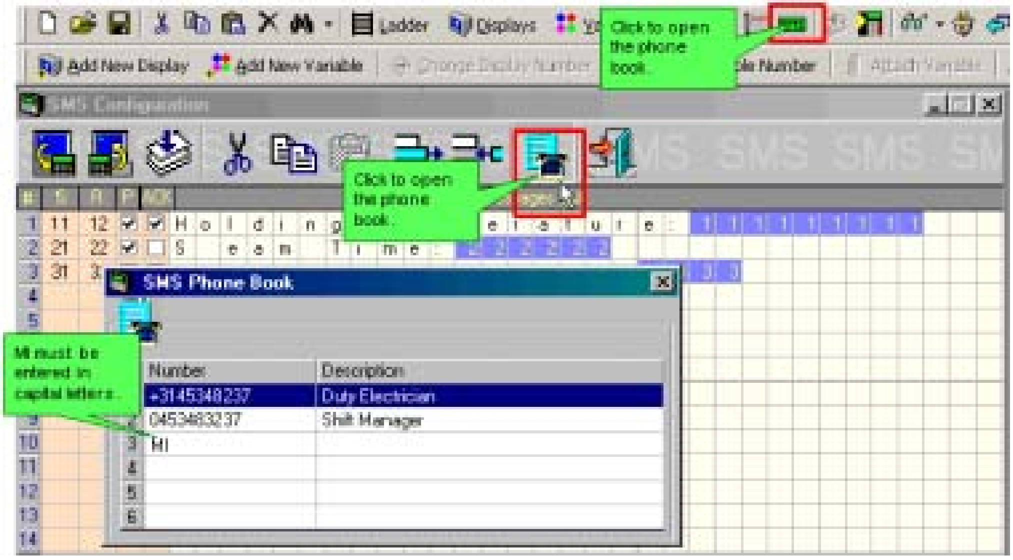
SMS Phone Number: via MI Pointer

Use this utility to use an MI vector as one of the phone numbers in the SMS phone book. This allows you to:

- Enable a number to be dialed via the M90's keypad.
- Exceed the 6 number limit of the SMS phone book.

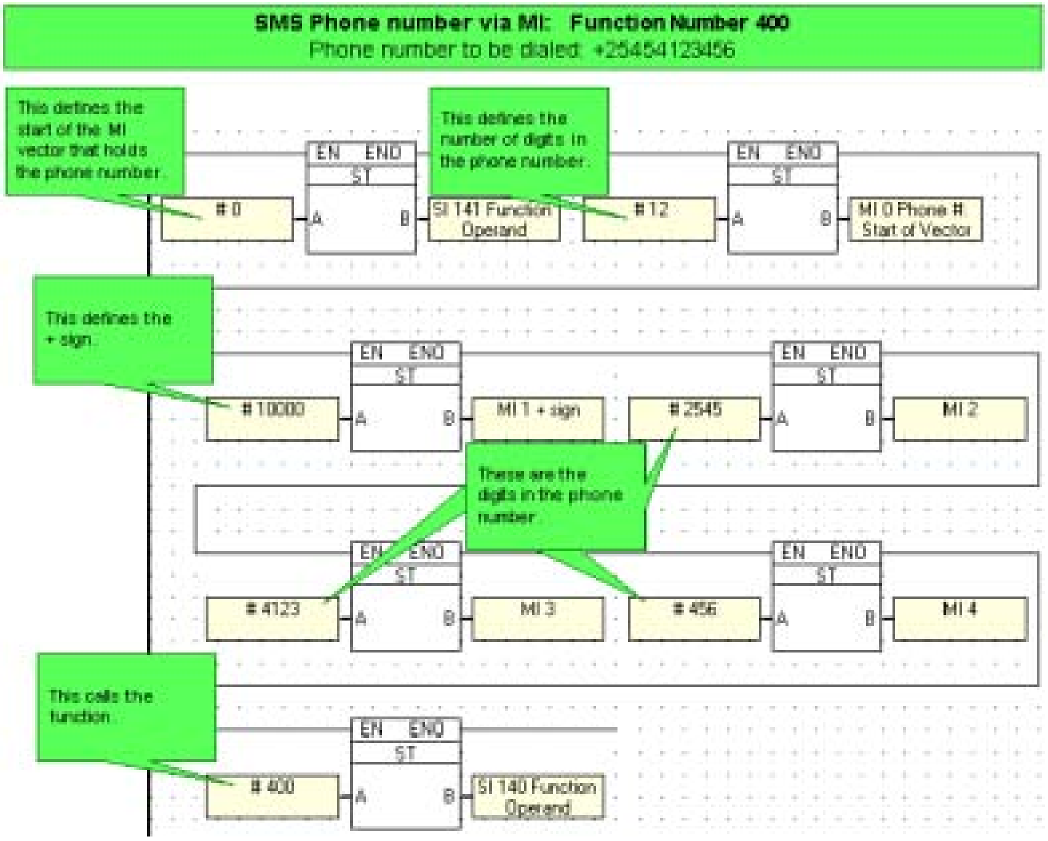
Note that since there is no Ladder element for this function; you perform it by:

- Storing the start address of the MI vector needed to contain the phone number into SI 141,
- Entering the characters MI, in capital letters, in the *SMS phone book*,



	Number	Description
1	+3145348237	Duty Electrician
2	0453483237	Shift Manager
3	MI	
4		
5		
6		

- Using the index number of that line to call the number, which enables the number in the MI vector to be called,
- Storing 400 into SI 140 to select the function. Storing the function number calls the function. In your application, call the function **after** you have entered all of the other parameters. Note that when you run Test (Debug) Mode, the current value in SI 140 will **not** be displayed.


SMS Phone number via MI: Function Number 400
Phone number to be dialed: +25454123456
This defines the start of the MI vector that holds the phone number.
# 0
EN ENO
ST
A B
SI 141 Function Operand
This defines the number of digits in the phone number.
# 12
MI 0 Phone #. Start of Vector
This defines the + sign.
# 10000
MI 1 + sign
# 2545
MI 2
These are the digits in the phone number
# 4123
MI 3
# 456
MI 4
This calls the function
# 400
SI 140 Function Operand

Communications

Configuring my PC's modem

You can configure your PC's modem to dial an M90's modem. Via a PC-modem-to-M90-modem connection, you can:

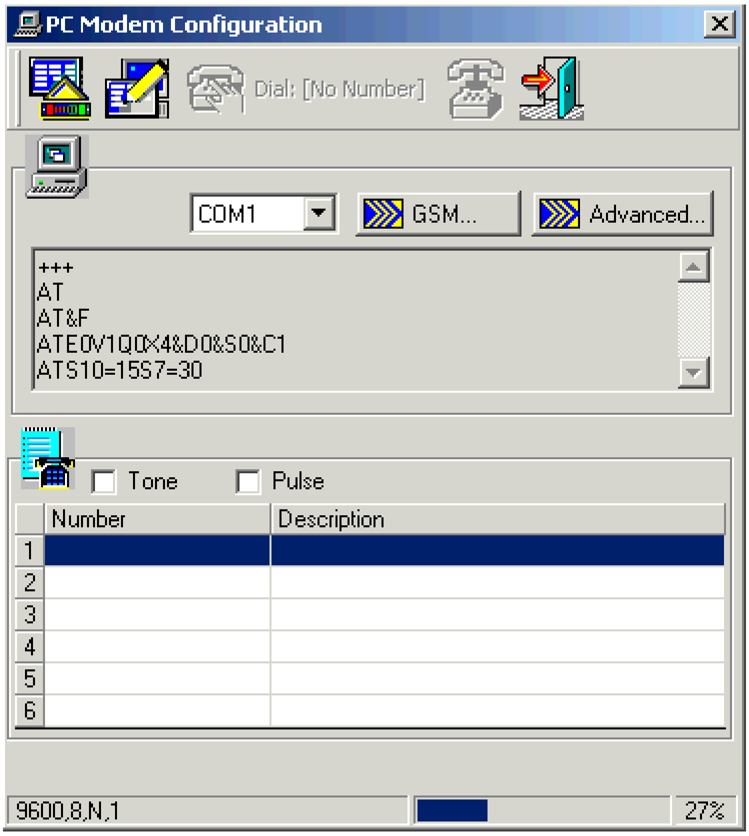
- Download and upload applications
- Test and troubleshoot problems in remote M90 units and applications.

Note ♦ PC-to-M90 communications are via Direct Com. This means that PC modem installation procedures are not necessary.

Configuring your PC's modem

1. Display the PC Modem Configuration box by selecting M90 OPLC Settings from the Controller menu, then clicking on the Modem Setup button.

   Note that the default port setting for internal PC modems is commonly COM 3 or COM 4. Most modems automatically match the parameters of incoming data: baud rate, data bits, parity & stop bits. The U90 Ladder fixed modem settings are: 9600, 8 data bits, no parity, 1 stop bit. You may need to manually change your modem's communication settings to match these.
   You can also select a GSM modem by clicking the GSM button and selecting a modem type.



COM1

```
+++
AT
AT&F
ATE0V1Q0X4&D0&S0&C1
ATS10=15S7=30
```

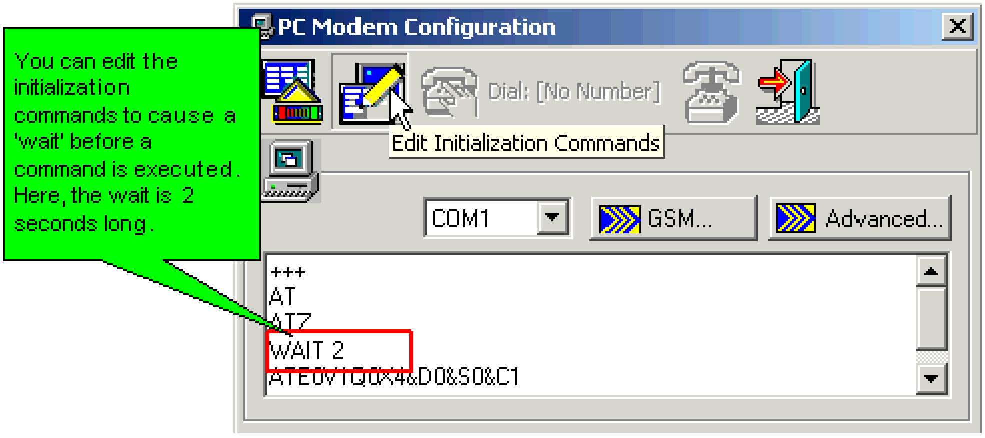
	Number	Description
1		
2		
3		
4		
5		
6		

9600,8,N,1 27%

   The default modem initialization commands that appear here are standard for most modems. If your modem requires different commands, you can edit them.

2. To edit initialization commands, click on the Edit Initialization Commands button shown below. The window containing the commands turns white; you can now add, delete or edit commands.

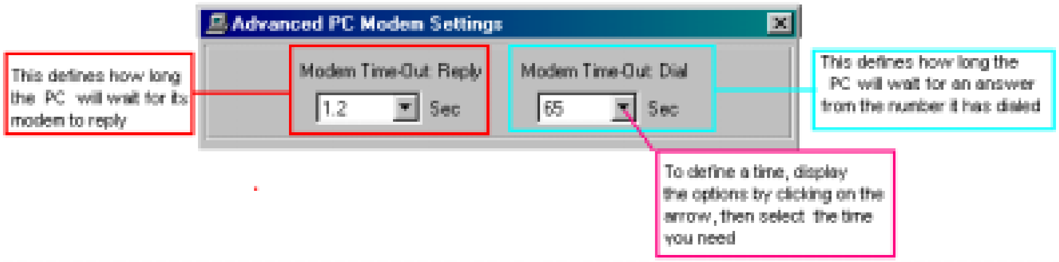
   Note that you can restore the default commands by clicking the Default Initialization button.



You can edit the initialization commands to cause a 'wait' before a command is executed. Here, the wait is 2 seconds long.

```
+++
AT
ATZ
WAIT 2
ATE0V1Q0X4&D0&S0&C1
```

1. Select whether to use pulse or tone dialing, as is required by the system, by clicking on the appropriate box. You can also leave both blank (default).
2. Click the Advanced button to edit Time-Out settings.



Modem Time-Out Reply: 1.2 Sec

Modem Time-Out Dial: 65 Sec

This defines how long the PC will wait for its modem to reply

This defines how long the PC will wait for an answer from the number it has dialed

To define a time, display the options by clicking on the arrow, then select the time you need

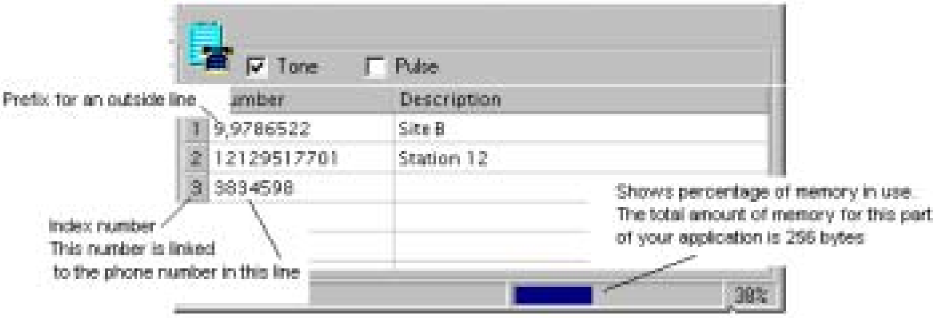
Phone Book

The Phone Book is where you define the list of numbers that the PC can dial. You can enter up to six numbers. Each phone number is automatically linked to an index number. Each phone number can be up to 18 characters long. You can also add a description to identify the location or other details of the number to be dialed.

Entering numbers in the Phone Book

1. Click on an empty line in the Phone Book, then type in the number, exactly as you would dial from a standard phone, including area codes. To dial an outside line, enter the prefix number required and follow it with a comma as shown below.

   This comma causes the short pause, or delay, that is required by many systems.


☑ Tone □ Pulse

	umber	Description
1	9,9786522	Site B
2	12129517701	Station 12
3	3834598	

38%

Prefix for an outside line

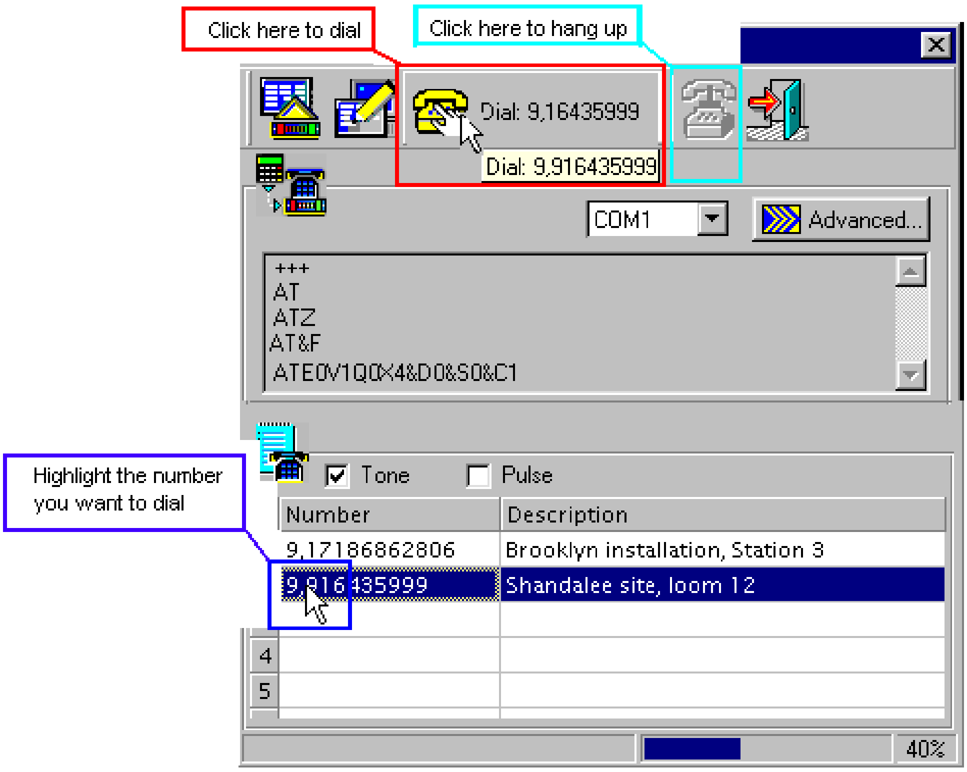
Index number This number is linked to the phone number in this line

Shows percentage of memory in use. The total amount of memory for this part of your application is 256 bytes

To edit the phone book, click in a number or description, then make your changes.

Dialing a remote M90

1. To dial, highlight the number you want to dial, then click on the Dial button as shown below.



```
+++
AT
ATZ
AT&F
ATE0V1Q0X4&D0&S0&C1
```

	Number	Description
	9,17186862806	Brooklyn installation, Station 3
	9,916435999	Shandalee site, loom 12
4		
5		

Note that this Phone Book is used only by the PC's modem, although it is similar in appearance to the M90's Phone Book.

Communication Log

When you dial a remote modem using U90 Ladder, a window opens up in the bottom of your screen. The log of events is quickly displayed in this window. This log is stored as a .txt file. You can view this log by navigating to the U90 folder and opening a file named U90ldxxx.txt.

This log is stored as a .txt file. You can view this log by navigating to Unitronics\U90_Ladder\U90Ldxxx and opening a file named ComLog.txt.

In this file, the most recent log of events appears last.

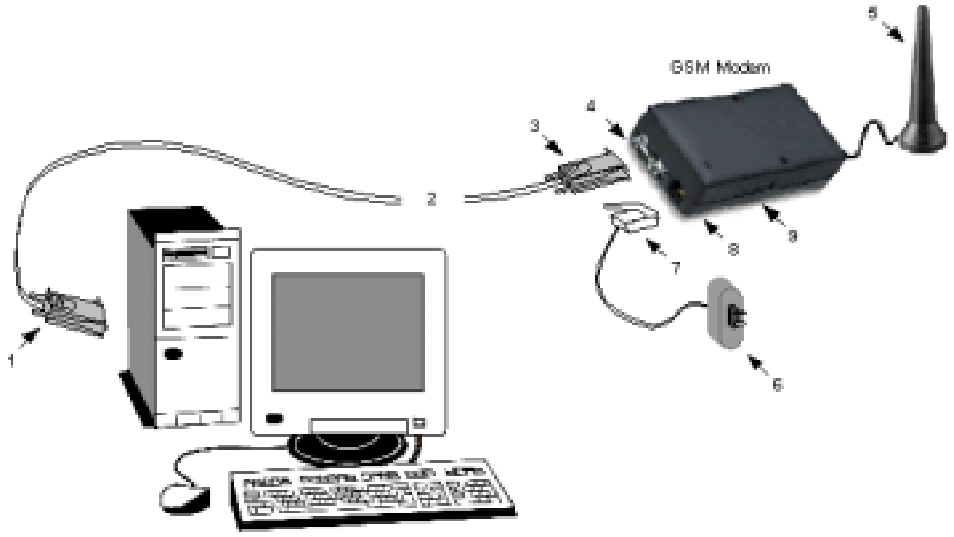
Note □ The PC-modem cable is not the same type of cable used to connect between the controller and the modem. Ensure that the cable used to connect the PC to the modem provides connection points for all of the modem's pins.
□ If call are routed via a switchboard, note that the switchboard settings may interfere with communications. Consult with your switchboard provider.
□ PC/PLC modem communications: Both PC and controller must use the same type of modem: either landline or GSM.
□ Internal PC modems must be used in conjunction with the driver provided by the modem's manufacturer.

Using a PC to access an M90 via GSM modem

To use a PC running U90 Ladder to access a remote M90 OPLC for programming and maintenance via GSM networks:

1. Connect your M90 to the GSM modem according to the instructions supplied with the GSM Modem Kit.

2. Connect your *PC to the GSM modem*.

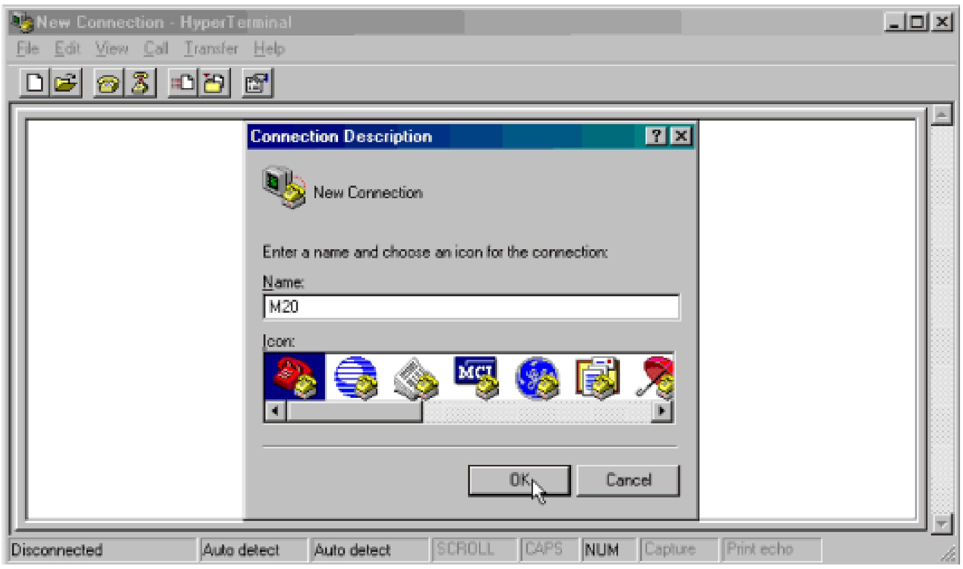


1.

1	RS232 connector
2	RS232 cable MJ10-22-CS28 (available by separate order)
3	RS232 connector
4	GSM Modem serial port
5	GSM antenna
6	Power supply PS-GSM modem (available by separate order)
7	RJ11 connector
8	GSM modem power supply
9	SIM card drawer

3. U90 Ladder's modem communication rate is set at 9600 bps. To enable the modem to communicate with U90 Ladder, *change the modem's default communication rate from 19200 bits per second (bps) to 9600 bps via Hyperterminal.*

   1. Open Hyperterminal. The program can generally be located by clicking the Start button in the lower left corner of your screen, then selecting Programs>Accessories>Communications>Hyperterminal. The New Connection window opens as shown below.
      Note ♦ Hyperterminal generally starts by pointing to the internal modem, if one is installed on the PC.



Connection Description

New Connection

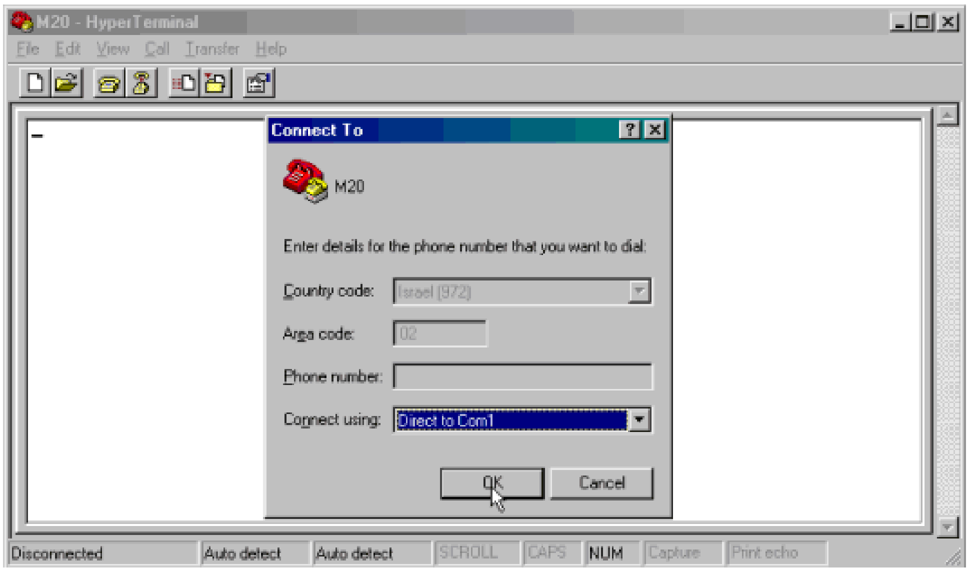
Enter a name and choose an icon for the connection:

Name: M20

Icon:

2. Enter a name for the new connection and select an icon, and then click OK. The Connect To box opens .

3. Select a COM port for the modem, and then click OK.



Connect To

M20

Enter details for the phone number that you want to dial:

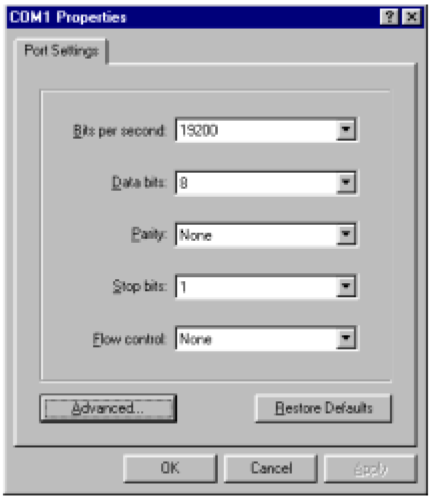
Country code: Israel (972)

Area code: 02

Phone number:

Connect using: Direct to Com1

4. The Port Settings box opens as shown below. To enable your PC to communicate with the modem, set the COM port parameters to a BPS of either 9600 or 19200, Data bits=8, Parity=N, Stop bits=1, Flow control=None, and then click OK.



COM1 Properties

Port Settings

Bits per second: 19200

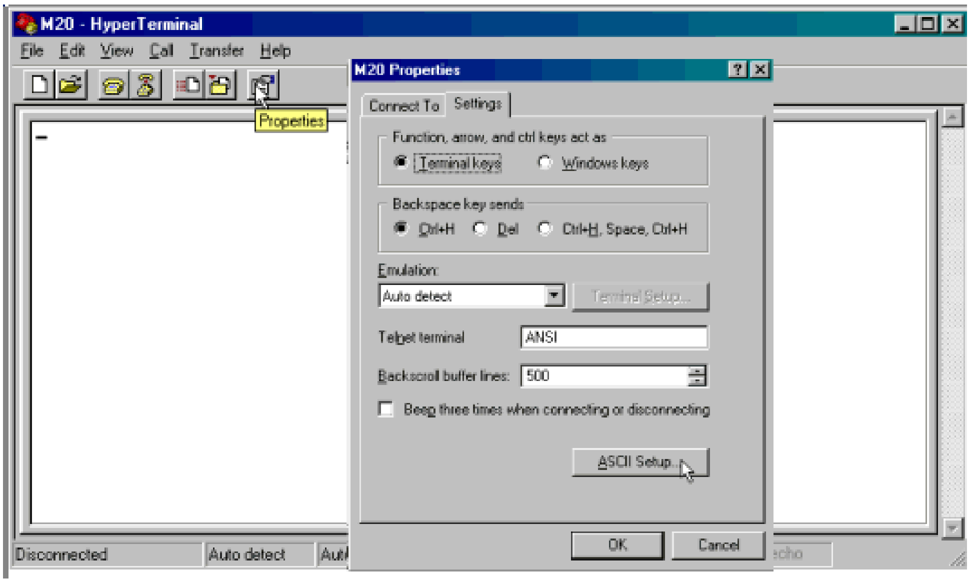
Data bits: 8

Parity: None

Stop bits: 1

Flow control: None

5. Open the modem's Properties box by clicking on the Properties button, then open ASCII Setup.



M20 Properties

Connect To | Settings

Function, arrow, and ctrl keys act as

● Terminal keys ○ Windows keys

Backspace key sends

● Ctrl+H ○ Del ○ Ctrl+H, Space, Ctrl+H

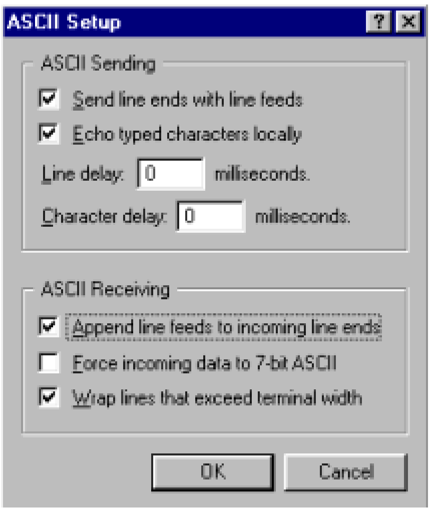
Emulation: Auto detect

Telnet terminal: ANSI

Backscroll buffer lines: 500

□ Beep three times when connecting or disconnecting

6. Select the options shown below, and then click OK.



ASCII Setup

ASCII Sending

☑ Send line ends with line feeds

☑ Echo typed characters locally

Line delay: 0 milliseconds.

Character delay: 0 milliseconds.

ASCII Receiving

☑ Append line feeds to incoming line ends

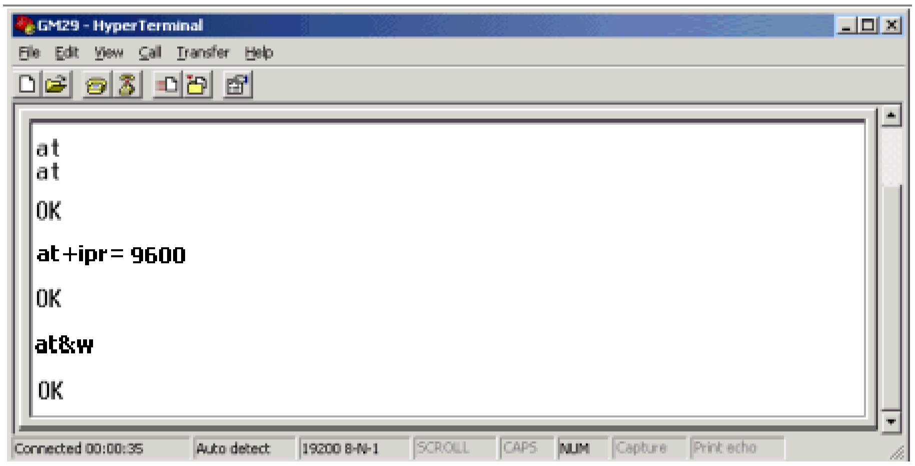
□ Force incoming data to 7-bit ASCII

☑ Wrap lines that exceed terminal width

Hyperterminal is now connected to your PC via Com 1; the ASCII settings now enable you to enter commands via the PC keyboard and see the replies from the modem within the Hyperterminal window.

To test the connection, type AT; if the connection is valid the modem will respond 'OK'.

To change the modem's baud rate, type AT+IPR=9600&W; the command '&W' burns the new baud rate into the modem's non-volatile memory.



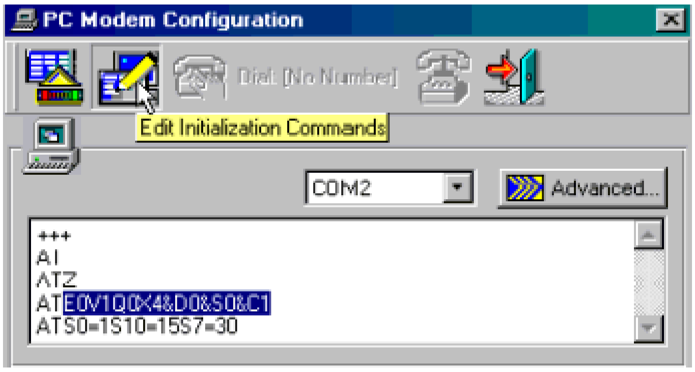
```
at
at
OK
at+ipr= 9600
OK
at&w
OK
```

Connected 00:00:35 | Auto detect | 19200 8-N-1 | SCROLL | CAPS | NUM | Capture | Print echo

You can reset the modem's communication rate by returning to this window and typing AT+IPR=19200&W.

4. Configure *U90 Ladder's modem initialization commands.*

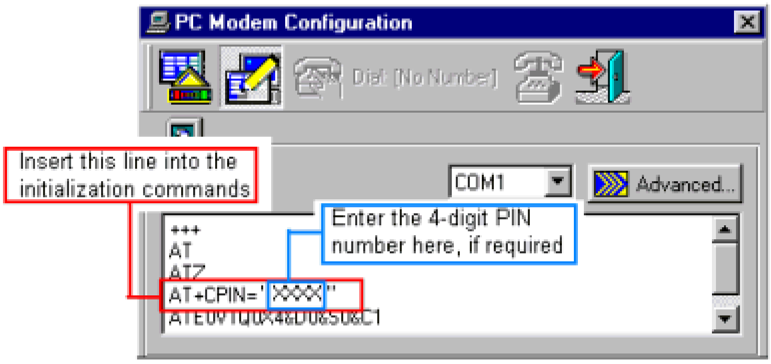
   1. Start U90 Ladder. Open the PC Modem Configuration box by selecting PC Modem Configuration from the Controller menu.
      To enable U90 Ladder to communicate with the GSM modem, you must edit the initialization commands.
   2. Access the initialization commands by clicking on the Edit Initialization Commands button shown below. The window containing the commands turns white; you can now edit commands.



COM2

```
+++
AI
ATZ
ATE0V1Q0X4&D0&S0&C1
ATS0=1S10=15S7=30
```

3. If you are using a SIM card that has a PIN number, enter a new initialization command AT+CPIN="XXXX", where XXXX is the 4-digit PIN #.

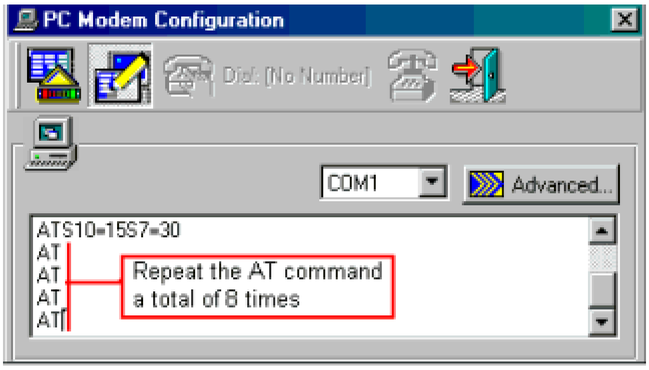


Insert this line into the initialization commands

Enter the 4-digit PIN number here, if required

```
+++
AT
ATZ
AT+CPIN="XXXX"
ATE0V1Q0X4&D0&S0&C1
```

4. End the list of commands by entering the AT command eight times as shown below.



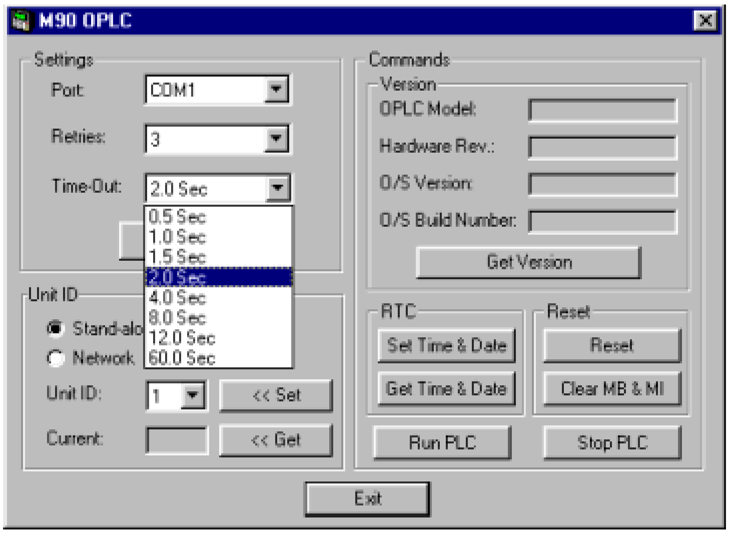
```
ATS10=15S7=30
AT
AT
AT
AT
```

Repeat the AT command a total of 8 times

5. After you have made these changes, close the PC Modem Configuration box.

6. Open the M90 OPLC box by selecting M90 OPLC from the Controller menu.

7. Set the M90 OPLC's Time-Out to 2 seconds as shown below. This should allow sufficient time for PC-to-M90 communications via the GSM modem.



Port: COM1

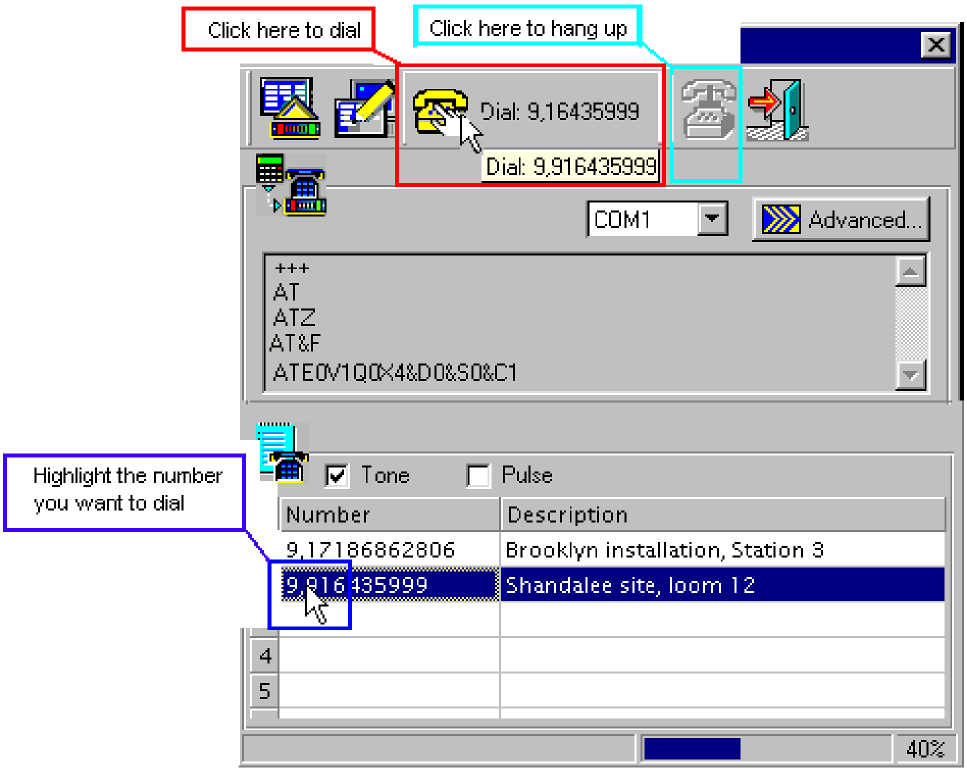
Retries: 3

Time-Out: 2.0 Sec

● Stand-alo
○ Network

Unit ID: 1

5. Dial the *remote M90 modem from your PC.*



Click here to dial

Click here to hang up

Highlight the number you want to dial

```
+++
AT
ATZ
AT&F
ATE0V1Q0X4&D0&S0&C1
```

	Number	Description
	9,17186862806	Brooklyn installation, Station 3
	9,916435999	Shandalee site, loom 12
4		
5		

Note

- Both GSM modems must contain SIM cards capable of data transfer. Check with your SIM card supplier to see if your SIM card is capable of data transfer.
- Note that only 3V SIM cards can be used with the GSM modem supplied with the Unitronics' GSM Modem Kits.

Modem Troubleshooting

General Information

Note

- The PC-modem cable is not the same type of cable used to connect between the controller and the modem. Ensure that the cable used to connect the PC to the modem provides connection points for all of the modem's pins.
- If calls are routed via a switchboard, note that the switchboard settings may interfere with communications. Consult with your switchboard provider
- If, within the modem initialization strings, the parameter S7 is too short to permit the PLC's modem to answer, an error will result.
  For example, if this parameter is set as S7=30, the PC modem will wait for 3 seconds to receive an answer from the PLC's modem. If the PLC modem does not answer before the 3 seconds have elapsed, the S7=30 parameter is exceeded, and the PC modem will return the No Carrier error.
- PC/PLC modem communications: Both PC and controller must use the same type of modem: either landline or GSM. Internal PC modems must be used in conjunction with the driver provided by the modem's manufacturer.

Modem commands

Note ♦ The modem must reply with either OK or READY to each command entered. If the modem fails to answer, the command has not been processed.

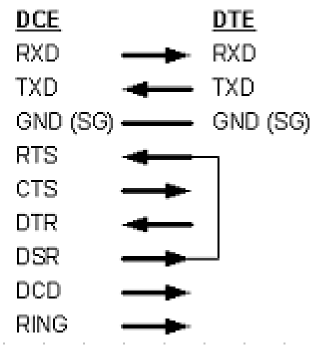
+++	Escape Sequence. This causes the modem to close connections and go back to command mode
AT	This command means Attention; and is used to begin a session
AT&F	Restores factory default settings
ATZ	Resets the modem. This command may take time to implement, so the response from the modem may be delayed
ATE0	No Echo
V1	Enable Verbose (long) response
Q0	Respond
X4	Detailed answers
&D0	Ignore DTR
&D2	Once DTR falls, disconnect and go to command mode
&D1	Once DTR falls, disconnect
&S0	DSR always ON. Since the DSR can be permanently set to ON, connecting it to the RTS causes the terminal always be ready to transmit\receive data
&S1	DSR OFF in command and test modes
&C1	Give the user a signal for the DCD
&C0	Don't give the user a signal for the DCD (refers to LED indications where relevant)
ATS0=1	Auto-Answer after 1 ring
S0=0	Modem doesn't answer. Forces PLC to answer with ATA (pickup)
S10=15	Sets the time (in units of 0.1 sec) from the time when CD is not detected, until the string NO CARRIER is shown. If the value is 255, then the CD signal will not fall—even if the modems are no longer connected
S7=30	Time-out: If this time is exceeded, the modem notifies that dial has failed
S12	The modem register that defines the time interval during which the line must remain clear, before and after the +++ command.
&W	Burn the configuration into the modem's non-volatile memory. **Note ♦** This is part of the COM Init FB's modem default initialization strings.

Modem Connections

This is the interface between the Data Communications Equipment (DCE; the modem) and the Data Terminal Equipment (DTE; the controller or PC). The arrows below show the direction of data flow.

Note ♦ Unitronics' controllers do not support the control lines. This is why the DTE side of the table comprises only 3 pins.

♦ Since the DSR can be permanently set to ON, connecting it to the RTS causes the terminal always be ready to transmit\receive data.



Data Flow Direction

Generally, when you transmit data, you send it out. Note, however, that transmitted data (TXD) is input to the DCE. A Receive Data signal (RXD) is input to the DTE, but output from the DCE.

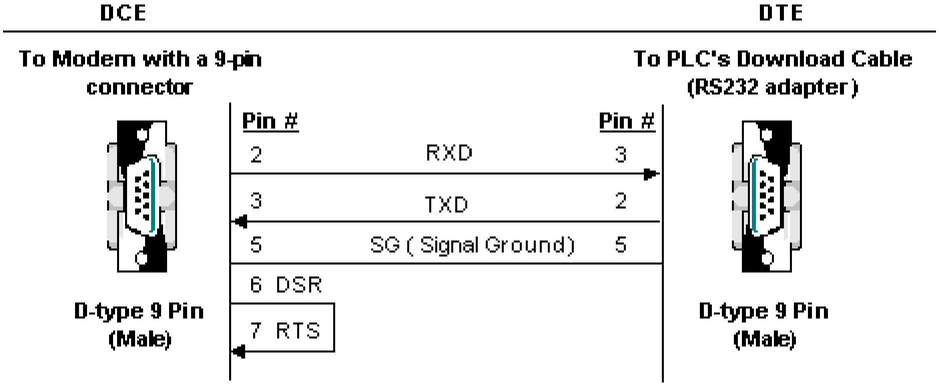
Therefore, the RXD and TXD signals are crossed within the majority of modems. This means that a straight through "one to one" cable is generally all that is necessary between a modem and a controller or PC serial port.

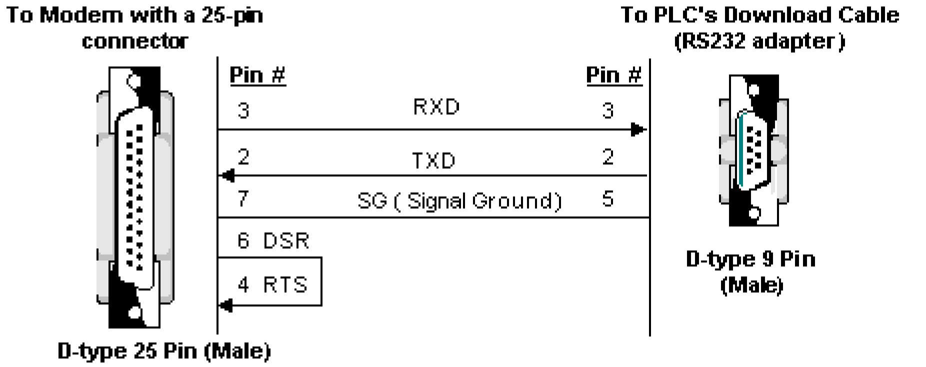
RS-232 signal information

RXD Receive Data	Input for DTE devices (Receive), output for DCE devices. This is the data channel from the DCE device to the DTE device.
TXD Transmit Data	Output for DTE devices (Send), input for DCE devices. This is the data channel from the DTE device to the DCE device.
GND Signal Ground	Signal return for all signal lines.
RTS Request To Send	Terminal is ready to receive data. When the DTE is ready to receive data, the DTE serial port RTS signal is ON.
CTS Clear To Send	Terminal is ready --not related to data transfer.
DTR Data Terminal Read	It is an output for DTE devices and an input for DCE devices. This signal is typically used in UNIX to show that the port has been activated or "opened".
DSR Data Set Ready	Detects if the RS232 is actually connected.
DCD Data Carrier Detect	Turns ON when the modems connect.
RING	Turns ON when someone is calling the DTE.

Cable Pin-out

The Unitronics' cable provided with modem kits does not provide a standard connection. This connection is adapted to support the fact that Unitronics controllers do not support the control lines. The cable shorts the DSR and the DTR together, which ensures that the terminal is always ready to receive data.





PC-side modem, error messages

This deals with errors that may result from the PC's modem

Message	Cause
Com Port not open, or modem does not exist	The PC was unable to access the PC port. The port may: -Already be in use -Be damaged.

Modem not connected	The PC receives no reply from the modem following the 'AT' command. Check that: -The modem is connected to the same PC port you have defined in PC-modem Configuration. -The PC-modem cable is in proper order.
Modem not initialized	The modem was not successfully initialized. Check the topic: Using Hyperterminal for Modem Troubleshooting
The messages below describe the modem 's status if the PC dial attempt (ATD+ number) fails. Any one of these errors aborts the Dial process.	
Modem Busy	
Modem Error	
No Dial Tone	
No Carrier	Note This can occur if, within the modem initialization strings, the parameter S7 TimeOut, is to short to permit the PLC's modem to answer. For example, if this parameter is set as S7=30, the PC modem will wait for 3 seconds to receive an answer from the PLC's modem. If the PLC modem does not answer before the 3 seconds have elapsed, the S7=30 parameter is exceeded, and the PC modem will return the No Carrier error.
Dial time-out exceeded	No reply was received from the modem within the defined time.
The messages below only relate to unsuccessful GSM modem initialization.	
GSM SIM card blocked	
GSM SIM card does not exist	
Illegal GSM PIN code	
GSM Network not found	
Time-out exceeded	

M90-side modems

These errors may result from problems in the PLC-side modem

Message	Possible cause	Recommended action
Modem Busy	Modem is engaged, or is being initialized	Check that the line is free. Use the SBs: Modem Initialization Status listed above to check the COM port status; communications cannot flow through the port during initialization. For more information check the topic How the M90 works with a modem.
Handshake between modems complete ('CONNECT'), PLC does not reply	Modem adapter cable	Check the PLC-to-modem connection and pin-out, particularly that the DSR is connected to the RTS on the modem side.

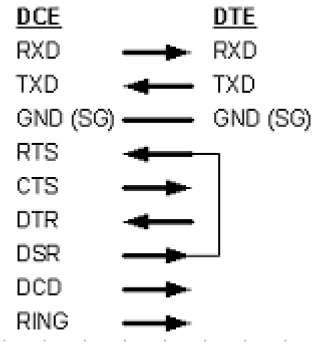
Problem	SB74	Possible Cause & Recommended Action
Modem fails to initialize	ON	1. PLC-to-modem cable: Make sure that the cable is securely connected. Check the modem connection and pin-out of the PLC-to-modem adapter cables. 1. Incompatible communication settings. Most modems automatically match the parameters of incoming data: baud rate, data bits, parity & stop bits. You may need to manually change your modem's communication settings via Hyperterminal.

Modem Connection

This is the interface between the Data Communications Equipment (DCE; the modem) and the Data Terminal Equipment (DTE; the controller or PC). The arrows below show the direction of data flow.

Note □ Unitronics' controllers do not support the control lines. This is why the DTE side of the table comprises only 3 pins.

Note□ Since the DSR can be permanently set to ON, connecting it to the RTS causes the terminal always be ready to transmit\receive data.



M90 modem communication problems

If your M90 is transferring data via modem, you can begin troubleshooting by entering Information Mode. You can then check the status of relevant System Bits and Integers to help diagnose the problem.

To begin diagnosing the problem, check the error code contained in SI 70. Refer to the error code table in How the M90 works with a modem.

The tables below show the more common causes of modem communication problems.

Problem	SI 70 value	Possible Cause & Recommended Action
Modem fails to initialize	2: Modem Did Not Reply	M90-to-modem cable: Make sure that the cable is securely connected. Check the M90 modem connection and pin-out of the M90-to-modem adapter cables. Note that if you use cables comprising this pin-out, you must set the RS232 parameter Flow Control to N (none). Incompatible communication settings. Most modems automatically match the parameters of incoming data: baud rate, data bits, parity & stop bits. The M90's embedded modem settings are: 9600, 8 data bits, no parity, 1 stop bit. You may need to manually change your modem's communication settings to match these via Hyperterminal.
	0: No Error	SB 72 OFF: In order to work with a modem, you must select 'Use modem' in the M90 Modem Configuration box . This causes SB 72 Initialize Modem to turn ON when the M90 powers up. Note that if the M90 has also been configured to use SMS messaging, that the M90 will not be able to connect to a modem because the SMS feature overrides the modem. Check too, that SB 72 is not disabled in your program.
	6: Modem Report Error	Check the modem initialization commands. Refer to Configuring the M90 to use a modem.

Other problems:

Problem	Possible cause	Recommended action
Modem is busy	Modem is engaged, or is being initialized	Check that the modem is free. Use the SBs: Modem Initialization Status listed above to check the COM port status; communications cannot flow through the port during initialization. For more information check the topic How the M90 works with a modem.
Handshake	Modem	Check the PLC-to-modem connection and pin-out,

between modems complete ('CONNECT'), PLC does not reply	adapter cable	particularly that the DSR is connected to the RTS on the modem side, as shown in Modem Connections above.
M90 does not dial	Incorrect phone number	Check the M90's phone book. Refer to Configuring the M90 to use a modem.

GSM modems

Problem	Possible Cause & Recommended Action
Cell phone does not receive message	Check the cell phone's SIM card; it may be full.

Check SI 180 for the error messages listed below.

Error Messages (SI 180)

Number	Error Message	Description
0	No error	No error found
1	GSM Modem Not Initialized	The GSM modem was not initialized. Before using the SMS feature the modem must be initialized. Refer to relevant help sections.
2	GSM Modem Did Not Reply	The GSM modem referred to is the one on the M90 side.
3	Modem Reports Unknown Message	Modem returns an unrecognized reply
5	Wrong PIN number	The Personal Identification Number that was given does not match that of the SIM card installed in the M90's GSM modem.
6	Failed Registration	GSM modem did not register successfully, for example if no network was found, or if the modem antenna is not functioning.
7	No Phone Number	SI 181 contains a number that is not linked to any phone number stored in the GSM phone book.
8	Transmit: Undefined String number	SI 182 contains a string number that is not linked to any string number stored in the SMS Messages List.
9	Unauthorized Origin	This SMS string has been transmitted from an unauthorized phone number.
11	Illegal String Received	The string received is not linked to any string stored in the SMS Messages List. SI 184 will contain 0.
14	RS232 Port Busy	The RS232 port is already in use; for example, the modem is currently connected.
16	SMS not successfully sent to all numbers	The SMS message was not successfully sent to all the phone numbers for which it was configured.
17	PUK number needed	The SIM card is locked due to too many attempts to enter an incorrect PIN number.

System Bits

SB	Symbol	Description
180	Initialize GSM Modem for SMS	This is necessary to enable use of the SMS feature. Note that the modem must first be initialized using SB 70.

181	SMS: Initialization Succeeded	Signals that GSM modem has been initialized. The modem is now ready to send and receive SMS messages.
182	SMS: Initialization Failed	Signals that GSM modem has failed. SI 180 contains the error code.
183	Send SMS	Send the string that is represented by the index number stored in SI 182, to the phone number represented by the index number stored in SI 181.
184	SMS: Transmission succeeded	Signals that SMS has been successfully transmitted
185	SMS: Transmission Failed	Signals that SMS has failed. SI 180 contains the error code
186	SMS Received	Signals that a defined SMS has been received. SI 183 contains the index number identifying the origin of the SMS, if this number has been stored in the SMS phone book. If the number is not found, SI 183 equals 0. SI 184 contains the index number of the SMS string that has been received. Only messages that have been defined in the SMS messages list can be received by the M90.
187	Error in Received SMS	This bit signals one of the errors listed below. SI 180 contains the error code.
188	Ignore Received SMS	Allows the user to block reception of SMS messages
189	Print SMS message	This prints a message with CR (Carriage Return) & LF (Line Feed)
190	Print SMS message	This prints a message with LF (Line Feed)
191	Print SMS message	This prints a message without CR (Carriage Return) or LF (Line Feed)
192	Get GSM antenna signal quality	Get GSM antennae signal quality. The signal quality is contained in SI 185 GSM Signal Quality.
193	Delete SMS messages from SIM	Deletes all of the SMS messages from the SIM card.
194	Print SMS message	This prints a message including STX and ETX.

System Integers

SI	Symbol	Description
180	SMS Error Code	Contains an error code resulting from a SMS error. The list is shown above.
181	SMS: Send to Phone Number	Contains the index number of a phone number within the GSM phone book. Use the Store Direct function to place the index number of the desired phone number in SI 181. Storing the value '0' into SI 181 causes a message to be sent to the **last** number to which an SMS message was sent. When auto-acknowledge is selected, the number 7 will be automatically placed into S1 181 when the SMS is acknowledged.
182	SMS: String Number to Send	Contains the index number that represents the SMS string to be sent. Use the Store Direct function to place the index number of the desired SMS string in SI 182.
183	Origin of Received SMS	Contains the index number that represents the phone number from which the SMS was sent. If this number is not defined in the GSM phone book, SI 183 will contain 0.
184	Received SMS String	Contains the index number that represents the SMS that has been received. If this number is not defined in the SMS message list, SI 184 will contain 0.
185	GSM Signal	GSM antenna signal quality. If this is less than 11, reposition the

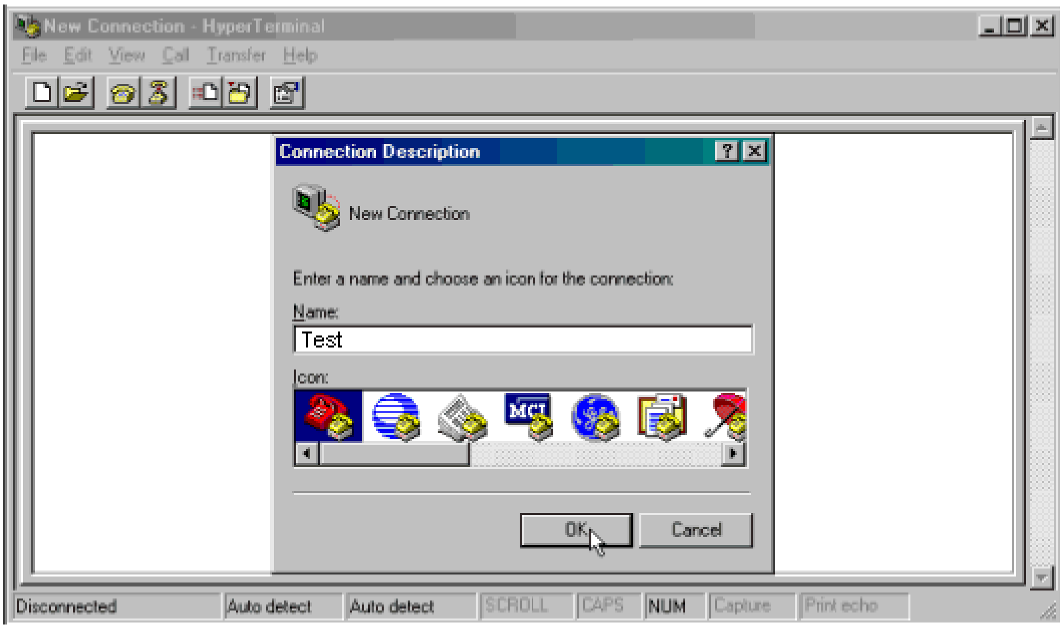
Quality	antenna. You can use SB 192, Get GSM antennae signal quality,together with this SI.

Using Hyperterminal to check PC-PLC direct communications

If the PC port is defective or in use by another application, you may be unable to access a directly connected controller with your PC.

Via Hyperterminal, you can check the PLC-PC communication connection by sending a simple text command, Get ID. If the connection is in order, the controller replies with its ID; if the connection is faulty, the controller will not reply.

1. Open Hyperterminal.



New Connection - HyperTerminal

Connection Description

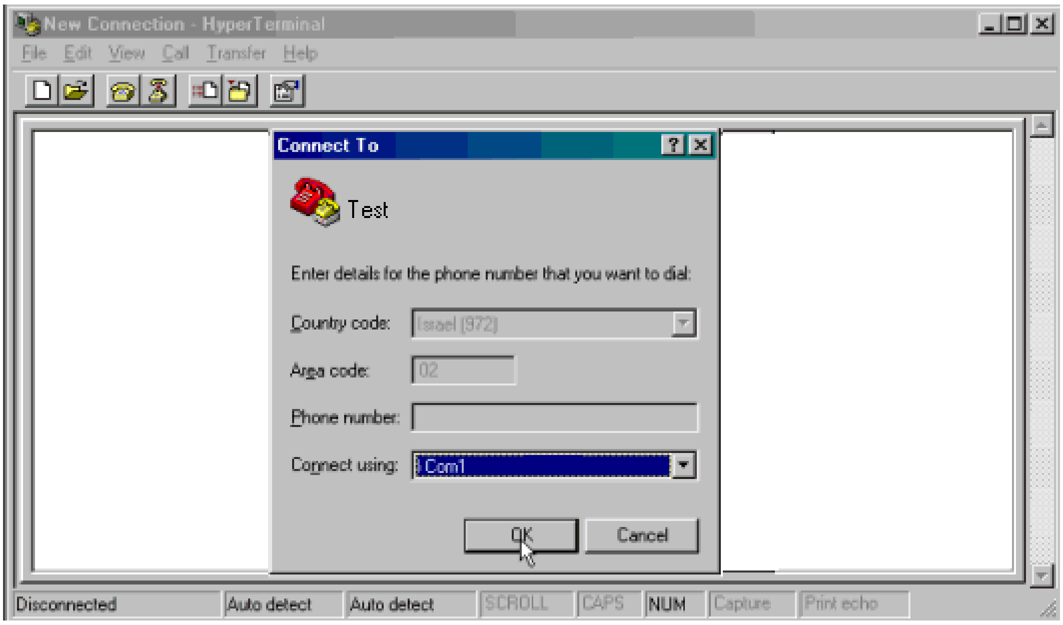
New Connection

Enter a name and choose an icon for the connection:

Name: Test

2. Enter a name for the new connection and select an icon, and then click OK. The Connect To box opens.

3. Select the PC COM port that connects the PC to the controller, and then click OK.



New Connection - HyperTerminal

Connect To

Test

Enter details for the phone number that you want to dial:

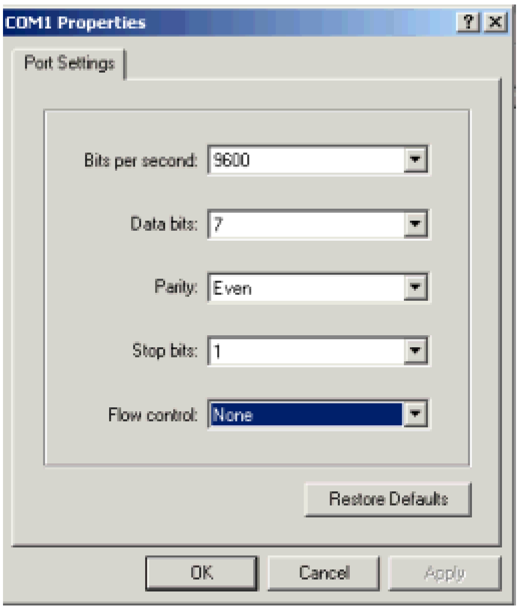
Country code: Israel (972)

Area code: 02

Phone number:

Connect using: Com1

4. The Port Settings box opens as shown below. To enable your PC to communicate with the controller, set the COM port parameters to the M90 default settings:BPS 9600, Data bits=7, Parity=N, Stop bits=1, Flow control=None, and then click OK.



COM1 Properties

Port Settings

Bits per second: 9600

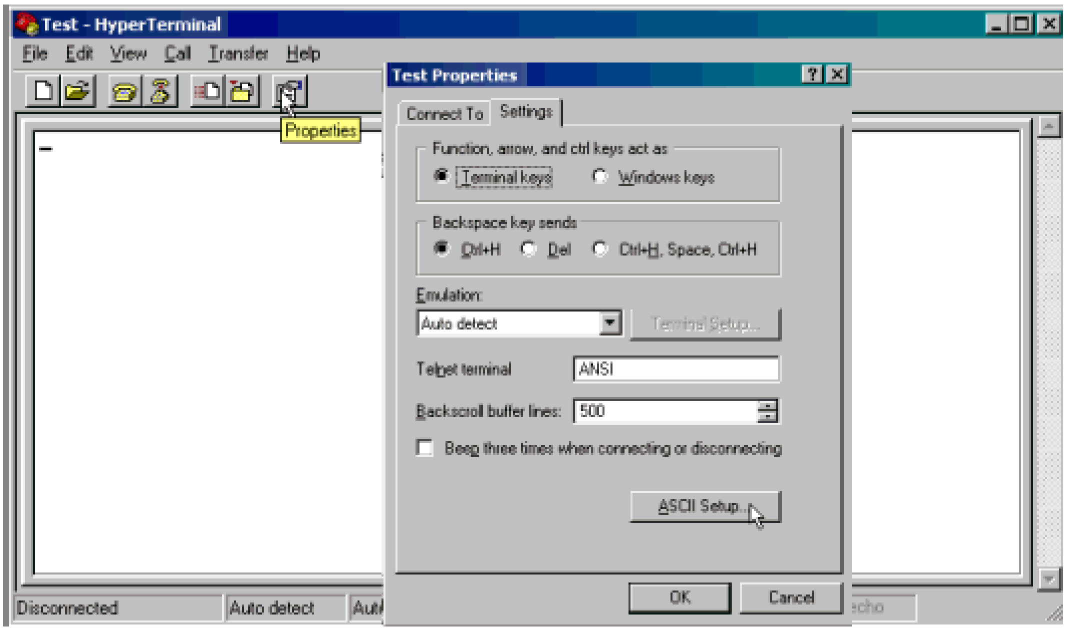
Data bits: 7

Parity: Even

Stop bits: 1

Flow control: None

5. Open the Properties box by clicking on the Properties button, then open ASCII Setup.



Test - HyperTerminal

Test Properties

Connect To | Settings

Function, arrow, and ctrl keys act as: ● Terminal keys ○ Windows keys

Backspace key sends: ● Ctrl+H ○ Del ○ Ctrl+H, Space, Ctrl+H

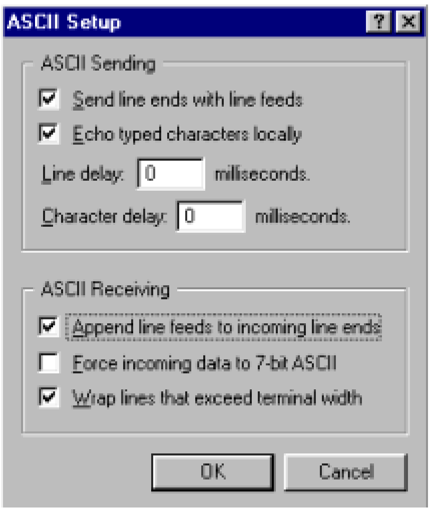
Emulation: Auto detect

Telnet terminal: ANSI

Backscroll buffer lines: 500

□ Beep three times when connecting or disconnecting

6. Select the options shown below, and then click OK.



ASCII Setup

ASCII Sending

☑ Send line ends with line feeds

☑ Echo typed characters locally

Line delay: 0 milliseconds.

Character delay: 0 milliseconds.

ASCII Receiving

☑ Append line feeds to incoming line ends

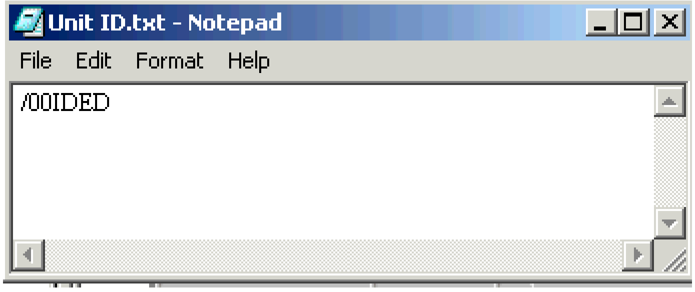
□ Force incoming data to 7-bit ASCII

☑ Wrap lines that exceed terminal width

7. To synchronize the controller's communication settings, enter Info mode. Navigate to System>RS232>Restore Defaults, and then press the Enter key.

8. Open Notepad, enter the text **/00IDED**, press Enter, and save the file. This is the Get ID command, where 00 is the 'placeholder' for the controller's Unit ID number. 00 enables any directly controller to answer, no matter what it's actual ID number is. ED is the command's checksum.

Note ♦ Pressing Enter places a Carriage Return command at the end of the text. Although the Carriage Return is not visible, the command will not be processed without it.

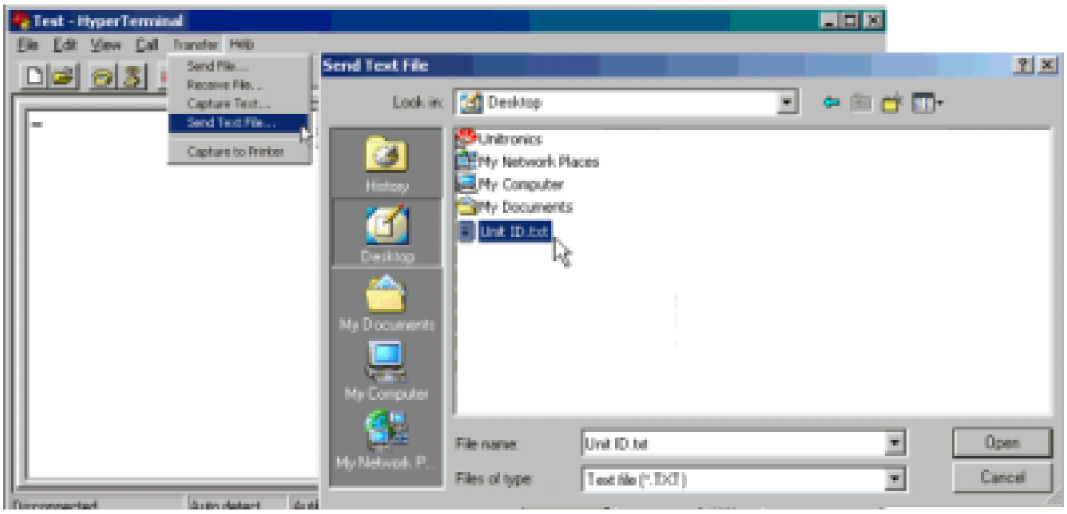


Unit ID.txt - Notepad

```
/00IDED
```

9. Select Send Text file from the Transfer menu, and open the text file

.



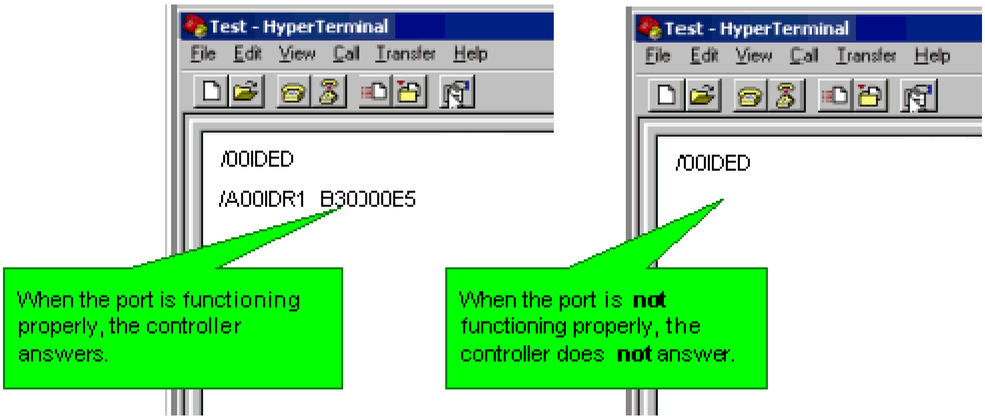
Send Text File

Look in: Desktop

File name: Unit ID.txt

Files of type: Text file (*.TXT)

10. If you have configured everything according to the instructions above, and the port is functioning properly, the controller with its ID number. If the port is out of order, the controller will not reply.


Test - HyperTerminal
File Edit View Call Transfer Help
Test - HyperTerminal
File Edit View Call Transfer Help

```
/00IDED
/A00IDR1  B30000E5
```

When the port is functioning properly, the controller answers.

```
/00IDED
```

When the port is **not** functioning properly, the controller does **not** answer.

In the figure above, the characters in the string that is returned by the controller, /A00IDR1 B30000E5 may be interpreted as follows:

A	00	ID (PLC model)	B (Hardware Version)	E5	CR (ETX)
Answer	Requested number	M90-R1	OS V3.00 (00)	Checksum	Not visible

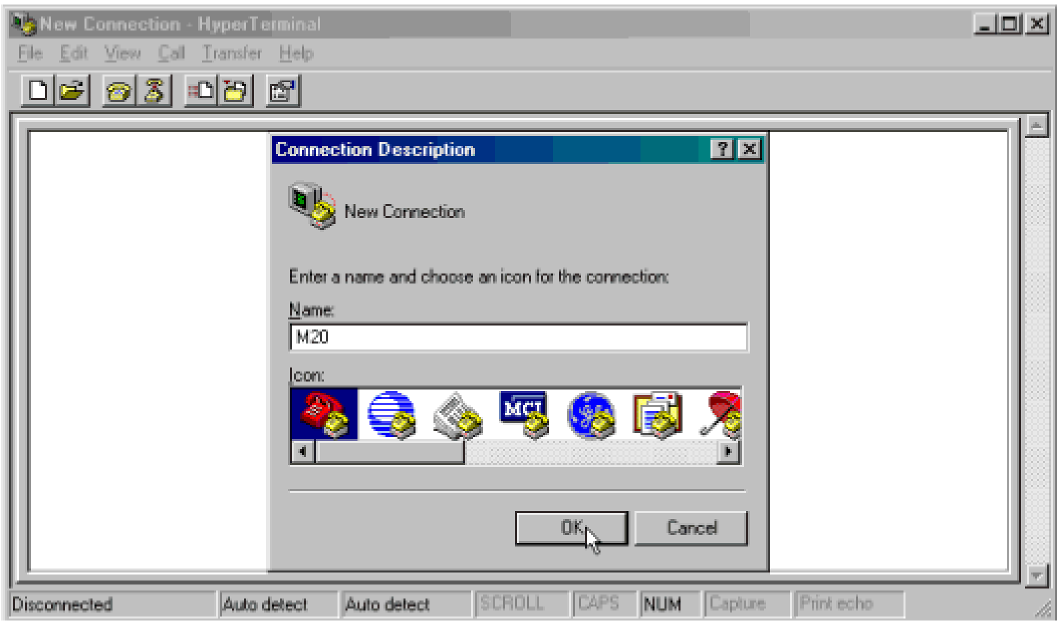
Using Hyperterminal for Modem Troubleshooting

You can use a standard Windows application called Hyperterminal to perform certain tasks, such as changing a modem's communication rate.

Note ♦ The modem driver does not need to be installed in order to access the modem via Hyperterminal.

Using Hyperterminal

2. Open Hyperterminal. The program can generally be located by clicking the Start button in the lower left corner of your screen, then selecting Programs>Accessories>Communications>Hyperterminal. The New Connection window opens as shown below.
Note ♦ Hyperterminal generally starts by pointing to the internal modem, if one is installed on the PC.



New Connection - HyperTerminal

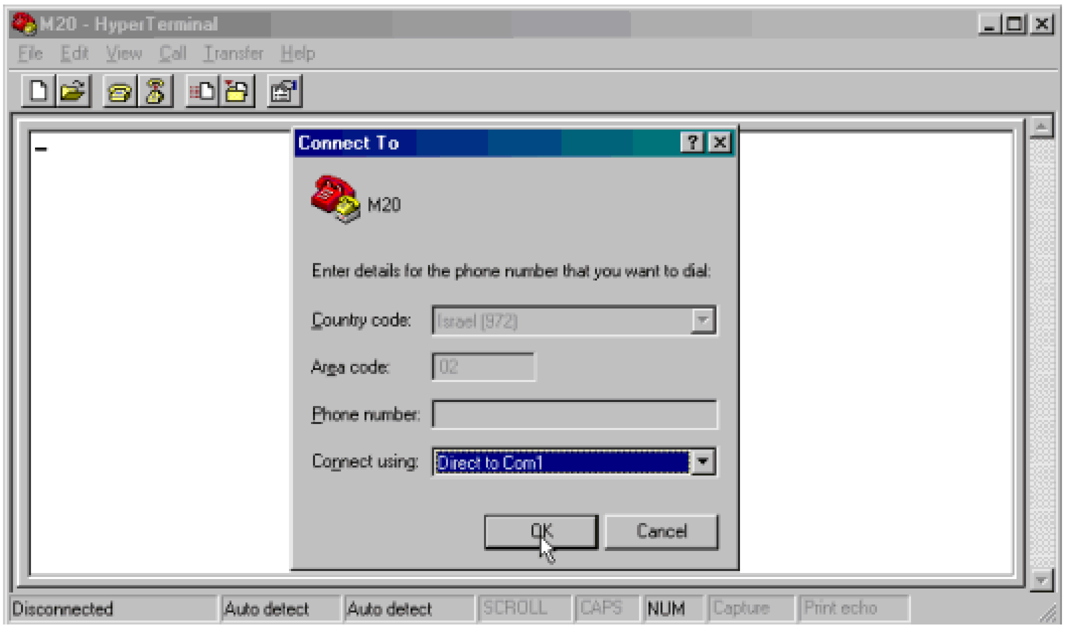
Connection Description

New Connection

Enter a name and choose an icon for the connection:

Name: M20

1. Enter a name for the new connection and select an icon, and then click OK. The Connect To box opens .
1. Select a COM port for the modem, and then click OK.



M20 - HyperTerminal

Connect To

M20

Enter details for the phone number that you want to dial:

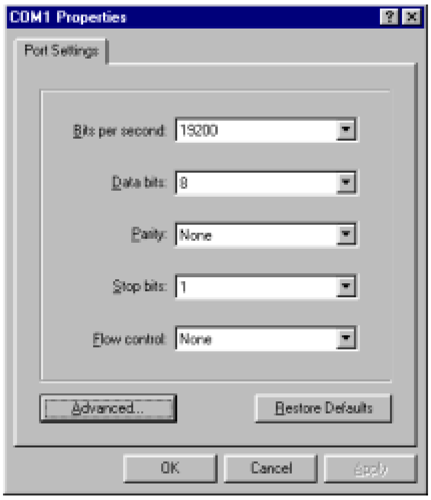
Country code: Israel (972)

Area code: 02

Phone number:

Connect using: Direct to Com1

1. The Port Settings box opens as shown below. To enable your PC to communicate with the modem, set the COM port parameters to a BPS of either 9600 or 19200, Data bits=8, Parity=N, Stop bits=1, Flow control=None, and then click OK.


COM1 Properties

Port Settings

Bits per second: 19200

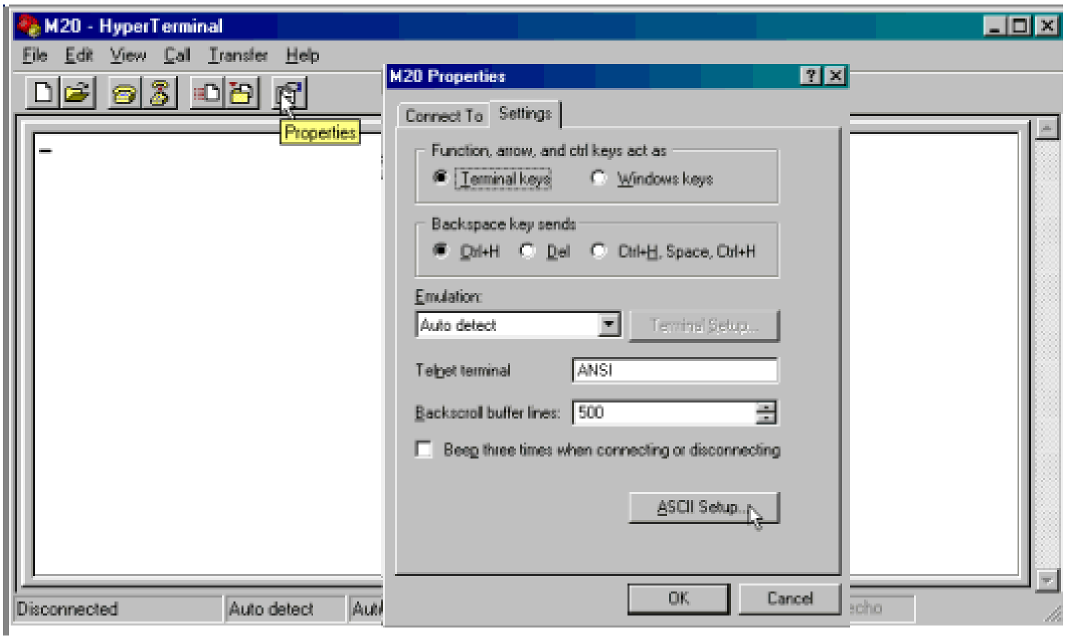
Data bits: 8

Parity: None

Stop bits: 1

Flow control: None

1. Open the modem's Properties box by clicking on the Properties button, then open ASCII Setup.


M20 - HyperTerminal

M20 Properties

Connect To | Settings

Function, arrow, and ctrl keys act as

● Terminal keys ○ Windows keys

Backspace key sends

● Ctrl+H ○ Del ○ Ctrl+H, Space, Ctrl+H

Emulation: Auto detect

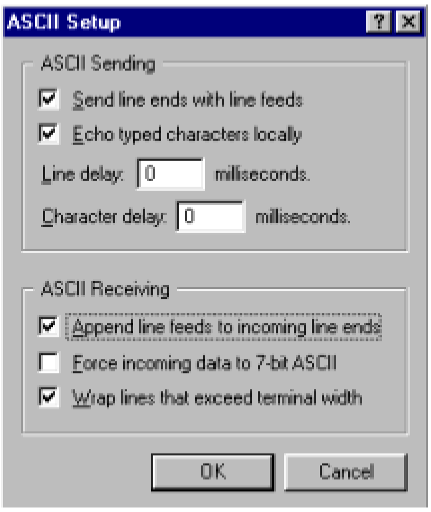
Telnet terminal: ANSI

Backscroll buffer lines: 500

□ Beep three times when connecting or disconnecting

Disconnected | Auto detect | Aut | echo

1. Select the options shown below, and then click OK.


ASCII Setup

ASCII Sending

☑ Send line ends with line feeds

☑ Echo typed characters locally

Line delay: 0 milliseconds.

Character delay: 0 milliseconds.

ASCII Receiving

☑ Append line feeds to incoming line ends

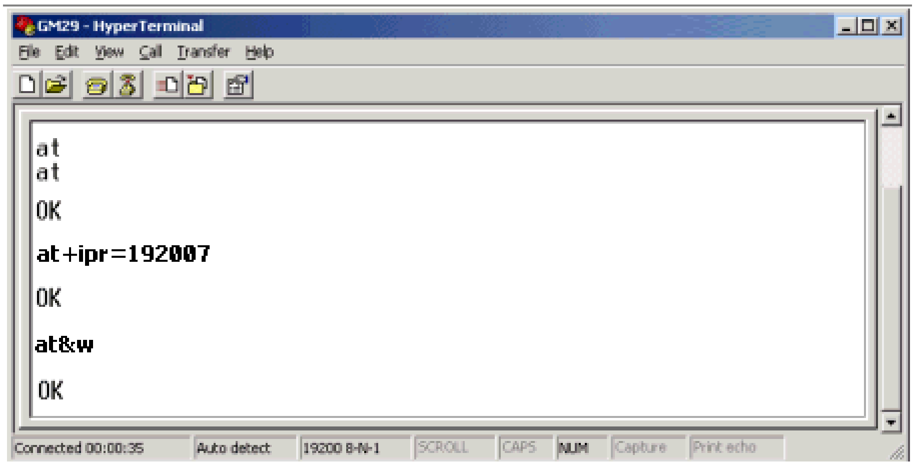
□ Force incoming data to 7-bit ASCII

☑ Wrap lines that exceed terminal width

Hyperterminal is now connected to your PC via Com 1; the ASCII settings now enable you to enter commands via the PC keyboard and see the replies from the modem within the Hyperterminal window.

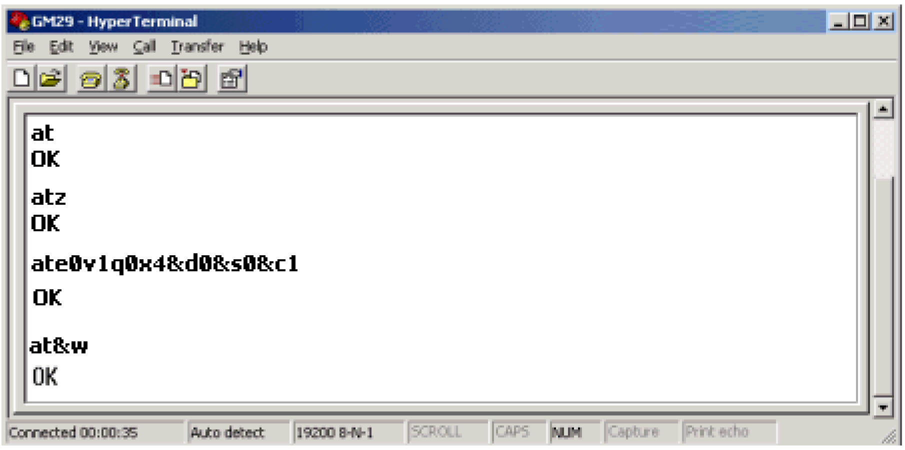
To test the connection, type AT; if the connection is valid the modem will respond 'OK'.

To change the modem's baud rate, type AT+IPR=19200&W; the command '&W' burns the new baud rate into the modem's non-volatile memory.



```
at
at
OK
at+ipr=192007
OK
at&w
OK
```

Typical initialization strings used with an Siemens M20-type modem are shown below.



```
at
OK
atz
OK
ate0v1q0x4&d0&s0&c1
OK
at&w
OK
```

Modem Commands

Note □ The modem must reply with either OK or READY to each command entered. If the modem fails to answer, the command has not been processed.

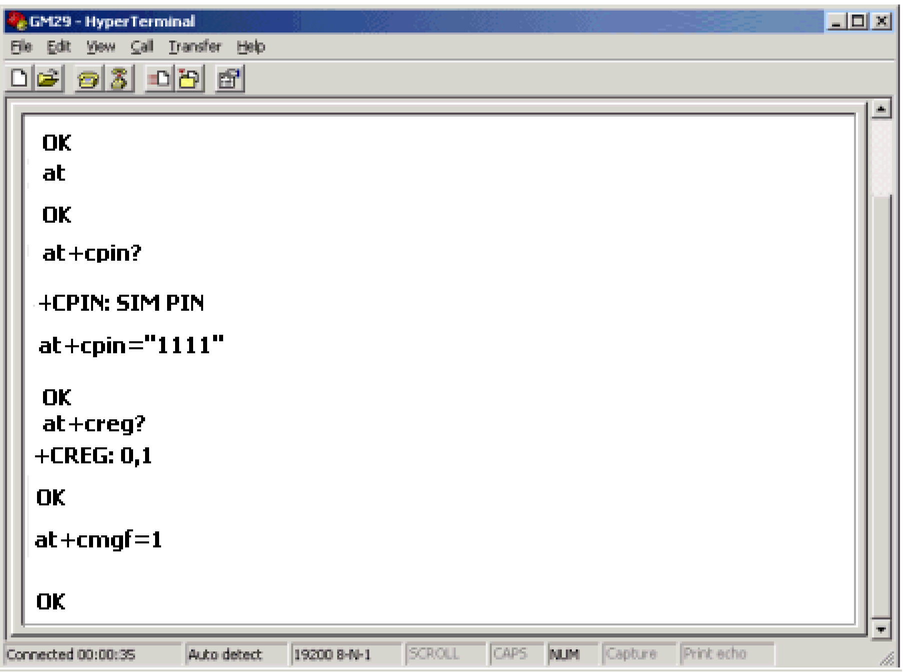
+++	Escape Sequence. This causes the modem to close connections and go back to command mode
AT	This command means Attention; and is used to begin a session
AT&F	Restores factory default settings
ATZ	Resets the modem. This command may take time to implement, so the response from the modem may be delayed
ATE0	No Echo
V1	Enable Verbose (long) response
Q0	Respond
X4	Detailed answers
&D0	Ignore DTR
&D2	Once DTR falls, disconnect and go to command mode
&D1	Once DTR falls, disconnect

&S0	DSR always ON. Since the DSR can be permanently set to ON, connecting it to the RTS causes the terminal always be ready to transmit\receive data
&S1	DSR OFF in command and test modes
&C1	Give the user a signal for the DCD
&C0	Don't give the user a signal for the DCD (refers to LED indications where relevant)
ATS0=1	Auto-Answer after 1 ring
S0=0	Modem doesn't answer. Forces PLC to answer with ATA (pickup)
S10=15	Sets the time (in units of 0.1 sec) from the time when CD is not detected, until the string NO CARRIER is shown. If the value is 255, then the CD signal will not fall—even if the modems are no longer connected
S7=30	TimeOut: If this time is exceeded, the modem notifies that dial has failed
S12	The modem register that defines the time interval during which the line must remain clear, before and after the +++ command. Note □ In the M90, this value is fixed on the M90 side and is not entered into the modem. If the controller cannot hang up, register S12 should be checked to ensure that the pause =1.2 sec
&W	Burn the configuration into the modem's non-volatile memory

Initializing the modem to SMS mode via Hyperterminal

Once the modem is successfully initialized, you can use Hyperterminal to initialize the modem to SMS mode.

Command	Description	Notes
at+cpin=?	Is a pin number required?	
at+pin="xxxx"	Is the pin number set in the application?	XXXX is the PIN number coming from the U90 application.
at+creg?	Has the SIM card been registered with the local cellular provider?	Should return one of two answers: • +CREG: 0,1 The SIM is registered with its local provider. • +CREG: 0,5 The SIM is in **roaming** mode.
at+cmgf=1	Go to text mode	



```
OK
at
OK
at+cpin?
+CPIN: SIM PIN
at+cpin="1111"
OK
at+creg?
+CREG: 0,1
OK
at+cmgf=1
OK
```

Notes ♦ Commands including question marks are run for verification **twice.** If the command is not verified during the second attempt, the attempts stop.

♦ If the SIM requests the PUK number, the SIM must be taken out of the modem and installed into a phone to enable the number to be entered.

- ♦ If the SIM is full, the SIM must be taken out of the modem and installed into a phone to enable the SIM to be cleared.
- ♦ The modem must be able to support Text mode. P.D.U. mode is not supported.

When a controller sends an SMS text message

- The controller uses the Send command, containing the number to be called: AT+CMGS= ”phone number”.
- The controller then waits for the reply '>'.
- When the '>' is received, the controller sends the message, ending the line with CTRL_Z
- If the message is successfully sent, the controller will receive a message of confirmation,+CMGS:xx. When this message is received by the controller, SB 184 turns ON. The confirmation message is acknowledged by OK.
- If :
  the message of confirmation is not received within 15 seconds, or
  the '>' is not received within 3 seconds, SB 185 turns ON.

When the controller receives an SMS text message:

- It receives the command: +CMTI: “SM” ,xx. Xx is a number in the controller's memory, 1 to 20.
- When the message is received, the controller asks the modem for the text via the command AT+CMGR=xx
- The modem replies with +CMGR, including the phone number, status, text, and concluding with OK.

Note ♦ When a Com port has been successfully initialized, the relevant bit turns ON: SB 80 , 82, 83 or 84.
If initialization fails, SB 81, 83, 85, or 87 will turn ON.

'The Sniffer'--Viewing communication strings

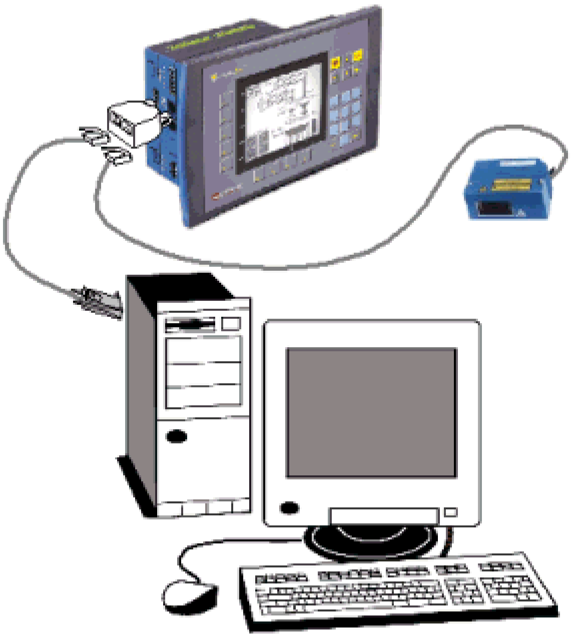
The instructions below show you how to construct a communications 'Sniffer'. This device enables you to use Hyperterminal to view communication strings flowing between a PLC and an external, connected device such as a bar code reader.

‘Sniffer’ is connected to the external device.

‘COM’ is connected to the PLC.

The completed Sniffer is connected to a PLC communication port, PC and external device.

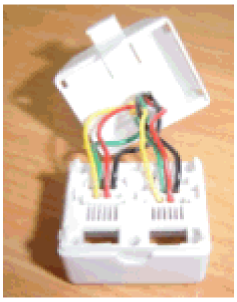
Note that communication cables are the programming cable provided by Unitronics.



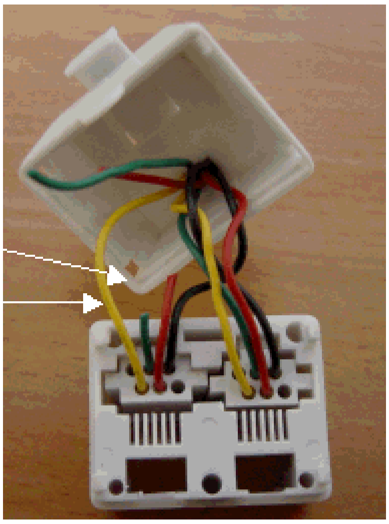
To make a Sniffer, you need:

- An adapter.
- Two 1N4148 or 1N914 diodes.

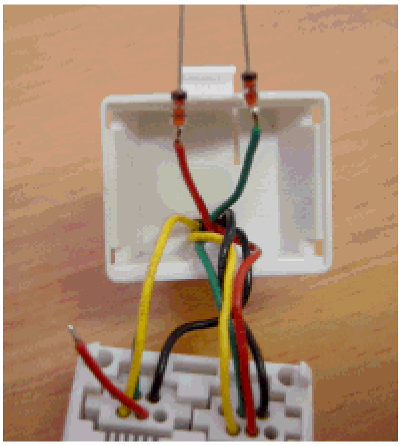
1. Open the adapter carefully via the 4 snaps in its sides.



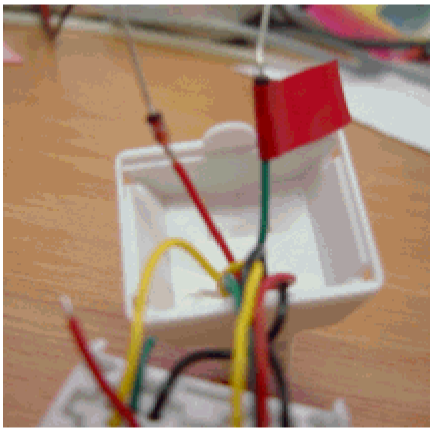
1. Cut the red and green wires as shown below.



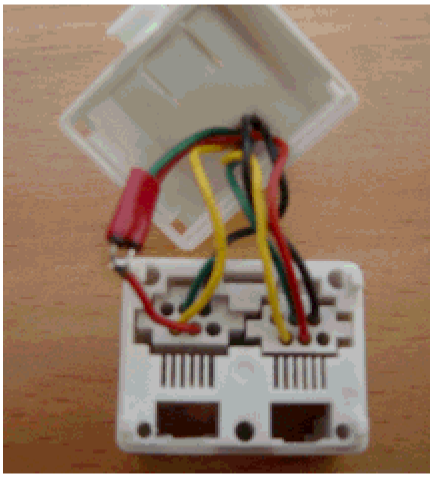
1. Solder one diode to the red wire, and one diode to the green wire. The soldered point provides the anode.



- Put isolating material on the soldered points.



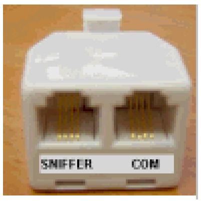
4. Solder both diodes' cathodes to the red wire.



3. Put isolating material on the solder.

5. Close the Sniffer.

3. Label the connectors as



shown.

Note ♦ In order to run view the strings in Hyperterminal, you must set the program to display ASCII strings as described above in Using Hyperterminal.

Troubleshooting

Direct Communication problems

If your PC is not able to establish direct communications with a locally connected M90, refer to the following table:

Problem	Possible Cause	Recommended Action
No Communications	M90 is not turned on	Turn M90 on. If the M90 does not turn on, click here.
	Communication cable	Check that: • You are using the correct communication cable. • The RS232 port of the M90 is connected to your PC's communication port according to the instructions in the M90 User Guide.
	Communication settings	Refer to M90 Communications Settings.
		If you are trying to communicate with an M90 network via a bridge, you must define your project as a Network project--however, U90 Ladder cannot automatically detect communication settings in a Network project. Make sure the current RS232 parameters in your project are the same as the parameters that are actually in the bridge. Remember: to download via an M90 bridge to a networked M90, you must select the unique ID of the networked M90.
	COM port is not enabled	Check that your PC communication port is enabled. This means checking your PC's BIOS/CMOS setup.
	COM port is defective	Refer to How do I use a PC to access an M90 via GSM modem?
	COM port is occupied	Close the application that is accessing the port. For more information, refer to How do I use a PC to access an M90 via GSM modem?

If you are still unable to establish communications:

Contact your local Unitronics distributor.

Why does the Controller display the 'Restart' message?

The most common reason for this event is a peak in electromagnetic (EMF) 'noise'. This may result from contactors, power relays, solenoid valves, etc. switching on and off, as well as from power transformers and motor speed drivers.

Recommendations

- Use different power supplies - highly recommended - one for the controller (CPU and inputs), and a different one for other electromagnetic devices;
- Use suppressors - reverse connected diodes for DC loads and RC filters for AC loads;
- Where possible, place the signal cables, including the 24V power supply, far away from power lines, especially from cables, coming in and out of motor drivers;

- If needed, use shielded cables for signals, including for 24 VDC and for power cables between the motor driver and the motor itself.

Taking these precautions should help prevent ' Controller Restart'. If the problem persists, contact support@unitronics or your local Unitronics representative.

M90 modem communication problems

If your M90 is transferring data via modem, you can begin troubleshooting by entering Information Mode. You can then check the status of relevant System Bits and Integers to help diagnose the problem.

To begin diagnosing the problem, check the error code contained in SI 70. Refer to the error code table in How the M90 works with a modem.

The tables below show the more common causes of modem communication problems.

Problem	SI 70 value	Possible Cause & Recommended Action
Modem fails to initialize	2: Modem Did Not Reply	M90-to-modem cable: Make sure that the cable is securely connected. Check the M90 modem connection and pin-out of the M90-to-modem adapter cables. Note that if you use cables comprising this pin-out, you must set the RS232 parameter Flow Control to N (none). Incompatible communication settings. Most modems automatically match the parameters of incoming data: baud rate, data bits, parity & stop bits. The M90's embedded modem settings are: 9600, 8 data bits, no parity, 1 stop bit. You may need to manually change your modem's communication settings to match these.
	0: No Error	SB 72 OFF: In order to work with a modem, you must select 'Use modem' in the M90 Modem Configuration box . This causes SB 72 Initialize Modem to turn ON when the M90 powers up. Note that if the M90 has also been configured to use SMS messaging, that the M90 will not be able to connect to a modem because the SMS feature overrides the modem. Check too, that SB 72 is not disabled in your program.
	6: Modem Report Error	Check the modem initialization commands. Refer to Configuring the M90 to use a modem.

Other problems:

Problem	Possible cause	Recommended action
Modem is busy	Modem is engaged	Check that the modem is free.
Connection established, but the M90 does not reply	Modem adapter cable	Check the M90 modem adapter cable pin-out, particularly that the DSR is connected to the RTS on the modem side.
M90 does not dial	Incorrect phone number	Check the M90's phone book. Refer to Configuring the M90 to use a modem.

PC modem communication problems

If your PC is unable to communicate with a remote M90 via modem, refer to the table below.

Note ♦ The PC-modem cable is not the same type of cable used to connect between the controller and the modem. Ensure that the cable used to connect the PC to the modem provides connection points for all of the modem's pins.
♦ Internal modems must be used in conjunction with the driver provided by the modem's manufacturer.

Problem	Possible Cause	Recommended Action
Modem fails to initialize	PC-to-modem cable	Make sure that the cable is securely connected, and that it is the original cable that was supplied with your modem.
	Incorrect initialization commands	To learn how to edit initialization commands, refer to Configuring my PC's modem.
	Incompatible communication settings	Most modems automatically match the parameters of incoming data: baud rate, data bits, parity & stop bits. The U90's modem communication settings are: 9600, 8 data bits, no parity, 1 stop bit. You may need to manually change your modem's communication settings to match these.
	Incorrect Com port	Assign the correct modem Com port. Refer to Configuring my PC's modem.
	Com port not enabled	Check that your PC communication port is enabled. This means checking your PC's BIOS/CMOS setup.
	Com port occupied	Close the application that is accessing the port.
Modem initializes, but no connection is established	Remote M90's modem failed to initialize	Refer to M90-to-Modem connections
	Remote M90 is not connected to modem	Check that the remote M90 is connected to the modem.
	M90-to-modem cable	Make sure that the cable is securely connected, and that it is the original cable that was supplied with your modem.
	Modem is not connected to telephone line	Check that the modem of both the local PC and the remote M90 is correctly connected to a functioning telephone line.
	Incorrect phone number	Check the PC modem Phone Book. Refer to Configuring my PC's modem.
	No power supply to modem	Check the power supply to both the PC's and the M90's modem.
Modems connect, but the M90's modem does not answer.	M90's modem did not initialize	Refer to M90 modem communication problems.
	M90-to-modem cable	Check the pin-out of the M90-to-modem adapter cables. Note that if you use cables comprising this pin-out, you

		must set the M90's RS232 parameter Flow Control to N (none).

CANbus network problems

Problem	Possible cause	Recommended Action
Failed communication	Baud rate settings	All M90's in the network must be set to the same CANbus baud rate. These may be edited in the M90 OPLC Advanced settings.
	Termination resistors	Check the M90 user guide for details.
	CANbus power supply	Check that the CANbus power supply is properly connected, and that the voltage is in the permissible range as described in the M90 User Guide.
	Incorrect ID number	You may not have assigned the correct unit ID number in your operand addresses (between 1-63). Check in the M90 OPLC settings.
PC cannot communicate with bridge	Incorrect ID number	When you communicate with the M90 unit that you are using as a bridge to the network, select Unit ID number 0, or select Stand-alone project in the M90 OPLC settings.
PC cannot communicate with network	Communication settings	If you are trying to communicate with an M90 network via a bridge, you must define your project as a Network project--however, U90 Ladder cannot automatically detect communication settings in a Network project. Make sure the current RS232 parameters in your project are the same as the parameters that are actually in the bridge.
	Incorrect baud rate	The bridge's RS232 port's baud rate cannot be set below 9600.

M90 does not turn on

When the M90 is turned on, the display screen is lit.

Note that the screen can display messages only after you download HMI displays to the M90. If you have not downloaded displays, check the screen by pressing the 'i' button for a few seconds to enter Information Mode. If no text appears on the screen, the M90 may not be receiving turned on.

If your M90 does not turn on

- Check that the power supply's voltage is in the permissible range in accordance to the technical specifications for your model.
- Check the M90's connections. The +24VDC must be connected to the + V terminal, and the ground connected to the 0V terminal.
- Make sure that the 24VDC output power supply is connected to a functioning AC power source.
- Check your fuses or circuit breakers. These must allow power flow.
- Make certain that the power is ON.

If you have checked all of the above, and the M90 does not turn on, contact your local distributor.

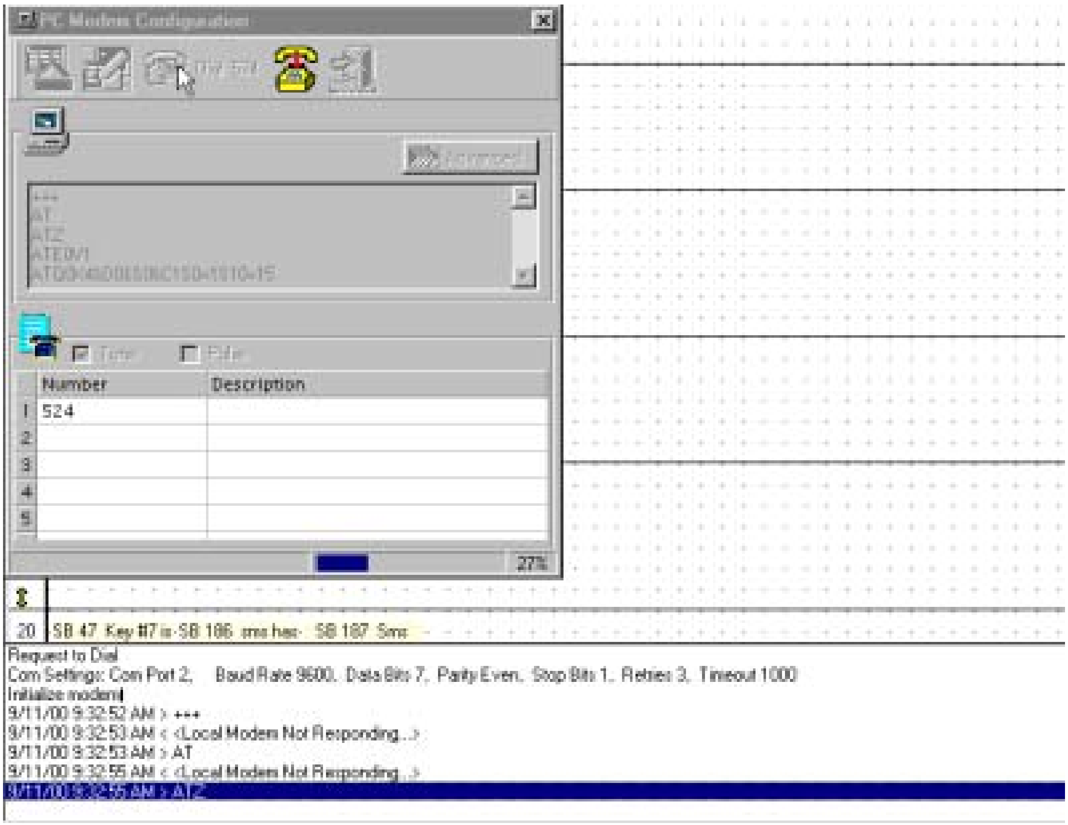
Power-up Modes

You can force the controller to enter Bootstrap or Stop mode by turning on the power supply while pressing specific keypad keys.

Mode	M90	M91
Bootstrap	< I > + < 7 >	< I > + < 7 >
Stop (O/S) Exit Stop Mode by entering Information Mode, and then selecting System> Reset.	< I >	< I >

Communication Log

When you dial a remote modem using U90 Ladder, a window opens up in the bottom of your screen. The log of events is quickly displayed in this window.



```
Request to Dial
Com Settings: Com Port 2,    Baud Rate 9600,  Data Bits 7,  Parity Even,  Stop Bits 1,  Retries 3,  Timeout 1000
Initialize modem
9/11/00 9:32:52 AM > +++
9/11/00 9:32:53 AM < <Local Modem Not Responding...>
9/11/00 9:32:53 AM > AT
9/11/00 9:32:55 AM < <Local Modem Not Responding...>
9/11/00 9:32:55 AM > ATZ
```

The log also appears during download and upload if there are communication problems.

This log is stored as a .txt file. You can view this log by navigating to Unitronics\U90_Ladder\U90Ldxxx and opening a file named ComLog.txt.

In this file, the most recent log of events appears **last**.

Index