

H-Designer

Reference Manual

English

H-Designer Reference Manual

Foreword

This manual is a description of H-Designer, the configuration tool used to create applications for the operator terminals in the H-series.

The manual assumes that the most recent versions of the system program (firmware) and configuration tool are used.

The operator terminal can be connected to many types of automation equipment, such as PLCs, servo and drives. In this document the expression “the controller” is used as a general term for the connected equipment.

© Beijer Electronics AB, MA00822, 2005-11

All examples in this manual are only intended to improve understanding of the functionality and handling of the equipment. Beijer Electronics AB cannot assume any liability if these examples are used in real applications. In view of the wide range of applications for this software, users must acquire sufficient knowledge themselves in order to ensure that it is correctly used in their specific application. Persons responsible for the application and the equipment must themselves ensure that each application is in compliance with all relevant requirements, standards and legislation in respect to configuration and safety.

Beijer Electronics AB will accept no liability for any damage incurred during the installation or use of this equipment.

Beijer Electronics AB prohibits all modification, changes or conversion of the equipment.

Contents

1	Installation.....	9
1.1	Installation Procedure	10
2	Instructions.....	13
2.1	H-Designer Programming Environment	13
2.2	File Menu	14
2.2.1	New	14
2.2.2	Open and Close	15
2.2.3	Save and Save as	15
2.2.4	Print	15
2.2.5	Upload Application and Download Application From	16
2.2.6	Upload Recipes and Download Recipes	16
2.2.7	Reconstruct Source	17
2.2.8	Exit	17
2.3	Edit Menu	18
2.3.1	Duplicate	18
2.3.2	Find/Replace Address.....	19
2.3.3	Decompose Shape.....	22
2.3.4	Align and Make Same Size	23
2.3.5	Nudge and Layer	24
2.3.6	Group and Ungroup	24
2.3.7	Object Attributes	25
2.3.8	State/Text Management.....	26
2.4	View Menu	30
2.4.1	Whole Screen and Whole Screen With I/O Labels.....	30
2.4.2	Language 1-5	30
2.4.3	Zoom In, Normal and Zoom Out	31
2.4.4	Toolbars.....	31
2.5	Screen Menu.....	32
2.5.1	New Screen.....	32
2.5.2	Screen Manager	32
2.5.3	Close Screen.....	36
2.5.4	Cut/Copy/Delete Current Screen and Paste Screen.....	36
2.5.5	OPEN Macro, CLOSE Macro and CYCLIC Macro.....	36
2.5.6	Screen Properties.....	37
2.6	Draw Menu	44
2.6.1	Geometric Shapes	44
2.7	Object Menu	56
2.7.1	Creating Objects.....	56
2.7.2	Specifying Object Properties	57
2.7.3	Buttons	61
2.7.4	Numeric Entry.....	73
2.7.5	Character Entry	76
2.7.6	List.....	77
2.7.7	Drop Down List	79
2.7.8	Indicators.....	80
2.7.9	Numeric Display.....	85
2.7.10	Character Display	87

2.7.11	Message Display Objects	88
2.7.12	Bar Graph	95
2.7.13	Trend Graph	98
2.7.14	XY Chart	100
2.7.15	Panel Meters.....	103
2.7.16	Pie Graph	105
2.7.17	Dynamic Graphics	106
2.7.18	Historical Display.....	118
2.7.19	Alarm Display	126
2.7.20	Sub Macro.....	131
2.8	Library Menu	132
2.8.1	Bitmap Library	132
2.8.2	Font Library	134
2.8.3	Save as Shape.....	135
2.8.4	Shape Library Manager.....	135
2.8.5	Text Pool.....	136
2.9	Application Menu	138
2.9.1	Workstation Setup	138
2.9.2	Tag Table.....	142
2.9.3	Alarm Setup	143
2.9.4	Slide-out Menu	144
2.9.5	System Message	145
2.9.6	Macros	147
2.9.7	Compile	148
2.9.8	Download Application and Download Firmware and Application...	149
2.9.9	File Protection.....	150
2.10	Tool Menu.....	150
2.10.1	Cross Reference	150
2.10.2	Off-line and On-line Simulation	153
2.10.3	View/Edit Recipes	154
2.11	Options Menu.....	155
2.11.1	Snap to Grid.....	155
2.11.2	Display Grid.....	155
2.11.3	Grid Attributes.....	155
2.11.4	Transmission Setup	155
2.11.5	Default Screen Background Style.....	156
2.11.6	Default Frame Styles	156
2.11.7	Default Text Styles	157
2.11.8	Numeric Keypad Setup	157
2.11.9	Editing Options	157
2.12	Window Menu.....	158
2.13	Help Menu.....	159
3	Recipes	161
3.1	Example	161
3.2	Recipe Operation Steps	162
3.3	Recipe Controlled by Controller.....	165
3.4	Recipe Controlled by Operator Terminal	168

4	Control and Status Block	171
4.1	Control Block	172
4.1.1	Screen Number Register.....	172
4.1.2	Command Flag Register.....	174
4.1.3	Logging Buffer Control Registers: LBCRs.....	176
4.1.4	RCPNO Number Register: RNR.....	179
4.1.5	General User Area Register.....	179
4.1.6	Determine the Control Block Size	179
4.2	Status Block	180
4.2.1	Screen Status Register	180
4.2.2	General Status Register	180
4.2.3	Logging Buffer Status Registers (LBSRs)	182
4.2.4	RCPNO Image Register.....	182
4.3	Recipe Register Block.....	183
4.3.1	Recipe Register Number - Enhanced Operator Terminals.....	183
4.3.2	Addressing Recipe Data - Enhanced Operator Terminals.....	184
4.4	Time Block	185
4.4.1	The Operator Terminal Writes to the Controller.....	185
4.4.2	The Controller Writes to the Operator Terminal.....	186
4.5	Read Cycle.....	186
5	Multi-Link: Normal Connection Port.....	187
5.1	Communication Parameters.....	188
5.2	Communication Efficiency	190
5.3	Important Notes	191
6	Ethernet Communication.....	193
6.1	Connection	193
6.2	IP Address Setup.....	193
6.3	Application Upload/Download over Ethernet.....	194
6.4	Communication with Ethernet-enabled controllers.....	196
6.5	Multi-Link - One Master and Multiple Slaves.....	197
6.6	Cross-Link over Ethernet (Data Sharing)	200
7	Multi-Channel Communication.....	205
7.1	Connection	205
7.2	Connection Setup	206
7.3	Read/Write Address Setup	210

8	Macros.....	211
8.1	Macro Function	211
8.2	Macro Classifications.....	211
8.2.1	Application Macros	211
8.2.2	Screen Macros	211
8.2.3	ON/OFF Macros	212
8.2.4	Sub-Macros	212
8.3	Macro Commands.....	213
8.3.1	Arithmetic	215
8.3.2	Logical.....	215
8.3.3	Data transfer.....	216
8.3.4	Comparison	216
8.3.5	Flow Control.....	220
8.3.6	Data Conversion	221
8.3.7	Bit Setting	222
8.3.8	Others	222
8.4	Cautions.....	226
8.5	Internal Memory	226
9	Communication between Operator Terminal and Controller.....	227
9.1	Allen Bradley Micrologic 1000/1500.....	228
9.2	Allen Bradley PLC-5	230
9.3	Allen Bradley SLC-503/504	233
9.4	Allen Bradley IQ Master Servo Controller.....	236
9.5	ABB Comli (Slave Mode)	237
9.6	Computer (as Master/Slave/V2/Null) /Modbus Master.....	238
9.7	Delta DVP	239
9.8	Ero TFS/THS/LFS.....	240
9.9	Facon FB.....	241
9.10	Festo FPC	243
9.11	Fuji NB	244
9.12	GE Series 90 CCM	246
9.13	GE-Fanuc 90-SNP	247
9.14	Hitachi EC.....	249
9.15	Hitachi H/EH1	250
9.16	Hust CNC Controller	252
9.17	Idec Micro-3	253
9.18	Jetter Nano_B	255
9.19	Jetter Delta.....	256
9.20	Klockner Moeller PS	257
9.21	Koyo SA/TI 325/330	258
9.22	Koyo Direct DL/Koyo SU Series/TI435.....	260

9.23	LG Glofa GM6.....	262
9.24	LG K10/60H/200H	263
9.25	LG K200S	265
9.26	LG K300S	266
9.27	LG Master-K10S/K30S/60S/100S	267
9.28	Matsushita FP	268
9.29	Mirle DX.....	270
9.30	Mitsubishi FX.....	272
9.31	Mitsubishi A	274
9.32	Mitsubishi QnA.....	277
9.33	Modbus Slave	279
9.34	Modicon PC984 or Modbus (ASCII) or TSX Quantum	280
9.35	Omron C.....	282
9.36	Omron CS1	284
9.37	Omron CV	285
9.38	Parker 6K.....	286
9.39	Side Mida 20/20D	288
9.40	Siemens Simatic S5	290
9.41	Siemens Simatic S5 3964R	292
9.42	Siemens Simatic S7-200 PPI.....	293
9.43	Siemens Simatic S7-300 CP340.....	295
9.44	Siemens Simatic S7-300 via MPI port.....	296
9.45	Siemens Simatic S7-300 MPI-Cable	298
9.46	Siemens Simatic S7-300 HMI-Cable	300
9.47	Taian TP01	302
9.48	Taian TP02	303
9.49	Taian N2	305
9.50	Telemecanique TSX Micro	306
9.51	Toshiba M20/M40	308
9.52	Toshiba T1/T2	309
9.53	Unidriver UD70.....	311
9.54	Vigor M.....	312
9.55	Yokogawa FA-M3.....	314
10	Appendix A - H-Designer Features and Operator Terminal Models	317

1 Installation

The basic H-Designer system requirements are as follows:

- PC - CPU 80586 or higher;
- Memory - 64 MB RAM or more;
- Hardware - 60 MB or more available hard drive space;
- Display - VGA or SVGA. Microsoft Windows with 256 colors or higher, and a resolution of 800×600 or higher.

Since all the programs in the H-Designer suite have been compressed, one needs to uncompress and install the software before using it.

The H-Designer software can be run under the following Windows operating systems:

- Windows 95
- Windows 98
- Windows ME
- Windows 2000
- Windows XP

The software is available from the following website or your local distributor.

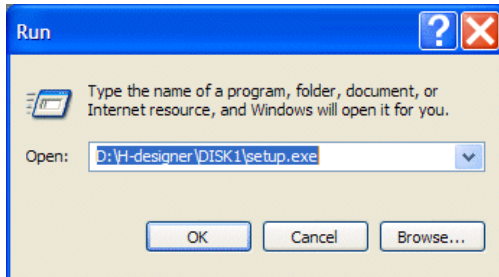
www.beijerelectronics.com

Note also that not all features provided by H-Designer are available for every operator terminal model.

For complete details of these H-Designer features and the applicable models, please refer to [Appendix A - H-Designer Features and Operator Terminal Models](#).

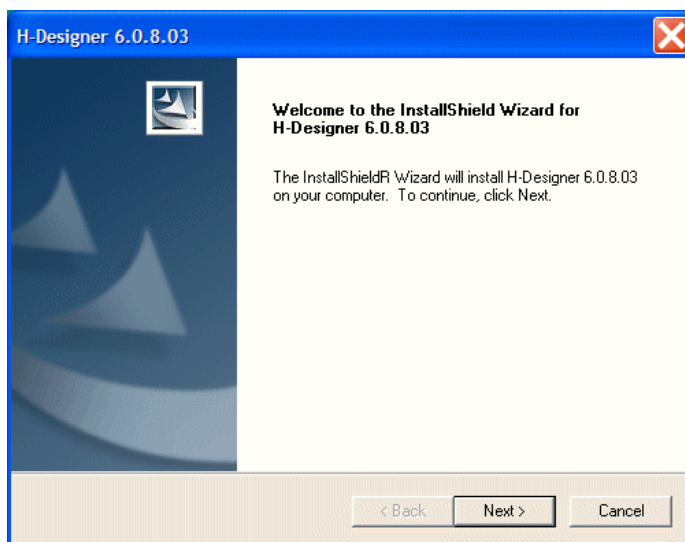
1.1 Installation Procedure

1. Start your computer in the Windows environment.
2. Click **Start** and select **Run**. When the Run dialog box appears on the screen, select **Browse** to locate the installation program **Setup.exe**.



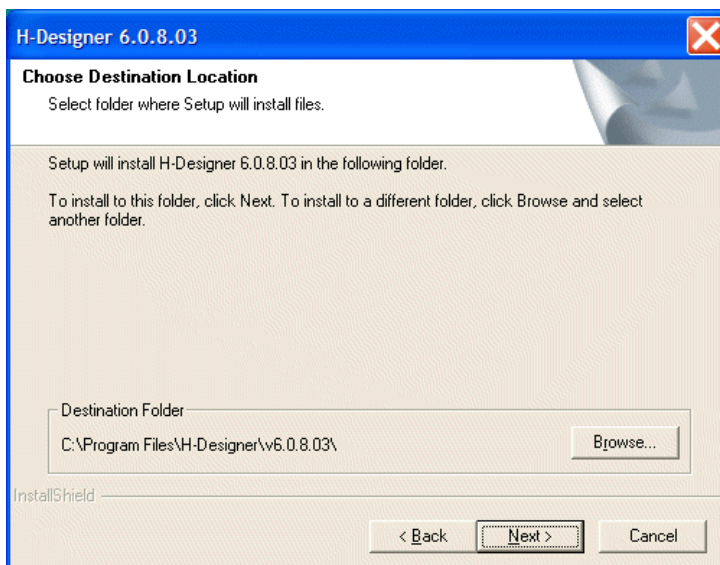
Running the H-Designer installation program "Setup.exe" in Windows

3. Click **OK** to start the installation.



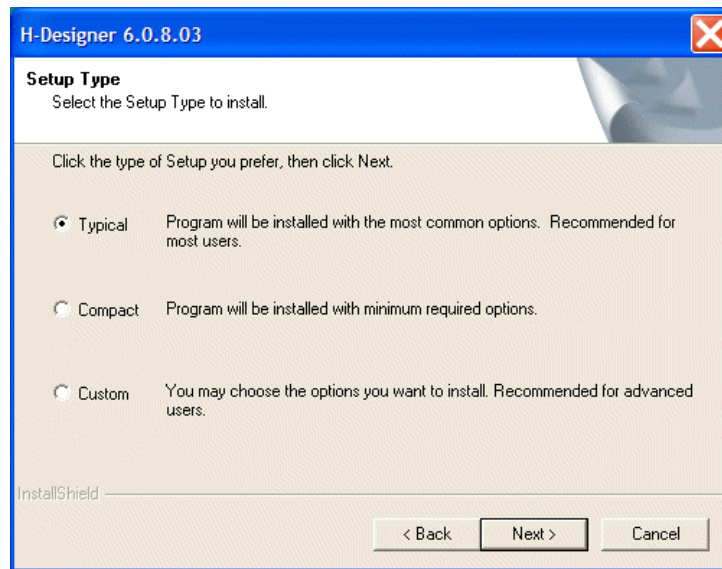
H-Designer is preparing to install

4. Follow the instructions and specify the hard drive and directory where H-Designer is to be installed.



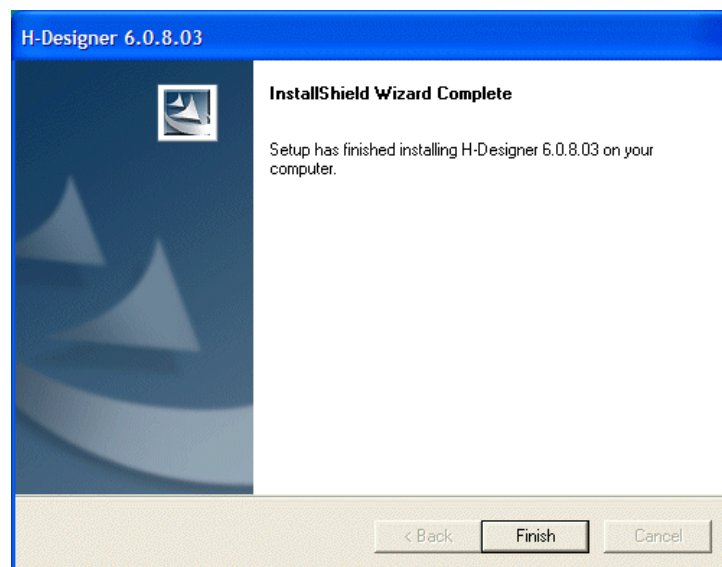
The destination hard drive and directory

- Click **Next** to select the type of setup. **Typical** is recommended for most users. **Compact** installs the program with basic options. **Custom** allows users to individually select the options to install and this is recommended for advanced users.



Selecting the type of setup

- Click **Next** to begin the installation. If the **Typical** option is selected, the following dialog box will appear on the screen.



H-Designer installation

- After installation, the system will create the H-Designer icon automatically.
- Once the installation is complete, the H-Designer software can be found in the specified directory. To launch the H-Designer program, simply click the H-Designer icon.

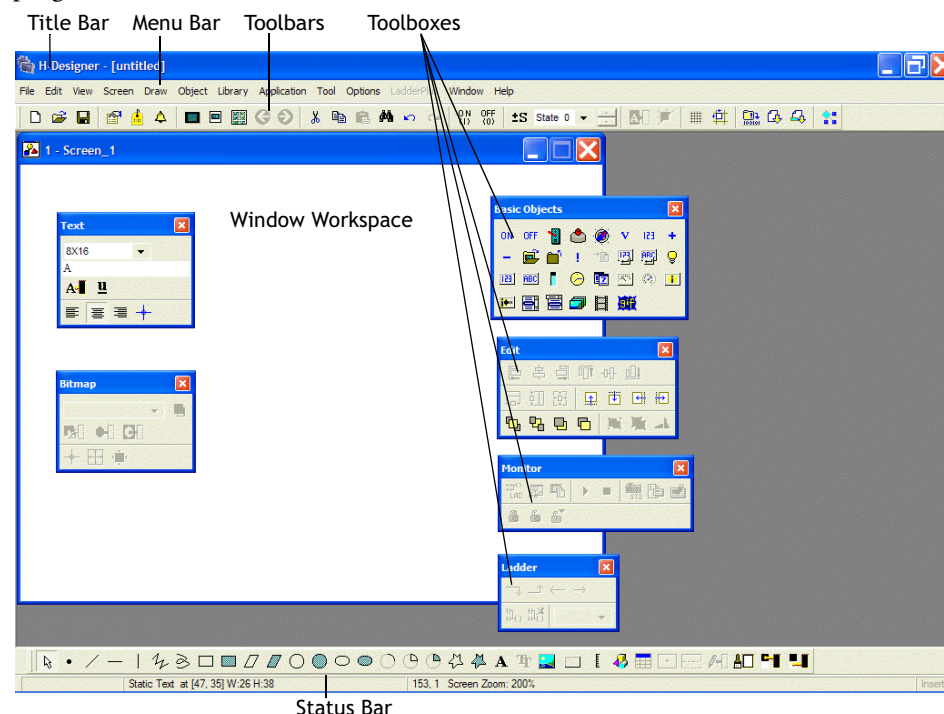
2 Instructions

Microsoft Windows™ is undeniably the predominant PC operating environment these days. H-Designer is designed to make full use of the Windows environment, using a “What You See is What You Get” Approach. Users can immediately see the objects they create on a PC screen with their specified attributes, such as font size, color, object location, picture, scale, frame, and so on. What the user sees on the PC screen will be the same as what is displayed on a operator terminal.

Furthermore, H-Designer utilizes the principles of object-oriented design to implement drag-and-drop editing. Users can use a mouse to conveniently drag objects to another location or change their shape and size as they wish.

2.1 H-Designer Programming Environment

The figure below illustrates the main menu bars and tool boxes in the H-Designer program environment.



H-Designer program environment overview

Title Bar

The Title Bar shows the name of the window and the directory where the current application is found.

For example: C:\Program Files\H-Designer\Project Files\Project1.V6F.

If an application file has not been saved, “Untitled” will be displayed in the Title Bar.

Menu Bar

There are 13 menus in the Menu Bar; they are File, Edit, View, Screen, Draw, Object, Library, Application, Tool, Options, LadderPlus, Window and Help.

Tool Bar

Users can easily create an H-Designer project by clicking the icons on the Tool Bar. This feature also helps new users to familiarize themselves with the software quickly and easily.

Window Workspace

This is the area where the user designs a screen. The objects or data created in this area will actually be displayed on the operator terminal screen.

In the following sections, we will talk about the function, application, and usage of each of the commands and menus in much more detail.

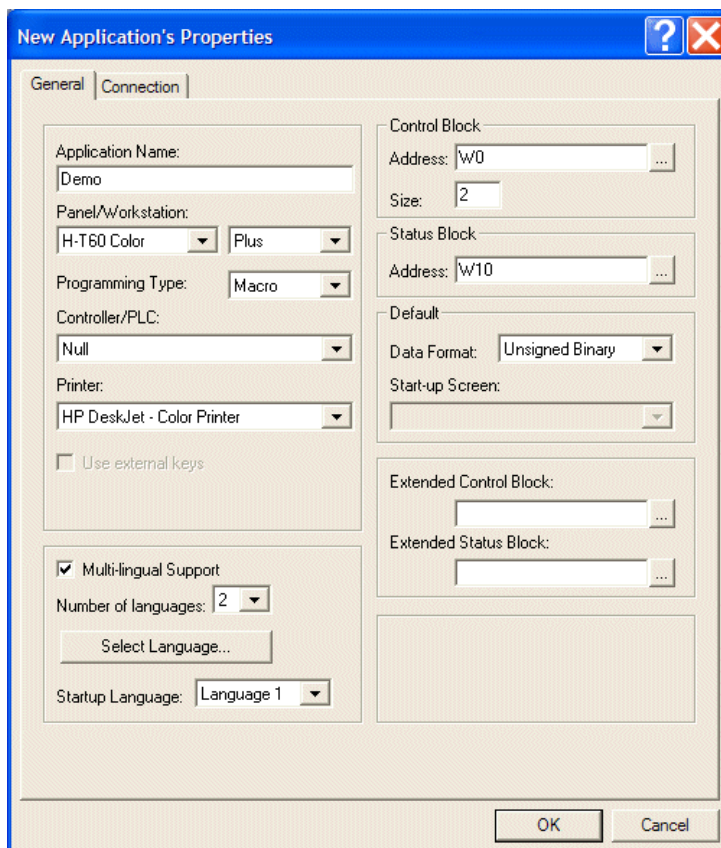
2.2 File Menu

The main purpose of the File menu is to manage the files created by H-Designer.

2.2.1 New

This command allows users to create a new application and specify its properties, including **Application Name**, **Panel/Terminal**, **Controller/PLC**, **Printer**, **Multilingual Support**, **Control Block**, **Status Block**, and so on.

To create a new application, select **File/New**. The **New Application's Properties** dialog box appears on the screen.



The New Application's Properties dialog box

The following are the basic properties a user needs to set up for a new application:

- In the **Application Name** box, enter the application name.
- In the **Panel/Terminal** list, select a operator terminal model.
- In the **Controller/PLC** list, select the type of controller which the operator terminal will communicate with.

Please see the sections [Application Menu](#) and [Connection Tab](#) for more details.

2.2.2 Open and Close

Open opens an existing application.

Close closes the application.

2.2.3 Save and Save as

Selecting **Save** saves an existing application, replacing the previous copy with the new copy.

Selecting **Save As** saves a new or existing application with a new name.

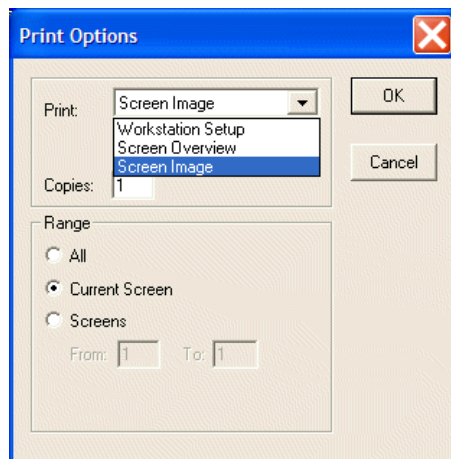
2.2.4 Print

An application file can be printed for the planning, management or storage.

Note:

The **Print** option is not available for all operator terminal models: please refer to [Appendix A - H-Designer Features and Operator Terminal Models](#) for complete details.

Select **File/Print** and the **Print Options** dialog box appears on the screen.



The Print Options dialog box

There are three options available: **Workstation Setup**, **Screen Overview**, and **Screen Image**.

Workstation Setup prints the operator terminal data, such as the controller type, configuration setup, and the logging buffer details.

Application Name:

Page 1

11/01/2005 14:58

Workstation Type:

PLC Type:

Printer Type:None

Default Startup Screen:(None)

Default Data Format:Unsigned Binary

Control Block Address:W0 Size:2

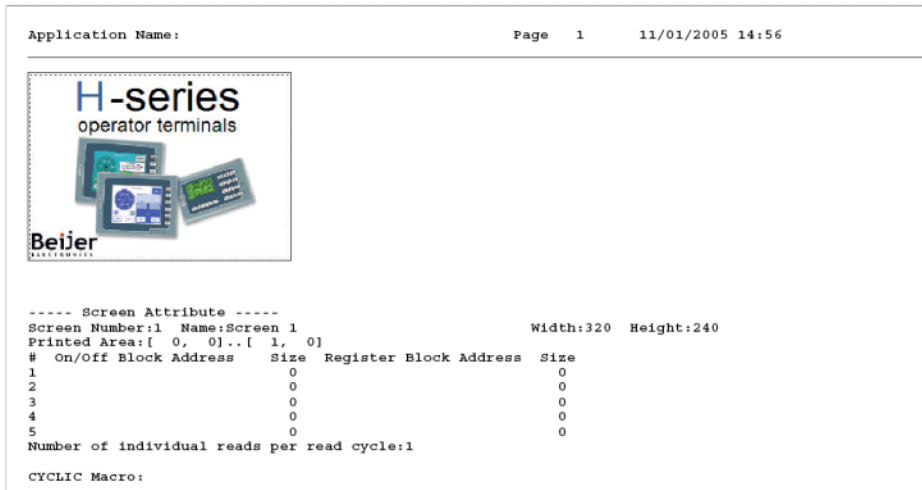
Status Block Address:W10

Logging Buffers

#	Source Address	Record Size	Total	Stamp Time	Auto Date	Triggered Stop	By	Time Interval
1	W30	2	200	No	No	No	Timer	2
2		0	0	No	No	No	PLC	0
3		0	0	No	No	No	PLC	0
4		0	0	No	No	No	PLC	0
5		0	0	No	No	No	PLC	0
6		0	0	No	No	No	PLC	0
7		0	0	No	No	No	PLC	0
8		0	0	No	No	No	PLC	0
9		0	0	No	No	No	PLC	0
10		0	0	No	No	No	PLC	0
11		0	0	No	No	No	PLC	0
12		0	0	No	No	No	PLC	0

An example of the Workstation Setup print option

Screen Overview prints a screen image with the controller location of each object.



An example of the Screen Overview print option

Screen Image prints a screen image without the controller locations.



An example of the Screen Image print option

Other options:

Copies: Specifies the number of copies to print.

Range: Only available for the Screen Overview and Screen Image options.

2.2.5 Upload Application and Download Application From

Upload Application is used to upload an application from a operator terminal to a PC and save the file as *.AF6.

Download Application From is used to download a program from a PC to a operator terminal. The file format is *.AF6.

2.2.6 Upload Recipes and Download Recipes

Upload Recipes is used to upload recipes from the operator terminal to a PC. The file will be saved as *.RCP.

Download Recipes is used to download recipes from a PC to a operator terminal. The file format used is *.RCP.

Note:

These functions are not available for all operator terminal models: please refer to [Appendix A - H-Designer Features and Operator Terminal Models](#) for complete details.

2.2.7 Reconstruct Source

This function allows users to directly reconstruct the source file of an application by uploading the application file from an operator terminal to a PC.

Reconstruct Source is used to reconstruct an uploaded application file from *.AA6 to *.V6F. The application will be displayed on a PC and the source file (*.V6F) can be saved for the future editing and application.

In H-Designer, an application downloaded to an operator terminal is “rebuildable”; **Reconstruct Source** enables a user to directly reconstruct a source file of an uploaded application on an operator terminal from *.AA6 to *.V6F.

Note:

This function is not available for all operator terminal models: please refer to [Appendix A - H-Designer Features and Operator Terminal Models](#) for complete details.

Steps to reconstruct a source file from *.AA6 to *V6F:

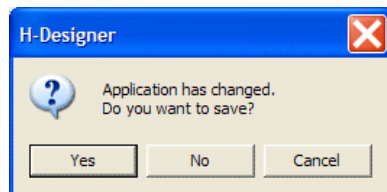
(Assuming that an application has been uploaded on the operator terminal.)

1. On the operator terminal, select **Upload Application**. In H-Designer, select **File/Upload Application**. The operator terminal will upload the application to the PC and the file will be saved as *.AF6.
2. Next, select **File/Reconstruct Source** and open the application file (*.C64 or *.AA6). The application should appear on the PC and the user can save the source file as *.V6F for future use.

2.2.8 Exit

Exit is used to close and exit H-Designer.

If any changes have been made, the following dialog box will appear on the screen, asking the user to save the changes or exit.



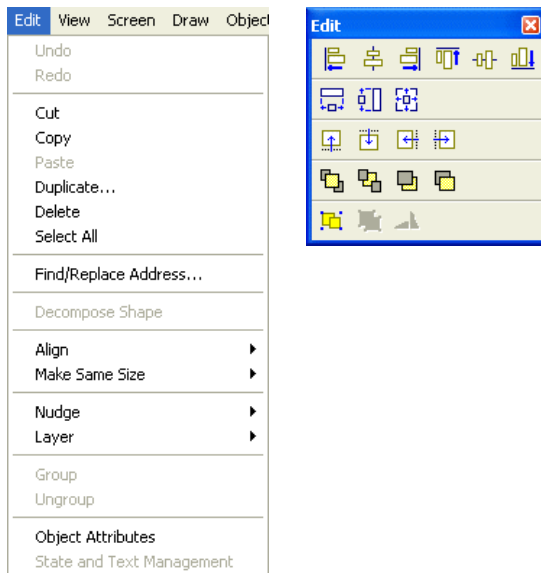
The Exit dialog box

2.3 Edit Menu

The primary purposes of the **Edit** menu include the common editing functions for H-Designer screen and objects. If a user does not open an image, all of the light-colored items in the **Edit** menu are unavailable.

The user can click the icons on the tool bar to speed edit or quickly gain use of the elements on the list by right-clicking with the mouse.

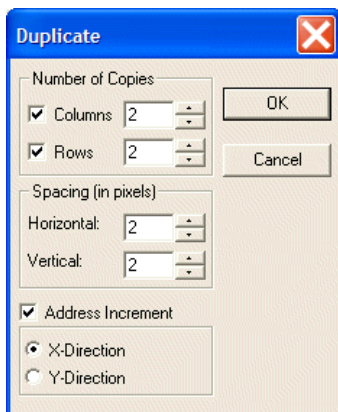
The following sections will explain the functions that are more specific to the H-Designer software.



*The **Edit** menu and the **Edit** toolbar*

2.3.1 Duplicate

Allows a user to make multiple copies of an object and simultaneously increase the corresponding addresses incrementally.



*The **Duplicate** dialog box*

Number of Copies

Columns: Specifies number of columns to duplicate.

Rows: Specifies the number of rows to duplicate.

Spacing

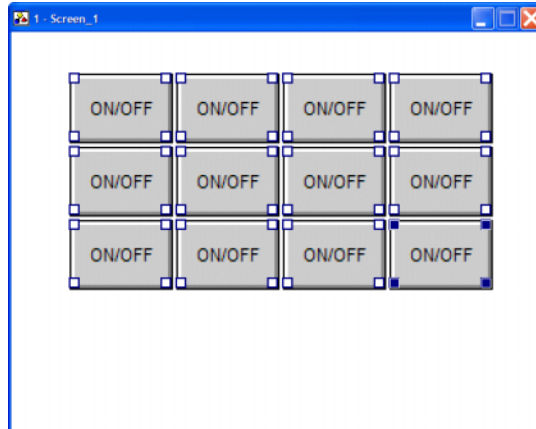
Horizontal: Specifies how many pixels to space duplicate objects horizontally.

Vertical: Specifies how many pixels to space duplicate objects vertically.

Address Increment

X-Direction: The address of the same dynamic objects in-creases from left to right.

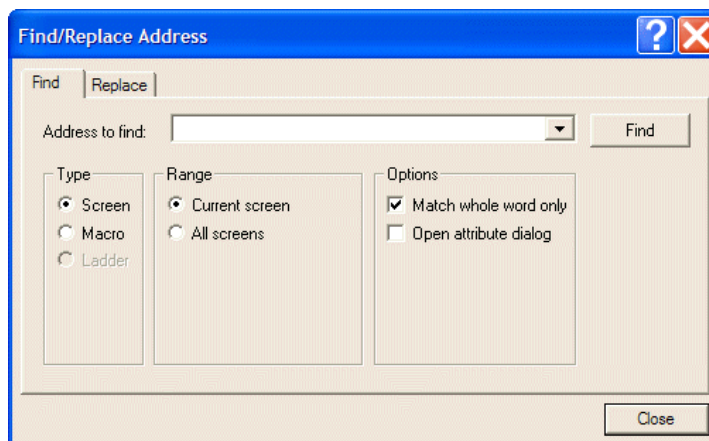
Y-Direction: The address of the same dynamic objects increases from top to bottom.



Duplicate

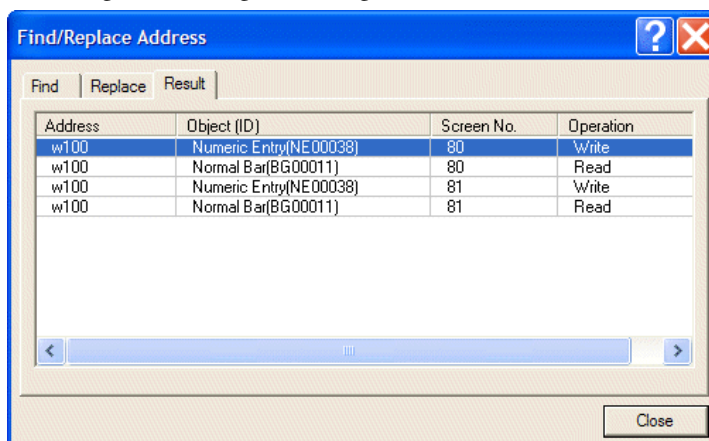
2.3.2 Find/Replace Address

The main function is to find or edit addresses in a created project and to replace the found addresses.



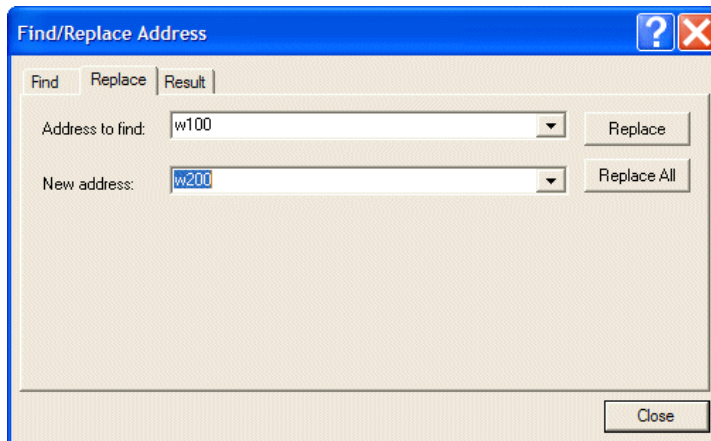
The Find/Replace Address dialog box

Enter the address in the blank space or select from the drop-down list directly (e.g. W100); then click the **Find** button to search. The **Result** tab displays a detailed list according to the designated range and address.



The Result tab

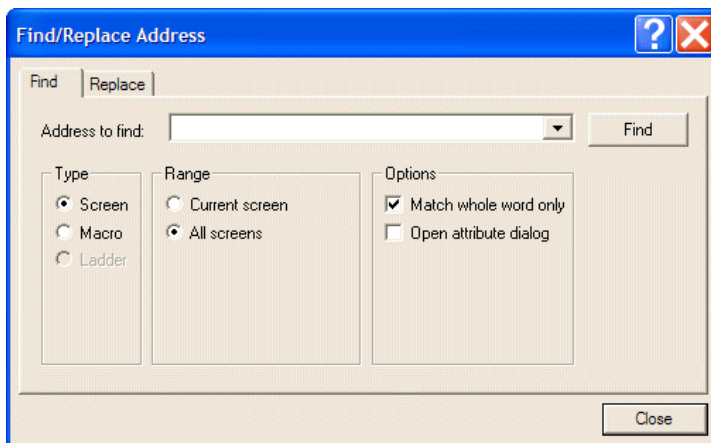
Use the **Replace** button to replace the address with a new one. The function of the **Replace** button is to replace the objects' addresses one by one; use **Replace All** to replace all of the objects' address at one time.



The Replace tab

Example 1: Find Screen Address

Select **Type/Screen** to find the screen address.



Find Screen address

Range:

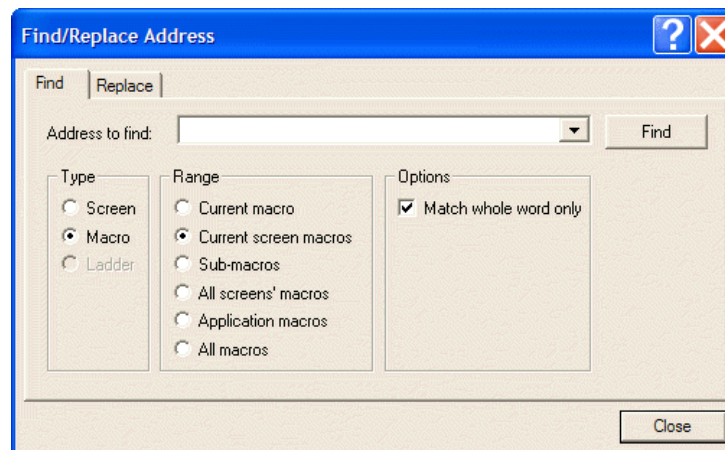
- **Current screen:** Only finds in the current open screen.
- **All screens:** Finds all the screens in the application program.

Option:

- **Match whole word only:** The findings match the entry address entirely. If this option is not selected, the initial findings match the entry address including partial and entire matches.
- **Open attribute dialog:** To display an object's attributes, double-click it in the list on the Result tab. If this option is not selected, the object attributes dialog box is unavailable.

Example 2: Find Macro Address

Select **Type/Macro** to find the macro address.



Find Macro address

Range

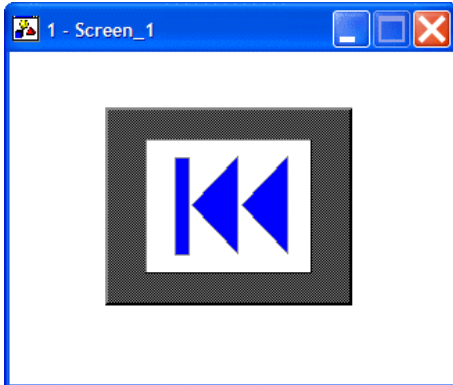
- **Current macro:** Finds the current macro.
- **Current screen macros:** Finds the current image/screen macros.
- **Sub-macros:** Finds all sub-macros.
- **All screens' macros:** Finds all image/screen macros.
- **Application macros:** Finds three types of macros in the **Application** menu.
- **All macros:** Finds all macros.

Option

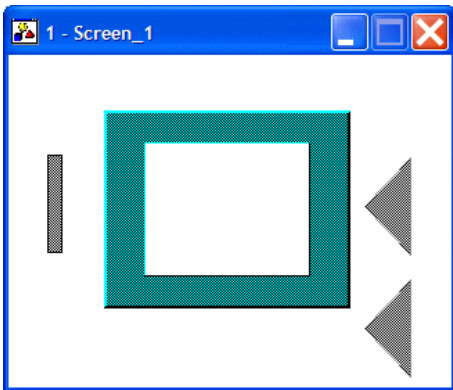
- **Match whole word only:** The findings match the entry address entirely. If this option is not selected, the initial findings match the entry address including partial and entire matches.

2.3.3 Decompose Shape

This function is primarily used to decompose the graphics created using **Shape** in the **Draw** command. Each decomposed graphic can be modified and edited.



*Selecting **Graph** in **Draw/Shape** to edit*

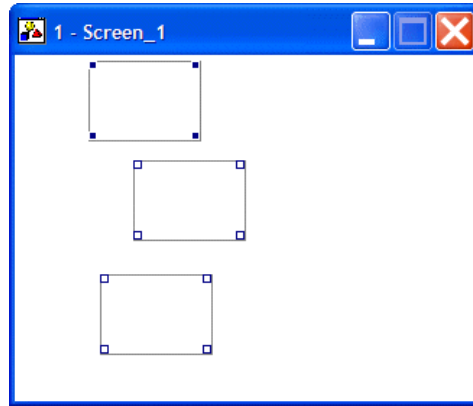


*Selecting **Edit/Decompose** to modify each decomposed object*

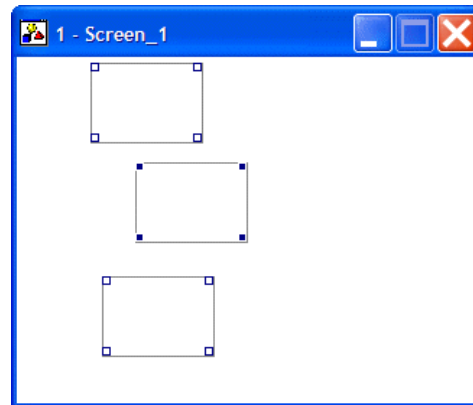
2.3.4 Align and Make Same Size

These two functions are primarily used to align created objects or to make them the same size, for example indicators, moving signs, message displays and lines. The following are the steps used for these two commands:

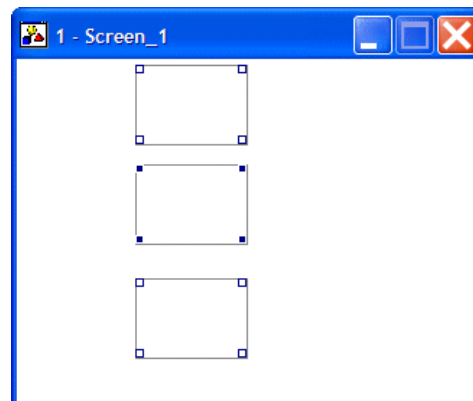
1. Select **Shift** and click on all objects which are to be aligned or made the same size.



2. Click on the master object; the four corners of this master object should then be shaded.



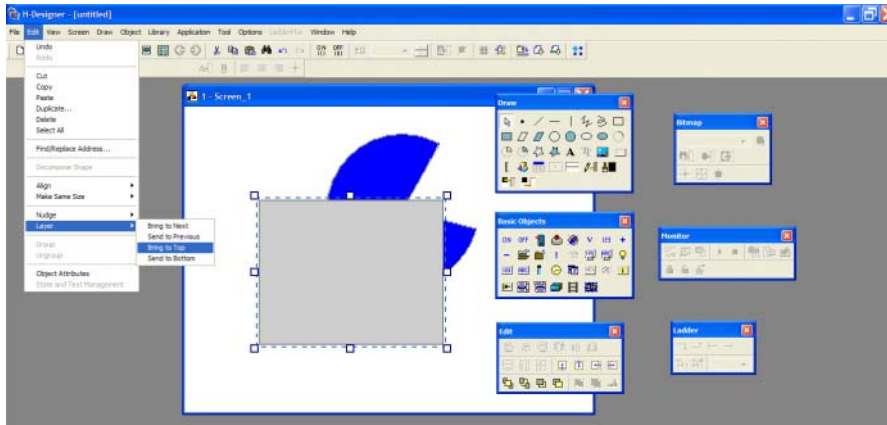
3. Select the command **Align** or **Make Same Size** to make the other objects aligned with or made same size as the master object.



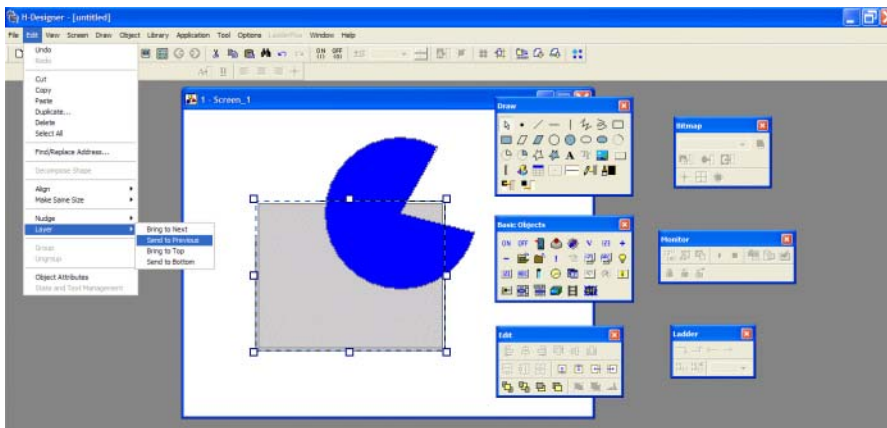
2.3.5 Nudge and Layer

Nudge: Choose objects to slightly shift location and adjust towards the specified direction.

Layer: If there are more than two objects, users can shift the objects' layer positions up and down.



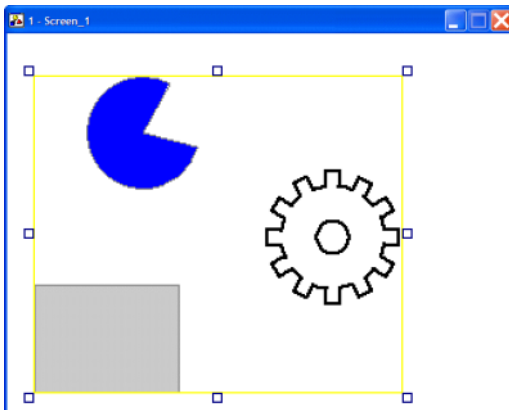
The Layer/Bring to Next command sends the rectangle to the top.



The Layer/Send to Previous sends the rectangle to the bottom.

2.3.6 Group and Ungroup

If there are more than two graphics or objects on the screen to be edited; frame (by using **Shift + left-click**) all objects to be grouped and then select **Group**. All of the framed objects will be moved to the appointed position as a single unit.



*Select **Edit/Group** to group the selected objects together.*

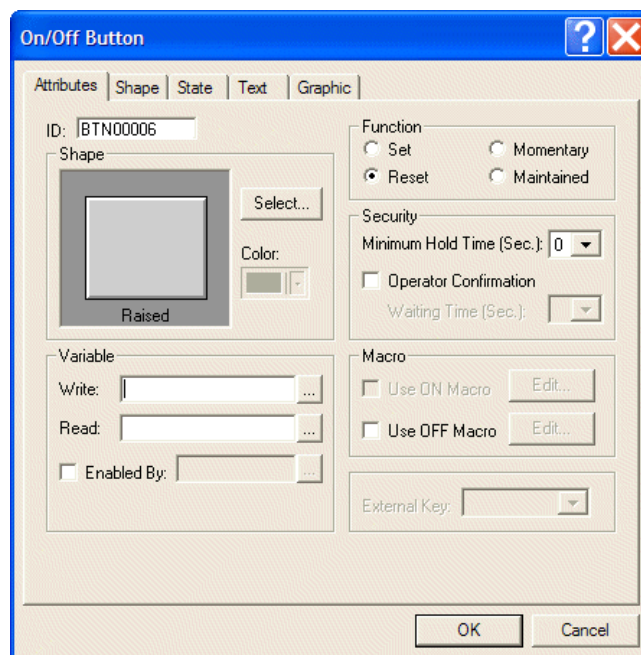
Ungroup is used to ungroup the selected group of objects.

2.3.7 Object Attributes

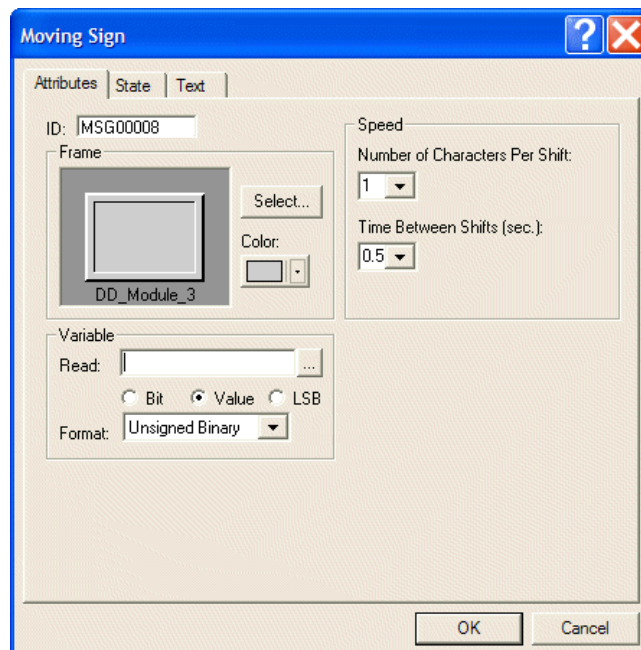
You can edit the content of the objects or modify the data location and formats associated with the controller.

Note that different objects (such as the push button, indicator, moving sign and message display) have their own object attributes.

Click on the object and then select **Edit/Object Attributes** and the dialog box will then be displayed on the screen. You can also double-click directly on the objects. For properties not explained in this section, please refer to section [Object Menu](#).



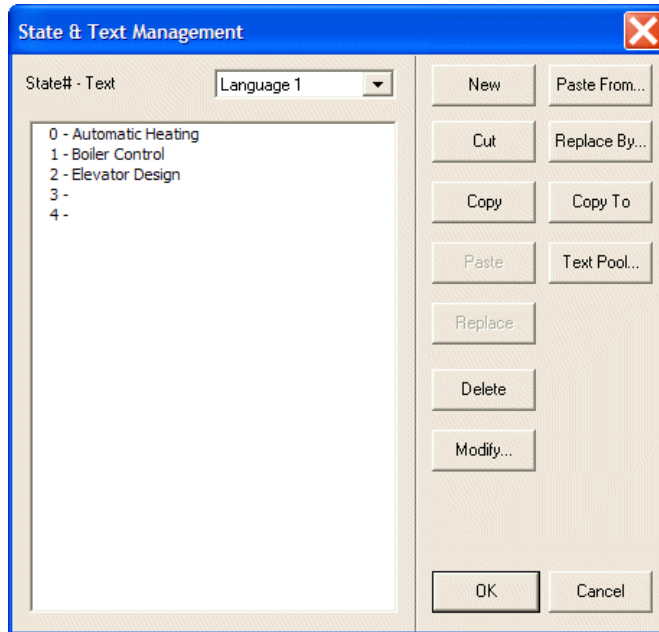
The ON/Off Button object attributes dialog box



The Moving Sign object attributes dialog box

2.3.8 State/Text Management

The main uses of **State/Text Management** include editing the text, colors, and type-faces used the object created. This function also provides users with easier operation, enabling such functions as copy, modify, line feed, and so on.

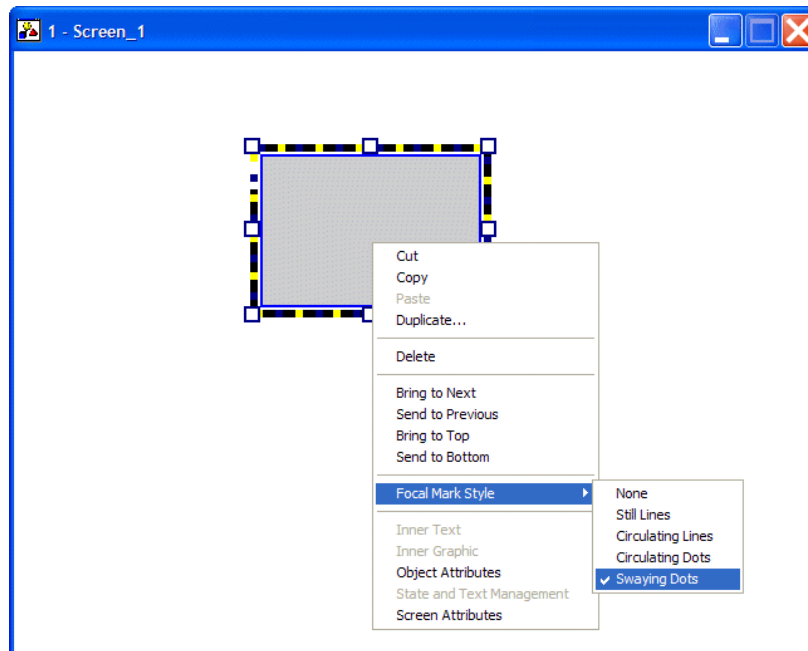


The State & Text Management dialog box

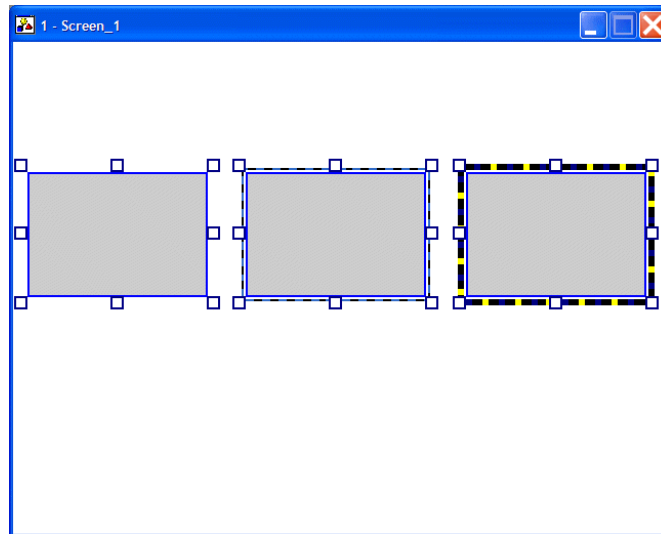
Focal Mark Style

There are five focal mark styles to choose from: **None**, **Still Lines**, **Circulating Lines**, **Circulating Dots** and **Swaying Dots**.

This function is not limited by any application images or files. It modifies the H-Designer editing environment.



The Focal Mark Style list

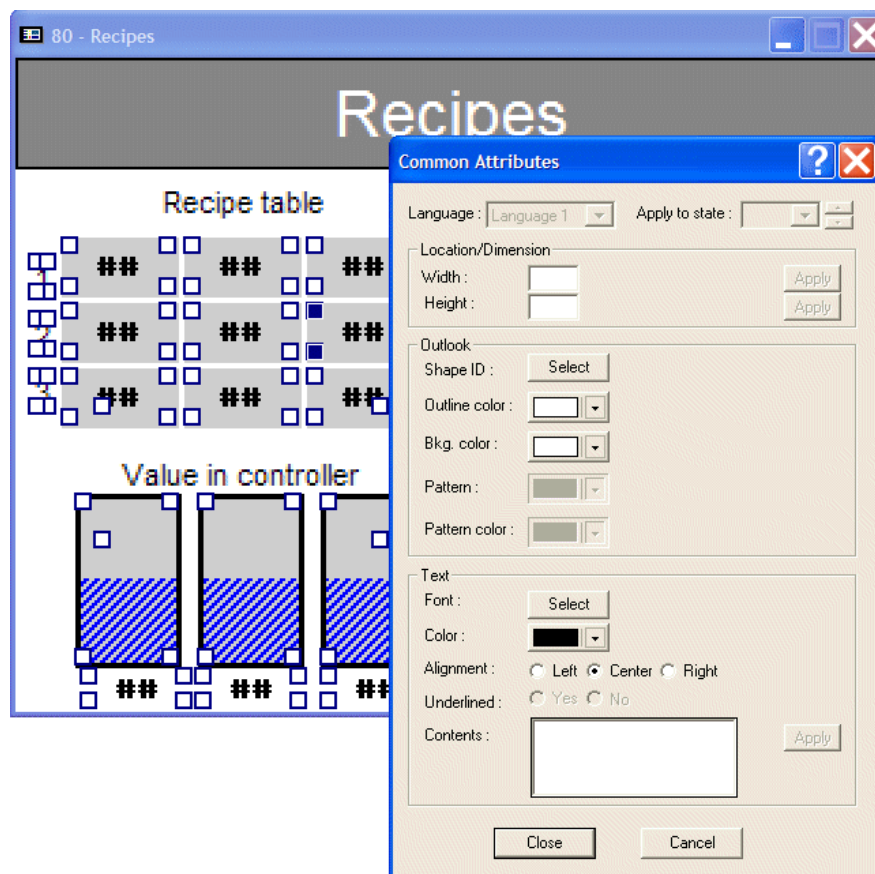


Objects with different focal marks

Common Attributes

This function makes it easier to edit objects with common attributes simultaneously. The user can modify attributes of all objects which are on the screen at one time.

First, select the objects to be edited at one time. The **Common Attributes** dialog box will be displayed on the screen when you double-click one of the objects.



Opening the Common Attributes dialog box

The user can edit the common attributes of selected objects, such as location/dimension, appearance and text.

Language

Specifies language to be used for the selected objects.

Apply to State

Specifies the state of the selected objects. The user can edit different states in this list.

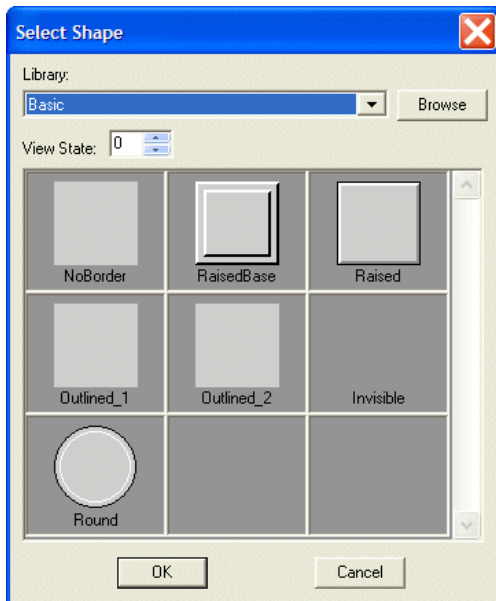
Location/Dimension

Specifies the width and height of the objects. Click the **Apply** button to modify.

Outlook

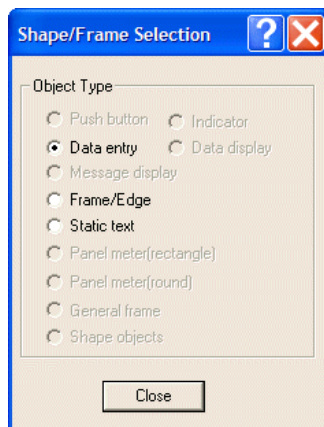
Specifies the apperance of selected objects.

Shape ID: Specifies the shape of different objects. When clicking the **Select** button, a dialog box with a selection of shapes will appear on the screen.



Selection of shape of the object

When the selection has been made, a dialog box with a selection of functions for the selected shape will appear on the screen as.



Selection of function of the object

The user can then select the next object to create.

Outline Color: Specifies the outline color of all selected objects.

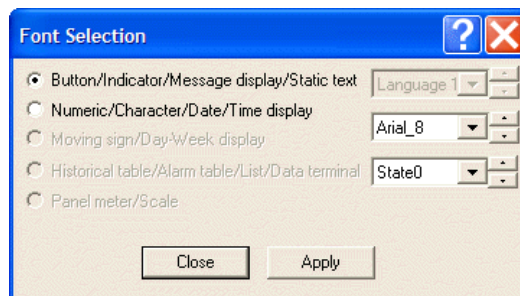
Background Color: Specifies the background color of the selected objects.

Pattern: Specifies the pattern of all selected objects which can be set.

Pattern Color: Specifies the pattern color of all selected objects which can be set.

Text

Font: Specifies the text font for different objects. The user can specify the text font for different types of objects. When using the **Selection** button, the **Font Selection** dialog box will be displayed.



Specifies the text font for objects

Color: Specifies the text color for all selected objects.

Alignment: Specifies the text alignment type for all selected objects.

Underlined: Specifies whether the text of all selected objects is to be underlined.

Contents: Specifies the text contents for all selected objects.

Remember to click the **Apply** button to apply changes.

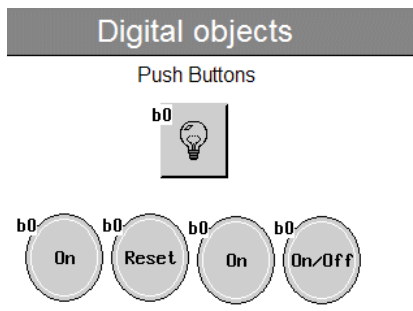
2.4 View Menu

The main functions of the **View** menu include the managing functions of whole screen, whole screen with I/O labels, five language selections, zoom in/out, and eight kinds of toolbars.

2.4.1 Whole Screen and Whole Screen With I/O Labels

Whole Screen: Full-screen view showing all edited objects. The user can return to the previous view by clicking on the screen.

Whole Screen With I/O Labels: To view the whole screen with dynamic objects with write/read addresses displayed on the top left side. The user can return to the previous view by clicking on the screen.

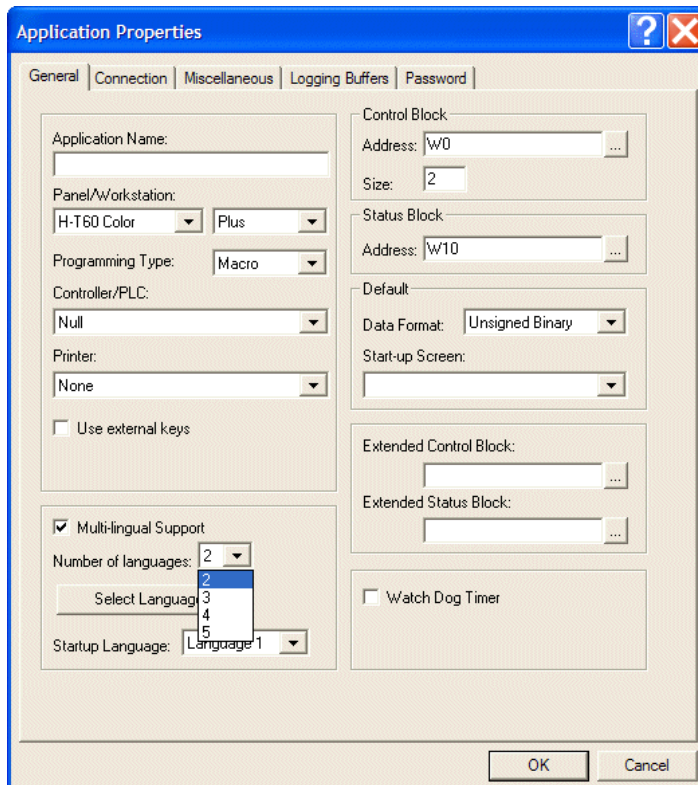


Displaying Whole Screen With I/O Labels

2.4.2 Language 1-5

There are five languages available to the user. Please follow the steps below to set up multiple languages:

1. Select **Application/Workstation Setup**. The **Application Properties** dialog box will appear on the screen.



The Application Properties dialog box

- On the **General** tab, check the **Multi-lingual Support** box to begin setup.

Number of Languages: Specify the number of languages required for the project. A maximum of 5 languages can be selected.

Select Language: Specifies the languages to use.

Startup Language: Specifies the language to display at startup. The user can select the language directly from the **View** menu if the screen language is to be switched later.

2.4.3 Zoom In, Normal and Zoom Out

Zoom In: To enlarge the image size by a set percentage of 150%, 175%, 200%, or 250%.

Normal: To adjust the image size to the actual size of the screen.

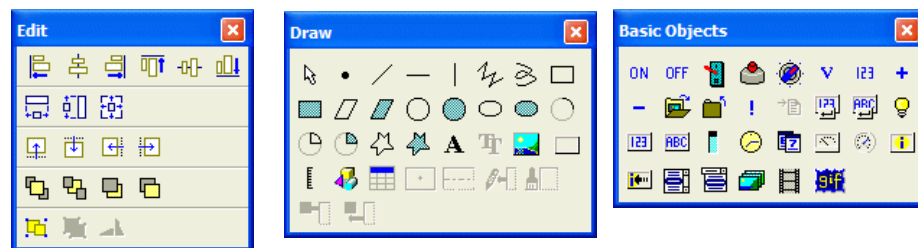
Zoom Out: To decrease the image size by a set percentage of 75%, 50%, or 25%.

2.4.4 Toolbars

There are eight different toolbars in the **View** menu. The user can decide to display the toolbars for quick-operation or not. The toolbars are shown as below:



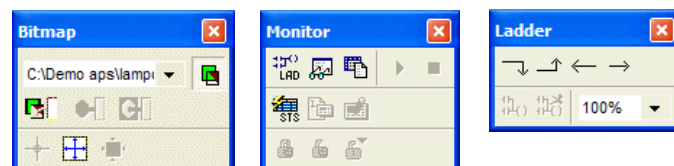
The Standard toolbar



The Edit toolbar, the Draw toolbar and the Basic Objects toolbar



The Text toolbar



The Bitmap toolbar, the Monitor toolbar and the Ladder toolbar

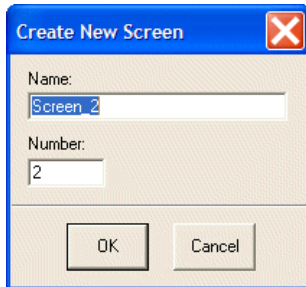
2.5 Screen Menu

The main purposes of the **Screen** menu are to name, number, edit, and manage screens. The following sections will explain these commands in detail.

2.5.1 New Screen

Select **New Screen** to create a new screen.


In the **Name** field, enter the name for the new screen. In the **Number** field, enter the screen number.

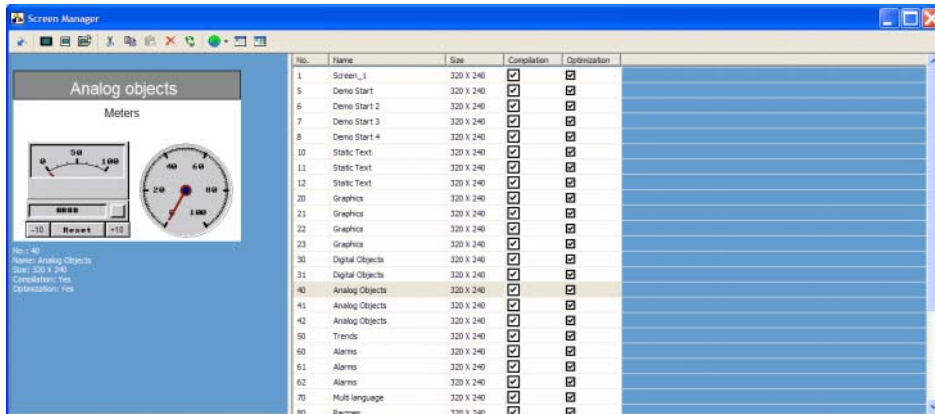


The Create New Screen dialog box

2.5.2 Screen Manager

The function of the Screen Manager command is to display all of the application files in detail view or thumbnail view in order to make it easy for a user to search, modify, edit, and so on.

Select **Screen/Screen Manager** or click . The entire Screen Manager will be displayed in the middle of the window or minimized on the left side.




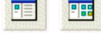









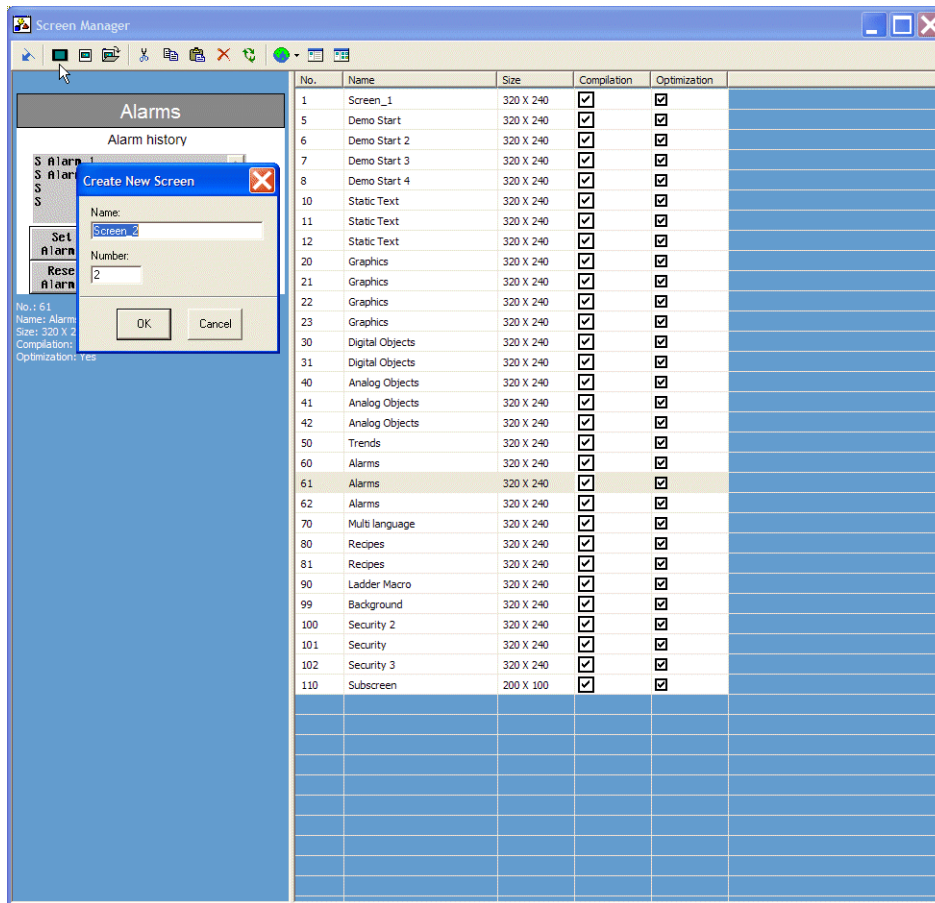
The Screen Manager

In the **Screen Manager** window, click the number or name to display the selected screen. If you want to open the image, just double-click on the image.

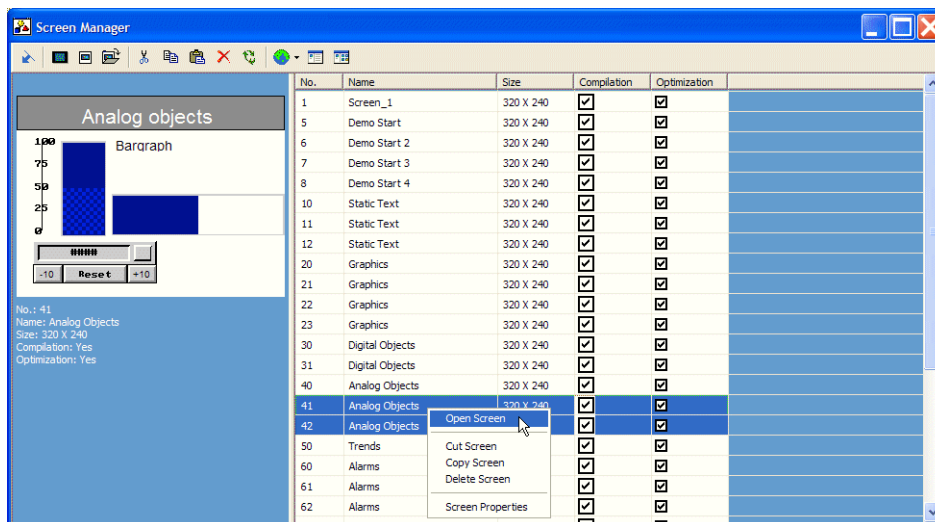
Screen Manager Icons

Screen Manager icons provide functions to manage screens and screen properties, as well as functions to increase ease of use of the **Screen Manager**, as explained below:

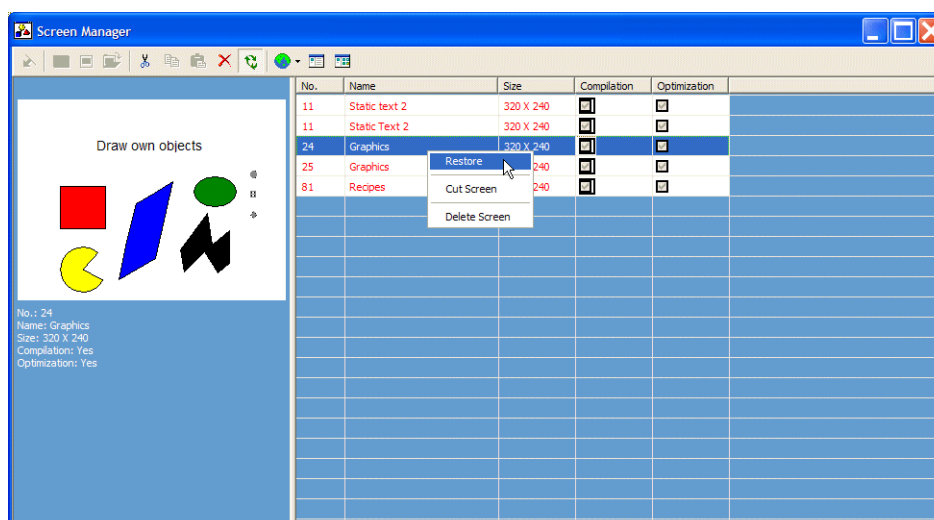
Icon	Name	Function
	Dock	The size of the Screen Manager window will be decreased, and the Screen Manager will be docked at the left side of the screen.
	Undock	The Screen Manager window is restored to full screen size.
	Close	Closes the Screen Manager .
	Detail View/ Thumbnail View	Displays screen data in different ways, either as detailed list data, or as thumbnails.
	New Screen	Adds a screen with No. and Name . A new screen appears in the Screen Manager list. Also described in image below.
	Screen Properties	Opens the Screen Properties dialog box. Please refer to section Screen Properties for complete details.
	Open Screen	Opens and displays the selected screen while the Screen Manager window will be minimized at the bottom. A screen can also be opened by double-clicking on it.
	Cut Screen/ Copy Screen/ Paste Screen/ Delete Screen	Used to edit the selected screens. A pop-up menu to select editing operations can also be displayed by right-clicking on the screen. Use the Ctrl -key while left-clicking to select several screens for editing at once. Also described in image below.
	View Recycle Bin	Lists all of the deleted screens marked with red. Right-click on the screen list and then select Restore from the list. Also described in image below. Click the  again to return to the previous Screen Manager window.
	Select Language	Specifies the language displayed in the screen. Also described in image below.



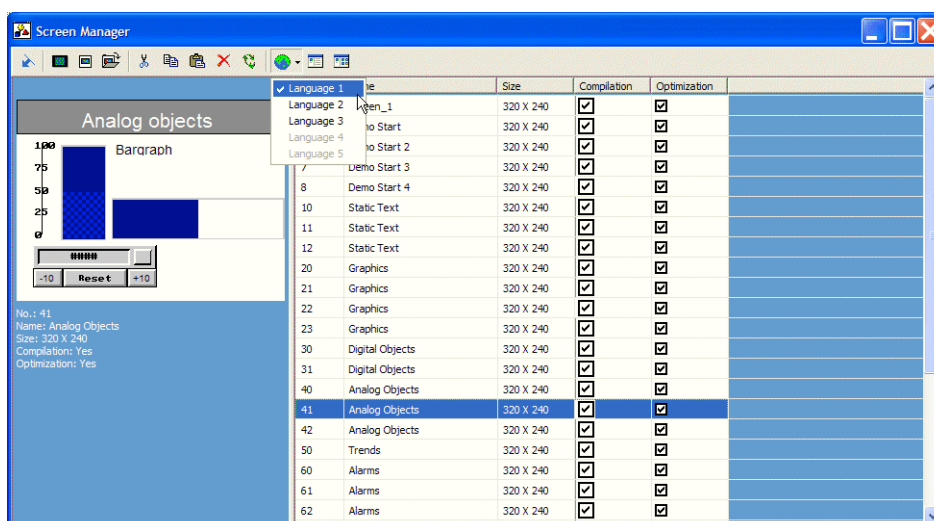
Adding a new screen from the Screen Manager.



Selecting two screens for editing at once. Right-clicking opens the edit pop-up menu.



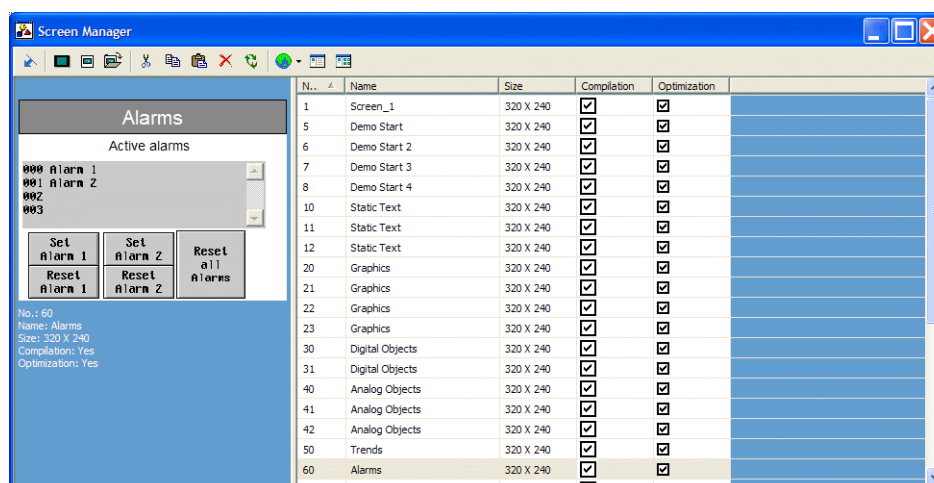
Restoring a deleted screen.



Switching between Language 1 and Language 2

Screen Table

In the **Screen Manager**, click the **Detail View** button. The detailed data will be listed in the table in five columns: **No.**, **Name**, **Size**, **Compilation** and **Optimization**. When you click the header, the data will be listed in increased or decreased order following numerical order, dimensional order, and so on.



List sorted in numerical order after clicking No. in the table header.

The **Screen Manager** can also help you manage and edit the screens quickly and easily. If the data in these columns is to be modified, you can click the **Screen Properties** button on the toolbar and modify the screen properties in the displayed dialog box. Alternatively, right-click for the **Screen Properties** selection.

The **Compilation** column is used to choose to check compilation or not. This can also be configured in the **Screen Properties** dialog box, on the **General** tab.

The **Optimization** column is used to choose to perform block optimization during compilation. This can also be configured in the **Screen Properties** dialog box, on the **Read Blocks** tab.

Please refer to section [Screen Properties](#) for complete details.

2.5.3 Close Screen

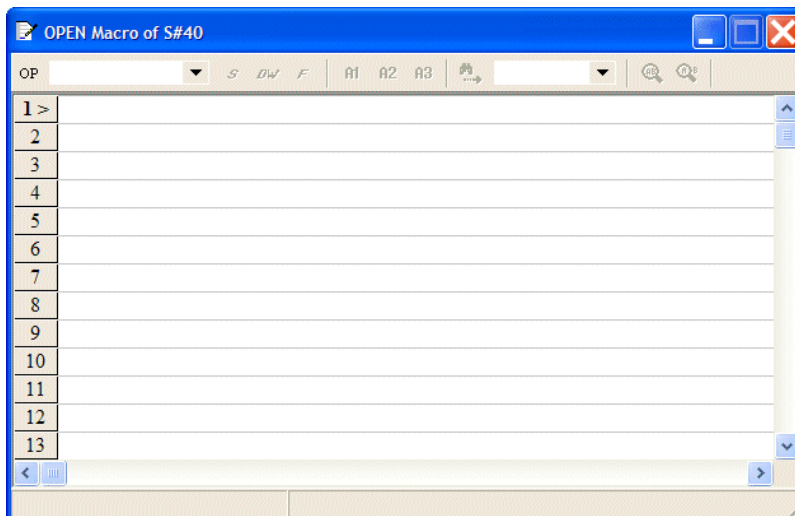
Closes the current screen.

2.5.4 Cut/Copy/Delete Current Screen and Paste Screen

Cuts, copies or deletes the current screen. A cut or copied screen can be pasted onto other screens.

2.5.5 OPEN Macro, CLOSE Macro and CYCLIC Macro

These three macros enable the operator terminal to execute data initialization, display control, and internal register or contact initialization. Once these commands are selected, the operator terminal will display the edit screen.



The OPEN Macro edit screen

OPEN Macro is executed when the screen open command is issued. A screen is not displayed until the **OPEN Macro** is completely executed. There is one **OPEN Macro** per screen.

CLOSE Macro is executed when the screen close command is issued. A screen is not closed until the **CLOSE Macro** is completely executed. There is one **CLOSE Macro** per screen.

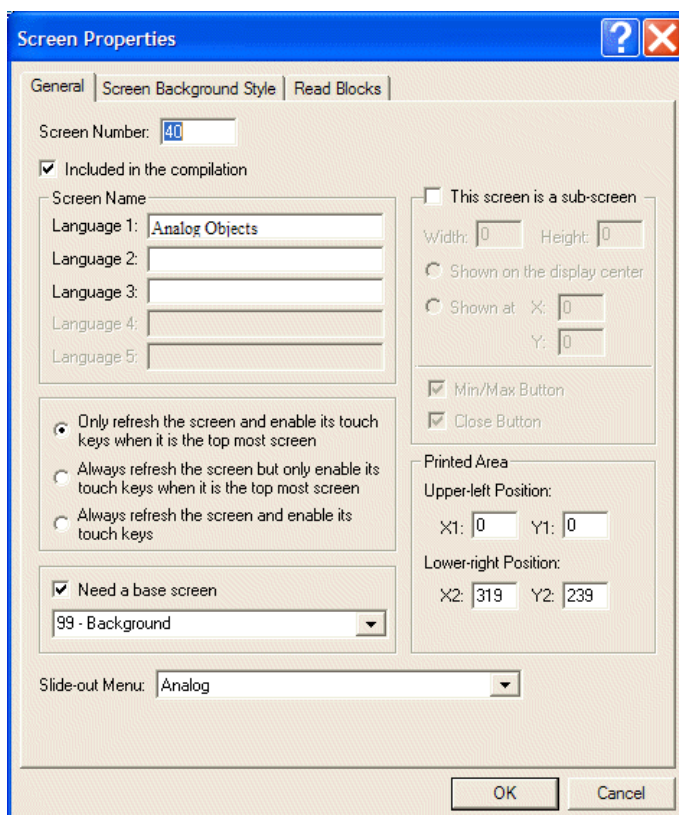
The **CYCLIC Macro** is executed cyclically when the screen is displayed. The operator terminal stops executing the macro when it encounters an **End** command or reaches the end of the macro.

Please refer to chapter [Macros](#) for complete details.

2.5.6 Screen Properties

The function of the **Screen Properties** command is to display the properties of the current screen including **General**, **Screen Background Style** and **Read Block** tabs.

General Tab



The General tab of the Screen Properties dialog box

Screen Number: Specifies the number of the current screen.

Included in the compilation: Use this option to compile the selected screen or not. This feature provides convenient screen editing but is unnecessary for draft screens.

Screen Name: Enters the name for a current screen.

Screen Update and Key Function: Specifies the types of screen updates.

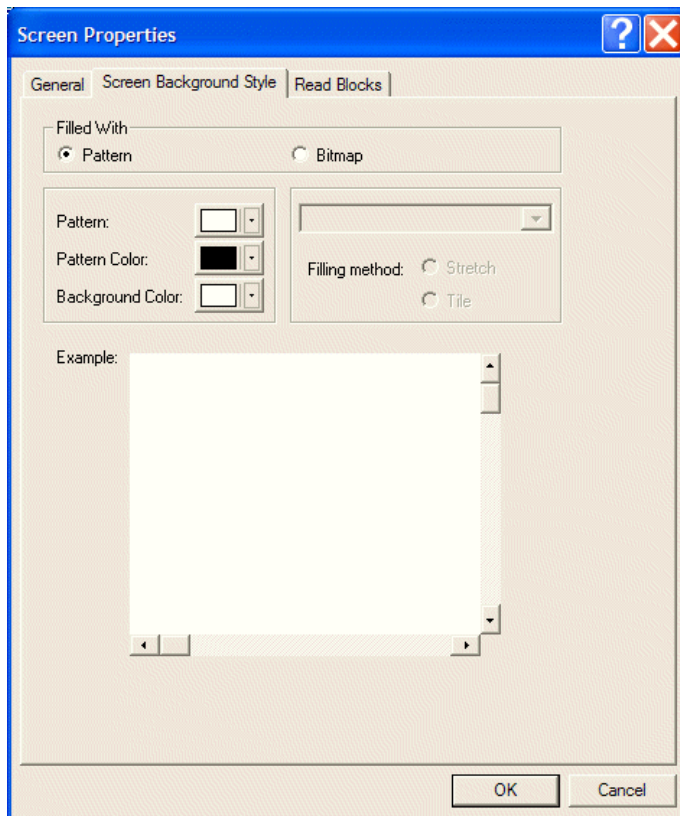
Need a base screen: Check this option for a base screen. A base screen can be the background for various screens.

This screen is a sub-screen: Check this option to display the selected screen as a sub-screen. You can designate the width, height, location and buttons of a sub-screen in this block. (Maximum width = 180; height = 160). For further information, please refer to section [Sub-screen](#).

Printed Area: Specifies the area to print. Note that this feature is available only on models with PRINTER PORT.

Slide-out Menu: Specifies the slide-out menu of the screen.

Screen Background Style Tab



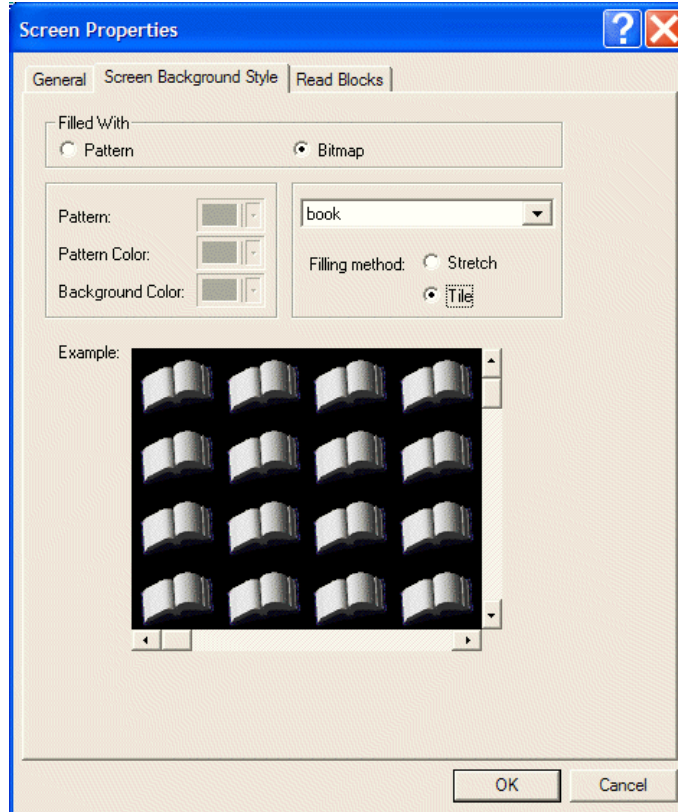
The Screen Background Style tab of the Screen Properties dialog box

Filled With:

When selecting **Pattern**, the options **Pattern**, **Pattern Color** and **Background Color** will be available for selection.

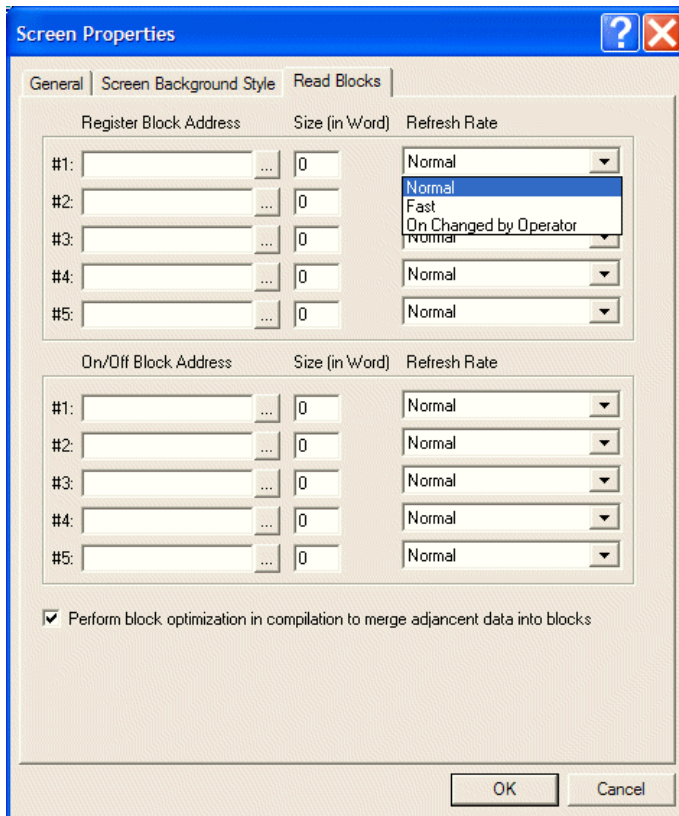
When selecting **Bitmap**, a bitmap can be selected from the drop-down list. There are two filling methods available:

- **Stretch**: Displays the bitmap entirely on the entire screen. When this option is selected, the bitmap can be enlarged to match the designed screen.
- **Tile**: Displays the duplicate bitmaps lined up on the designed screen. Each bitmap keeps its original size.



Selecting the Tile option

Read Blocks Tab



The Read Blocks tab of the Screen Properties dialog box

The function of the **Read Block** tab is to specify register block addresses, on/off block addresses, size (in words), and refresh rate.

Register Block Address: Specifies the register block address according to the controller. A screen can have a maximum of five specified register addresses.

On/Off Block Address: Specifies the on/off block address. A screen can have a maximum of five specified register addresses.

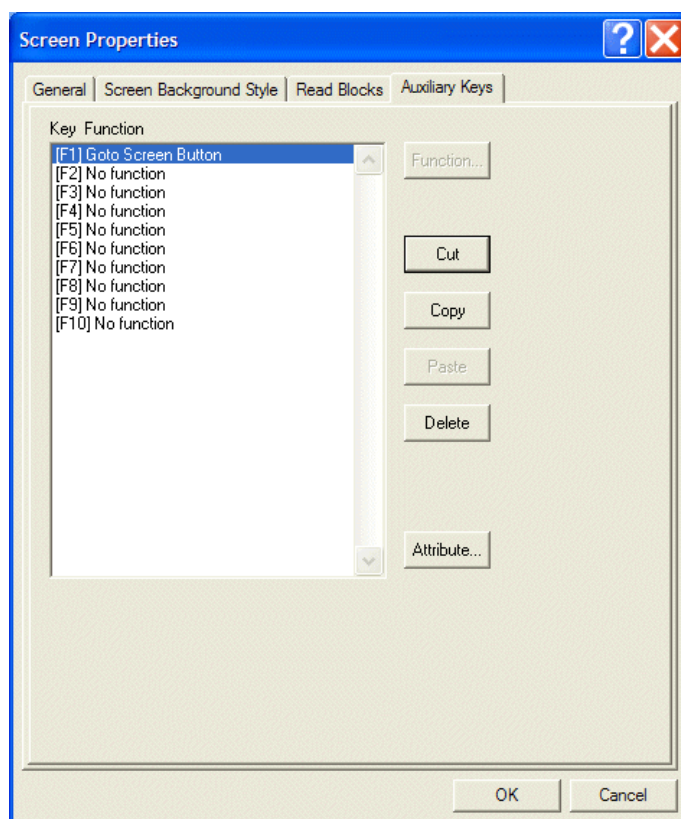
Size: Specifies the size of block.

Refresh Rate:

- **Normal:** Reads data at normal speed for the controller.
- **Fast:** Reads data at twice the speed of **Normal**.
- **On Unchanged by Operator:** Enables users to change the value on the operator terminal but not change the value in the controller

Perform block optimization: Check this option to merge adjacent data into blocks. Note that it is recommended that the data addresses of the controller are continuous to ensure good communication with the controller.

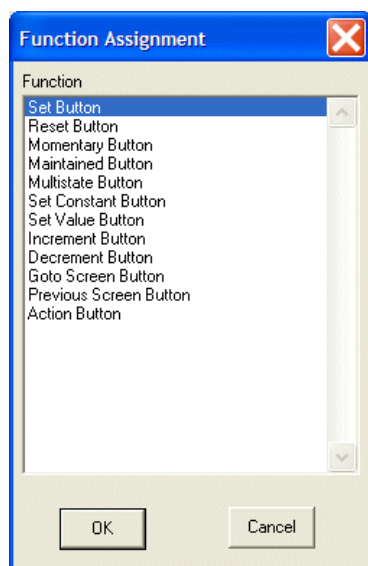
Auxiliary Keys Tab



The Auxiliary Keys tab of the Screen Properties dialog box

The main function of the **Auxiliary Keys** is to create the attributes of external buttons. The buttons defined on the **Auxiliary Keys** tab are only available for the current screen. If the F1 key is defined in screen 5 as **Goto Screen 1**, this function is only available on screen 5.

Click the **Function** key to display the function assignment dialog box.



The Function Assignment dialog box

Note:

This function is not available for all operator terminal models: please refer to [Appendix A - H-Designer Features and Operator Terminal Models](#) for complete details.

Sub-screen

A sub-screen is a screen that is smaller than the normal screen. The operator terminal displays a sub-screen in the center of the screen without destroying the existing display and adds a raised frame to it automatically.

The following are the steps required to create a sub-screen:

1. Create a new screen; enter the screen name and number.
2. Open the **Screen Properties** dialog box and check the **This screen is a sub-screen** box.
3. Enter the width and height of the sub-screen.
4. Specify the position of the sub-screen display (shown on the display center or designated specific position).

Screen Properties

General | Screen Background Style | Read Blocks | Auxiliary Keys

Screen Number: 40

☒ Included in the compilation

Screen Name: Analog Objects

Language 1: Analog Objects

Language 2:

Language 3:

Language 4:

Language 5:

☒ This screen is a sub-screen

Width: 0 Height: 0

☒ Shown on the display center

☐ Shown at X: 0 Y: 0

☐ Min/Max Button

☐ Close Button

☒ Only refresh the screen and enable its touch keys when it is the top most screen

☐ Always refresh the screen but only enable its touch keys when it is the top most screen

☐ Always refresh the screen and enable its touch keys

☐ Need a base screen

99 - Background

Printed Area

Upper-left Position: X1: 0 Y1: 0

Lower-right Position: X2: 0 Y2: 0

OK Cancel

Modifying attributes of a sub-screen

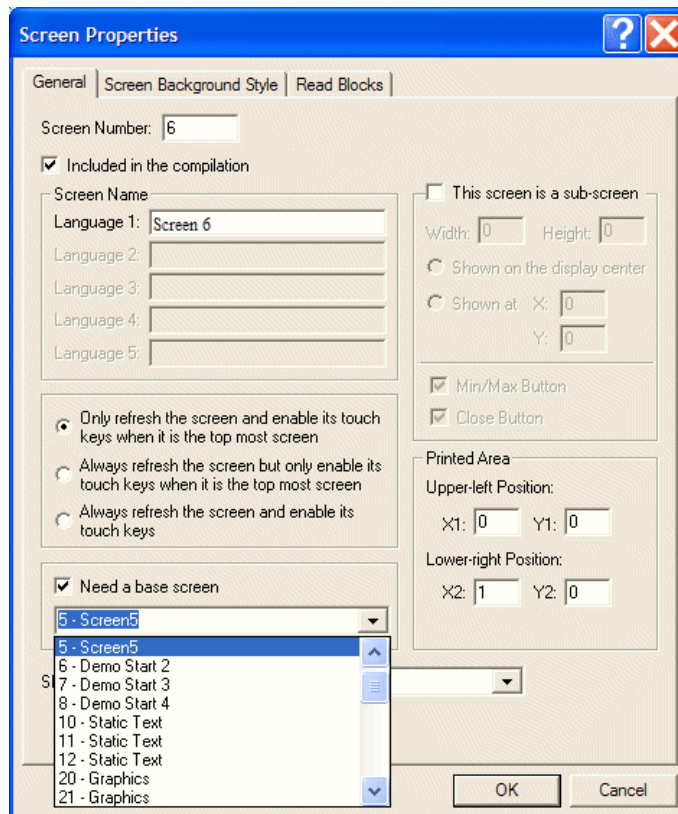
5. After clicking **Enter**, the screen will be minimized to the specified size.

Base Screen

A base screen is a screen which may be used as a template for many different screens. Once you have edited a base screen, all of the same base screens in the application will be changed at the same time.

The following are the steps required to create a base screen:

1. Create a base screen first, named for example **Screen 5**.
2. Create a new screen (**Screen 6**). Then check the **Need a base screen** box and specify the base screen (**Screen 5**) in the **Screen Properties** dialog box.



Specifying a base screen






















2.6 Draw Menu





To strengthen the effectiveness of the display of the created objects, it is often helpful to draw a rectangle, a line, or a scale to label the object data. This will help users to read and take note of the data.

The geometric shapes are static, and are not influenced by dynamic controller data.

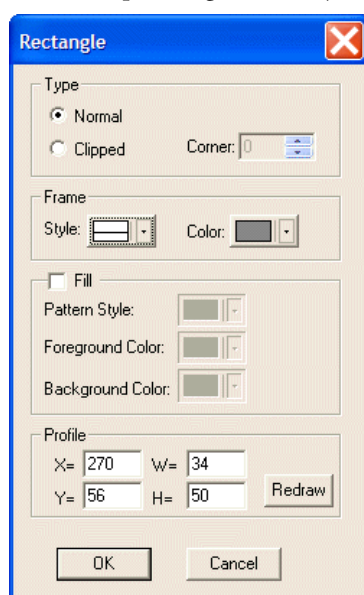
2.6.1 Geometric Shapes

The following geometric shapes are included:

Toolbar button	Function	Described in section
	Draws a dot	Dot
	Draws a line	Line, Horizontal Line and Vertical Line
	Draws a horizontal line	
	Draws a vertical line	
	Connects lines with mouse movement	Connected Lines and Free Form
	Connected lines and curves with mouse movement	
	Draws a rectangle	Rectangle and Solid Rectangle
	Draws a solid rectangle	
	Draws a parallelogram	Parallelogram and Solid Parallelogram
	Draws a solid parallelogram	
	Draws a circle	Circle, Solid Circle, Ellipse and Solid Ellipse
	Draws a solid circle	
	Draws an ellipse	
	Draws a solid ellipse	
	Draws an arc	Arc, Pie and Solid Pie
	Draws a pie segment	
	Draws a solid pie segment	
	Draws a polygon	Polygon and Solid Polygon
	Draws a solid polygon	
	Static text display and design	Static Text
	Displays a bitmap	Bitmap

Toolbar button	Function	Described in section
	Draws a frame/edge	Frame/Edge
	Draws a scale	Scale
	Draws a table	Table
	Displays a shape	Shape

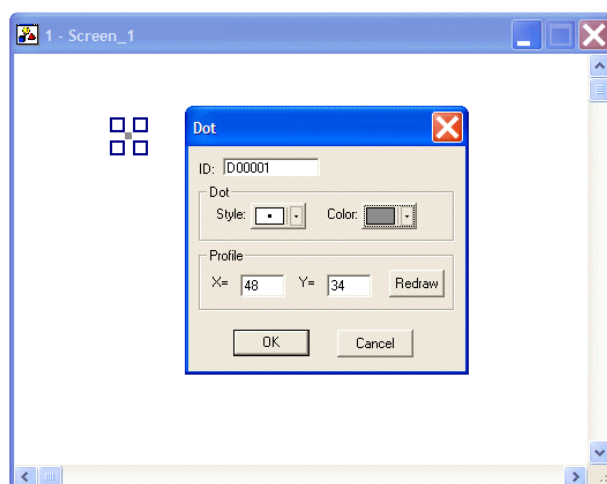
Double-click on the object, or right-click and select **Object Attributes**. The dialog box corresponding to the object will appear.



The Rectangle dialog box

Dot

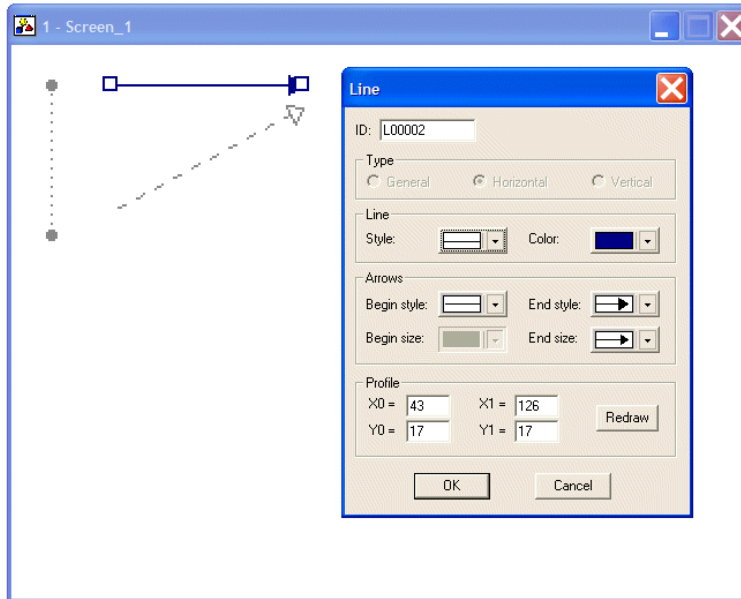
Style, Color and Profile can be specified in the **Dot** dialog box.



Drawing a Dot

Line, Horizontal Line and Vertical Line

Type, Color, Arrows and Profile can be specified in the **Line**, **Horizontal Line** and **Vertical Line** dialog boxes.



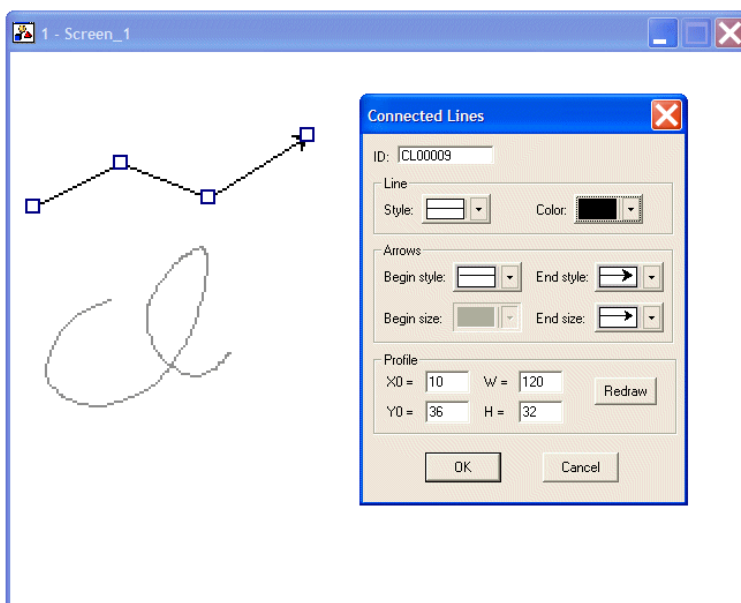
*Drawing a **Line** with arrows*

Connected Lines and Free Form

Connected Lines are used to connect lines with the movement of the mouse cursor. Click mouse button once, and then move the cursor somewhere else on the screen, to draw a straight line between the two points. This will continue until you right-click with the mouse.

Free Form is used to create lines and curves by holding down the left mouse button while moving the mouse cursor. This will continue until you right-click with the mouse.

Style, Color, Arrows and Profile can be specified in the **Connected Lines** and **Free Form** dialog boxes.

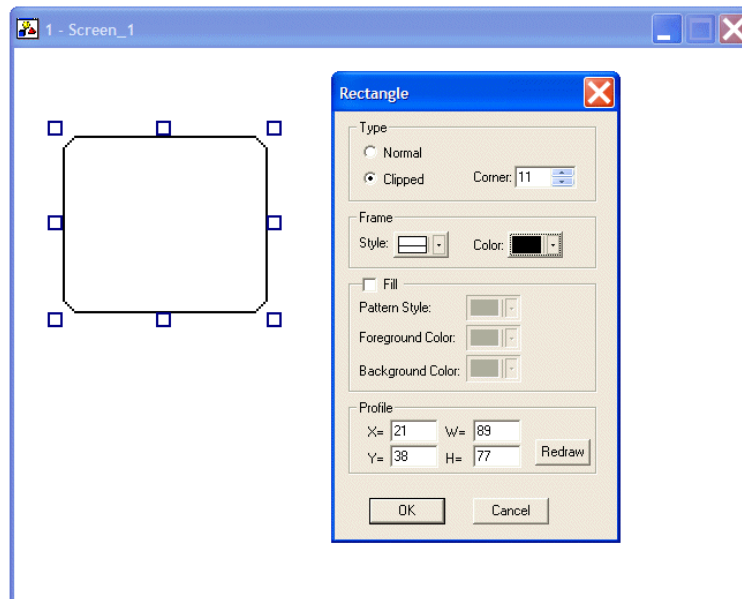


*Drawing **Connected Lines** with arrows*

Rectangle and Solid Rectangle

Rectangle and **Solid Rectangle** are used to draw rectangular shapes.

Type, **Frame** and **Profile** can be specified in the **Rectangle** and **Solid Rectangle** dialog boxes. **Fill** is used for the **Solid Rectangle**.

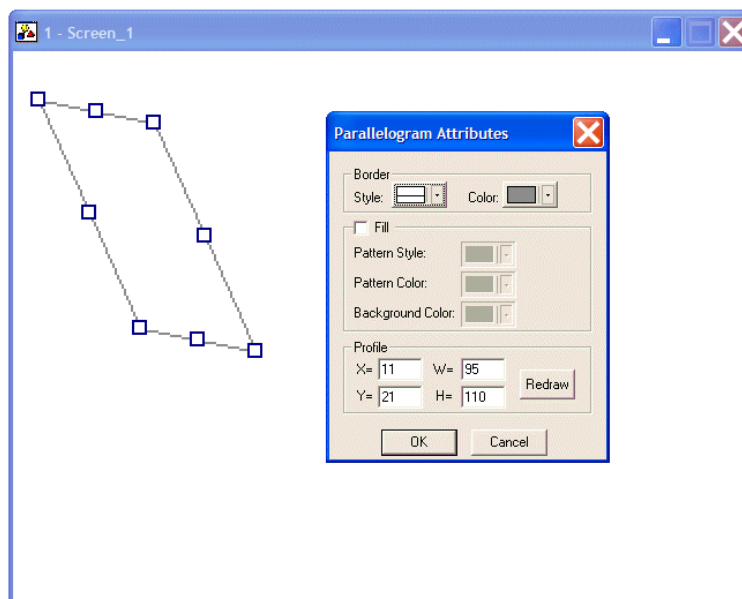


Rectangle with clipped corners

Parallelogram and Solid Parallelogram

Parallelogram and **Solid Parallelogram** are used to draw a parallelograms by holding down the left mouse button to draw a side. This side will continue until you left-click with the mouse. Then you drag this side to configure a rectangle until you right-click with the mouse.

Border and **Profile** can be specified in the **Parallelogram** and **Solid Parallelogram** dialog boxes. **Fill** is used for the **Solid Parallelogram**.

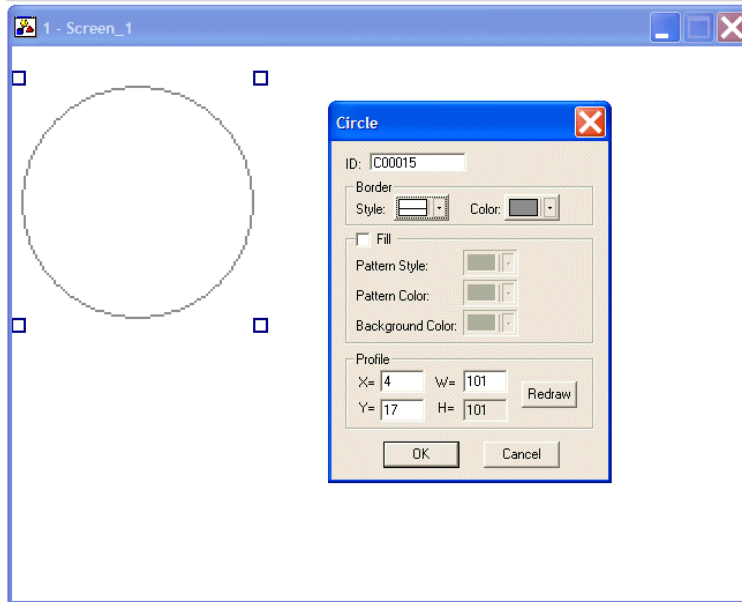


Drawing a Parallelogram

Circle, Solid Circle, Ellipse and Solid Ellipse

Circle, Solid Circle, Ellipse and Solid Ellipse are used to draw circles and ellipses.

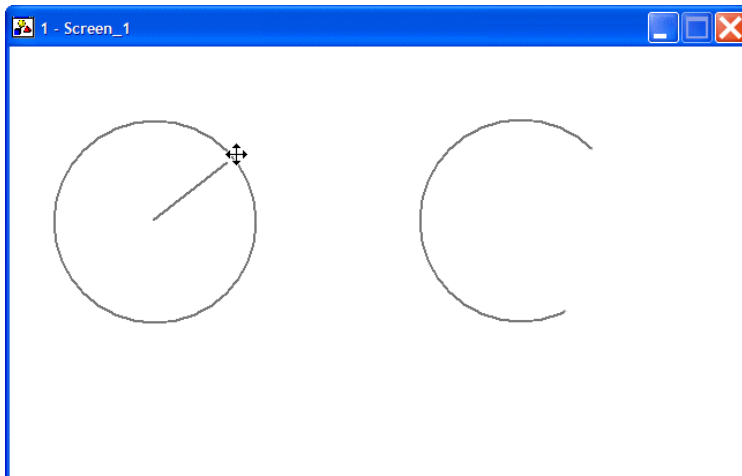
Border and **Profile** can be specified in the Circle and Ellipse dialog boxes. **Fill** is used for the Solid Circle and Solid Ellipse.



The Circle dialog box

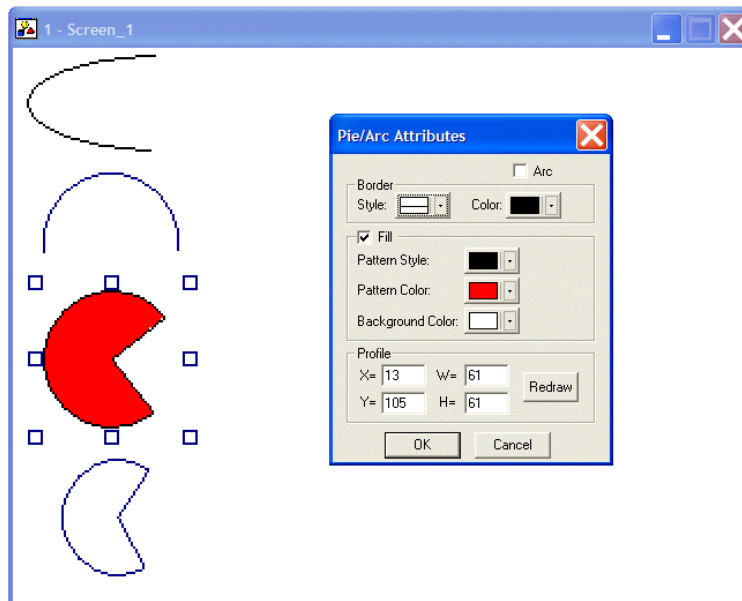
Arc, Pie and Solid Pie

Arc is used to draw a circle by holding down the left mouse button. Continue until you have configured the desired size, then right-click. Left-click for a radius display. You can drag the radius to configure a desired arc until left-clicking again.



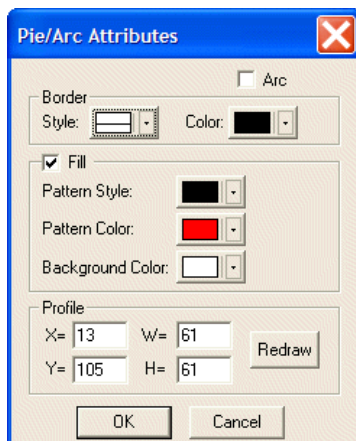
Drawing an Arc

Drawing a pie segment or a solid pie segment is similar to drawing an arc. The difference between a pie and an arc is that two lines are connected between the two sides of an arc and the center.



Arc, Pie and Solid Pie

The Arc, Pie and Solid Pie dialog boxes are the same:



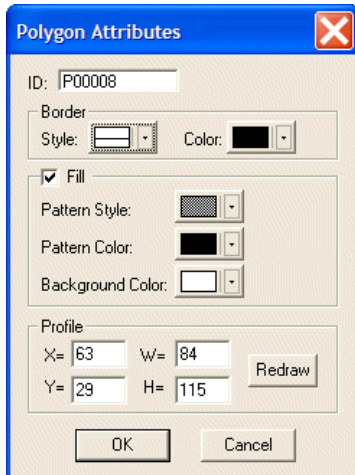
The Pie/Arc Attributes dialog box

Check **Arc** to draw an arc; check **Fill** to draw a solid arc and specify the **Pattern** there; check **Arc** and adjust the width and height under **Profile** to configure an arc from an ellipse.

Polygon and Solid Polygon

Polygon is used to draw sides of a polygon by moving the mouse cursor and to connect the lines between the starting point and the terminal point with the shortest distance. Draw a polygon by dragging the mouse cursor, then left-clicking on the turning point and right-clicking to form a polygon.

Border and **Profile** can be specified in the **Polygon** and **Solid Polygon** dialog boxes. **Fill** is used for the **Solid Polygon**.



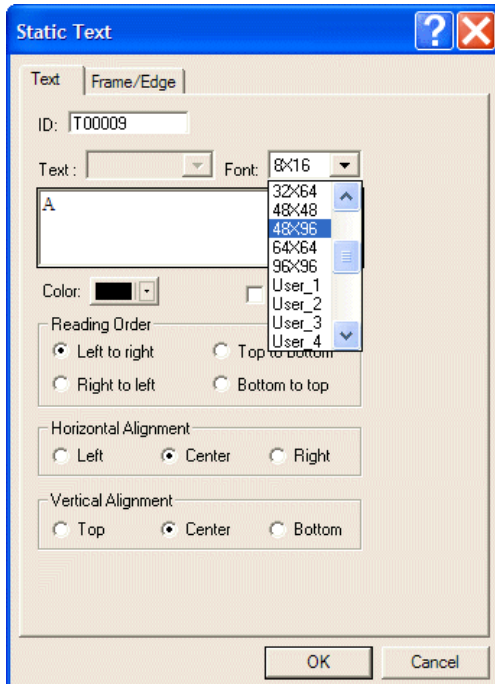
The Polygon Attributes dialog box

Static Text

The **Static Text** object allows you to select text, font, color, reading order, alignment, and frame/edge. Double-click on the object to display the Static Text dialog box on the screen.

Text tab

Enter text in the **Text** area. 16 different types of fonts can be selected.

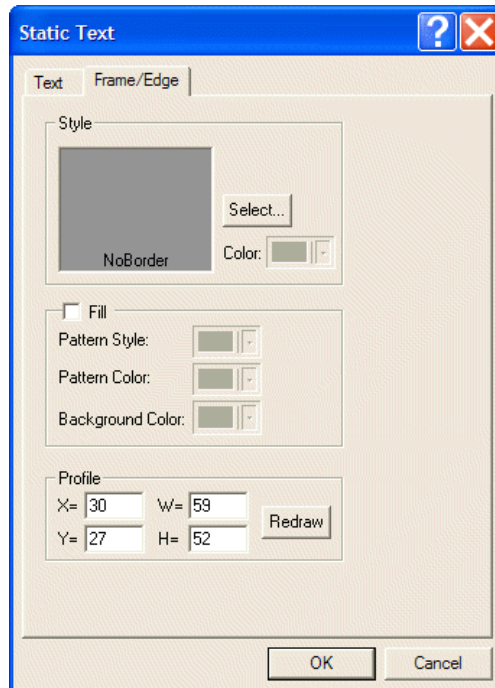


The Text tab of the Static Text dialog box

For properties not explained in this section, please refer to the section [Font Library](#).

Frame/Edge tab

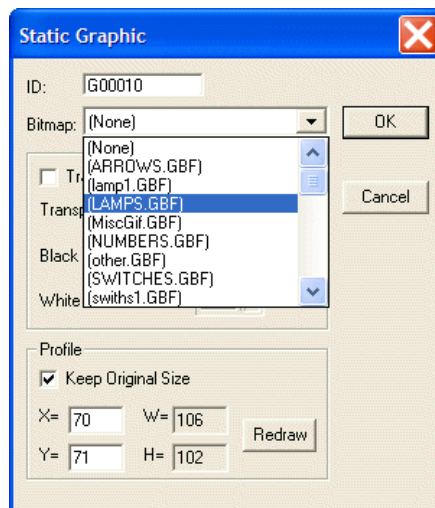
Click the **Select** button to specify the style of frame/edge.



The Frame/Edge tab of the Static Text dialog box

Bitmap

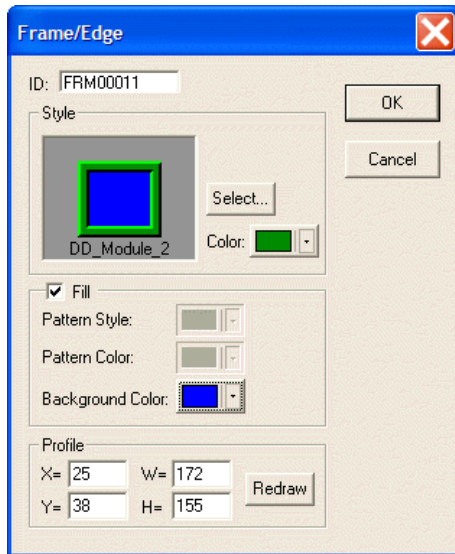
The purpose of **Bitmap** is to provide graphics for selection. Double-click on the object; you can then select the bitmap from the drop-down list in the **Static Graphic** dialog box.



The Static Graphic dialog box

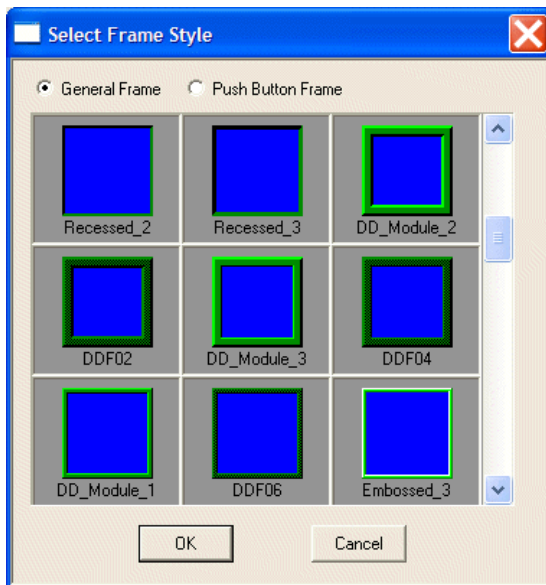
Frame/Edge

Frame/Edge enables you to choose style, pattern style, background and color.



The Frame/Edge dialog box

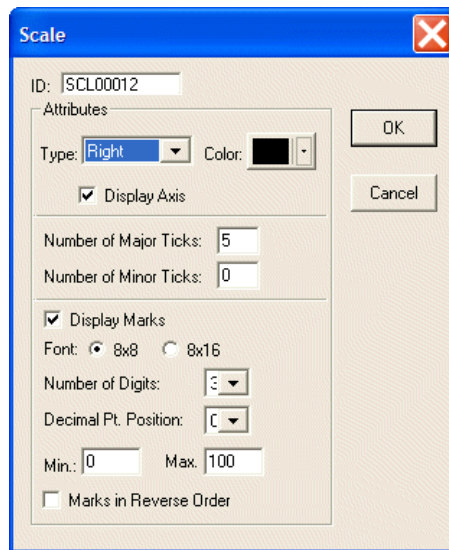
Clicking Select displays flow chart styles for selection.



Selection of Frame Style

Scale

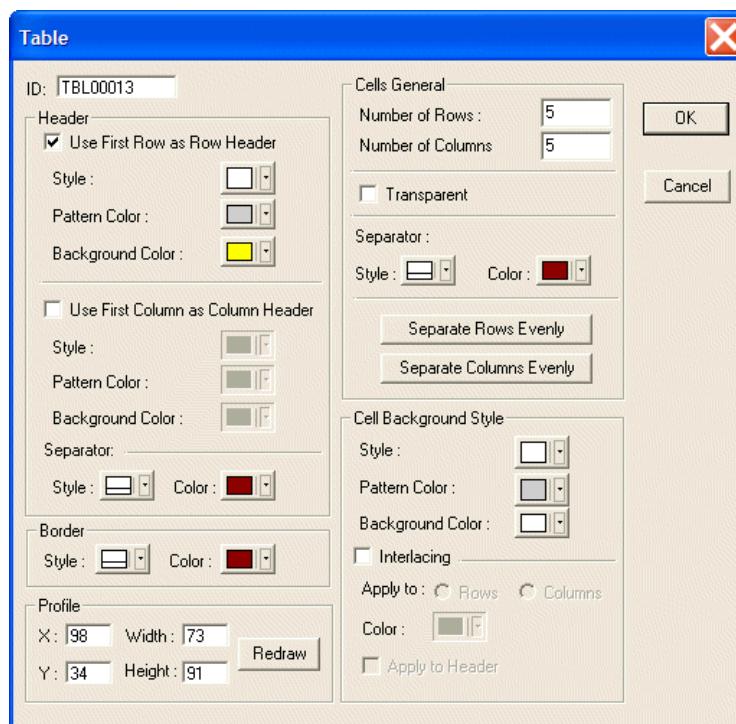
Scale provides scales directed left, right, up and down, as well as color, number of ticks, and display marks for scales.



The Scale dialog box

Table

Table is used to create tables.



The Table dialog box

Use First Row as Row Header: Specifies the pattern style and color of row header.

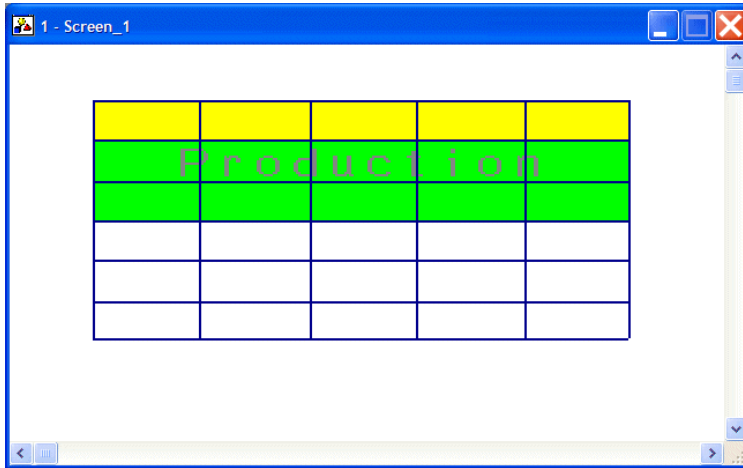
Use First Column as Column Header: Specifies the pattern style and color of column header.

Border: Specifies the style and color of border.

Profile: Specifies the location and size of a table.

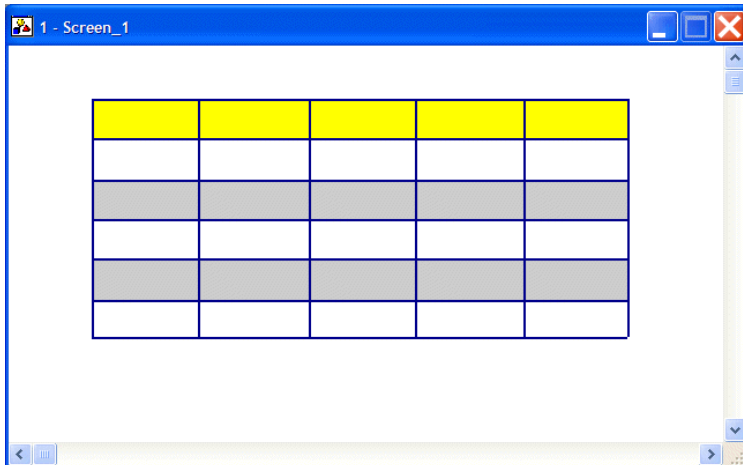
Cells General: Specifies the number of rows/columns, and style.

Transparent: Lets other object(s) be displayed under the table.



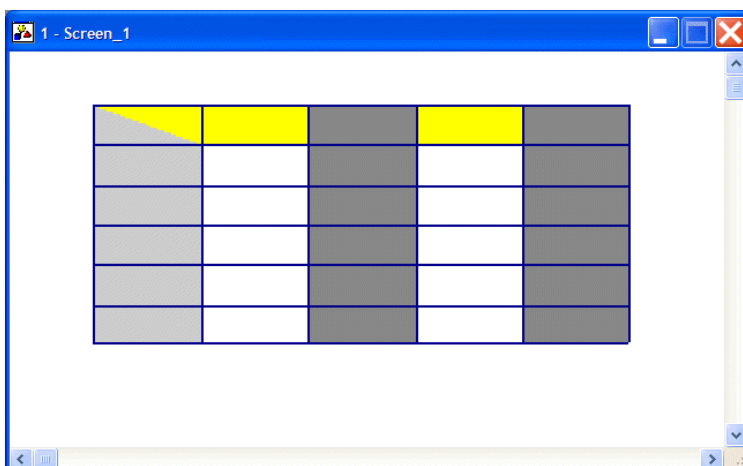
Example: Static text displayed under the table.

Interlacing: Interlaces rows or columns. Not available with transparent table.



Interlacing

Apply to Header: Interlace applied to header. Not available with transparent table.



Interlacing applied to header

Shape

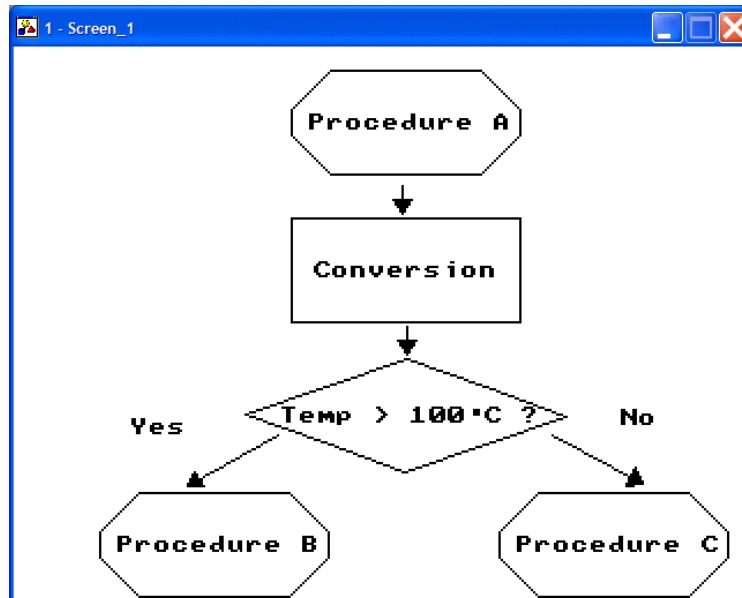
Shape provides graphics for selection. Double-click on the object, then click **Select** to access the shape library in the Shape dialog box.

Flow Chart

Flow chart is one of the applications in **Draw** used for lines, geometric graphs and frame/edge editing. You can illustrate an applied flow chart clearly to facilitate operations.

Example:

Convert the boiler temperature in **Procedure A** into centigrade ($^{\circ}\text{C}$). If the temperature is less than 100°C , **Procedure B** will be entered; if the temperature is 100°C or more, **Procedure C** will be entered. The following flow chart is made up of polygons, rectangles, lines with arrows and static text:



Flow chart example

2.7 Object Menu

A screen object is an item placed on the screen to perform a particular function. Each object has its unique user configurable properties and the object can be set to perform in exactly the method desired.

Objects are divided into four categories:

1. Related to screen buttons and dynamic data, e.g. **Push Button**, **Numeric Entry** and **List**
2. Unrelated to screen buttons but related to dynamic data, e.g. **Numeric Display** and **Bar Graph**
3. Related to dynamic controller data and operator terminal memory buffer zone; **Historical Display** and **Alarm Display**
4. Related to application; data contents are connected with the entire system. If one of the contents is modified, such as text display or controller data format, the other objects with the same application will be changed simultaneously as **Sub-macro**.

Object	Library	Application
Push Button		
Numeric Entry		
Character Entry (V)		
List		
Drop Down List		
Indicator		
Numeric Display		
Character Display (W)		
Message Display		
Bar Graph		
Trend Graph		
X-Y Chart		
Panel Meter		
Pie Graph		
Dynamic Graphic		
Historical Display		
Alarm Display		
Sub-macro (Z)...		

Related to screen button and dynamic data

Irrrelative to screen button but related to dynamic data

Related to dynamic controller data and operator terminal memory buffer zone

Related to application

The Object menu

2.7.1 Creating Objects

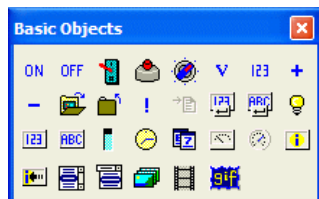
You can select the object type from the **Object** menu for editing. Some objects, such as the **Push Button**, have sub-command lists.

Object	Library	Application	Tool	Options	LadderPlus
Push Button					
Numeric Entry					
Character Entry (V)					
List					
Drop Down List					
Indicator					
Numeric Display					
Character Display (W)					
Message Display					
Bar Graph					
Trend Graph					
X-Y Chart					
Panel Meter					
Pie Graph					
Dynamic Graphic					
Historical Display					
Alarm Display					
Sub-macro (Z)...					

Set
Reset
Maintained
Momentary
Multistate
Set Value
Set Constant
Increment
Decrement
Goto Screen
Previous Screen
Action
Data to Text Conversion

The Push Button sub-commands

Some of the objects are provided in the **Basic Objects** toolbar for editing.



The Basic Objects toolbar

Select a desired object from the list, for example **Push Button/Set Button**, and you will get a cursor (+) that allows you to drag the object to the desired size by clicking the left mouse button and then clicking again when finished.

Once created, the object can be resized by dragging one of the object's handlebars. If the object's handlebars are not visible, clicking anywhere on the object will display the handlebars). To move the object, click at the center of the object and then drag it.

2.7.2 Specifying Object Properties

There are three ways to specify the properties of an object:

1. You can select **Object Attributes** from the **Edit** menu
2. You can double-click on the object
3. You can right-click on the object and then select **Object Attributes** from the pop-up menu.

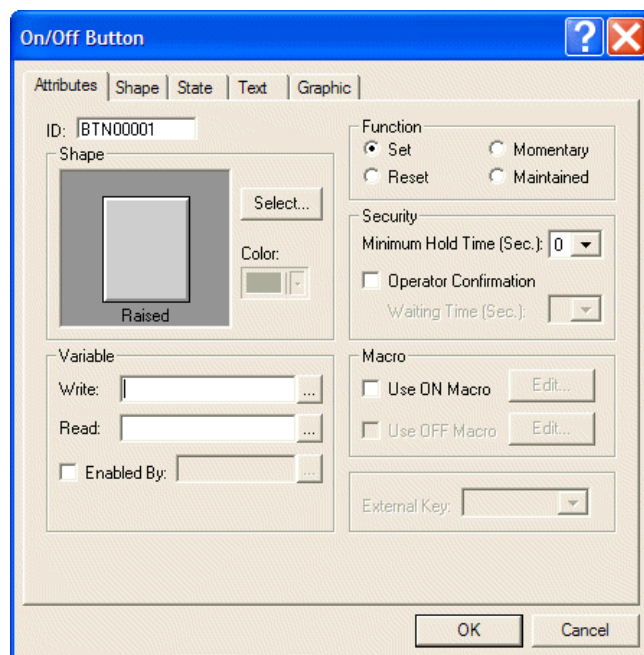
Each of the above methods will bring up the dialog box for the properties specified.

In the H-Designer software, each object has a corresponding dialog box. For example, there is the **On/Off Button** dialog box in the **Set Button** object; there is the **Numeric Entry** dialog box in the **Numeric Entry** object.

The following five tabs are available for most objects; certain specific properties will be explained later.

Attributes Tab

Major properties are specified on the **Attributes** tab. Each object has attributes that define its use.



The Attributes tab of the On/Off Button dialog box

Shape

Select: Specifies shape from library.

Color: Specifies the color of the selected shape.

Variable

Write: Writes to the specified controller register.

Read: Reads the value from the specified controller register. If the location is not specified, the operator terminal reads from the **Write** location.

Enabled By: Specifies the controller register to the ON button. This is not available in OFF state labeled.

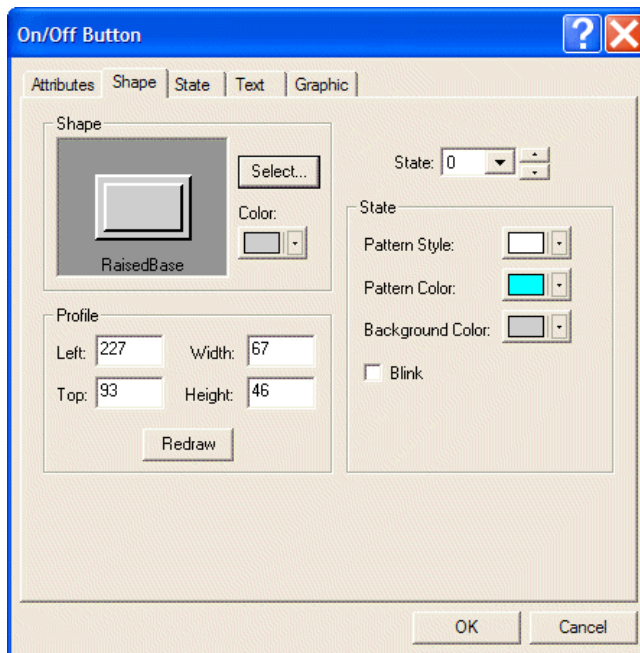


Specified State = OFF / Specified State = ON

This feature is only available on objects with input text/numeric or specific states.

Shape Tab

The shape style of a selected object is defined on the **Shape** tab.



The Shape tab of the On/Off Button dialog box

Shape

Select: Selects shape from library.

Color: Specifies the color of the shape.

Profile

Specifies the location, width, and height of the object.

State

Pattern Style: Specifies the pattern style for the object.

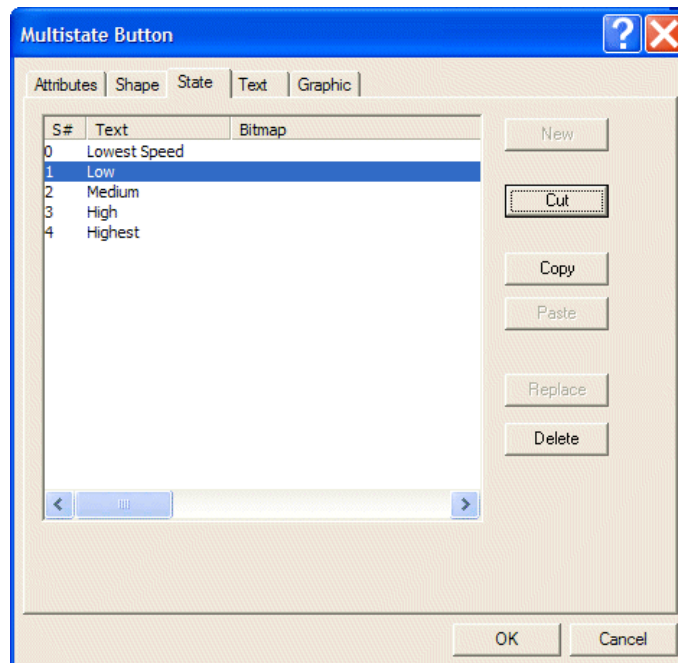
Pattern Color: Specifies the color of the pattern for the object.

Background Color: Specifies the background color of the object.

Blink: Specifies whether the object blinks.

State Tab

The states of the object are defined on the **State** tab.



The State tab of the Multistate Button dialog box

New: Adds a new state to the object.

Cut: Cuts the specified state to the clipboard.

Copy: Copies the specified state of the object and keep the original state.

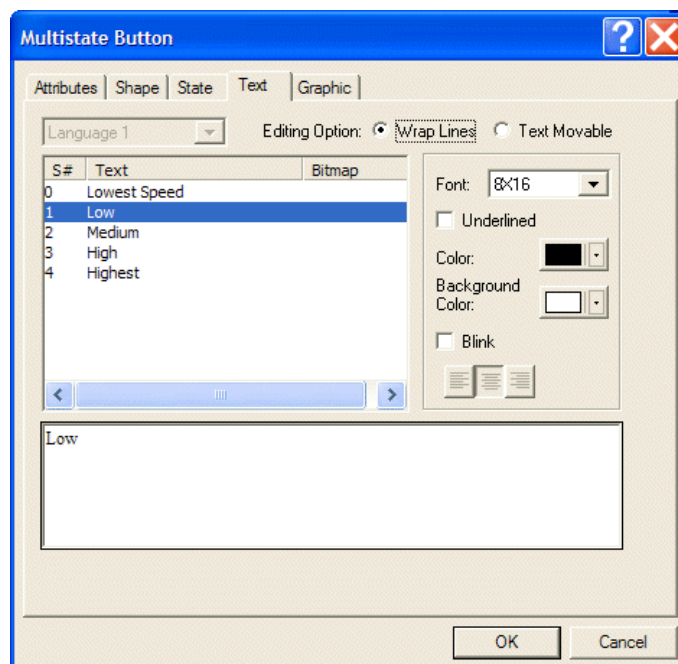
Paste: Pastes the state from the clipboard.

Replace: Replaces the current specified state from the clipboard.

Delete: Deletes the current specified state.

Text Tab

Font and appearance of the text in the object is specified on the **Text** tab.



The Text tab of the Multistate Button dialog box

Editing Option

Wrap Lines: If the length of the text is longer than the width of the button, it will be executed in wrapped lines.

Text Movable: If the length of the text is longer than the width of the button, it will not be executed in wrapped lines. Click the text on the selected object, to make the text surrounded with the handlebars for dragging.

Appearance

Font: Specifies the size of the font.

Underlined: Specifies whether the text is to be underlined.

Color: Specifies the text color.

Background Color: Specifies the background text color.

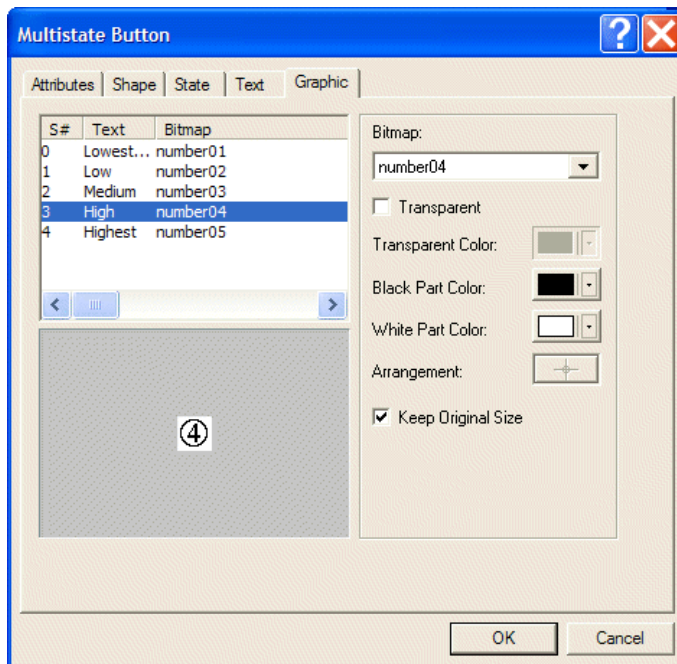
Blink: Specifies whether the text blinks.



Aligns the text to the left/center/to the right.

Graphic Tab

Bitmap style, color etc for each state is defined on the **Graphic** tab.



The Graphic tab of the Multistate Button dialog box

Bitmap: Specifies the bitmap to display.

Transparent: Specifies whether the bitmap is transparent.

Transparent Color: Specifies the color of the bitmap when transparent.

Black Part Color: Replaces the black part color (only available for monochrome displays).












White Part Color: Replaces the white part color (only available for monochrome displays).

Arrangement: Arranges the moveable bitmap to a previous location.

Keep Original Size: Keeps the bitmap's original size.

2.7.3 Buttons

There are 13 buttons in the sub-command list for the **Push Button** menu:

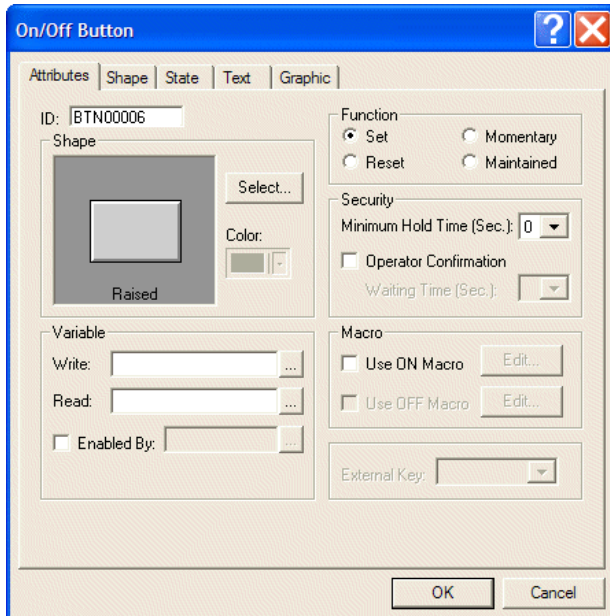
Icon	Function	Described in section
	Set Button: Click to set the contact as ON, release or re-click still to set ON.	Set Button
	Reset Button: Click to set the contact as OFF, release or re-click still to set OFF.	Reset Button
	Maintained Button: Click to set the contact as ON, released still on; re-click to set OFF.	Maintained Button
	Momentary Button: Click to set the contact as ON; release be OFF.	Momentary Button
	Multistate Button: Click to change a register to the next (previous) state of a referenced register. S0 → S1 → S2 → S3 → S4 → S0 (a straightforward cycle) or S0 → S4 → S3 → S2 → S1 → S0 (a reverse cycle).	Multistate Button
	Set Value Button: Click to a numeric keypad display. Click ENTER button to write a numeric entry to corresponding controller register.	Set Value Button
	Set Constant Button: Click to write a constant to a register.	Set Constant Button
	Increment/Decrement Button: Click to write the value obtained by adding/subtracting a constant to/from the corresponding register value corresponding controller register.	Increment Button , Decrement Button
	Goto Screen Button: Click to change the current screen to the specified screen.	Goto Screen Button
	Previous Screen Button: Click to change the current screen to the previous screen.	Previous Screen Button
	Action Button: Performs a built-in action.	Action Button

Set Button

When pressed, the operator terminal sets the controller's corresponding bit location to ON. A **Set Button** will be ON whether pressed or released.

Attributes Tab

Function: Select Set.

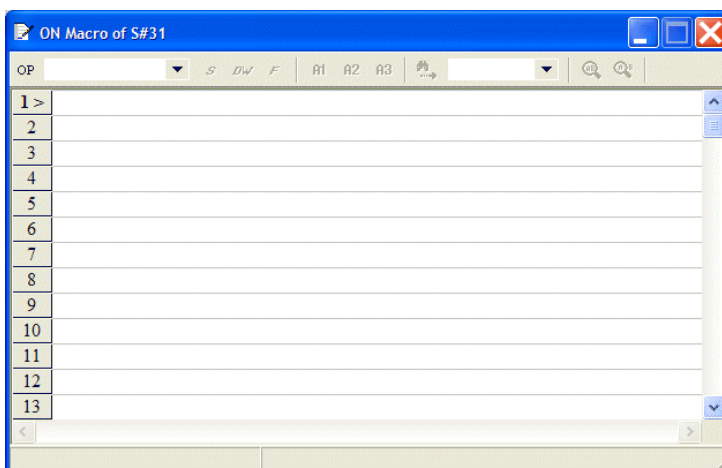


Selecting the Set function for the Set Button

Security:

- **Minimum Hold Time (Sec.):** Specify how long to activate the button (0-10 seconds).
- **Operator Confirmation:** If any changes have been made, this dialog box will appear on the screen to ask the user to confirm the desired operation. A waiting time of 5-60 seconds can be selected.

Macro: Check the **Use ON Macro** for a **Set Button**. Click the **Edit** button to display the **ON Macro** dialog box.



The ON Macro edit window

For properties not explained in this section, please see chapter [Macros](#).

- **Use ON Macro:** When the **Set Button** is selected, the operator terminal will run the program created as ON macro. This feature is for data control, screen display, controller register, bits initialized and so on.

For properties not explained in this section, please see the section [Specifying Object Properties](#).

Example of designing a Set Button

On the **Attributes** tab of the **On/Off Button**:

1. **Shape**: Select **Raised**.
2. **Write**: Specify the controller register **B0** to write in.
Read: Specify **B0**. (The controller model is NULL.)

On the **Text** tab of the **On/Off Button**:

3. Enter text **OFF** for state 0; the font **16 x 16**, the color **White** and the background color **Black**.
4. Enter text **ON** for state 1 ; the font **24 x 24**, the color **Black** and the background color **White**.



The button is white displaying ON in state 1, and black displaying OFF in state 0.

Reset Button

This button is contrary to the **Set Button**. A **Reset Button** sets a bit-location to OFF whenever pressed or released.

Attributes Tab

Function: Select **Reset**.

All other properties are the same as for the **Set Button**; please see section [Set Button](#).

Maintained Button

The function of the **Maintained Button** is to change the button states by pressing. Click to be ON and when released will still be ON until re-clicked to be OFF.

Attributes Tab

Function: Select **Maintained**.

Macro: There are **Use ON Macro** and **Use OFF Macro** options for the **Maintained Button**. For properties not explained in this section, please see chapter [Macros](#).

All other properties are the same as for the **Set Button**; please see section [Set Button](#).

Momentary Button

The function of the **Momentary Button** is to change the state by clicking and releasing. When the button is pressed, the bit-location is ON; when it is released, it is OFF.

Attributes Tab

Function: Select **Momentary**.

Macro: There are **Use ON Macro** and **Use OFF Macro** options for the **Momentary Button**. For properties not explained in this section, please see chapter [Macros](#).

To create two states as an ON button simultaneously, please see section [Set Button](#).

Multistate Button

When the button is pressed, the operator terminal will write the command to a corresponding controller bit-location or register. The option **Change to Next State** is used to change states in a straightforward cycle (S0 → S1 → S2 → S3 → S4 → S0); the option **Change to Previous State** is used to change states in a reverse cycle (S0 → S4 → S3 → S2 → S1 → S0). Click to change a register to the next (previous) state of a referenced register.

Attributes Tab

Variable:

Write: Writes the specified command to a corresponding controller bit-location and register.

- **Bit:** Only two states; enables you to enter multistated text but only two states can be displayed on the operator terminal.
- **Value:** 256 (0-255) states in all, 0 represents state 0; 1 represents state 1...etc.
- **LSB:** 16 states in all represented by bit. The operator terminal takes the bit number of the lowest bit that is on as the state number.

Format: Only available when **Value** has been selected. There are **BCD**, **Signed Binary** and **Unsigned Binary** options.

- **Read:** Specifies a register/bit location to read from; if the location is not specified, then the operator terminal reads from the **Write** location.

Function:

- **Change to Next State:** Changes the **Write** location to its next state in a forward cycle S0 → S1 → S2 → S3 → S4 → S0.
- **Change to Previous State:** Changes the **Write** location to its previous state in a reverse cycle S0 → S4 → S3 → S2 → S1 → S0.

Note that the number of states can be edited on the **State** tab.

For properties not explained in this section, please see the section [Specifying Object Properties](#).

Example of designing a Multistate Button

On the **Attributes** tab of the **Multistate Button**:

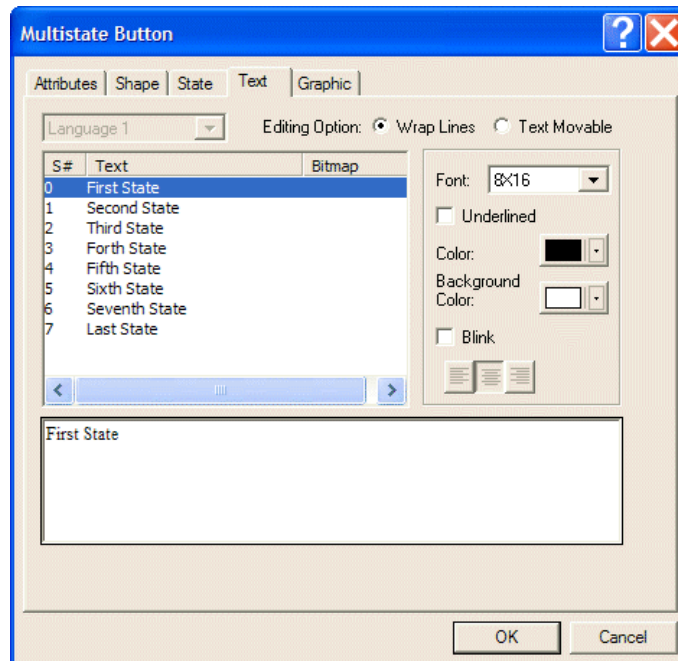
1. **Write:** Specify controller register **W50** to write in.
Read: Specify **W50**. (The controller model is NULL.)
2. **Format:** **Value**.
3. **Function:** **Change to Previous State**.
4. **Shape:** Select **Outlined _1** and the color **Black**.

On the **State** tab of the **Multistate Button**:

5. Add 8 States.

On the **Text** tab of the **Multistate Button**:

6. Enter the corresponding texts.



Texts for the 8 different states of the Multistate Button in the example

The above steps will create a **Multistate Button** that displays **First State** in state 0; **Second State** in state 1 etc.

Set Value Button

The function of the **Set Value Button** is to enable numeric entry. When pressed, the operator terminal displays a numeric keypad on the screen. When pressing ENT, the operator terminal will store the input value to the corresponding controller register.

Note that the corresponding controller value is not available in the **Set Value Button**.

Attributes Tab

Variable:

- **Word:** The entered value is 16-bit data; the maximum 65,535.
- **Double Word:** The entered value is 32-bit data; the maximum is 4,294,967,295.
- **Format:** BCD, Signed Binary, Unsigned Binary and Hexadecimal can be selected.
- **Notification:** Specify a register/bit location to be notified; the operator terminal will set the bit to ON.
 Select **Before Writing** to make the operator terminal set the notification to ON when the numeric keypad appears, and set the location to OFF when the numeric keypad disappears.
 Select **After Writing** to make the operator terminal set the notification location to ON after writing the input value to the **Write** location.

Display Format:

- **Decimal Pt. Position:** Specifies the number of digits after the integral part of the number. The maximum is based on the specified format.
- **Integral Digits:** The number of the integral part in a number.
- **Fractional Digits:** The number of decimal digits.
- **Display Asterisk Instead of Number:** Displays an asterisk instead of input value for security reasons.

Validation and Security:

- **Input Min.:** Sets the minimum input value. A number less than the minimum input value will be warned and rejected.
- **Input Max.:** Sets the maximum input value. A number greater than the maximum will be warned and rejected.
- **User Level:** There are 9 user levels, the order is 1 > 2 ... > 8 > 9.
- **Operation Confirmation:** Check this box to display a dialog box on the screen asking for **User Confirmation** during a waiting time of 5-60 seconds.

For properties not explained in this section, please see the section [Specifying Object Properties](#).

Example of designing a Set Value Button

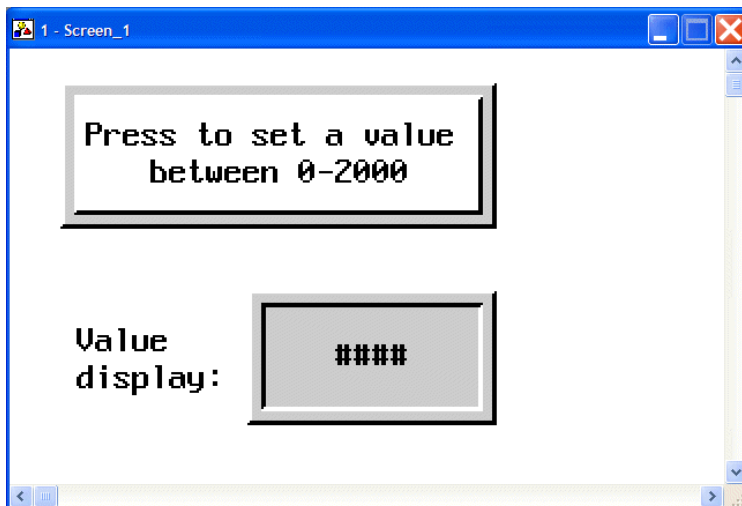
On the **Attributes** tab of the **Set Value Button**:

1. **Shape:** Select **RaisedBase**.
2. **Write:** Specify controller register **W100** to write in. **Numeric Entry:** **Word Notification:** Specify **B10** and **After Writing**. (The controller model is NULL.)
3. **Display Format:** Check **Display Asterisks Instead of Number**.
4. **Validation and Security:** Select **0** for **Input Min.**, and **2000** for **Input Max**. Check **Operator Confirmation** to require a confirmation after numeric entry.

On the **Text** tab of the **Set Value Button**:

5. Enter the desired text, for example **Press to set a value between 0-2000**, and select **White** for background color.

The steps above will create a **Set Value Button**. When the button is pressed, the numeric keypad will be displayed on the screen. After numeric entry, the input value cannot be displayed directly on the **Set Value Button**. Instead, you can create a **Numeric Display** object to display the input value. When the value is changed, B10 is set.



The Set Value Button and Numeric Display object

Set Constant Button

The **Set Constant Button** is used to make the operator terminal write a constant to the corresponding controller register. The numeric keypad will not be displayed on the screen since the constant has been set in the controller.

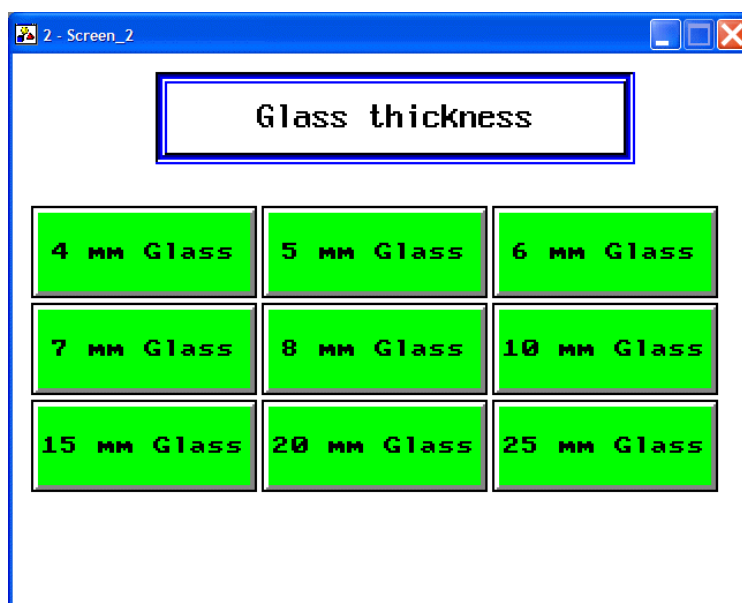
Attributes Tab

Value: Specifies the constant value.

For properties not explained in this section, please see the section [Specifying Object Properties](#).

Example of designing a Set Constant Button

Here we use a glass list as an example. When one of the **Set Constant Buttons** is pressed, the operator terminal will write the specified constant value to the corresponding controller register. In this example, **4 mm Glass** represents the constant value **400**, **5 mm Glass** represents the constant value **500** etc.



The Set Constant Buttons - glass list example

On the **Attributes** tab of the **Set Constant Button**:

1. **Variable:**
Write: Specify controller register **W10**.
Notification: Specify **B10**. (The controller model is NULL.)
2. **Set Value:** Select **Word** for numeric entry.
Value: Specify the constant value **400** for the **4 mm Glass** button.

On the **Text** tab of the **Set Constant Button**:

3. Enter the text **4 mm Glass** and select **Green** for background color.

The above steps will create a **4 mm Glass** button. When clicked, the operator terminal will store the constant value **400** to the register **W10**.

Follow the same steps to create other glass buttons, using the corresponding constant values.

Increment Button

The **Increment Button** is used to make the operator terminal read a constant variable stored in a controller register. Then a specified constant will be added to the value, before writing it back to the controller register.

Attributes Tab

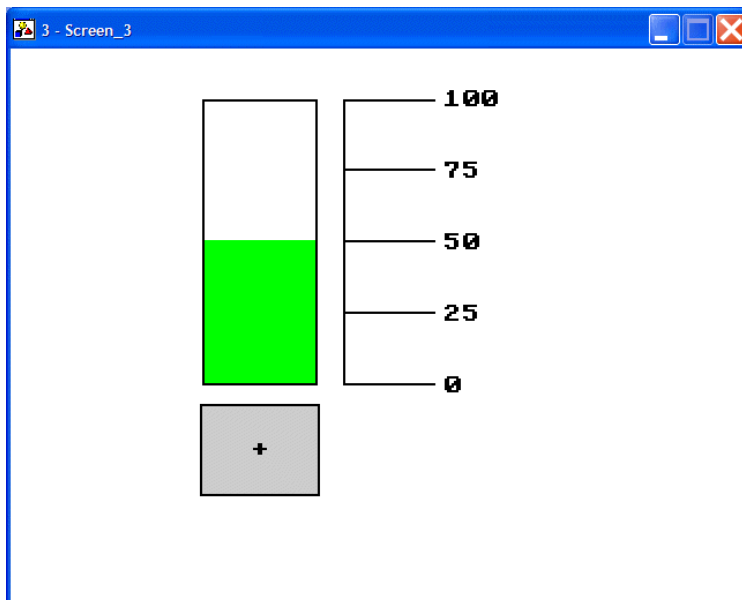
Function:

- **Increment:** Select **Increment** to create an **Increment Button**; one click increases a specified constant.
- **Jog Step:** The **Increment Button** is used to add a specified constant by clicking.
- **Limit:** Specifies the maximum value written to the register when using the **Increment Button**.

For properties not explained in this section, please see the section [Specifying Object Properties](#).

Example of designing an Increment Button

When clicking the **Increment Button**, the increased value stored in the controller register will be displayed in the bar graph. The **Bar Graph** is an object used to display the dynamic data.

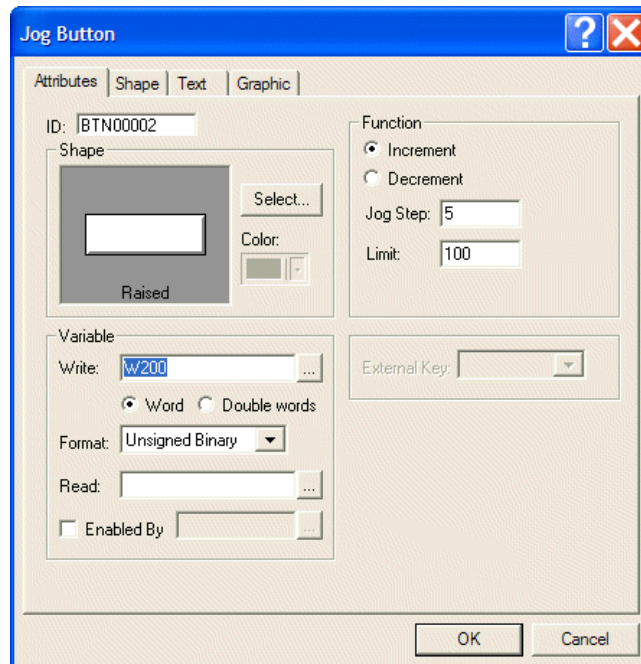


*When the **Increment Button** is clicked, the level in the **Bar Graph** is increased*

On the **Attributes** tab of the **Increment Button**:

1. **Variable:**
Write: Specify controller register **W200** to write in.
Read: Specify **W200**. (The controller model is NULL.)
2. **Shape:** Select **Raised**.

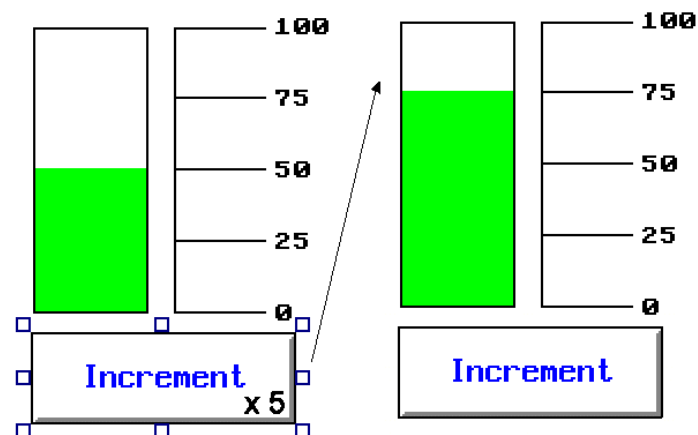
3. **Function:** Select **Increment**, **Jog Step:** 5 and **Limit:** 100.



Attribute settings for the Increment Button

On the **Text** tab of the **Increment Button**:

4. Enter the desired text, for example **Increment**, and select **White** for background color.



*Clicking the **Increment Button** five times adds 25 to the controller constant W200*

Decrement Button

The Decrement Button is used to make the operator terminal read a constant variable stored in a controller register. Then a specified constant will be subtracted from the value, before writing it back to the controller register.

Attributes Tab

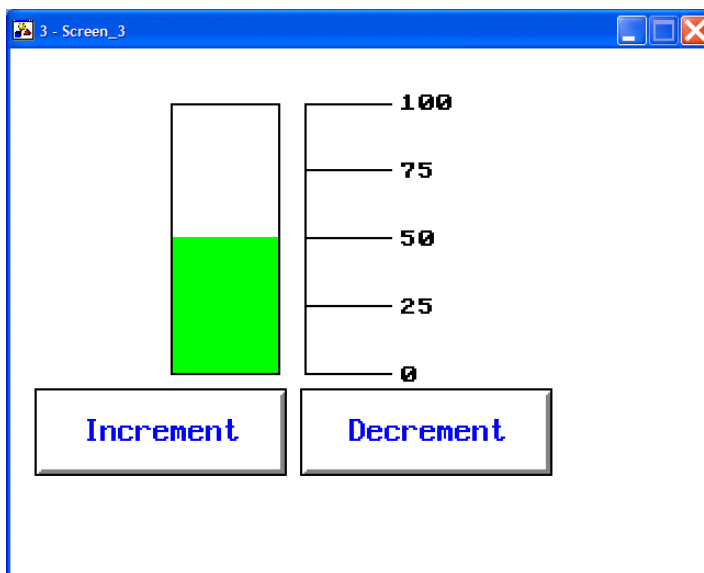
Function:

- **Decrement:** Select **Decrement** to create a **Decrement Button**; one click is one subtraction.
- **Jog Step:** The **Decrement Button** is used to subtract a specified constant by clicking.
- **Limit:** Specifies the minimum value written to the register when using the **Decrement Button**.

For properties not explained in this section, please see the section [Specifying Object Properties](#).

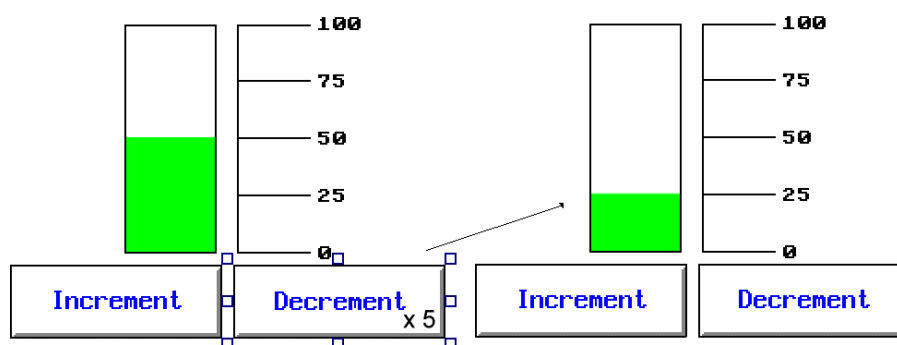
Example of designing a Decrement Button

When clicking the **Decrement Button**, the subtracted constant value stored in the controller register will be displayed in the bar graph. The **Bar Graph** is an object used to display the dynamic data..



*When the **Decrement Button** is clicked, the level in the **Bar Graph** is decreased*

The steps used to create a **Decrement Button** are the same as in section [Example of designing an Increment Button](#), but remember to change **Limit** to 0.



*Clicking the **Decrement Button** five times subtracts 25 from the controller constant*

Goto Screen Button

The **Goto Screen Button** is used to change the current screen to another screen.

Attributes Tab

Function:

- **Open/Go To:** Check this option to create a **Goto Screen Button**. You can select which screen to open from the drop-down list.
- **Enabled By:** Only change the screen when the controller register is ON.

Execution:

- **On Press:** Executes the command (changes the screen) when button is pressed.
- **On Release:** Executes the command (changes the screen) when button is released.

Appended Functions:

- **Change to the Lowest User Level:** Sets the current user level as the lowest level (User Level 9).
- **Acknowledge Alarm:** Acknowledges the current active alarm when the button is clicked.
- **Notify:** Notifies the specified bit-location after clicking the button.

Security:

- **User Level:** There are 9 user levels, the order is 1 > 2 ... > 8 > 9.

For properties not explained in this section, please see the section [Specifying Object Properties](#).

Example of designing a Goto Screen Button

In this example, clicking the **Goto Screen Button** will open **Screen_3**, assuming that the **Screen_3** has been created in the project.

On the **Attributes** tab of the **Goto Screen Button**:

1. **Shape:** Select **Raised**.
2. **Function:** Select **Screen_3** from the drop-down list.
3. **Execution:** Select **On Press**.

On the **Text** tab of the **Goto Screen Button**:

4. Enter the desired text, for example **Go to Screen 3**.

Previous Screen Button

The **Previous Screen Button** is used to return to the previous screen in the operator terminal.

Attributes Tab

Function:

- **Close/Previous:** Check this option to create a **Previous Screen Button**. You can select which screen to open from the drop-down list.

For properties not explained in this section, please see the sections [Goto Screen Button](#) and [Specifying Object Properties](#).

Example of designing a Previous Screen Button

The steps used to create a **Previous Screen Button** are the same as in section [Example of designing a Goto Screen Button](#), but remember to check the **Close/Previous** option for **Function**.

Action Button

The Action button is used to perform a built-in function.

The following actions are available:

Action	Description
Contrast Up	Increases the contrast or brightness of the display.
Contrast Down	Decreases the contrast or brightness of the display.
Save contrast	Saves the setting of contrast or brightness.
Password Table	Displays the password table.
Reenter Password	Displays the password table to reenter.
Set Lowest User Level	Changes to the lowest user level (level 9).
Print Screen	Prints the specified region (HARDCOPY) of current screen.
Goto System Menu	Changes to the system menu.
Turn off Backlight	Turns off the backlight.
Alarm Ack	Acknowledges the current active alarm to continue.
Set Time & Date	Sets the time and date.
Select Language #1-#5	Displays the screen in the specified language, 5 languages available for selection.

All features are not available on all operator terminal models; please refer to [Appendix A - H-Designer Features and Operator Terminal Models](#).

For properties not explained in this section, please see the section [Specifying Object Properties](#).

Example of designing an Action Button

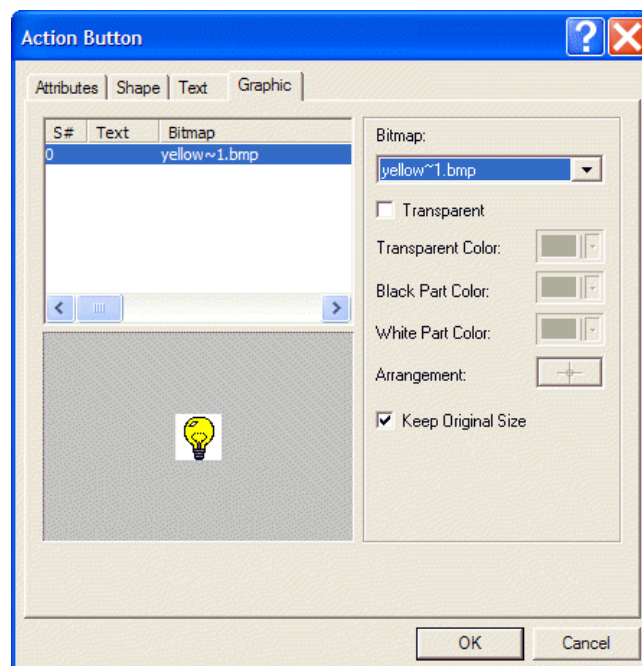
In this example, the Action Button will be used to change the contrast of the display.

On the **Attributes** tab of the Action Button:

1. **Action:** Select **Contrast Up** from the drop-down list.

On the **Graphic** tab of the Action Button:

2. **Bitmap:** Select **Symbol - ContrastUp**.



Selecting a symbol for the Action Button

2.7.4 Numeric Entry

The **Numeric Entry** is used to write an input value to a controller register, and to display the value on the operator terminal screen. When the button is clicked, a numeric keypad will be displayed on the screen. Enter a value and then press **ENT** on the keypad. The operator terminal will then write the input value to the specified controller register.

Variable

Write: Specifies the controller register to write to.

Format: BCD, Signed Binary, Unsigned Binary, Hexadecimal, 32-bit Floating-point and Octal can be selected.

Notification: Specifies a register/bit location to be notified; the operator terminal will set the bit to ON.

- **Before Writing:** The operator terminal sets the notification location to ON when the numeric keypad appears and sets the location to OFF when the numeric pad disappears.
- **After Writing:** The operator terminal sets the notification location to ON after writing the input value to the write location.

Display Format

- **Fill Leading Zeroes:** Select this option to add leading zeros; for example, 5902.1 is displayed as 005902.1.
- **Decimal Pt. Position:** Specifies the number of digits after the integral part of the number. There are 0-10 digits for selection.
- **Fractional Digits:** The number of decimal digits.

(Fractional Digits + Integral Digits or Decimal Pt. Position ≤ the maximum number of digits.)

- **Integral Digits:** The number of the integral part in a number.

(Fractional Digits + Integral Digits or Decimal Pt. Position ≤ the maximum number of digits.)

- **Scaling:** The formula is $Y = aX + b$.
Note that only the formats Signed Binary, Unsigned Binary and 32-bit Floating-point support this option.

Gain: $Y = aX$, where X = the value stored in controller and Y = operator terminal displayed value.

Offset: If the initial value is not zero, then set the Offset.

Validation and Security

- **Variable Input Limits:** Set the input limits as variable. The minimum is stored in the bit following the write location; the maximum is stored in the bit following the minimum input value.
For example, if the write location is **W10**, then the minimum is stored in **W11**; the maximum is stored in **W12**.
- **Min:** Set the minimum input value. Values less than the minimum input value will be warned and rejected.
- **Max:** Set the maximum input value. Values greater than the maximum will be warned and rejected.

For properties not explained in this section, please see the sections [Set Value Button](#) and [Specifying Object Properties](#).

Example of designing a basic Numeric Entry button

Perform the following steps to create a basic Numeric Entry button:



A basic Numeric Entry button

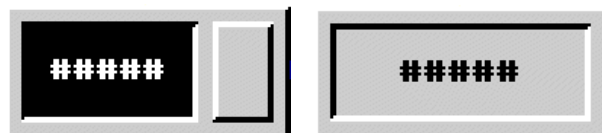
1. **Frame:** Select **DE_Module_2** and **Black** for background color.
2. **Variable:** Specify the controller register **W100** for **Write** to store the value. (The controller model is NULL.) Select **Unsigned Binary** for **Format**.
3. **Display Format:** Select **White** for character color. Allow **5 Digits** in a number.

Validation and Security: Set the minimum input value to **0** and the maximum input value to **60000**.

Numeric Entry properties in this example

The steps above will create a **Numeric Entry** button. When the button is clicked, a numeric keypad will be displayed on the screen. After entering a value, the operator terminal will show the input value on the button.

It is also possible to create a **Numeric Display** object to display the value stored in the controller. Therefore, for this basic numeric entry button, if one enters **10** on the operator terminal, then both the **Numeric Entry** button and the **Numeric Display** object will show 10.



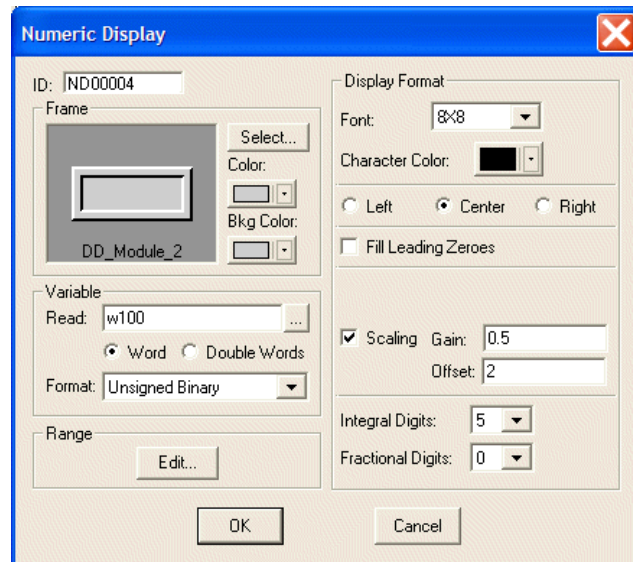
A basic Numeric Entry button and a Numeric Display object

Example of designing a Numeric Entry button with scaling feature

Perform the following steps to create a **Numeric Entry** button with the **Scaling** feature.

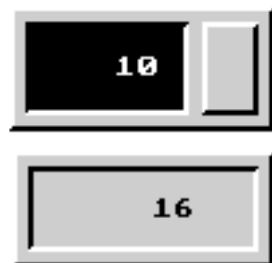
1. **Display Format:** Check **Scaling**. Select **Gain: 0.5** and **Offset: 2**.

All other properties are the same as in the previous example.



Numeric Entry properties in this example

After entering a value, the operator terminal will show the input value on the button. It is also possible to create a **Numeric Display** object to display the value stored in the controller. Therefore, for this example with a numeric entry button, if one enters **10** on the operator terminal, then the **Numeric Entry** button will show 10 and the **Numeric Display** object will show 16.



A Numeric Entry button with Scaling feature and a Numeric Display button

($Y = aX + b$; X is the value stored in the controller, Y is the input value on an operator terminal; where $a = 0.5$ and $b = 2$ here)

2.7.5 Character Entry

The function of the **Character Entry** is used to provide users with alphabetic input and display.

When the button is clicked, an alphabetic keypad will be displayed on the screen. Enter character(s) and then press **ENT** on the keypad. The operator terminal will then write the input entry in ASCII to the specified controller register.

The object is not available on all operator terminal models; please see [Appendix A - H-Designer Features and Operator Terminal Models](#) for complete details.

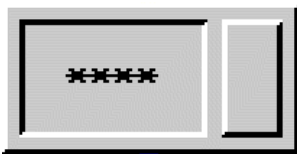
Variable

Number of Characters: Specifies the number of characters; the maximum is 28.
(2 words in ASCII = 1 word in a controller register)

For properties not explained in this section, please see the sections [Set Value Button](#) and [Specifying Object Properties](#).

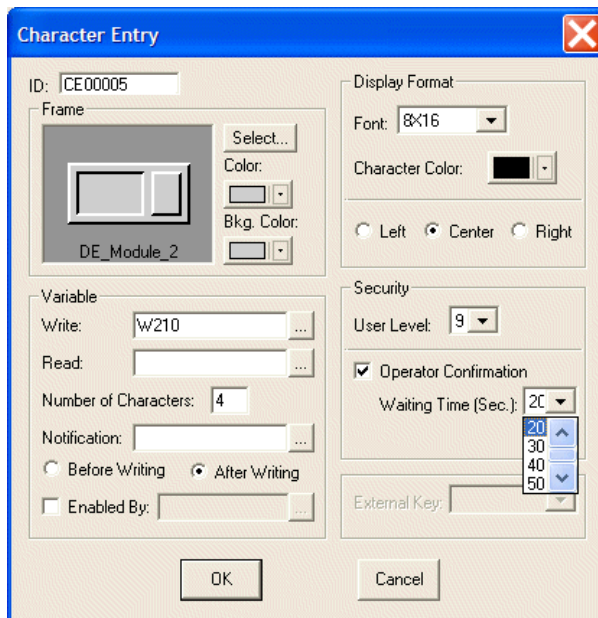
Example of designing a Character Entry button:

Perform the following steps to create a basic **Character Entry** button:



A Character Entry button

1. **Variable:** Specify the controller register **W210** for **Write** to store the value. (The controller model is **NULL**.) Select **4** for **Number of Characters**.
2. **Security:** Set the **Waiting Time (Sec.)** to **20** seconds.



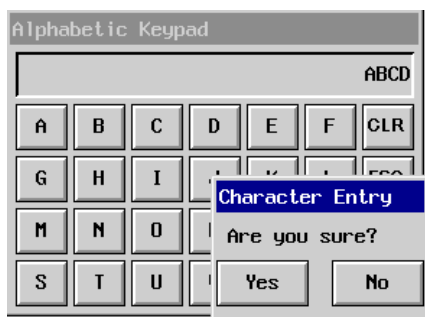
Character Entry properties in this example

The steps above will create a **Character Entry** button. When the button is clicked, an alphabetic keypad will be displayed on the screen.

Note:

The **Alt**-key on the keypad is used for **Shift** functionality.

After entering the characters, press ENT on the keypad. A dialog box asking for user confirmation appears on the screen.



Using the Character Entry button with user confirmation

2.7.6 List

Each item in the **List** object corresponds to a designated register value in the controller. The first item represents the register value as 0. The second item represents the register value as 1, and so on. When the user chooses one of the items in the list, the operator terminal will store the corresponding value in the controller register.

The corresponding item will be highlighted in the **List** object. Furthermore, you can change the value of a controller register by making a selection from the **List** object.

The object is not available on all operator terminal models; please see [Appendix A - H-Designer Features and Operator Terminal Models](#) for complete details.

Variable

Read Only: For display purposes. Users are not able to make a selection from the list.

Write: Writes the value to the specified controller register.

Type of State:

- **Value:** There are 256 states (0-255). The value of 0 represents state 0; the value of 1 represents state 1, and so on.
- **LSB:** There are 16 states. If more than 2 bits are to be ON, the controller register will store the value of the lower bit.

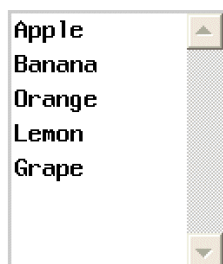
Format: This is only available when the **Value** option is selected. There are three selections: **BCD**, **Signed Binary** and **Unsigned Binary**.

Read: Reads the value from the specified controller register. If the location is not specified, then the operator terminal reads from the **Write** location.

For properties not explained in this section, please see the section [Specifying Object Properties](#).

Example of designing a List object

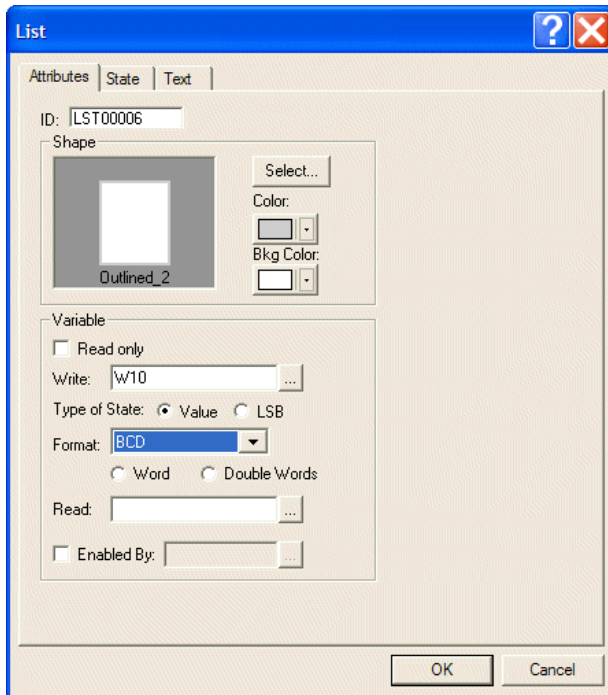
Perform the following steps to create a **List** object:



A List object

On the **Attributes** tab of the **List** object:

1. **Shape:** Select **Outlined_2** and **White** for background color.
2. **Variable:** Specify the controller register **W10** for **Write** to store the value. (The controller model is NULL.) Select **Value** for **Type of State** and **BCD** for **Format**.



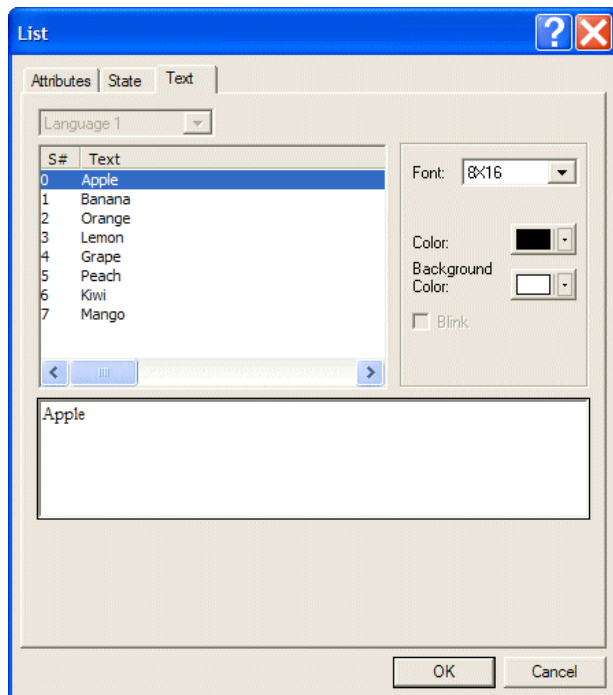
List properties in this example

On the **State** tab of the **List** object:

3. Add new states. There are 8 states in this object.

On the **Text** tab of the **List** object:

4. Enter the text and set up the format of the display.



Adding one text for each state on the Text tab of the List object

The steps above will create a **List** object. When an item from the **List** object is selected, the operator terminal will write the value associated with the item to the specified controller register. In this example, if the item **Peach** is selected, the value of the controller register will be 5.



The List object and the display of the controller register value of the item

2.7.7 Drop Down List

Each of the items in the **Drop Down List** object corresponds to a value of a controller register. Therefore, for a **Drop Down List** object, the value associated with the displayed item is the current value of the controller register.

Click the object to display the list. A list of items to choose from is dropped down. You can also change the value of a controller register by making a selection from the **Drop Down List** object. Once a selection is made from the object, the list will disappear.

The object is not available on all operator terminal models; please see [Appendix A - H-Designer Features and Operator Terminal Models](#) for complete details.

The properties of the **Drop Down List** object are similar to those of the **List** object; please see the sections [List](#) and [Specifying Object Properties](#).

Example of designing a Drop Down List object:

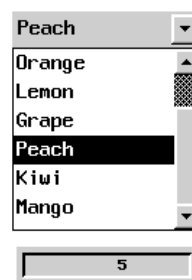
The steps used to create a **Drop Down List** object are similar to those for a **List** object. Please see the [Example of designing a List object](#) for the complete details. Remember to adjust the length of a list accordingly so that you are able to display all items of the list.



A Drop Down List object

The example below shows a **Drop Down List** object in the operator terminal. Click the object to display the list. The object will then drop down a list of items from which you can choose. When you select an item from the list, the operator terminal will write the value associated with the item to the controller register. Therefore, if the item **Peach** is selected, then the value of the controller register will be 5.

Notice that once a selection is made from the object, the list will disappear.



The Drop Down List object and the display of the register value of the item

2.7.8 Indicators

There are two types of indicators; the **Multistate Indicator** and the **Range Indicator**.

Multistate Indicator

The **Multistate Indicator** is used to indicate which state exists with text and/or graphics. Therefore, as the operator terminal reads the contact status or the register value from the controller, it can automatically display the corresponding designed content on the operator terminal screen according to the indicator.

The number of states is as follows:

1. **Bit:** The maximum number of states is 2
2. **Value:** The maximum number of states is 256
3. **LSB:** The maximum number of states is 16

Variable:

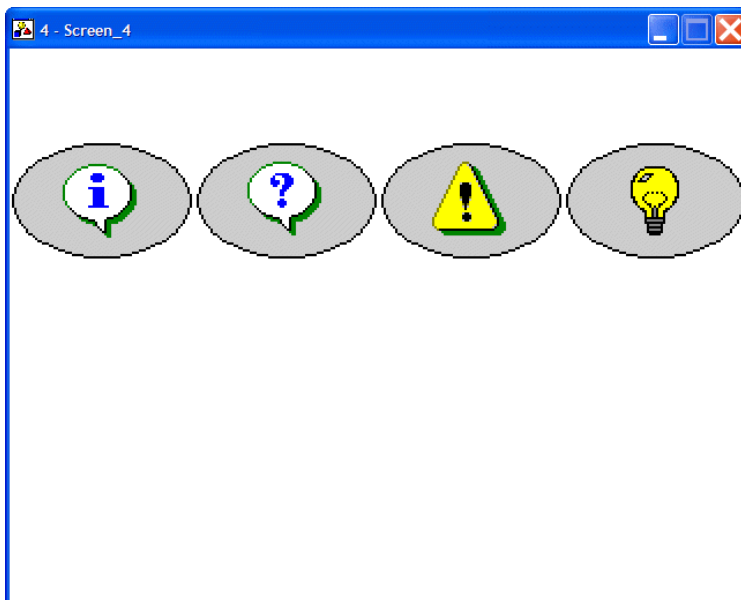
Read: Specifies the register/bit location.

- **Bit:** Two states in all. (Users can input more than two states but only two states will be displayed in the operator terminal.)
- **Value:** 256 states (0-255) in all ; value 0 represents state 0; value 1 represents state 1; value 2 represents state 2, and so on.
- **LSB:** 16 states in all; the operator terminal takes the bit number of the smallest bit that is on as the state number.

Format: Specifies the data format. There are three options: **BCD**, **Signed Binary**, and **Unsigned Binary**.

For properties not explained in this section, please see the section [Specifying Object Properties](#).

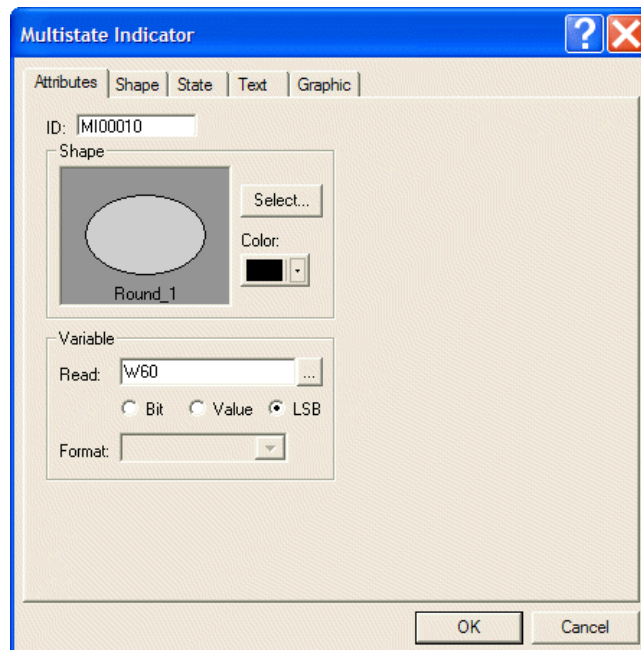
Example of designing a Multistate Indicator object



The Multistate Indicator object

On the **Attributes** tab of the **Multistate Indicator** object:

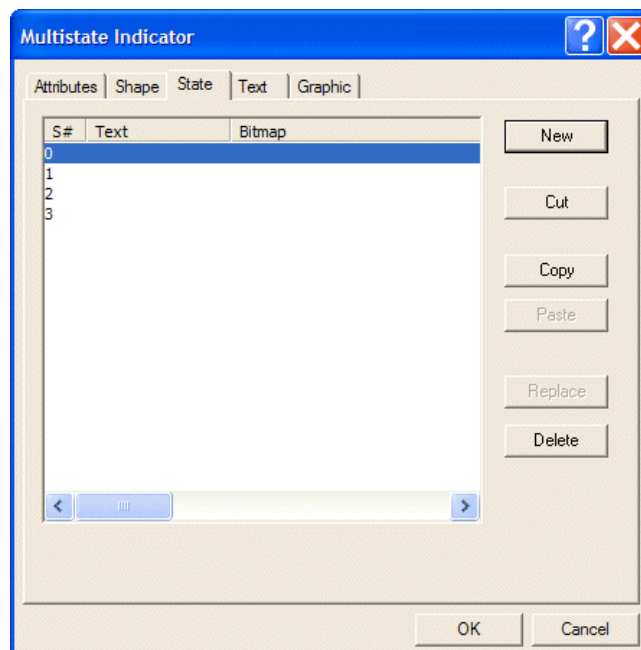
1. **Shape:** Select **Round_1** and **Black** for border color.
2. **Variable:** Specify controller register **W60** to read from. (The controller model is **NULL**). Select **LSB** for **Format**.



The Multistate Indicator attributes in this example

On the **State** tab of the **Multistate Indicator** object:

3. Add new states. There are 4 states in this object.



Adding states for the Multistate Indicator

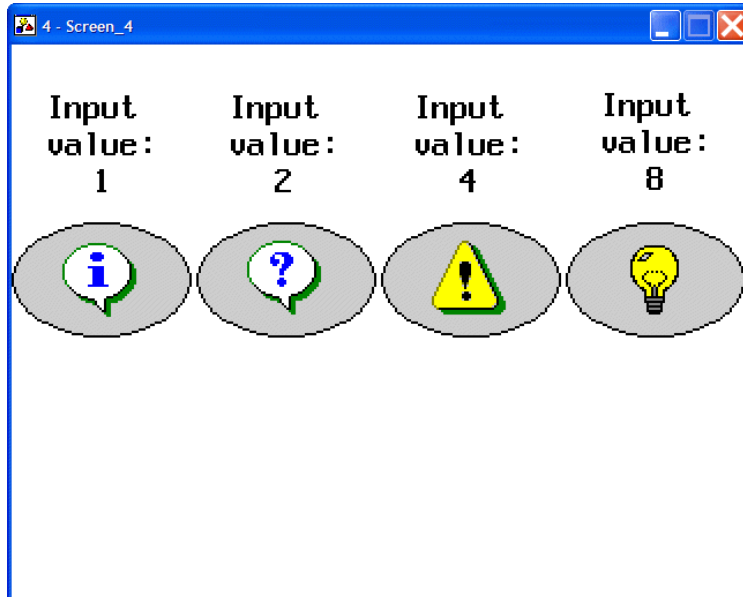
On the **Graphic** tab of the **Multistate Indicator** object:

4. Select bitmaps for the different states. This example does not display text, but graphics. These graphics are available in the **SYMBOLS.GBF** library.

On the **Text** tab of the **Multistate Indicator** object:

5. Select **White** for background color.





The above-mentioned steps will create a **Multistate Indicator** object. In this example, **Numeric Entry** buttons are created for the numeric entry written in the controller register. The **Multistate Indicator** objects will display the corresponding states according to the register value.



*The **Multistate Indicator** object displays the corresponding graphic*

Therefore, for this example of the object, if one enters 1 in the operator terminal, the **Multistate Indicator** object will show state 0; if one enters 4, the object will show state 2; if one enters 8, the object will show state 3.

The **Format** selected in this example is **LSB**; please refer to the following table:

Numeric Entry (LSB)	Bit State	Multistate Indicator	Graphics
1	0 bit is ON; the others are OFF	State 0	
2	1 bit is ON; the others are OFF	State 1	
4	2 bit is ON; the others are OFF	State 2	
8	3 bit is ON; the others are OFF	State 3	

Range Indicator

A **Range Indicator** displays one of several indicator labels depending on the register value. The operator terminal reads register values from the controller and automatically calculates the difference according to the boundary value of current states. Then the contents of current status are displayed on the operator terminal screen according to the calculated results.

Read value from controller → Calculated result → Display the corresponding states

Ranges:

Variable Limits: Specifies the minimum value of the ranges to be read from registers following the read location. If the **Read** address is **W_n**, the minimum value of Range # 0 is stored in **W_{n+1}**, the minimum of Range # 1 is stored in **W_{n+2}**, and so on.

Constant Limits: The minimum of the ranges is constant.

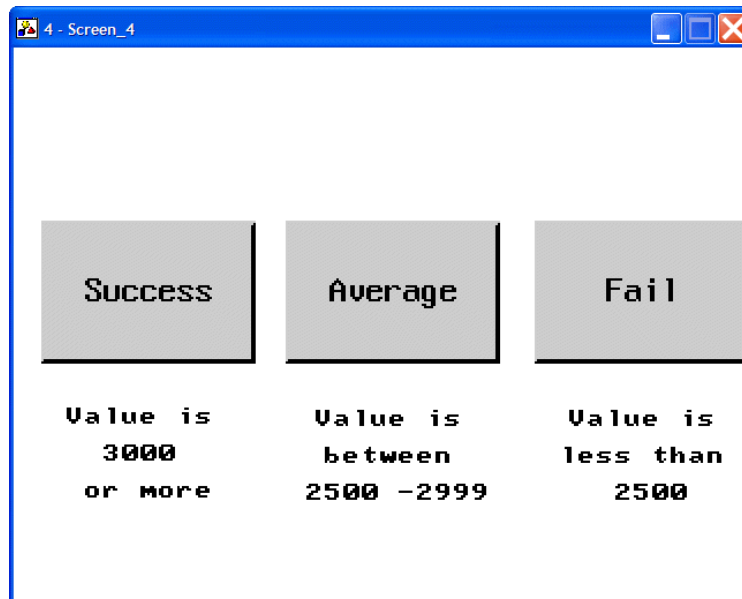
- **Range #:** The number of ranges, 15 ranges at most.
- **Minimum:** The minimum of ranges.

Note:

Number of ranges = number of states - 1.

For properties not explained in this section, please see the section [Specifying Object Properties](#).

Example of designing a Range Indicator object

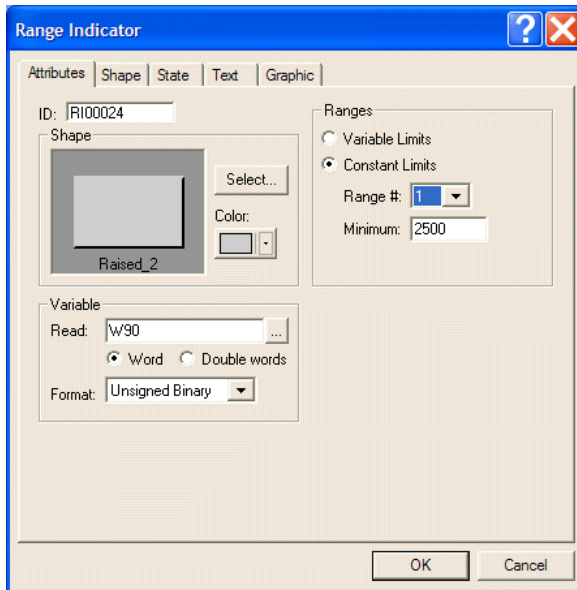


The Range Indicator object

On the **Attributes** tab of the **Range Indicator** object:

1. **Shape:** Select **Raised_2**.
2. **Variable:** Specify controller register **W90** to read from. (The controller model is **NULL**).

3. **Ranges:** Select **Constant Limits** and minimum 3000 of **Range # 0**; minimum 2500 of **Range # 1**.



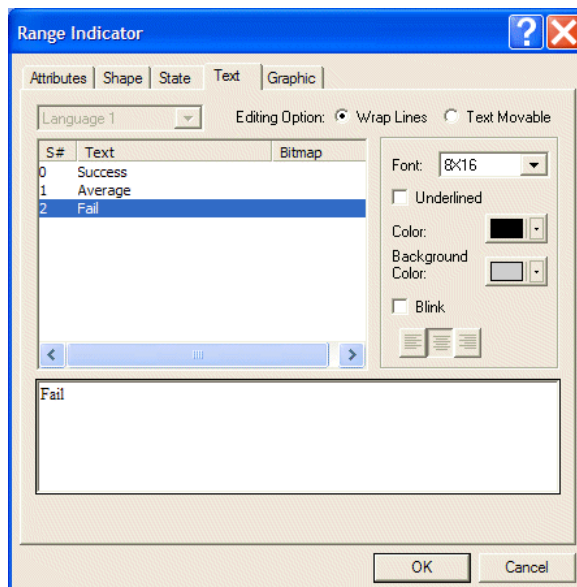
The Attributes tab of the Range Indicator in this example

On the **State** tab of the **Range Indicator** object:

4. Add new states. There are 3 states in this object.

On the **Text** tab of the **Multistate Indicator** object:

5. Enter the texts for the states.



Entering texts for the different states

The above-mentioned steps will create a **Range Indicator** object. In this example, a **Numeric Entry** button is used to input value in the controller register, and a **Range Indicator** object is used to calculate the result and display its corresponding state associated with the specified range.



The Range Indicator object displays the corresponding state

If you enter the input value 3500, the corresponding range is **Range # 0**. Therefore, the **Range Indicator** object will show **Success**.

2.7.9 Numeric Display

The **Numeric Display** object is used to display the register value stored in controller. This object does not support click-button functionality.

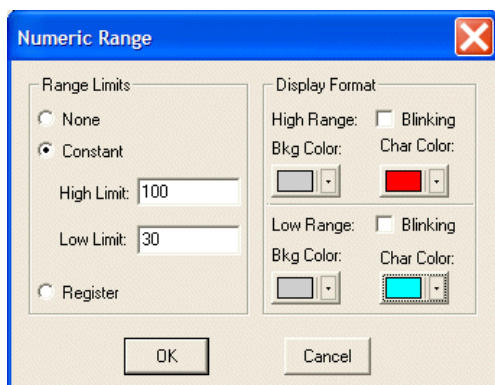
Variable

Read: Specifies a register/bit location.

Format: There are BCD, Signed Binary, Unsigned Binary, Hexadecimal, 32-bit Floating-point and Octal formats.

Range

Edit button: When clicking this button, the displayed dialog box provides a display showing high/low range.



Editing Numeric Range

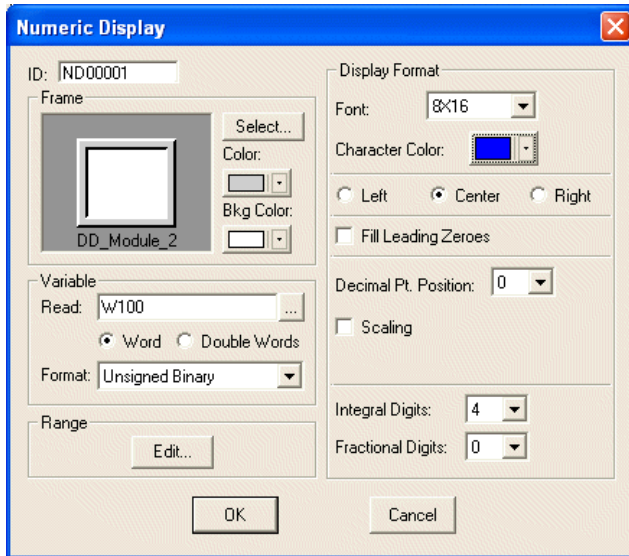
- **None:** No high/low limit.
- **Constant:** Enter the constant variable of high/low limit.
- **Register:** Read high/low limit from register. If the read location is W_n , the high limit is stored in W_{n+1} and the low limit is stored in W_{n+2} .
- **Display Format:** Specifies the format to display when the variable is equal or more/less than high/low limit.

For properties not explained in this section, please see the sections [Set Value Button](#) and [Specifying Object Properties](#).

Example of designing a Numeric Display object

1. **Frame:** Select DD_Module_2 and White for background.
2. **Variable:** Specify controller register W100 to read from. (The controller model is NULL)

3. **Display Format:** Select **Blue** for character color.



Numeric Display properties in this example

4. **Range:** Click the **Edit** button and select **Constant**. Specify **100** for **High Limit** and **30** for **Low Limit**. Select **Red** for character color for **High Range**, and **Light Blue** for character color for **Low Range**.



Numeric Entry buttons are used to influence Numeric Display objects.

In this example, a **Numeric Entry** button is used to input a value in the controller register. The variable will display different text colors on the screen according to its range: If the variable is less than or equal to **30**, it shows **Light Blue** text; if the variable is greater than or equal to **100**, it shows **Red** text ; if the variable is between **30** and **100**, it shows the original setting **Blue** text.

2.7.10 Character Display

The **Character Display** object is used to provide an alphanumeric display for an ASCII variable in the controller register. Note that it does not support click-button functionality.

Variable

Number of Characters: Specifies the number of characters to display. It can have up to 28 characters, limited by the width of the object.

For properties not explained in this section, please see the section [Specifying Object Properties](#).

Example of designing a Character Display object

The following steps are used to create a **Character Display** object; this function can be applied in the factory for product management.

Bar code reader
ASCII result



A Character Display object

1. **Frame:** Select **Outlined_2** , **Blue** for border color and **White** for background.
2. **Variable:** Specify controller register **W20** to read from. (The controller model is **NULL**). Specify **10** for **Number of Characters**.
3. **Display Format:** Select **Dark Blue** for character color.

2.7.11 Message Display Objects

There are six types of Message Display objects: **Prestored Message**, **Moving Sign**, **Data Terminal**, **Time Display**, **Data Display** and **Day-of-Week Display**.

Note:

Message Display objects contain only text; **Indicator Buttons** can have both text and graphic, however.

Prestored Message Display

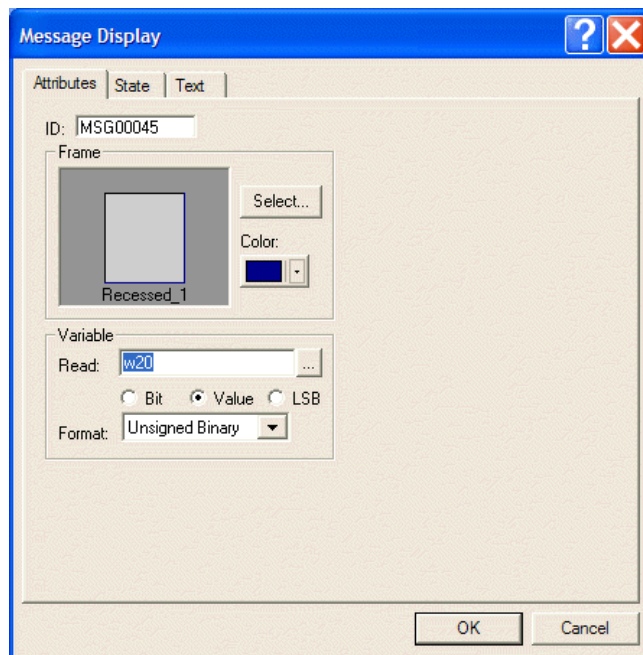
The **Prestored Message Display** object is used to make the operator terminal read the contact state (ON/OFF) or the register value from the controller and automatically display designed content on the operator terminal screen according to the state/value.

Please see the section [Multistate Indicator](#) for complete details.

Example of designing a Prestored Message Display object

On the **Attributes** tab of the Message Display object:

1. **Frame:** Select **Recessed_1** and **Dark Blue** for border color.
2. **Variable:** Specify **W20** to read from.



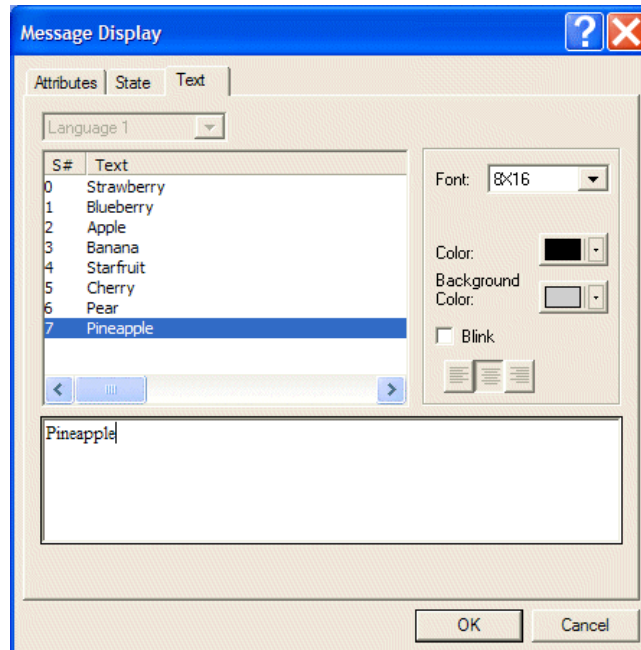
Setting attributes for the Message Display object in this example

On the **State** tab of the Message Display object:

3. Add 8 new states.

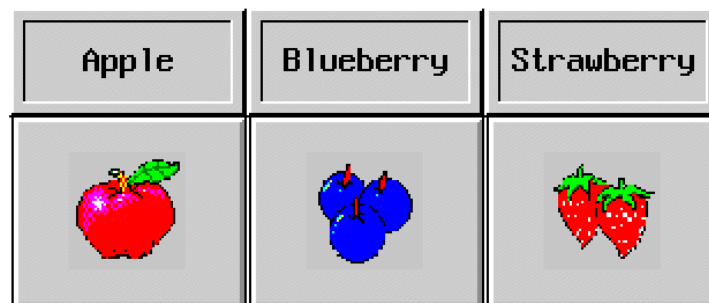
On the **Text** tab of the **Message Display** object:

4. Enter the desired text and specify the text format.



Adding texts for the 8 states in this example

The above-mentioned steps will create a **Prestored Message Display** object. When the **Multistate** button (under the **Prestored Message Display** object) is clicked once, the operator terminal writes the command to the controller for state change. When the state is changed, the **Prestored Message Display** will display the corresponding state.



Prestored Message Display objects used together with Multistate buttons

For instance, when the state is **Blueberry**; the **Prestored Message Display** object will display the corresponding content **Blueberry**. When the state is **Starfruit**, the **Prestored Message Display** object will display the corresponding content **Starfruit**.

Moving Sign

The **Moving Sign** object is used to display content one by one, from right to left.

When the operator terminal reads the value from a bit-location (ON/OFF) or register in the controller, the **Moving Sign** object will display its contents or message according to the corresponding state on the screen.

Speed

Number of Characters Per Shift: Specifies the number of characters per shift.

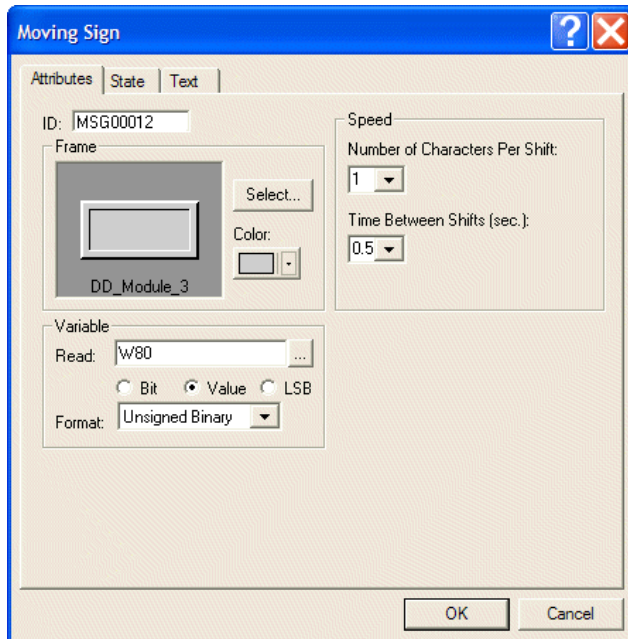
Time Between Shifts (sec.): Specifies the time between shifts in seconds.

For properties not explained in this section, please see the section [Specifying Object Properties](#).

Example of designing a Moving Sign object

On the **Attributes** tab of the **Moving Sign** object:

1. **Frame:** Select **DD_Module_3**.
2. **Variable:** Specify **W80** for **Read**. (The controller model is **NULL**). Select **Value** and **Unsigned Binary** format.
3. **Speed:** Select **1** for **Number of Characters Per Shift** and **0,5** seconds for **Time Between Shifts**.



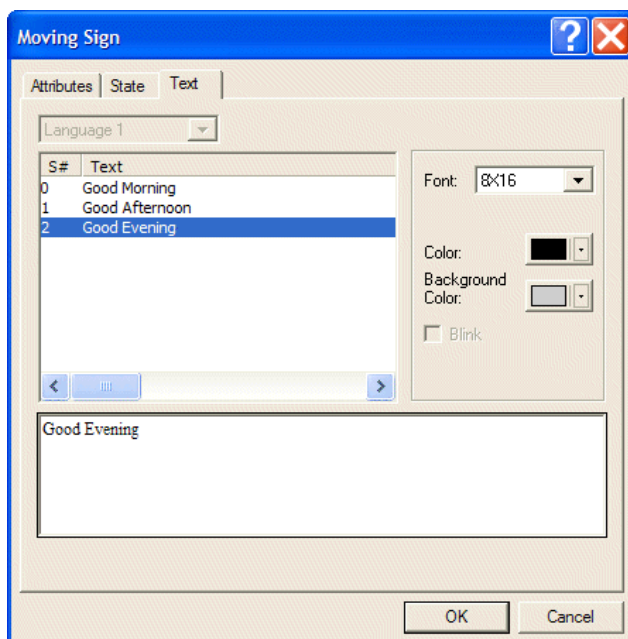
*Setting attributes for the **Moving Sign** object in this example*

On the **State** tab of the **Moving Sign** object:

4. 3 states are used in this example.

On the **Text** tab of the **Moving Sign** object:

5. Enter the desired text and specify the text format for the states.



Adding texts for the 3 states in this example

The above-mentioned steps will create a **Moving Sign** object.



*A **Moving Sign** object used together with a **Multistate** button*

The **Multistate** button is designed to write the command to the controller when clicked. Then, the **Moving Sign** object will display the corresponding content according to the current state.

For example, if the state is **Morning**, the **Moving Sign** object will display **Good Morning Good Morning Good Morning**. The text here is displayed revolving, moving through characters from right to the left.

Data Terminal

The **Data Terminal** object is used to simulate an ASCII terminal. The operator terminal can be connected with another specified communication port and the specified communication parameters to the data terminal displayed using ASCII /HEX mode.

Remember to set the ASCII Device to communicate and specify the communication port.

The object is not available on all operator terminal models; please see [Appendix A - H-Designer Features and Operator Terminal Models](#) for complete details.

Variable

Read: Specifies a bit-location to read from. The ASCII Device provides RX, RXSTS, TX and TXSTS contacts.

Display

Mode: Displays the terminal data in ASCII/HEX mode.

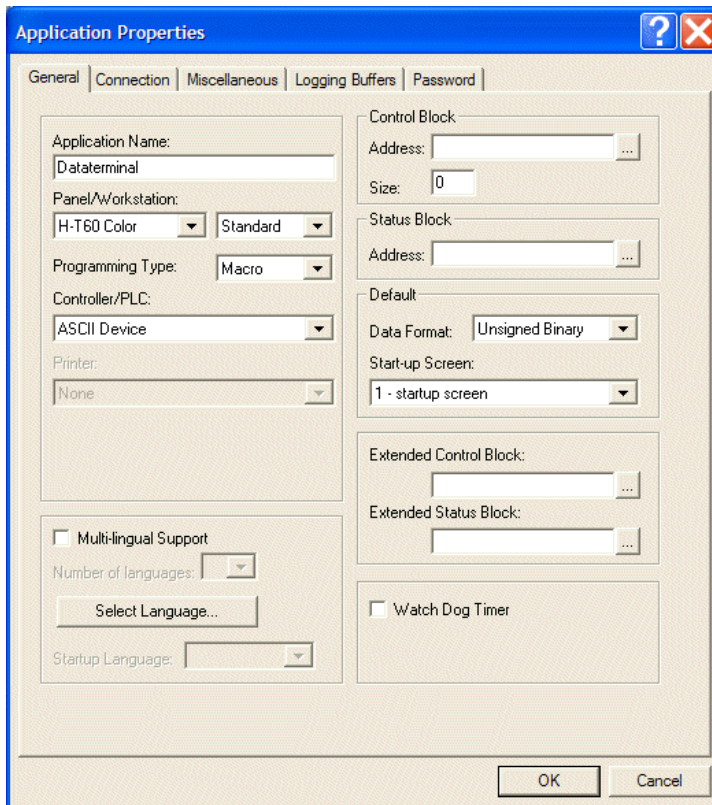
Data Buffer

Type: Selecting **Local** displays the current terminal data, but the last displayed data is not included when the screen is changed. Selecting **Global** displays the terminal data, including the last one when the screen is changed.

Size: Specifies the number of rows of terminal data.

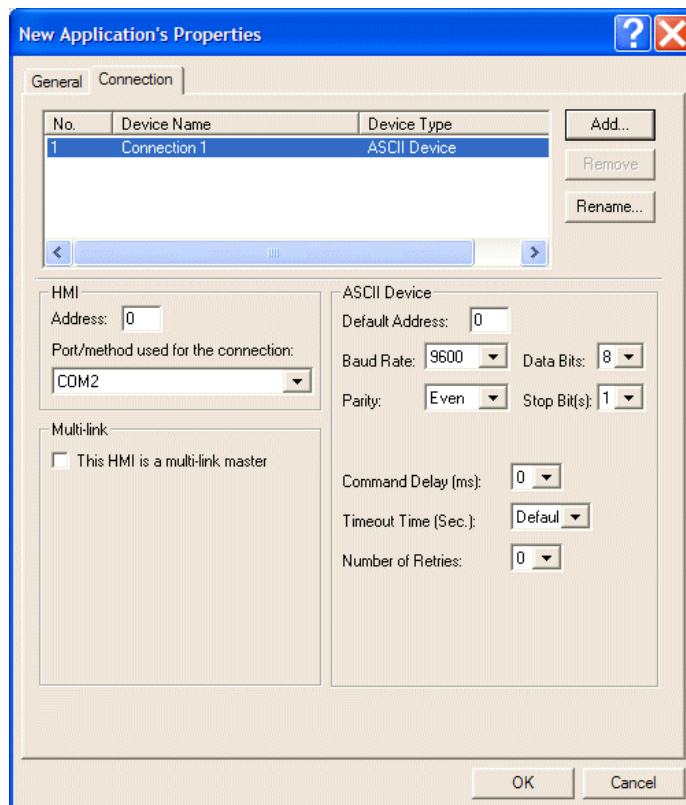
Example of designing a Terminal Data object

1. Click **Application/Workstation Setup** to set ASCII Device as **Controller/PLC** on the **General** tab of the **Application Properties** dialog box.



The General tab of the Application Properties dialog box

2. Specify the port/method used for the connection on the **Connection** tab.



The Connection tab of the Application Properties dialog box

Data Terminal attributes:

3. **Shape:** Select **Outlined_2** and **Black** for color.
4. **Variable:** Select **RX**. (The controller is an ASCII Device)
5. **Display:** Select **ASCII** mode to display terminal data.
6. **Data Buffer:** Select **Local**.

The above-mentioned steps will create a **Data Terminal** object that displays terminal data in ASCII mode.

Time Display

The **Time Display** object is used to make the operator terminal read the time value of the internal real time clock (RTC) and to display the content directly on the operator terminal screen.

Display Format

HH:MM:SS: Displays Hours:Minutes:Seconds

HH:MM: Displays Hours:Minutes

Example of a Time Display object

See section [Example of the Time, Date and Day-of-Week Display objects](#).

Date Display

The **Date Display** object is used to make the operator terminal read the date value of the internal real time clock (RTC) and to display the content directly on the operator terminal screen.

Display Format

MM/DD/YY: The format is Month/Date/Year.

DD/MM/YY: The format is Date/Month/Year.

DD.MM.YY: The format is Date.Month.Year.

Example of a Date Display object

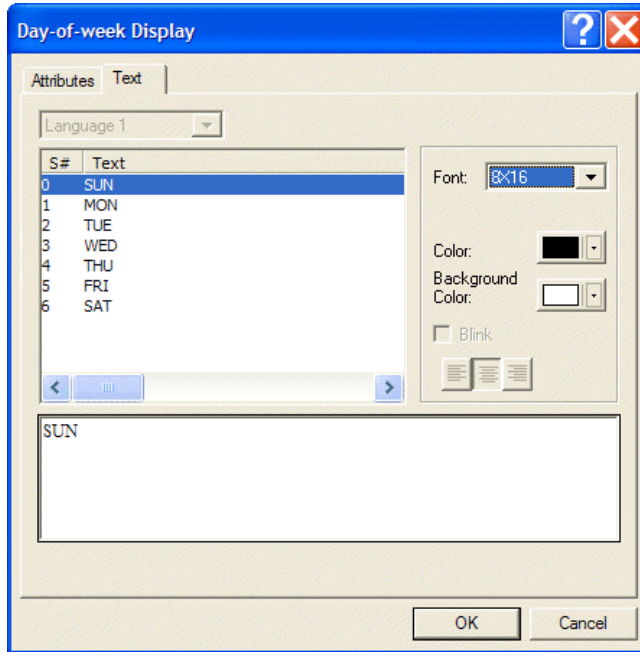
See section [Example of the Time, Date and Day-of-Week Display objects](#).

Day-of-Week Display

The **Day-of-Week Display** object is used to make the operator terminal read the day of week value from the internal real time clock (RTC) and to display the content directly on the operator terminal screen.

The object is not available on all operator terminal models; please see [Appendix A - H-Designer Features and Operator Terminal Models](#) for complete details.

The operator terminal will display the date of week automatically.



The Text tab of the Day-of-Week Display object

Example of the Time, Date and Day-of-Week Display objects

WED
11/02/05
19:17:02

The Time Display, Date Display and Day-of-Week Display objects

2.7.12 Bar Graph

There are two types of Bar Graphs; Normal and Deviation.

Normal Bar Graph

The **Normal Bar Graph** is used to make the operator terminal read the value of the controller register, to convert its data into a bar graph, and then to display the bar graph in the operator terminal.

Variable

Min.: Specifies the minimum value the bar graph can display.

Max.: Specifies the maximum value the bar graph can display.

Variable target/range limits: Select this option if the target value and the range limits are read from the controller.

The **Target Variable** is stored in the bit-location that follows the **Read**. The **High Limit** is stored in the bit location that follows the **Target Variable**. The **Low Limit** is stored in the bit location that follows the **High Limit**.

Register	X	X+1	X+2	X+3
	Read	Target Variable	High Limit	Low Limit
Example:	W10	W11	W12	W13

Display Format

Upward, Downward, Rightward, and Leftward: Selects the direction of the bar graph.

Color: Specifies the color of the bar graph.

Pattern: Specifies the pattern style to display.

Target: Sets the target to display.

- **Value:** Specifies the constant target value.
- **Color:** Specifies the color of target line.

Ranges: Select this option to fill the graph with a different color when the register value is beyond a normal range.

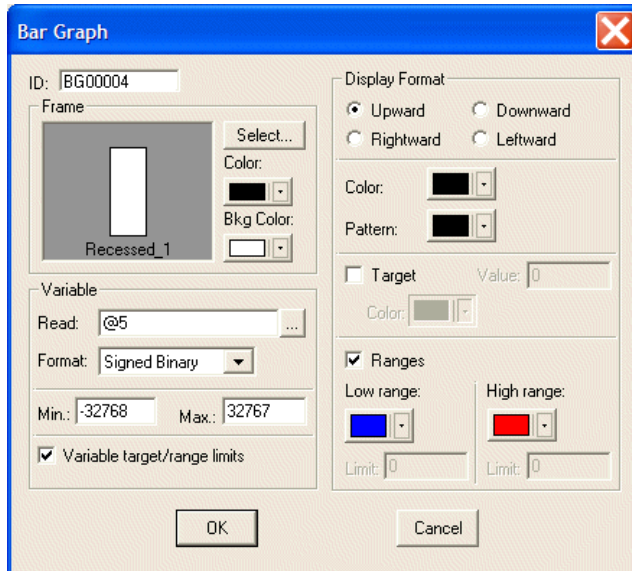
- **Low Range:** Specifies the color to fill the graph with if the register value is equal to or less than the low range limit.
- **Limit:** Specifies a constant for the low range limit.
- **High Range:** Specifies the color to fill the graph with if the register value is greater than or equal to the high range limit.
- **Limit:** Specify a constant for the high range limit.

For properties not explained in this section, please see the section [Specifying Object Properties](#).

Example of designing a Bar Graph object

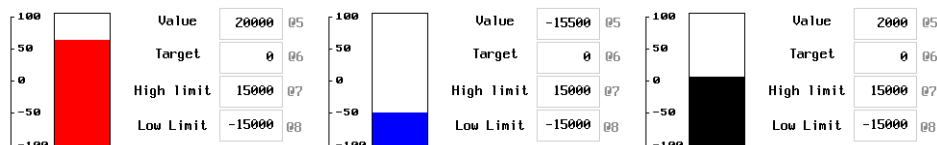
1. **Frame:** Select **Recessed_1** and **White** for background color.
2. **Variable:** Specify **@5** (Local Internal Memory) for **Read** and **Unsigned Binary** for format.
3. Set **Min.** to **-32,768** and **Max.** to **32,767**.
4. Check the option **Variable target/range limits**.

5. Check **Ranges** and select **Blue** for **Low range** and **Red** for **High range**.



*The attributes of the **Bar Graph** object in this example*

When the register value is equal to or less than the low limit, the graph will be filled with blue color; when the register value is equal or greater than the high limit, the graph will be filled with red color.



*Bar Graphs indicating different limits using **Numeric Entry** objects*

The **Numeric Entry** objects are used to set the high/low limits. There is a **Scale** to the left of the bar graphs. The **Low Limit** here is -15000 and the **High Limit** is 15000 using the following configuration:

Variable \leq -15000 fills graph with blue
 Variable between -15000 and 15000 fills graph with black
 Variable \geq 15000 fills graph with red

Deviation Bar Graph

The **Deviation Bar Graph** is used to make the operator terminal read the values of the controller register and to compare them with the normal value. Then the operator terminal converts the difference and presents it on a **Deviation Bar Graph** in the operator terminal.

Variable

Variable Std Value/Deviation Limit: Select this option if the standard value and deviation limit are read from the controller. If **Read** address is **W10**, **Standard Value** will be stored in **W11**; **Deviation Limit** will be stored in **W12**.

Display Format

Vertical, Horizontal: Selects the direction of the bar graph.

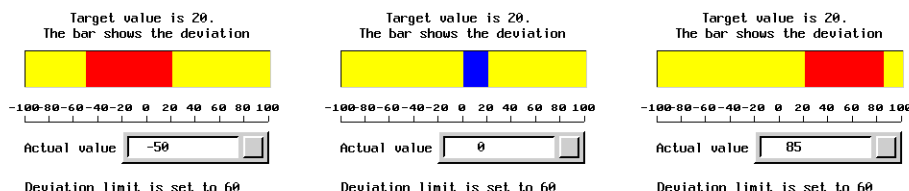
Standard Value: Specifies the constant standard value. The standard value will be a date line on the bar graph.

Display Deviation Limit: Select this option to fill the bar graph with selected color when the difference of the register value and the standard value is beyond the limit.

- **Limit:** Specifies the constant limit.
Difference Value: $=|\text{Variable} - \text{Standard Value}|$
- **Color:** Specifies the color to fill the graph with when the difference of the register value and the standard value is beyond the limit.

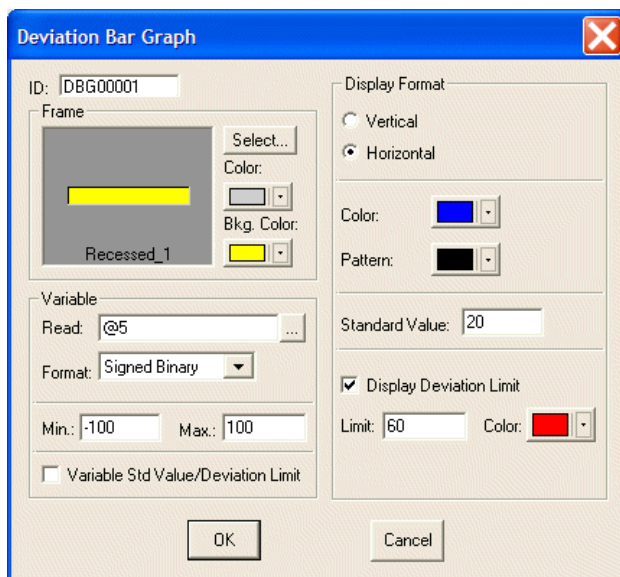
For properties not explained in this section, please see the sections [Normal Bar Graph](#) and [Specifying Object Properties](#).

Example of designing a Deviation Bar Graph



A Deviation Bar Graph shows the difference to the set standard value

1. **Frame:** Select **Recessed_1** and **Yellow** for background color.
2. **Variable:** Specify **@5** (Local Internal Memory) for **Read** and **Signed Binary** for **Format**.
3. Set **Min.** to **-100** and **Max.** to **100**.
4. **Display Format:** Select horizontal direction for the **Deviation Bar Graph**.
5. The **Standard Value** is **20** on this deviation bar graph.
6. Specify that the **Red** color is to fill the graph when the difference between the register value and the standard value is beyond the limit **60**.



The attributes of the Deviation Bar Graph in this example

2.7.13 Trend Graph

The **Trend Graph** is used to read a series of values from the related controller register. Then the operator terminal converts these values and presents them in a trend graph in the operator terminal.

Suppose that the register is **W_n** to be read from, with three curves in total. The data will be read as follows format:

Value in **W_n = m** is the real sampling points;
 Value in **W_{n+1}** is the first point of Y direction on the curve #1;
 Value in **W_{n+2}** is the first point of Y direction on the curve #2;
 Value in **W_{n+3}** is the first point of Y direction on the curve #3;
 Value in **W_{n+4}** is the 2nd point of Y direction on the curve #1;
 Value in **W_{n+5}** is the 2nd point of Y direction on the curve #2;
 Value in **W_{n+6}** is the 2nd point of Y direction on the curve #3;

:
:

and so on. The operator terminal reads all controller registers from **W_n** to **W_{n+3m}**.

For example, if the value in **W_n** is **m = 25** sampling points; the operator terminal will read data from 76 ($= 3 \times 25 + 1$) controller registers.

Control

To control the trend graph via the controller.

Trigger Flag #: The trigger flag number of the trend graph is 12-15 bits in CFR. The operator terminal reads data from the controller and displays the trend graph when the trigger flag turns on.

Clear Flag #: The clear flag number of the trend graph is bits 8-11 in CFR. The operator terminal clears the trend graph when the clear flag turns on.

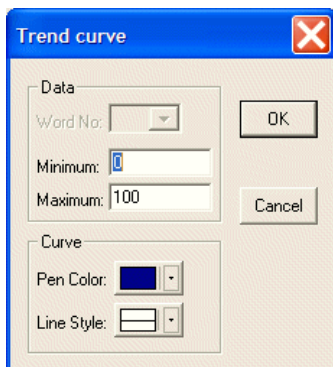
Display Format

Number of Points: Specifies the maximum number to display on the Y-direction.

Number of Grids: Specifies the number of evenly spaced horizontal grids to be displayed.

Grid Color: Specifies the color of the horizontal grids.

Curve # 1 - Curve # 4: Provides four curves for editing. When the **Edit** button is clicked, the dialog box below will appear:



Editing Display Format of Trend Curves

Minimum: Specifies the value corresponding to the lowest point on the trend graph. When the data is equal to or less than the **Minimum**, the operator terminal places the dot at the bottom pixel of the drawing area of the trend graph.

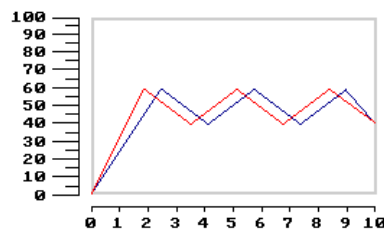
Maximum: Specifies the value corresponding to the highest point on the trend graph. When the register is equal to or greater than the **Maximum**, the operator terminal places the dot at the top pixel of the drawing area of the trend graph.

Pen Color: Specifies the color of the trend curve.

Line Style: Specifies the line style of the trend curve.

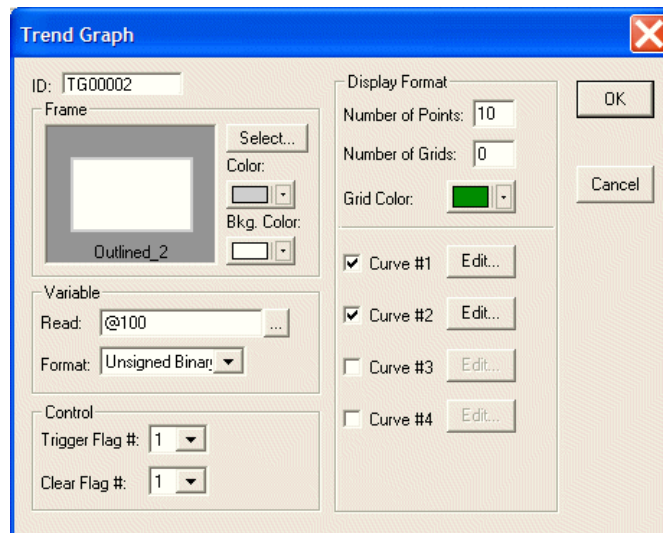
For properties not explained in this section, please see the section [Specifying Object Properties](#).

Example of designing a Trend Graph object



A Trend Graph

1. **Frame:** Select **Outlined_2** and **White** for background color.
2. **Variable:** Specify **@100** (Local Internal Memory) for **Read** and **Unsigned Binary** for **Format**.
3. **Control:** Select **1** for **Trigger Flag #** and for **Clear Flag #**.
4. **Display Format:** Specify **10** for **Number of Points**.



Trend Graph attributes in this example

5. Check **Curve #1** and click **Edit**. Set **Minimum** to **0** and **Maximum** to **100**. Select **Blue** for **Pen Color**.
6. Check **Curve #2** and click **Edit**. Set **Minimum** to **0** and **Maximum** to **100**. Select **Red** for **Pen Color**.

2.7.14 XY Chart

The XY Chart is used to read a series of values from the related controller register. Then the operator terminal converts these values and presents them on an XY chart in the operator terminal.

Suppose that the register to read from is W_n , with two curves in all. The data will be read as follows:

Value in $W_n = m$ are the real sampling points;
 Value in W_{n+1} is the first point of the X-axis on curve #1;
 Value in W_{n+2} is the first point of the Y-axis on curve #1;
 Value in W_{n+3} is the first point of the X-axis on curve #2;
 Value in W_{n+4} is the first point of the Y-axis on curve #2;
 Value in W_{n+5} is the 2nd point of the X-axis on curve #1;
 Value in W_{n+6} is the 2nd point of the Y-axis on curve #1;
 Value in W_{n+7} is the 2nd point of the X-axis on curve #2;
 Value in W_{n+8} is the 2nd point of the Y-axis on curve #2;
 :
 :

and so on. The operator terminal reads all controller registers from W_n to W_{n+2m} .

For example, if the value in W_n is $m = 15$ sampling points, the operator terminal will read data from 61 ($= 2 \times 2 \times 15 + 1$) controller registers.

Control

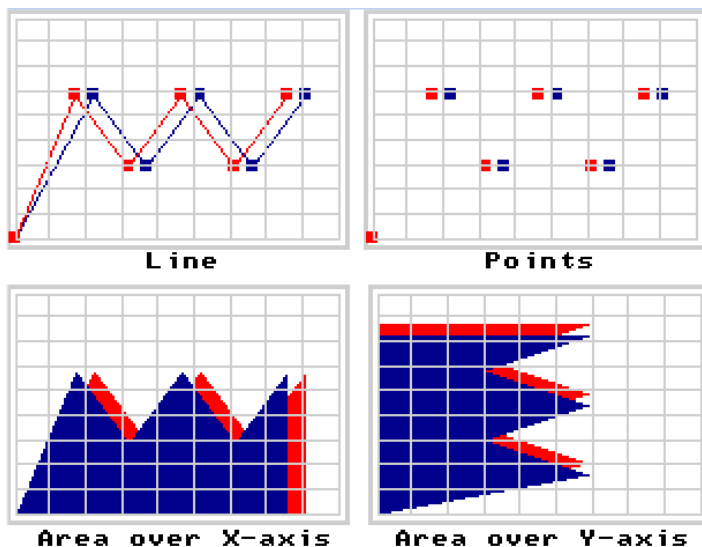
To control the trend graph by controller.

Trigger Flag #: The trigger flag number of the trend graph is bits 12-15 in CFR. The operator terminal reads data from controller and displays the XY chart when the trigger flag turns on.

Clear Flag #: The clear flag number of the trend graph is bits 8-11 in CFR. The operator terminal clears the trend graph when the clear flag turns on.

Display

Points, Line, Area over X-axis and Area over Y-axis: See the illustration below.

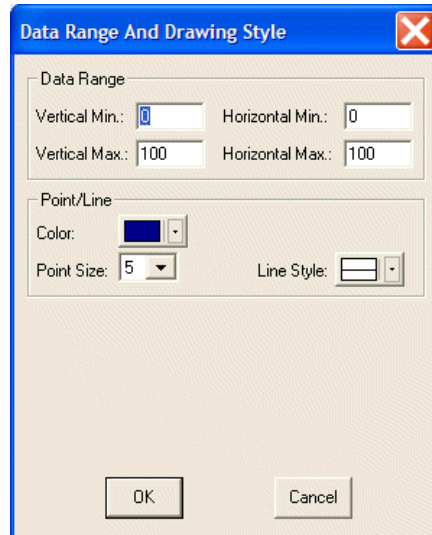


Maximum Number of Points: Specifies the maximum number of points to display on the XY chart.

Grids:

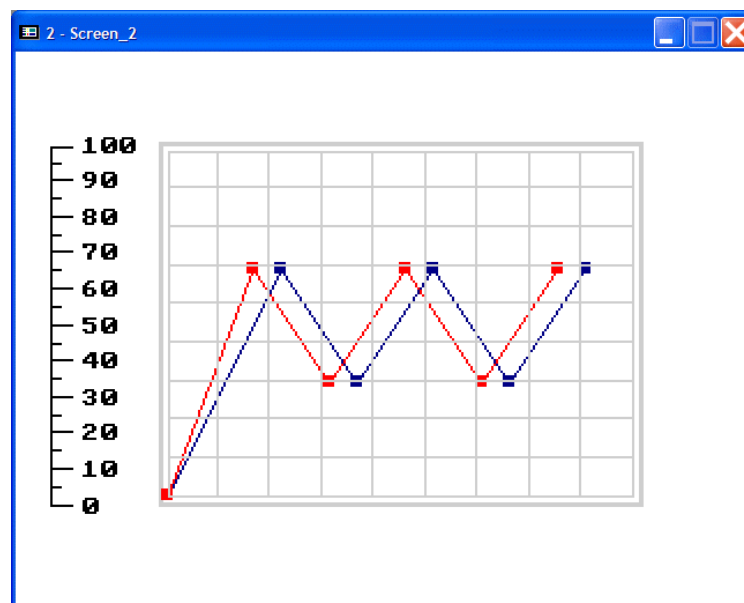
- **Number of H Lines:** Specifies the number of horizontal lines.
- **Number of V Lines:** Specifies the number of vertical lines.
- **Color:** Specifies the color of the lines.

Data Set # 1 - Data Set # 2: When the **Edit** button is clicked, the dialog box below will appear:



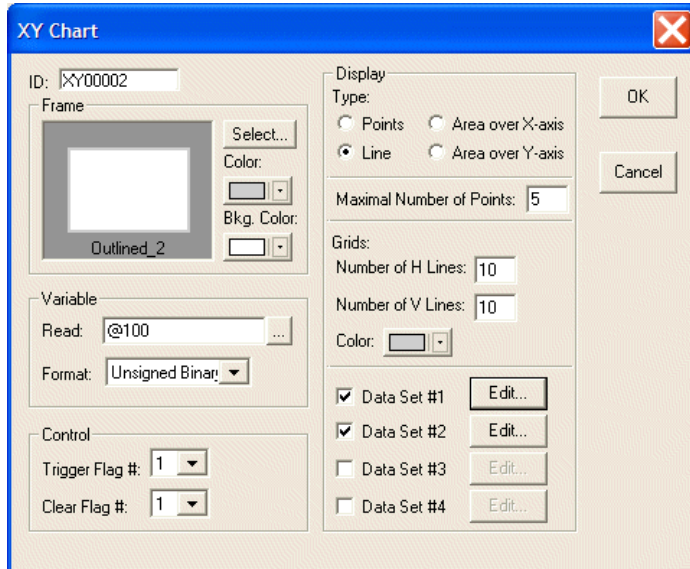
- **Vertical Min. and Vertical Max.:** Specifies the minimum and maximum values for the Y-axis.
- **Horizontal Min. and Vertical Max.:** Specifies the minimum and maximum values for the X-axis.
- **Color:** Specifies the color for the point/line.
- **Point Size:** Specifies the size of the point to display.

For properties not explained in this section, please see the section [Specifying Object Properties](#).

Example of designing an XY Chart object

An XY Chart

1. **Frame:** Select **Outlined_2** and **White** for background color.
2. **Variable:** Specify **@100** (Local Internal Memory) for **Read** and **Unsigned Binary** for **Format**.
3. **Control:** Select **1** for **Trigger Flag #** and **Clear Flag #**.
4. **Display:** Select **Line**.
5. **Maximum Number of Points:** 5. .



XY Chart attributes in this example

6. Check **Data Set #1** and click **Edit**. Set **Vertical Min.** and **Horizontal Min.** to **0**, and **Vertical Max.** and **Horizontal Max.** to **100**. Select **Blue** for **Point/Line Color**.
7. Check **Data Set #2** and click **Edit**. Set **Vertical Min.** and **Horizontal Min.** to **0**, and **Vertical Max.** and **Horizontal Max.** to **100**. Select **Red** for **Point/Line Color**.

2.7.15 Panel Meters

There are two types of panel meters - **Round** and **Rectangular**.

Round Panel Meter

The Round Panel Meter is used to make the operator terminal read the value from the controller register and to reflect the value on the **Round Panel Meter** object on the screen.

Needle

Color: Specifies the needle's color.

Sweep Angles (deg.): 300 or 360 degrees can be selected.

Scale

Color: Specifies the color used to display the scale.

Number of major ticks: Specifies the number of major ticks on the scale. If the number is less than 2, no ticks are displayed.

Number of minor ticks: Specifies the number of minor ticks on the scale.

Display axis: Check this option to display an arc as the axis of the scale.

Display mark: Check this option to display marks on the scale.

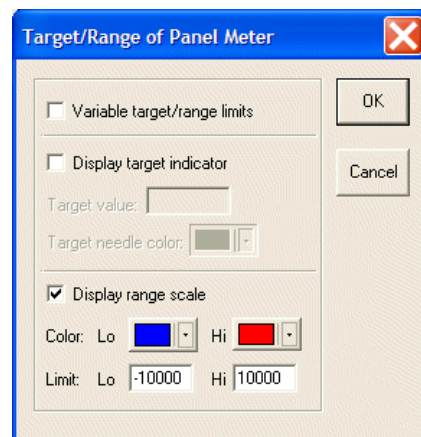
Font: 8 x 8 or 8 x 16 can be selected.

Number of digits: Specifies the number of digits, including precision and scale.

Decimal point position: Specifies the position of the mark's digit. If the number is 0, no decimal point is displayed.

Min. and Max.: Specifies the minimum and maximum number of marks.

Target/Range: Click the **Edit** button to display the dialog box below:



- **Variable target/range limits:** The target value and range limits are read from the controller. The target value is stored in a bit-location which is next to the **Read** location. The low limit is next to the target value. The low limit is next to the high limit. When the **Read** location is specified as **W10**, the target value is stored in **W11**; the low limit is stored in **W12**; the high limit is stored in **W13**.
- **Display target indicator:** Check this option to display target indicator.

Target value: Specifies the target value.

Target needle color: Specifies the color of the needle.

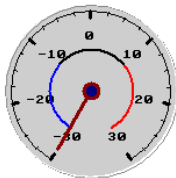
- **Display range scale:** Specifies the color of the range scale to display.

Low Range Color and High Range Color: Specifies the color to display on the scale when the value is less/greater than low/high range.

Low Limit and High Limit: Specifies the low limit and high limit constants.

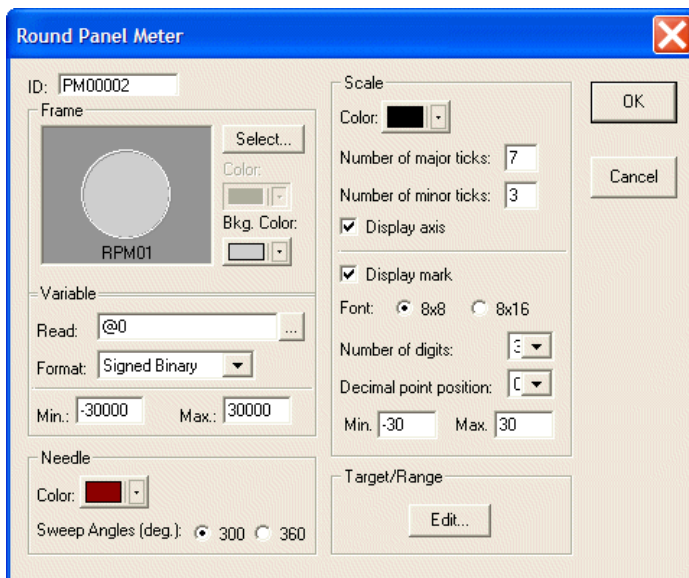
For properties not explained in this section, please see the section [Specifying Object Properties](#).

Example of designing a Round Panel Meter object



A Round Panel Meter

1. **Variable:** Specify @0 (Local Internal Memory) for **Read** and **Signed Binary** for **Format**.
2. Set **Min.** to -30000 and **Max.** to 30000.
3. **Needle:** Select 300 degrees for **Sweep Angle**.
4. **Scale:** Set **Number of Major Ticks** to 7 and **Number of Minor Ticks** to 3.
5. Check **Display mark** and set **Min.** to -30 and **Max.** to 30.



The attributes of the Round Panel Meter in this example

6. Click **Edit** for **Target/Range** and check **Display range scale**. Set the **Low range limit** to -10000 and **Blue** color; the **High range limit** to 10000 and **Red** color.

Rectangular Panel Meter

The properties of the **Rectangular Panel Meter** are the same as for the **Round Panel Meter**; please see section [Round Panel Meter](#).

2.7.16 Pie Graph

The **Pie Graph** is used to make the operator terminal read the register values in the controller. Then it converts the values into a 360 degree pie graph and displays the graph on the operator terminal screen.

Display Format

Starting angle: Specifies the original angle of the pie graph.

Pie color: Specifies the color used to fill the pie graph.

Bkg color: Specifies the color of the unfilled part of the pie graph.

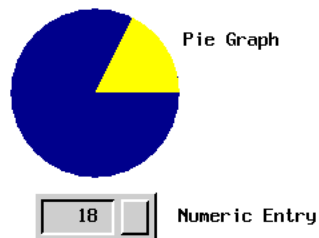
Style: Specifies the pattern style used for filling the pie graph.

Border: Check this option to display the pie graph with a border.

Color: Specifies the border color of the pie graph.

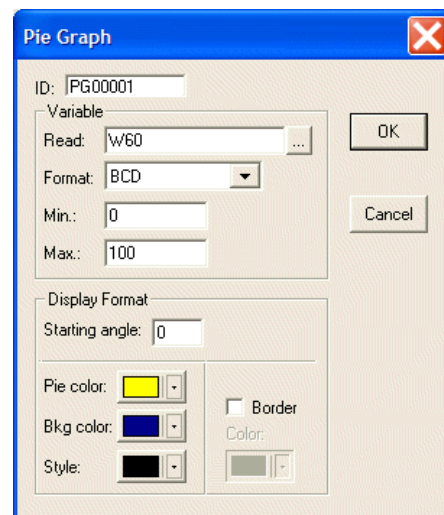
For properties not explained in this section, please see the sections [Bar Graph](#) and [Specifying Object Properties](#).

Example of designing a Pie Graph object



The Pie Graph object

1. **Variable:** Specify W60 for Read and BCD for Format.
2. Set 0 for Min. and 100 for Max.
3. **Display Format:** Set Starting Angle to 0.
4. Select Pie color, Bkg color, Style and Border color.



Attributes for the Pie Graph object in this example

A **Pie Graph** object is drawn from (+) X-axis (starting angle = 0). When the input value is 20, the area of the pie is one-fifth of a circle (= 20/100).

2.7.17 Dynamic Graphics

There are five types of dynamic graphics: **Animated Graphic**, **GIF Graphic**, **State Graphic**, **Dynamic Circle** and **Dynamic Rectangle**.

Animated Graphic

The **Animated Graphic** object enables you to control a graphic, including its position and moving-path on the operator terminal screen, whether via controller or not. For instance, the controller can control the graphic movement along the X-axis, Y-axis, or displaying different graphics.

Attributes Tab

Not Controlled by PLC: Check this option and the graph is not controlled by the controller.

Graphic State:

- **PLC controlled:** Displays the graphic states controlled by the controller.
- **Location dependent:** Displays the different states according to the location.
- **Auto change:** Changes the graphic state automatically.
Rate (Once per): Specifies the rate at which to change the graphic state.

Path:

- **PLC controlled:** Controls the object's movement path via the controller.
- **Horizontal line:** Moves the object along a horizontal line.
- **Horizontal marquee:** Moves the object along a horizontal line with marquee.
Check **Duplication** to move duplicated graphics along a horizontal line.
- **Vertical Line:** Moves the object along a vertical line.
- **Vertical marquee:** Moves the object along a vertical line with marquee.
Check **Duplication** to move duplicated objects along a vertical line.
- **Connected Lines:** Moves the object along the route of connected lines.

Path: Double-click the left key on the object to display the movement path.

Path Point: Right-click on the object to select **Add Path Point** or **Delete Path Point** from the drop-down list for the connected lines setup. Please see section [Example of creating horizontal/vertical line as a path](#) for complete details.

Note that a curve path can be configured with various path points.

- **Still:** Changes the state without movement.

Movement:

- **Rate (pixels/sec.):** Specifies the rate of movement.

One-way: Moves the object in one-way mode.

Two-way: Moves the object in two-way mode.

Duplication: Select this option to move duplicate objects with marquee; the number of copies can be specified as well.

For properties not explained in this section, please see the section [Specifying Object Properties](#).

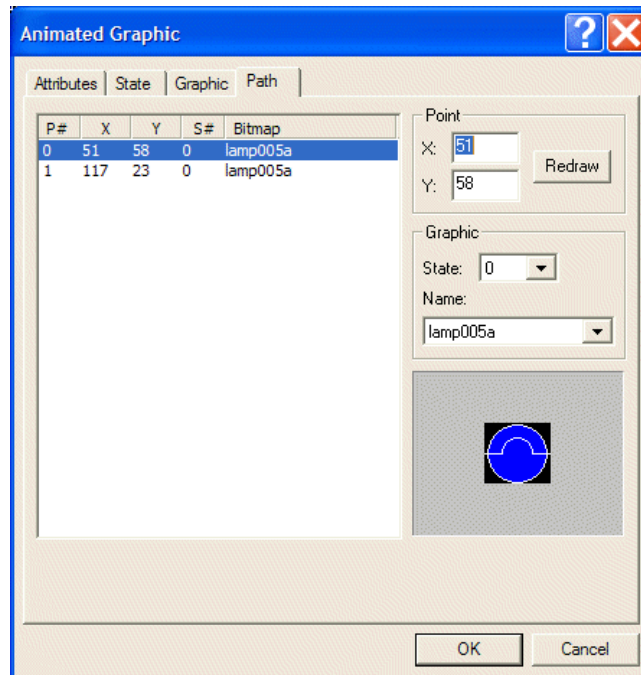
State and Graphic Tab

Please see section [Specifying Object Properties](#) for complete details.

Path Tab

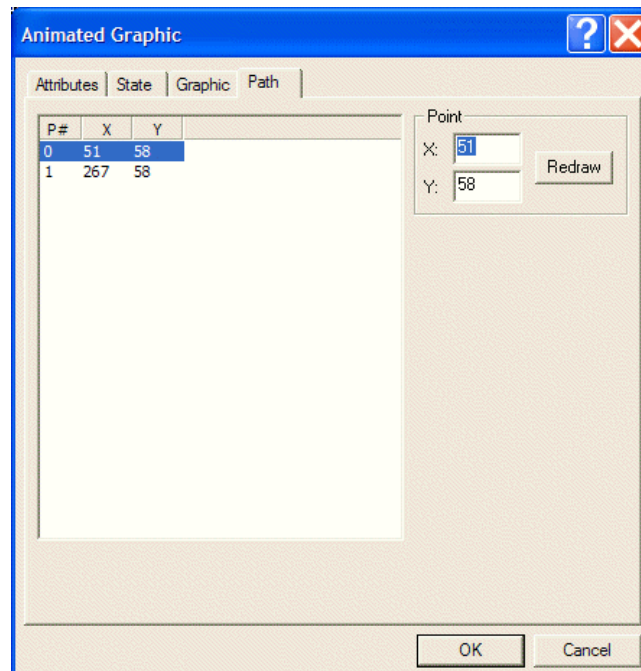
- The **Graphic State** is **Location dependent**:

On the **Path** tab, this feature enables you to specify the graphic states to be changed according to different locations. For instance, the point # 0 displays the graphic state in state # 0; the point # 1 displays the graphic state in state # 1.



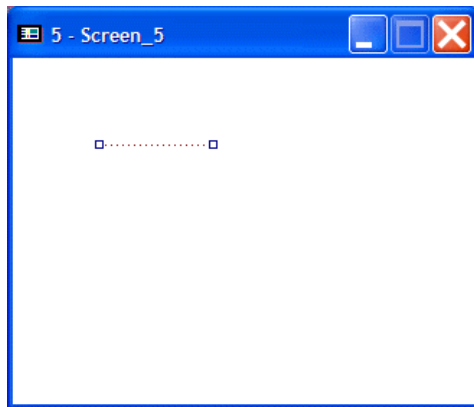
- The **Graphic State** is **Auto Change**:

On the **Path** tab, the object changes its state along the specified path. You can set the starting point and ending point of the path to be displayed on this tab.

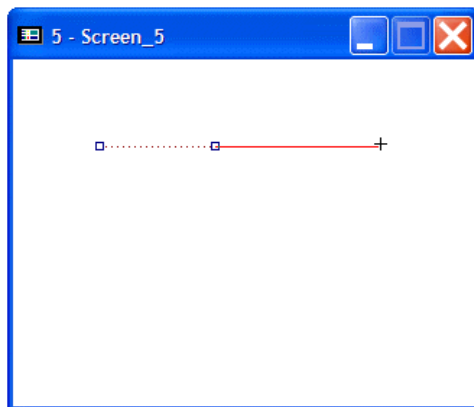


Example of creating horizontal/vertical line as a path

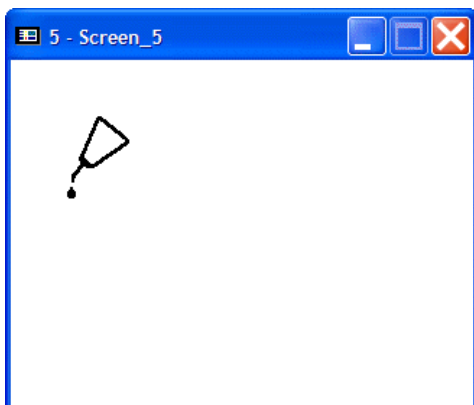
1. Double-click on the object to display the movement path.




2. Move the cursor to the point, then click to draw the movement path. The revised path is marked with a red line.

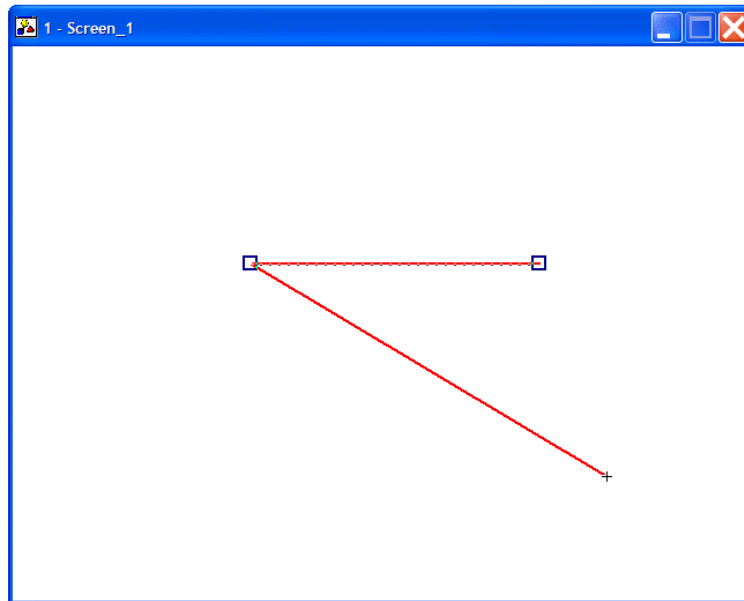


3. Click elsewhere on the screen to display the object's graphics.

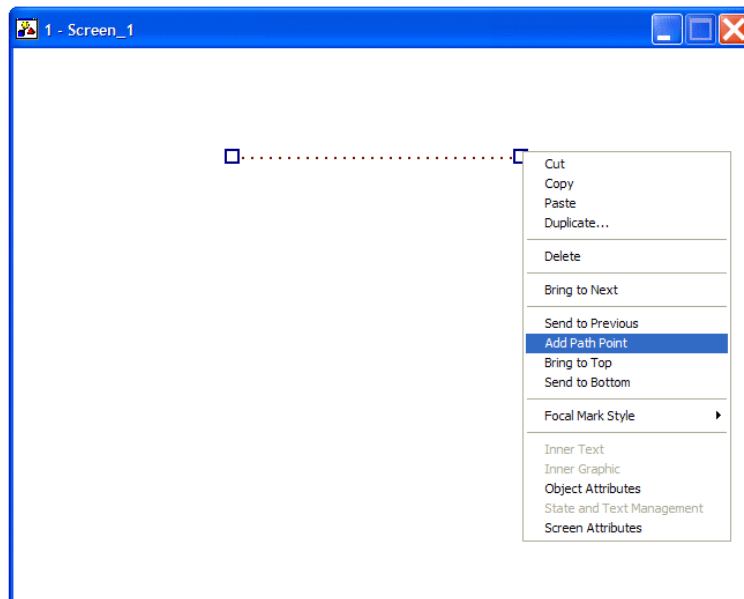


Example of creating connected lines as a path

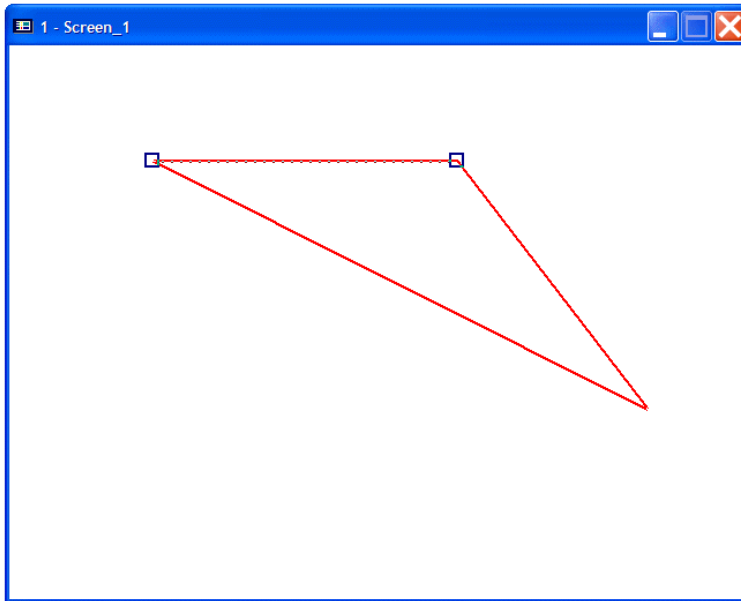
1. Double-click on the object to display the movement path.
2. Click on the points of the connected lines (icon is ) to draw the movement path. The revised path is marked with a red line.



3. Move the cursor to any one of the points and right-click to select **Add Path Point** or **Delete Path Point** from the drop-down list.



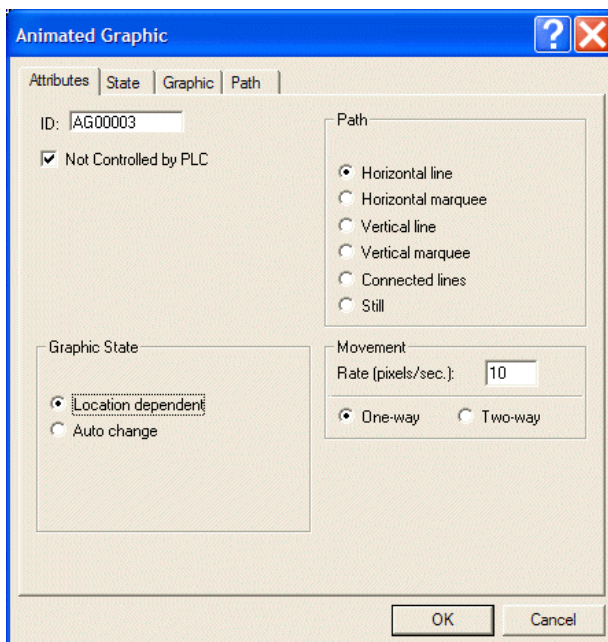
4. In the same fashion, the way to edit a new path is to click the added path point to drag a movement path, which is marked with red lines.



Example of Animated Graphic object Not Controlled by controller, One-way, Horizontal Line

On the Attributes tab of the Animated Graphic object:

1. Check the **Not Controlled by PLC** option.
2. The graphic state is **Location dependent**.
3. The path is **Horizontal line**.
4. The movement rate is **10 pixels/sec.** and the direction is **One-way**.



On the **Path** tab of the **Animated Graphic** object:

5. This object is one state; and the graphic is **oil.bmp**.

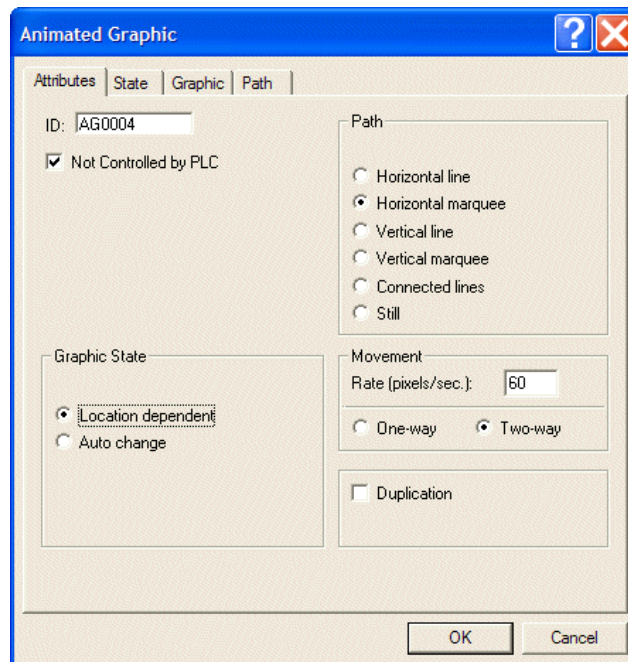


Therefore, this animated graphic is not controlled by the controller and moves along horizontal line at 10 pixels/sec. in one-way mode; the graphics are both **oil.bmp**.

Example of Animated Graphic object Not Controlled by controller, Two-way, Horizontal Marquee

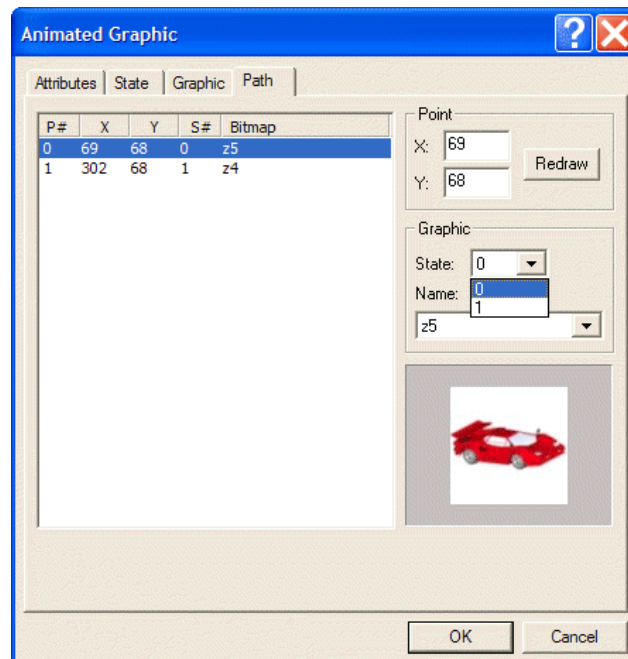
On the **Attributes** tab of the **Animated Graphic** object:

1. Check the **Not Controlled by PLC** option.
2. The graphic state is **Location dependent**.
3. The path is **Horizontal marquee**.
4. The movement rate is **60** pixels/sec. and the direction is **Two-way**.



On the **Path** tab of the **Animated Graphic** object:

5. Since the graphic state is location dependent, two states are set up.



Therefore, this object is not controlled by the controller and moves along a horizontal marquee at 60 pixels/sec. This object will move back and forth when it comes to the end point.



Moving to the right along a horizontal marquee

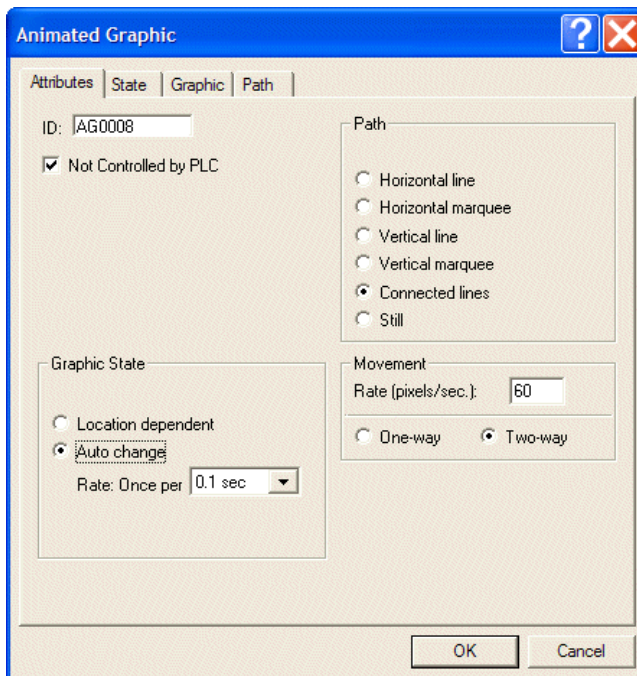


Moving to the left along a horizontal marquee

Example of Animated Graphic object Not Controlled by controller, Auto Change, Connected Line

On the **Attributes** tab of the **Animated Graphic** object:

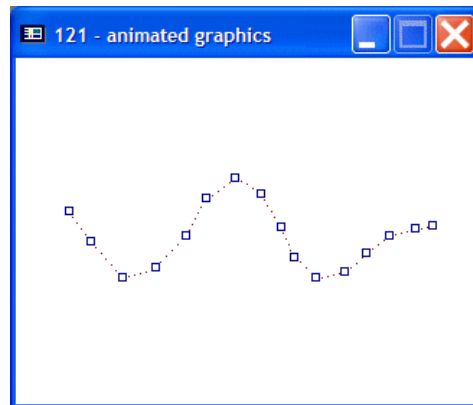
1. Check the **Not Controlled by PLC** option.
2. The graphic state is **Auto change** and the **Rate** is **0.1 sec**.
3. The path is **Connected lines**.
4. The movement rate is **60 pixels/sec.**; the direction is **Two-way**.



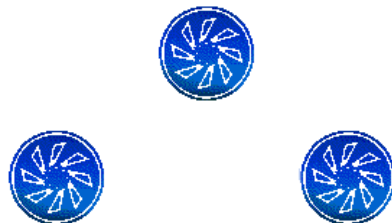
On the **Graphic** tab of the **Animated Graphic** object:

5. Setup 8 graphics for auto change. The example uses 8 wheels with different angles to create a rolling image when the graphic state is changed automatically.

6. Add more points to make the movement path smoother.



Therefore, this animated graphic object is not controlled by the controller and moves at 60 pixels/sec. back and forth along the curve. The graphic state is set to auto change to show a rolling effect.



Moving along the curve with auto change in two-way

GIF Graphic

The GIF Graphic displays GIF graphics, controlled by the controller or not.

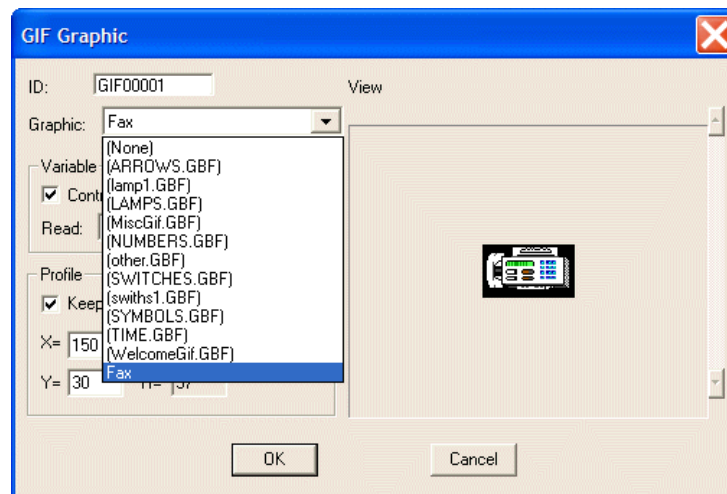
Graphic: Selects the graphic to display from the drop-down list. The graphic will appear in the **View** window.

Profile: Modifies the location and size of objects.

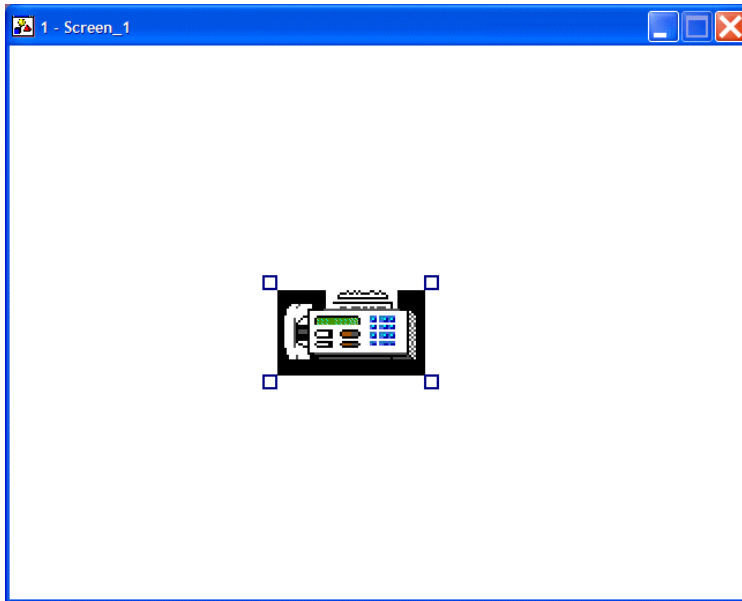
For properties not explained in this section, please see the section [Specifying Object Properties](#).

Example of designing a GIF Graphic object

1. Select a GIF graphic from the **Graphic** drop-down list; specify the controller register from which to read (if the option **Controlled by PLC** is checked) and modify its profile.



Selecting a GIF graphic



The GIF graphic appears on the object

State Graphic

The **State Graphic** constantly displays one of several bitmaps depending on the state of the controller register.

Attributes

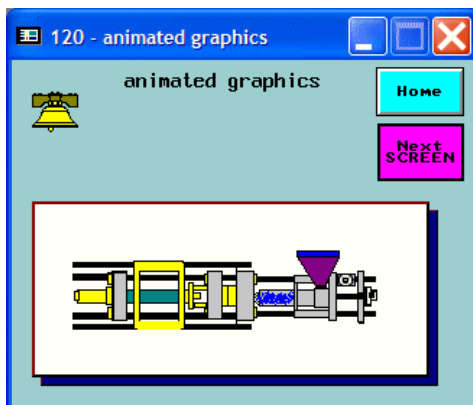
States:

- **Auto Change:** Check this option to change the graphic automatically.

Change Rate (Hz): Specifies the rate of change.

For properties not explained in this section, please see the section [Specifying Object Properties](#).

Example of creating a State Graphic object

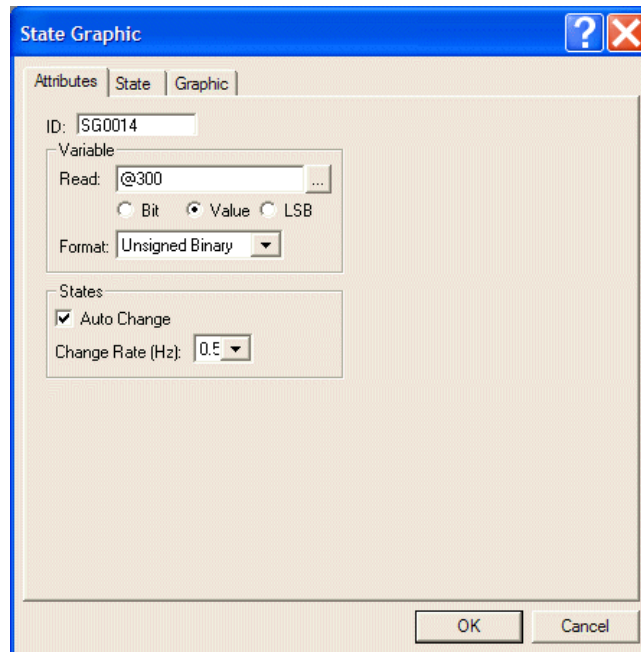


A State Graphic object

On the **Attributes** tab of the **State Graphic** object:

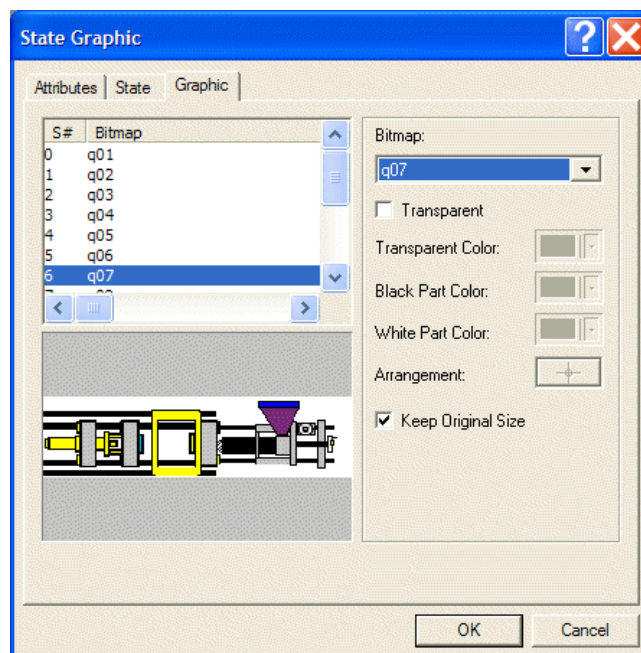
1. **Variable:** Specify **@300** (Local Internal Memory) for **Read** and **Value** to be displayed.
2. Select **Unsigned Binary** for **Format**.

3. Check the **Auto Change** option; **Change Rate (Hz)** is 0.5.



On the **State** and **Graphic** tabs of the **State Graphic** object:

4. There are 14 states in all. Add states on the **State** tab and select specific graphic to display on the **Graphic** tab.



Therefore, the object changes its state repeatedly every 0.5 sec and is controlled by the controller. A pumping effect will be generated on the operator terminal screen.



State 0



State 1

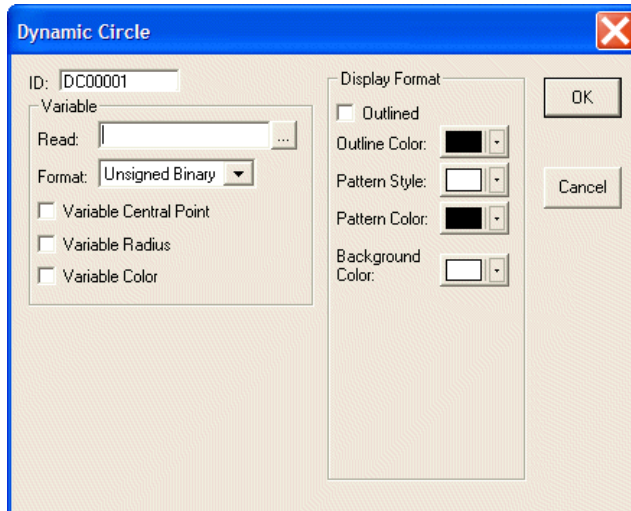


State 2

The object shows auto change (ex. state 0-2)

Dynamic Circles

The **Dynamic Circles** object changes its position, radius and color according to controller registers.



Variable Central Point: The position of the central point is controlled by the controller.

Variable Radius: The length of the radius is controlled by the controller.

Variable Color: The color of the object is controlled by the controller.

Display Format: Specifies the format of the object to display.

For properties not explained in this section, please see the section [Specifying Object Properties](#).

Example of using Dynamic Circle

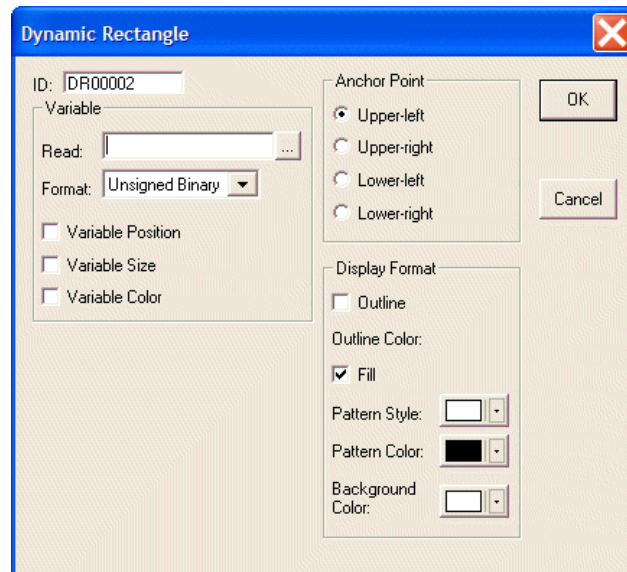
Suppose that the dynamic circle's central point, radius, and color are controlled by the controller. The **Read** address is W430.

The operator terminal can read four pieces of data simultaneously at most, and the read addresses here are W430, W431, W432, and W433. The following is the table for controller addresses and graphic properties.

Re-central Point	Re-central Point	Re-central Point	Re-central Point	Fix Central Point	Re-central Point	Fix Central Point
Re-radius	Re-radius	Re-radius	Re-radius	Re-radius	Fix Radius	Fix Radius
Fix Color	Fix Color	Re-Coloring	Fix Color	Fix Color	Fix Color	Re-Coloring
Wn= Radius	Wn= Radius	Wn= Radius	Wn= Radius	Wn= X	Wn= X	Wn= Color
Wn+1= X	Wn+1= X	Wn+1= Color		Wn+1= Y	Wn+1= X	
Wn+2= Y	Wn+2= Y			Wn+2= Color		
Wn+3= Color						

Dynamic Rectangle

The **Dynamic Rectangle** object changes its position, radius, and color according to controller registers.



Variable Position: The position of the object is controlled by the controller.

Variable Size: The length of the object is controlled by the controller.

Variable Color: The color of the object is controlled by the controller.

Anchor Point: Specifies the anchor point for the dynamic rectangle whose position is variable and size is fixed.

Display Format: Specifies the format of the dynamic rectangle to display.

For properties not explained in this section, please see the section [Specifying Object Properties](#).

Example of using Dynamic Rectangle

Suppose that the position, size, and color are variable and are controlled by the controller. The **Read** address is **W420**.

The operator terminal reads five simultaneous pieces of data from the controller at most. The read addresses here are W420, W421, W422, W423, and W424. The following is the table of controller addresses and graphic properties.

Re-position	Re-position	Fix Position	Fix Color	Fix position	Re-position	Fix Position
Re-size	Re-size	Re-size	Re-size	Fix Size	Fix Size	Fix Size
Re-Coloring	Fix Color	Re-Coloring	Re-Coloring	Re-Coloring	Fix Color	Fix Color
W420=Width	W420=Width	W420= Width	W420=Width	W420=X	W420=X	W420= Color
W421=Height	W421= Height	W421= Height	W421=Height	W421=Y	W421=Y	
W422=X	W422=X	W422=Color		W422=Color		
W423=Y	W423=Y					
W424=Color						

2.7.18 Historical Display

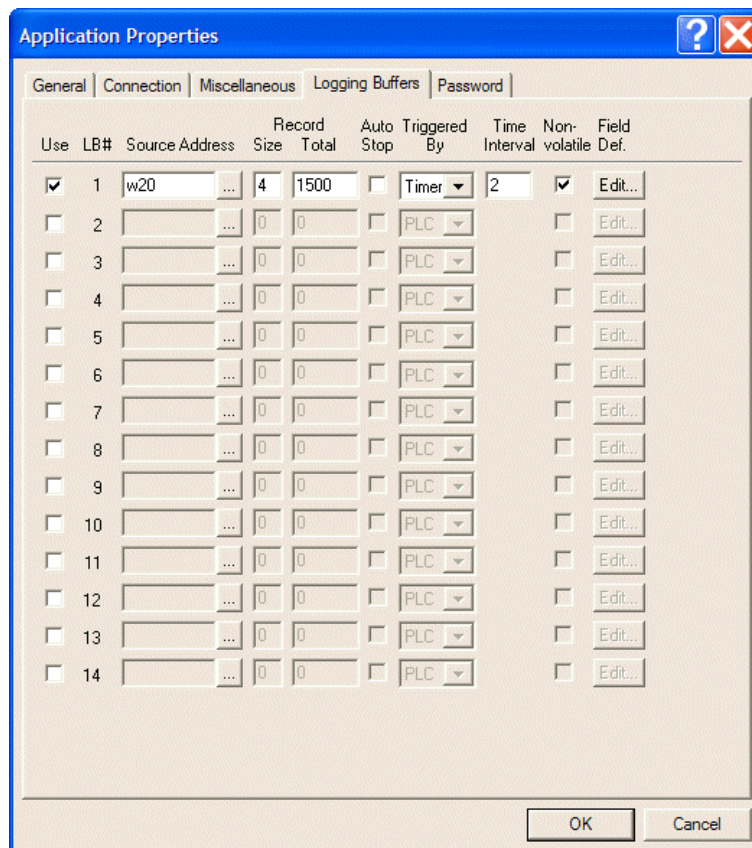
The **Historical Display** drop-down list includes the following: **Historical Trend Graph**, **Historical Data Table** and **Historical Event Table**.

The **Historical Display** data is stored in **logging buffers**, so you should assign its area and size first. The logging buffer is used to store the sampling data in the operator terminal battery backup RAM.

The object is not available on all operator terminal models; please see [Appendix A - H-Designer Features and Operator Terminal Models](#) for complete details.

Logging Buffer

Select the **Logging Buffers** tab from **Application/Workstation Setup**.



The Logging Buffers tab

Source Address: Specifies the starting address to read from, e.g. W20 is starting address of a block of controller registers from which the logging buffer reads from.

Size: Specifies the size of a record to read from at one time, e.g. **Size = 4** represents 4 words = W20, W21, W22, W23.

Total: Specifies the total to store in, e.g. **1500** represents that the operator terminal reads 4 words each time, sampling 1500 times in total.

Time/Date: Check these boxes to record the **Time/Date** while sampling.

Auto Stop: Check this box to stop the sampling when it reaches the specified total = 1500. If this option is not selected, the first piece of data will be overwritten when the 1501st piece of data records is recorded.

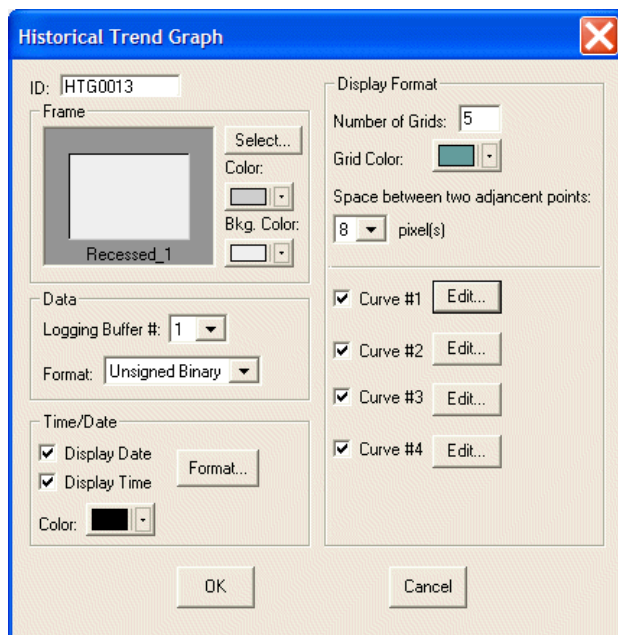
Triggered By: Select **Timer** to trigger periods of fixed time or select controller to be triggered by the controller. If **PLC** is selected for triggering, it is triggered by the specified corresponding bit-locations, Wn+2, Wn+3, and Wn+4.

Time Interval: Specify how often the logging buffer gets a record of data from the timer. Unit: seconds.

After completing the setup on the **Logging Buffers** tab, you can create three types of **Historical Display** objects:

Historical Trend Graph

The operator terminal may have a fixed sampling period or the controller may initiate that data be read from the specified registers and then store the data in the logging buffers in the operator terminal memory. After a sampling period, the data is then converted to continuous curve(s) and displayed on the operator terminal.



Data

Logging Buffer #: Specifies the number of the logging buffer where the historical data is stored, numbered 1 to 12.

Format: BCD, Signed Binary or Unsigned Binary.

Time/Date

Display Date and **Display Time:** Checking the boxes displays date and time; click **Format** button to setup.

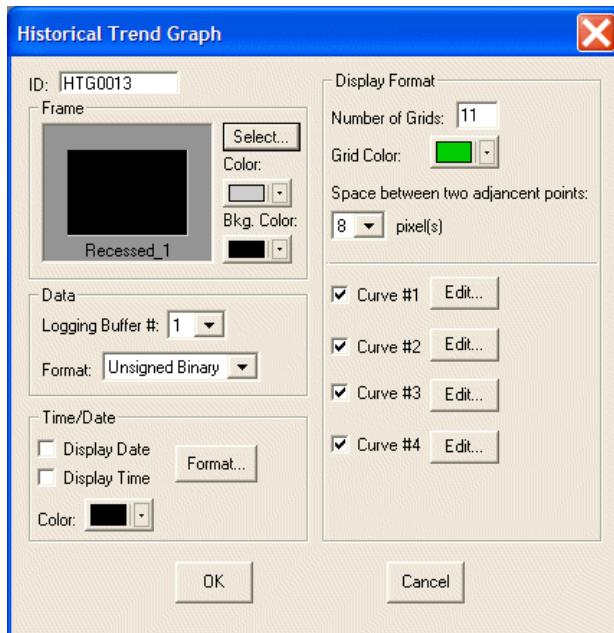
Color: Specifies the color of characters to be displayed.

Display Format

Number of Grids: Specifies how many evenly spaced horizontal lines shall be displayed.

Grid Color: Specifies the color of the horizontal grids.

Curve #1 - #4: There are four curves to be selected. Click the **Edit** button to display the following dialog box:



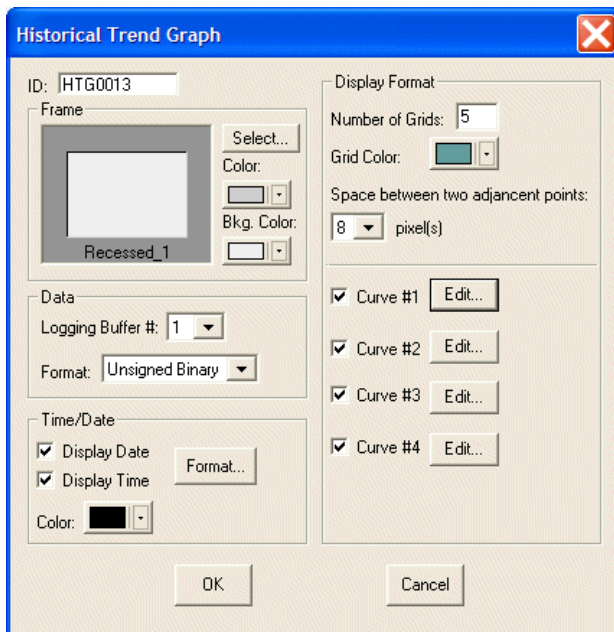
Word No: Specifies the number of the words to display on the historic trend curve.

Minimum and Maximum: Specifies the value corresponding to the lowest and highest point on the historical trend curve.

Pen Color: Specifies the color used to draw the trend curve.

Line Style: Specifies the line style of the trend curve.

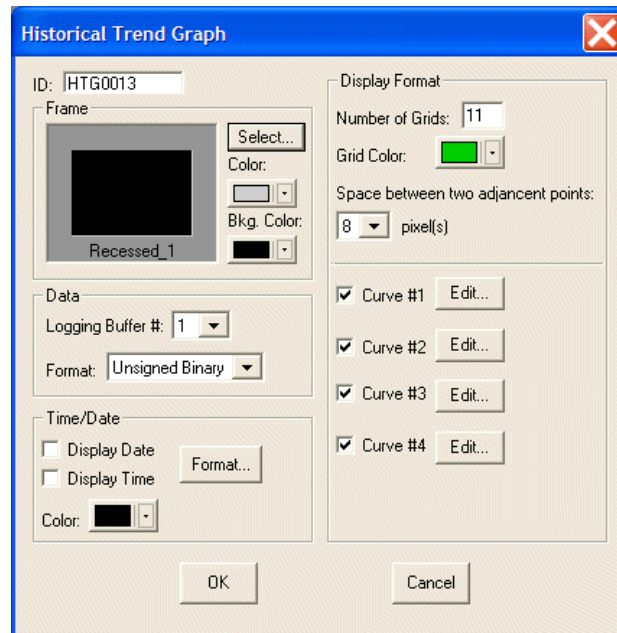
Example of designing a Historical Trend Graph



The Historical Trend Graph

1. First assign logging buffer size and area for **Logging Buffer #1** on the **Logging Buffers** tab in **Application/Workstation Setup**:
Source Address: @20
Size: 4
Total: 3000
Check **Time and Date**
Triggered By: Time
Time Interval: 1
Check **Non-volatile**.
Please see section [Logging Buffer](#).

For the **Historical Trend Graph** object, select the following properties:



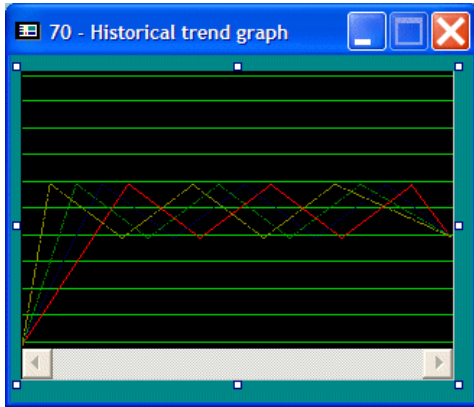
The Historical Trend Graph properties

2. **Frame:** Select **Recessed_1** and **Black** for background color.
3. **Data:** Select **Logging Buffer #1** and **Unsigned Binary** for **Format**.
4. **Display Format:** Specify 11 grids and **Green** for **Grid Color**.
5. Check four curves to display; Curve #1 displays the data record stored in the word 0, Curve #2 displays the data record stored in the word 1 etc.



Properties for Historical Trending Curve #1

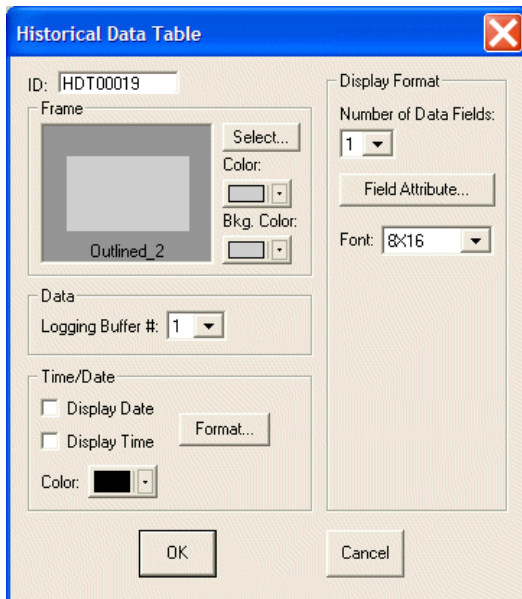
6. Set **Minimum** to **0** and **Maximum** to **65535**. Specify a different color for each curve.



The result of the settings in the above example: Displaying data stored in logging buffer #1.

Historical Data Table

The operator terminal may have a fixed sampling period or the controller may initiate a data read from the specified registers and then store the data in the logging buffers in the operator terminal memory. After a sampling period, the data is then converted to numeric data tables and displayed in the operator terminal.



Historical Data Table properties

Display Format

Number of Data Field: Specify how many data fields to display; up to 10 data fields.

Clicking the **Field Attribute** button displays the dialog box below:

The **Field Attribute** dialog box contains a table with 10 rows, each representing a data field. The columns are: Field No., Starting Position, Word No., Data Size, Data Format, Display Color, Leading Zero, Decimal Pt. Position, Int. Digits, and Frac. Digits. Field 1 is pre-filled with Starting Position 0, Word No. 0, Data Size 1, Data Format BCD, and Display Color black. Fields 2 through 10 are empty.

Field No.	Starting Position	Word No.	Data Size	Data Format	Display Color	Leading Zero	Decimal Pt. Position	Int. Digits	Frac. Digits
1	0	0	1	BCD	Black	<input type="checkbox"/>	0	4	0
2						<input type="checkbox"/>			
3						<input type="checkbox"/>			
4						<input type="checkbox"/>			
5						<input type="checkbox"/>			
6						<input type="checkbox"/>			
7						<input type="checkbox"/>			
8						<input type="checkbox"/>			
9						<input type="checkbox"/>			
10						<input type="checkbox"/>			

At the bottom of the dialog are **OK** and **Cancel** buttons.

Starting Position: Specifies the position of a data field to display.

Note:

If the starting position is 0 for **Field No. 1**; the time will be displayed in **Field No. 1**, the date will be displayed in **Field No. 2**, and the first data field will be displayed in **Field No. 3**. If there is no time/date displayed, the first data field will be displayed in **Field No. 1**.

Word No.: 0-31 characters can be specified.

Data Size: 1 represents one-word; 2 represents double-word.

Data Format: Select **BCD**, **Signed Binary**, **Unsigned Binary** or **Hexadecimal**.

Display Color: Specifies the character color of a data field.

Leading Zeros: Checking this box to displays leading zeros.

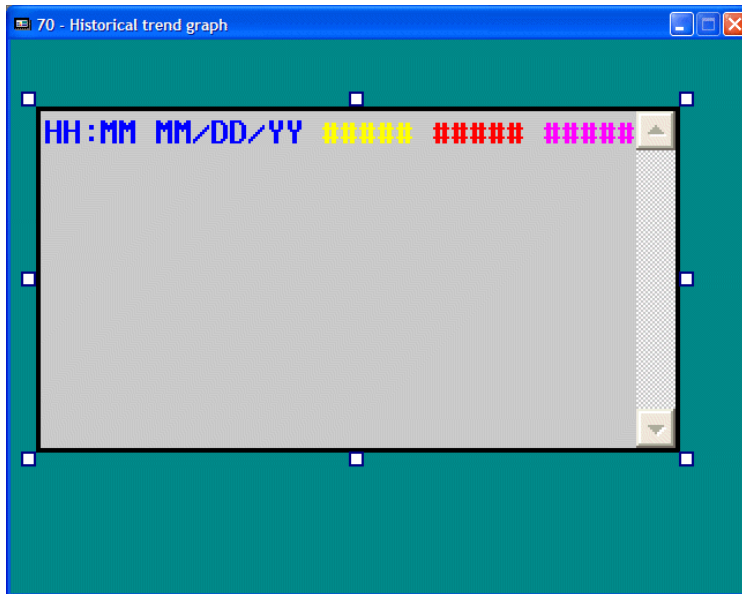
Decimal Pt. Position: Specifies the number of digits following the decimal point.

Int. Digits: Specifies how many digits to the left of the decimal point.

Frac. Digits: Specifies how many digits to the right of the decimal point.

For properties not explained in this section, please see the sections [Historical Trend Graph](#) and [Specifying Object Properties](#).

Example of designing a Historical Data Table

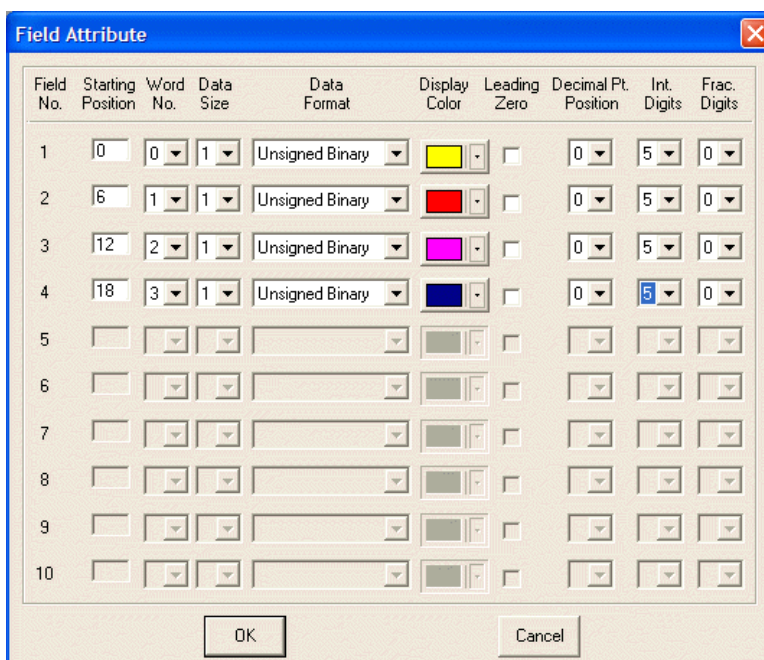


A Historical Data Table

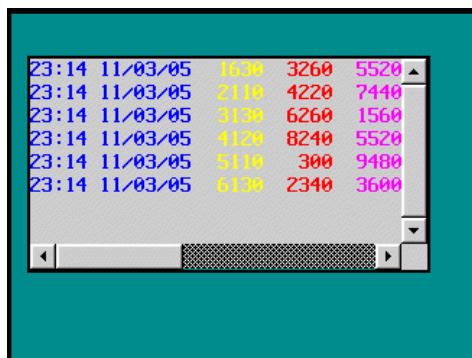
Suppose that there is historical data stored in logging buffer #1.

Select the following properties for the **Historical Data Table** object:

1. **Frame:** Select **Outlined_2** and **Black** for border color.
2. **Data:** Select **Logging Buffer #1** to read data from.
3. **Time/Date:** Check both boxes.
4. **Display Format:** Select 4 data fields. Click **Field Attribute**.



5. Make settings according to above.

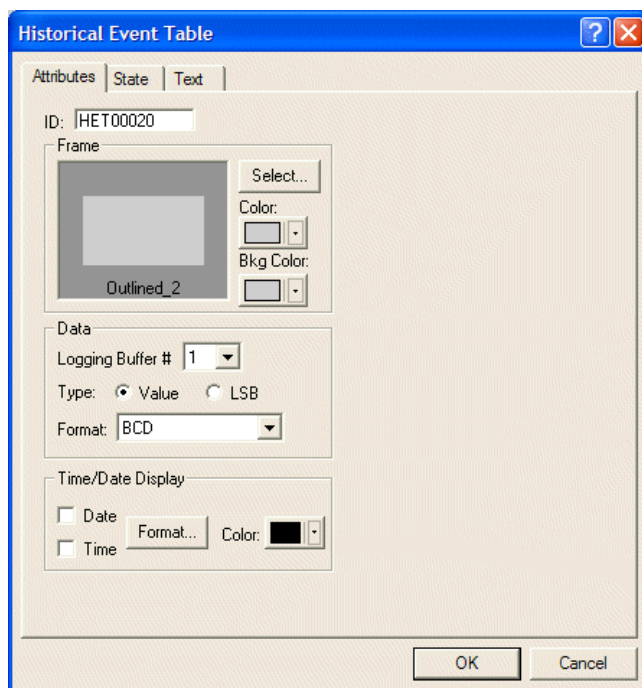


23:14	11/03/05	1530	3260	5520
23:14	11/03/05	2110	4220	7440
23:14	11/03/05	3130	6260	1560
23:14	11/03/05	4120	8240	5520
23:14	11/03/05	5110	300	9480
23:14	11/03/05	6130	2340	3600

The result of the settings in the above example: Displaying data stored in logging buffer #1.

Historical Event Table

The operator terminal may set a fixed sampling period or the controller may initiate a data read from the specified registers or the related bits in LSB. Thereafter, the data is converted into pre-defined message text(s) and then displayed line-by-line in the operator terminal.



The Historical Event Table

Data

Logging Buffer #: Specifies the number of the logging buffer where the historical data is stored, numbered 1 to 12.

Type: Select **Value**; 256 states in all (0-255), where 0 represents state 0; 1 represents state 1 etc. or **LSB**; 16 states in all, the operator terminal takes the bit number of the lowest bit that is ON as the state number.

Format: Only available when **Value** is selected, and the formats are **BCD**, **Unsigned Binary** or **Signed Binary**.

For properties not explained in this section, please see the sections [Historical Trend Graph](#) and [Specifying Object Properties](#).

2.7.19 Alarm Display

There are four types of **Alarm Display**: **Alarm History**, **Active Alarm List**, **Alarm Frequency Table** and **Alarm Marquee**.

To use the **Alarm Display** objects, you must set up the address of the alarm block and its parameters. The operator terminal reads the value stored in the controller and displays its corresponding messages. Up to 512 messages can be set. Please see section [Alarm Setup](#) for information about setting up alarms.

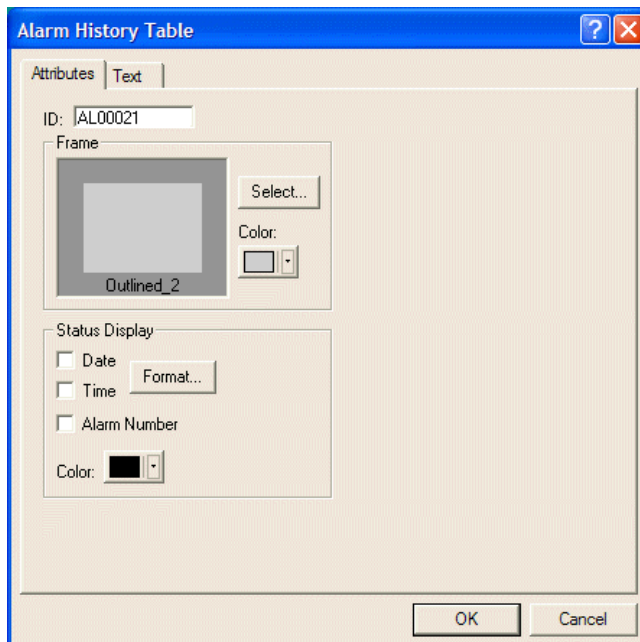
The objects are not available on all operator terminal models; please see [Appendix A - H-Designer Features and Operator Terminal Models](#) for complete details.

After completing alarm setup, the four types of **Alarm Display** objects can be used.

Alarm History Table

The operator terminal reads the reference bits in the controller in fixed periods and then activates the corresponding alarm messages. Thereafter, this the operator terminal can show the alarm history as an **Alarm History Table**.

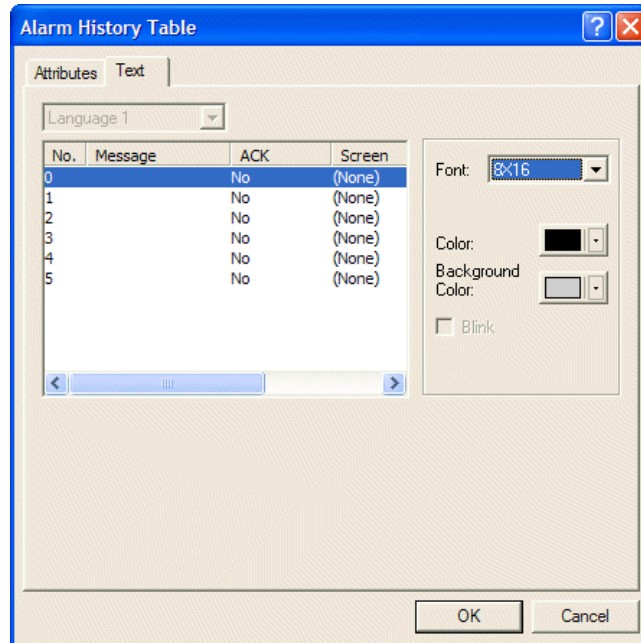
Attributes Tab



Status Display

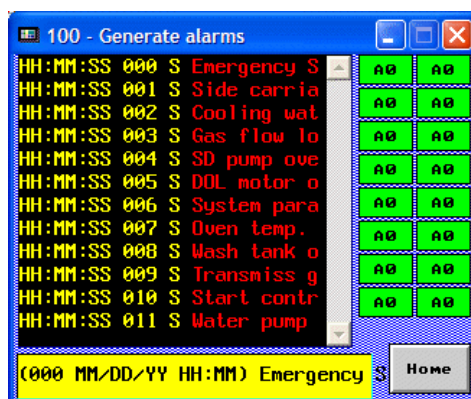
- **Date and Time:** Check the boxes to display the date and time and click the **Format** button to specify formats.
- **Alarm Number:** Check this box to display the alarm number.
- **Color:** Specifies the color of the message.

Text Tab



On this tab, Font, Color and Background Color for alarm messages can be selected.

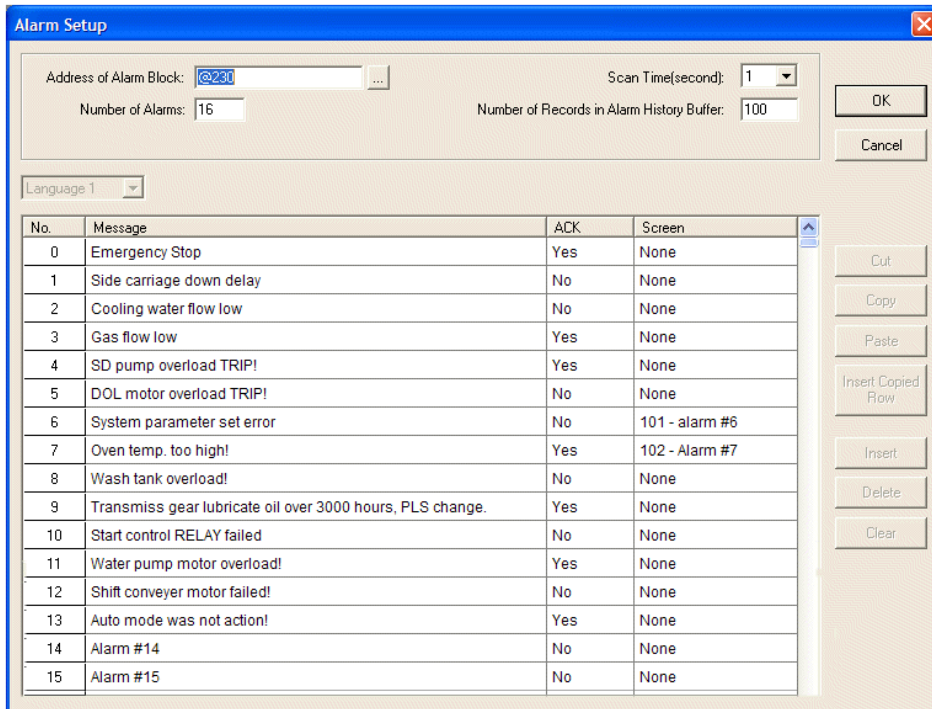
Example of designing an Alarm History Table



Alarm Setup:

1. Select **Application/Alarm Setup**.
2. Specify **@230** for Address of Alarm Block to and **16** as Number of Alarms.
3. Set the scan time to **1** second for sampling the controller data, and maximum number of records to **100**.

4. Enter texts in the message block, select to acknowledge the alarm and which screen to display.



The **Alarm Setup** dialog box contains the following fields and table:

Address of Alarm Block: @230
 Number of Alarms: 16
 Scan Time(second): 1
 Number of Records in Alarm History Buffer: 100

Language 1

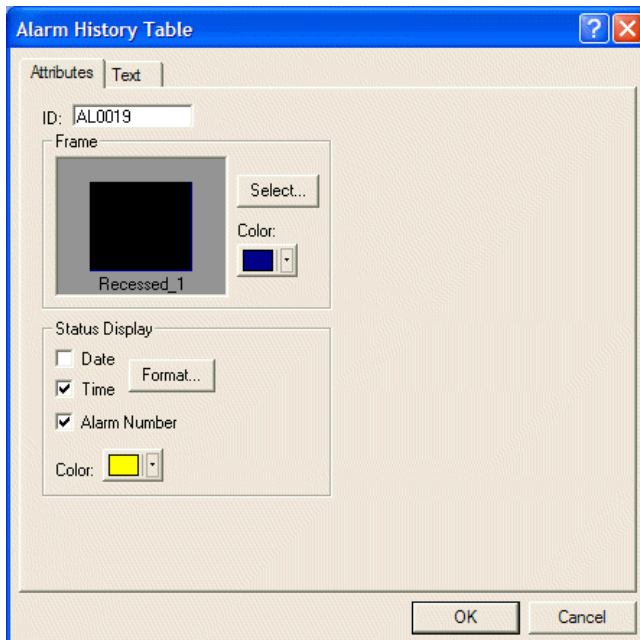
No.	Message	ACK	Screen
0	Emergency Stop	Yes	None
1	Side carriage down delay	No	None
2	Cooling water flow low	No	None
3	Gas flow low	Yes	None
4	SD pump overload TRIP!	Yes	None
5	DOL motor overload TRIP!	No	None
6	System parameter set error	No	101 - alarm #6
7	Oven temp. too high!	Yes	102 - Alarm #7
8	Wash tank overload!	No	None
9	Transmiss gear lubricate oil over 3000 hours, PLS change.	Yes	None
10	Start control RELAY failed	No	None
11	Water pump motor overload!	Yes	None
12	Shift conveyer motor failed!	No	None
13	Auto mode was not action!	Yes	None
14	Alarm #14	No	None
15	Alarm #15	No	None

Buttons: OK, Cancel, Cut, Copy, Paste, Insert Copied Row, Insert, Delete, Clear.

The Alarm Setup dialog box


Setting up properties for the Alarm History Table:

5. **Frame:** Select **Recessed_1** and **Blue** for frame color.
6. **Status Display:** Check **Time** and **Alarm Number**, and select **Yellow** for color.



The **Alarm History Table** dialog box, **Attributes** tab, shows the following settings:

ID: AL0019

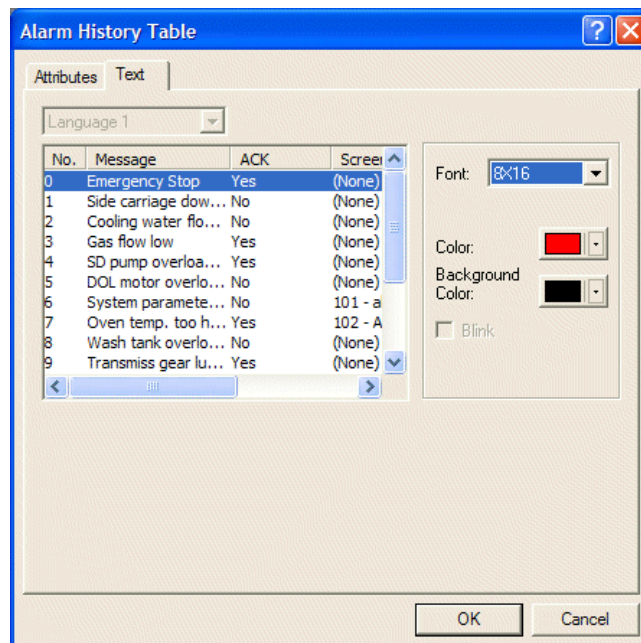
Frame:  Select... Color: Blue

Status Display:
☐ Date
☒ Time
☒ Alarm Number
 Color: Yellow

Buttons: OK, Cancel.

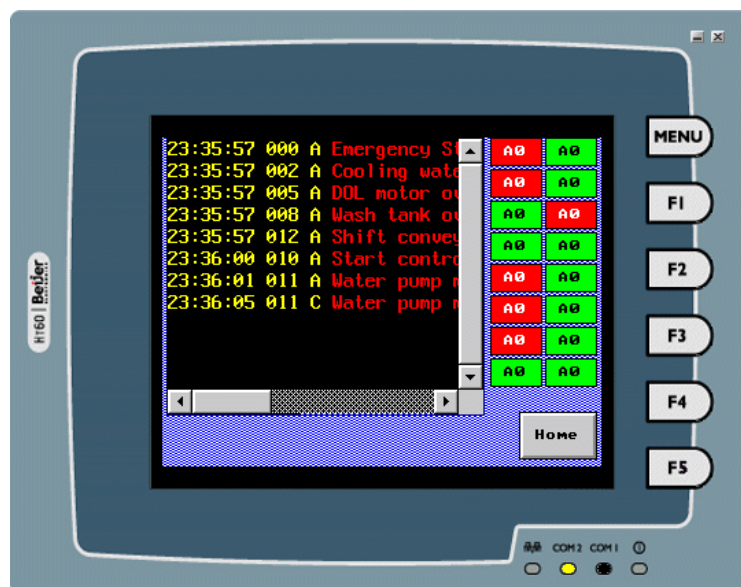
The Attributes tab of the Alarm History Table

7. Note that the properties for **Message**, **ACK** and **Screen** set in the **Alarm Setup** dialog box will be shown on the **Text** tab. Select **Red** for message color and **Black** for background color.



The Text tab of the Alarm History Table

The operator terminal will read the reference bits in the controller at fixed periods, then convert the data into its corresponding messages in sequence and display them on the screen.



The Alarm History Table, using 16 On/Off Buttons to send alarm messages.

Note that the Alarm State A represents Activate; Alarm State C represents Clear.

Active Alarm List

The operator terminal displays only the active alarms according to its reference bit in controller = ON and sorts the data according to the order of the state number.

All the attributes are the same as for the Alarm History Table object, please see the section [Alarm History Table](#).

Example of designing an Active Alarm List

The steps used to create an Active Alarm List are the same as for an Alarm History Table object. You must complete the alarm setup first, and then specify its properties. Please see section [Example of designing an Alarm History Table](#).



The Active Alarm List displays only the active alarms, and in numerical order.

Alarm Frequency Table

The operator terminal summarizes the number of occurrences of each alarm which are to be monitored and displayed on the screen.

All the attributes are the same as for the Alarm History Table object, please see the section [Alarm History Table](#).

Example of designing an Alarm Frequency Table

The steps used to create an Alarm Frequency Table are the same as for an Alarm History Table object. You must complete the alarm setup first, and then specify its properties. Please see section [Example of designing an Alarm History Table](#).



The Alarm Frequency Table displays the number of occurrences of each alarm

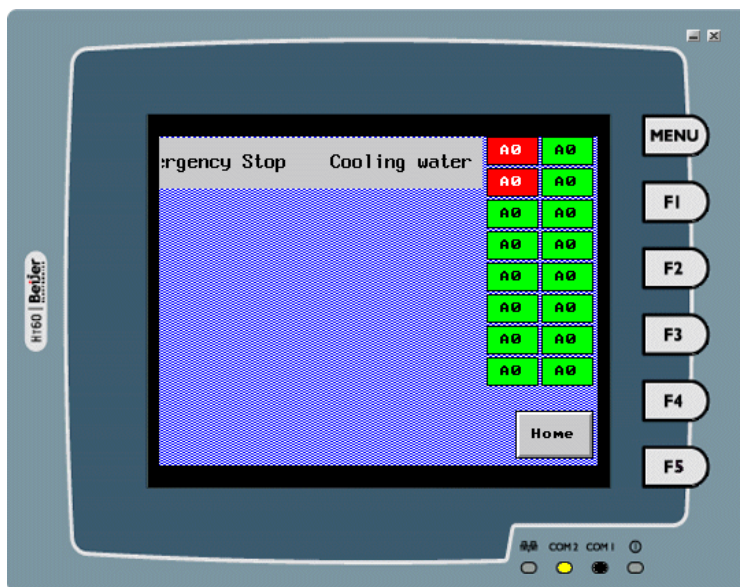
Alarm Marquee

The operator terminal displays alarm messages from active alarms as a moving sign.

All the attributes are the same as for the **Alarm History Table** object, please see the section [Alarm History Table](#).

Example of designing an Alarm Marquee

The steps used to create an **Alarm Marquee** are the same as for an **Alarm History Table** object. You must complete the alarm setup first, and then specify its properties. Please see section [Example of designing an Alarm History Table](#).

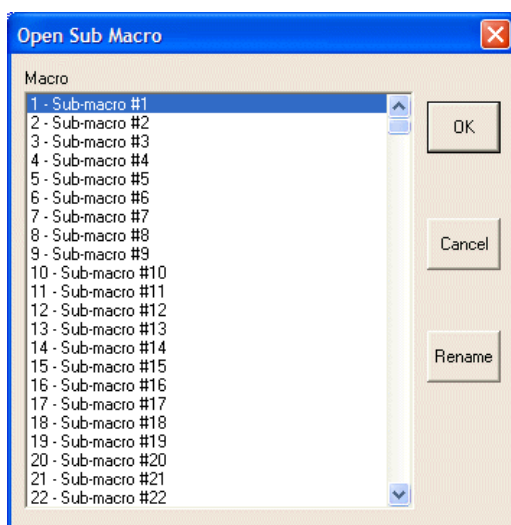


The Alarm Marquee displays the active alarm message as a moving sign on the screen

2.7.20 Sub Macro

A **Sub Macro** is the macro's sub-application. The main function is to call commands directly. Some common functions or operation commands which are used frequently can be edited and saved as sub macros for call commands.

There are 512 options for a Sub Macro, please see the chapter [Macros](#) for complete details.



The Sub Macro edit window

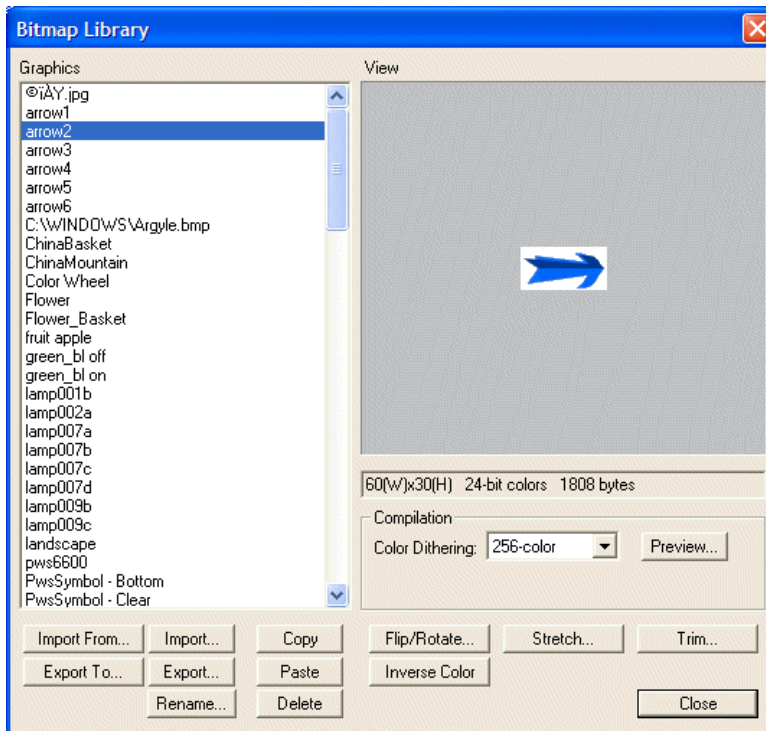
2.8 Library Menu

There are five options in the **Library** menu: **Bitmap Library**, **Font Library**, **Save as Shape**, **Shape Library Manager** and **Text Pool**. The main function is to edit, import and export bitmaps, shapes, fonts or text pools.

2.8.1 Bitmap Library

The **Bitmap Library** is used mainly to import, export, and edit bitmaps.

Select **Library/Bitmap Library**, to open the dialog box below:



The Bitmap Library dialog box

Graphics: Lists all the graphics available for selection.

View: Displays the selected graphic.

Compilation

Color Dithering: Processes the graphic (16-bit, 24-bit or JPEG) to display the image as vividly as the original on the screen. There are **8-color**, **16-color** and **256-color** options. The higher the color selected, the higher the contrast of the figure displayed.

Import from: Imports bitmaps from a selected library, *.GBF or *.GIF graphics.

Export to: Exports bitmaps to a selected library, *.GBF.

Import: Imports the graphic into the bitmap library from a computer. The importable graphic formats include Bitmap Image (*.BMP), Jpeg Image Files (*.JPG), AutoCad Files (*.DWG or *.DXF) and GIF Files (*.GIF).

Export: Exports the graphic stored in the bitmap library to a computer.

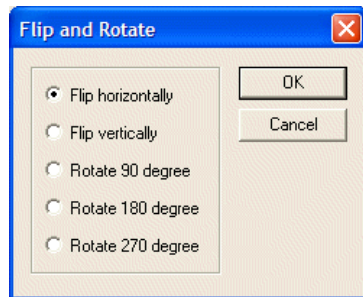
Rename: Modifies the name of the graphic.

Copy: Copies the selected bitmap to the clipboard.

Paste: Imports a bitmap from the clipboard. When clicked, a dialog box will appear and ask for the name of the imported graphic.

Delete: Deletes the selected bitmap.

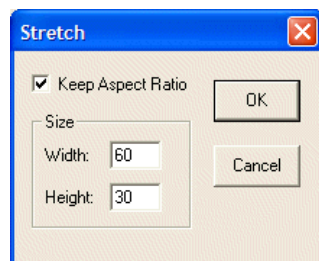
Flip and Rotate: Allows changing a bitmap's orientation. When clicked, a dialog box with flip or rotate degree options is displayed.



The Flip and Rotate dialog box

Inverse Color: Inverts the selected bitmap's colors.

Stretch: Adjusts the width and height of the bitmap.



The Stretch dialog box

Trim: Allows cutting unused area around a bitmap.



The Trim dialog box

2.8.2 Font Library

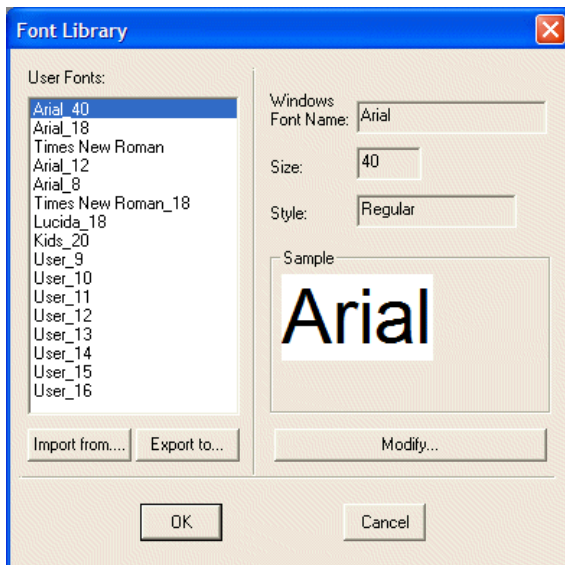
The **Font Library** supports all Windows fonts, and enables defining up to 16 types. You can define your preferred fonts to create a more attractive interface.

Times New Roman

Kids *Lucida*
 16x32 Arial

User-defined fonts

Select **Library/Font Library** to display the **Font Library** dialog box.

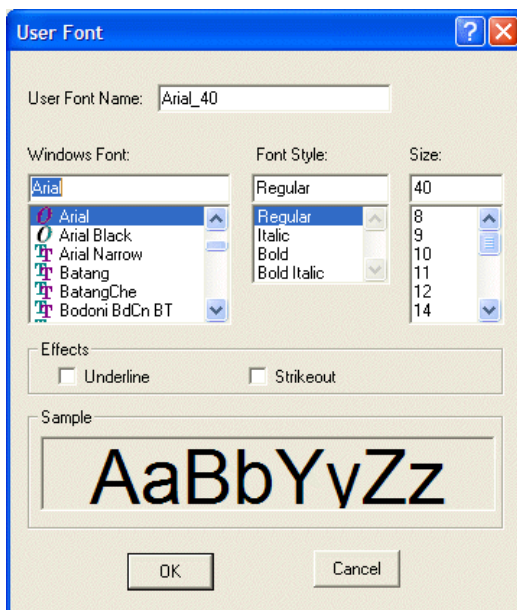


The Font Library dialog box

Import from: Click to import fonts into the font library.

Export to: Click to export fonts to the font library.

Modify: Click to modify the format of selected font.

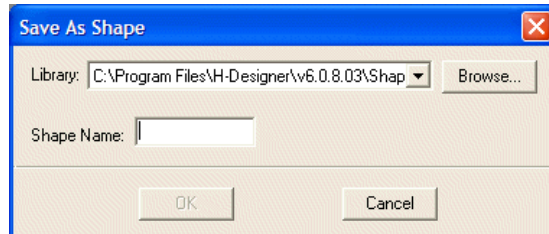


Modifying a font in the Font Library

2.8.3 Save as Shape

The **Save as Shape** command allows the user to save basic objects such as **Line**, **Rectangle**, **Circle**, **Polygon**, **Pie**, **Arc**, **Scale** or multiple shapes as a file in the **Library**. A shape must be selected before saving it as a shape. Multiple shapes may be selected simultaneously.

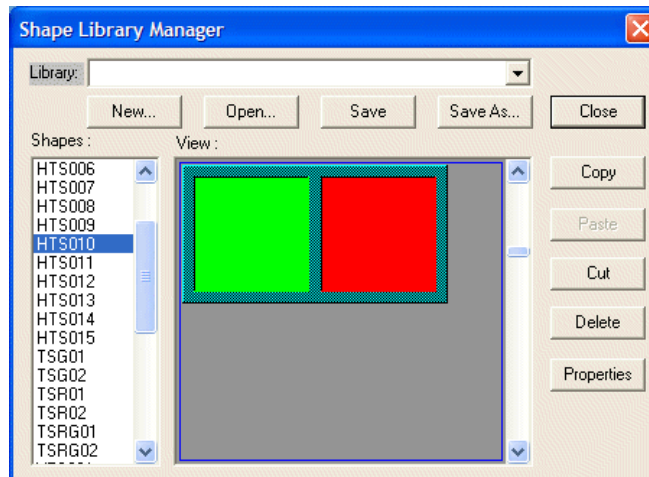
Select **Library/Save as Shape**. You can select the library to save in from the drop-down list and name the shape in the **Shape Name** block.



The Save as Shape dialog box

2.8.4 Shape Library Manager

The **Shape Library Manager** is used to manage shapes in the **Shape Library**.



The Shape Library Manager dialog box

New: Creates a new shape library.

Open: Opens an existing shape library.

Save: Saves the active shape library to a file.

Save As: Allows selection of which file to save to.

Close: Closes the **Shape Library Manager**. If changes have not been saved, a dialog box will be displayed, asking if you want to save.

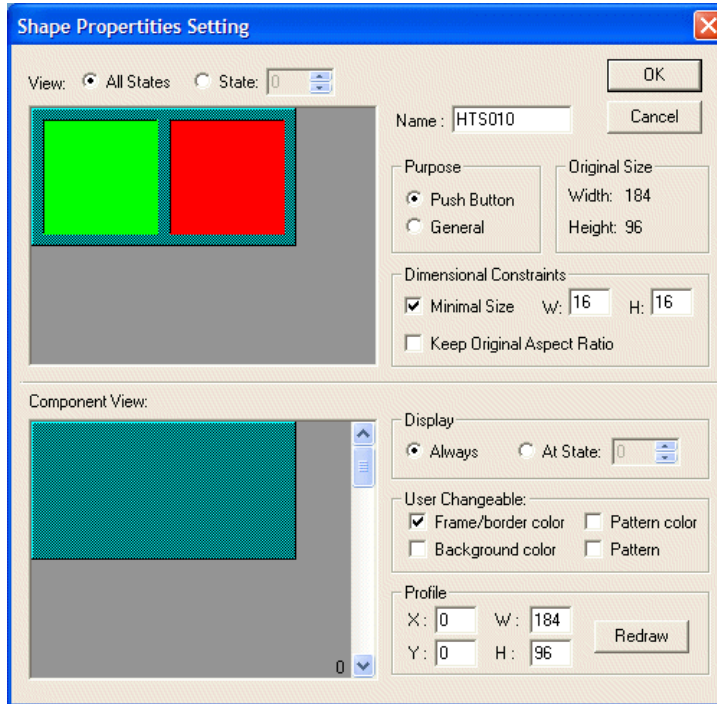
Copy: Copies the selected shape to the clipboard.

Paste: Imports the shape from the clipboard.

Cut: Exports the selected shape to the clipboard.

Delete: Deletes the selected shape.

Properties: Displays the properties of the selected shape.



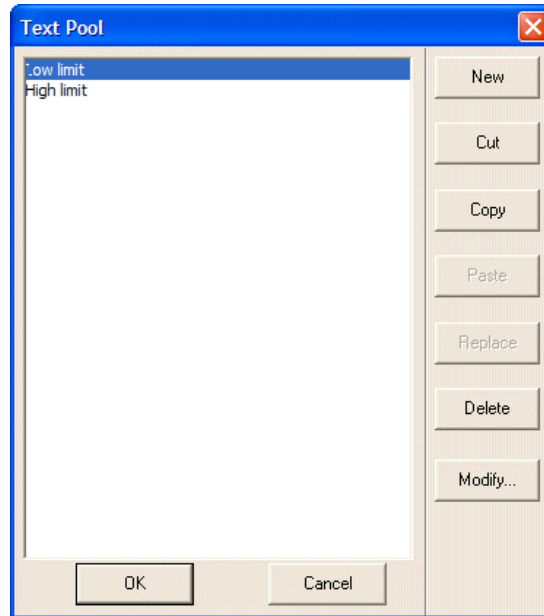
The Shape Properties dialog box

- **View:**
All States: Displays all states of the selected shape.
State: Displays individual states of the selected shape.
- **Name:** Specifies the name of the selected shape.
- **Purpose:** The function of the selected shape; **Push Button** or **General**.
- **Original Size:** Displays the width and height of the original shape.
- **Minimum Size:** Specifies the minimum width and height of the selected shape.
- **Keep Original Aspect Ratio:** Selecting this option keeps the size of the shape in its original ratio.
- **Display:** Sets the state of the selected graphic; **Always** or **At State**.
- **User Changeable:** Allows changing shape properties, including frame/border color, background color, pattern color, and pattern.
- **Profile:** Specifies the position of the component object and displays the specified view here.

2.8.5 Text Pool

The function of the **Text Pool** is mainly to provide common management and editing of the texts used in the application.

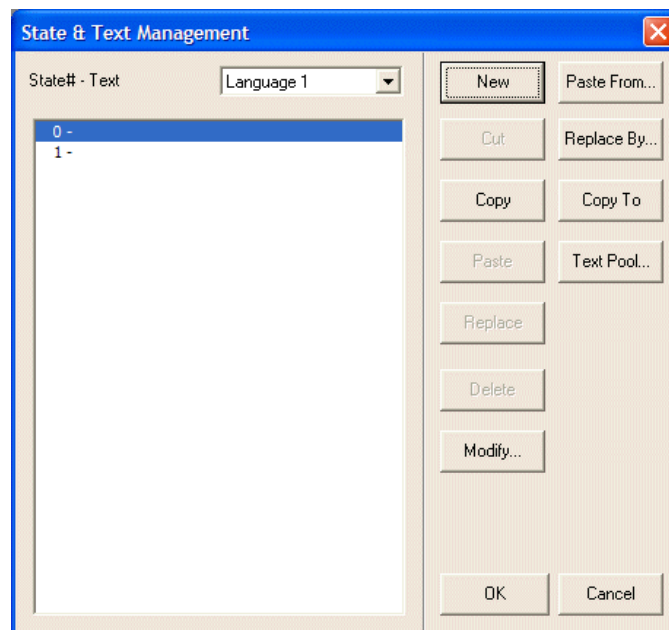
Select Library/Text Pool.



The Text Pool dialog box

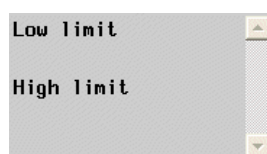
Perform the following steps to edit texts in the **Text Pool**:

1. Enter the desired texts in the **Text Pool** dialog box to save in.
2. Click on the object to edit an object which has texts, as in **Text Pool**, then select **Edit/State and Text Management**.



The State & Text Management dialog box

3. Select **Replace By** to edit. You can select the desired text which was edited in **Text Pool** to display.



The following List object displays the text which was edited in Text Pool.

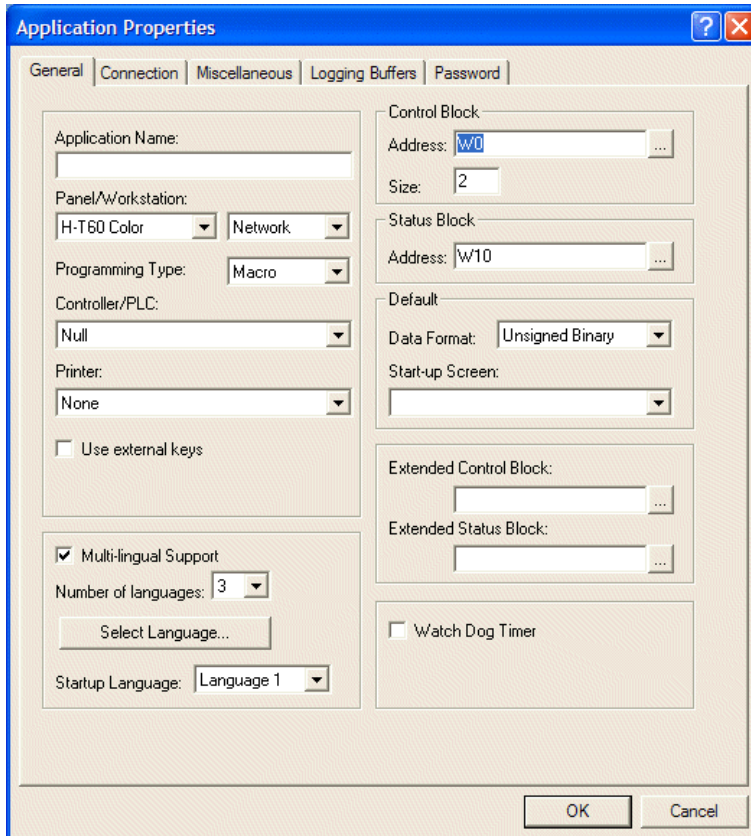
2.9 Application Menu

The **Application** menu is used for general management of the operator terminal and the software. You can set up configuration parameters for the operator terminal, such as controller type, operator terminal type, logging buffers and alarm setup. In addition, **Compile** and **Download** are also available in this menu.

2.9.1 Workstation Setup

Workstation Setup is used to set up the parameters of the operator terminal.

Select **Application/Workstation Setup** to display the following dialog box.



The General tab of the Application Properties dialog box

General Tab

On the **General** tab, you can set up the operator terminal and controller model, startup language and screen, and control block and status block.

Application Name: The name of the application.

Panel/Workstation: Specifies the model of operator terminal. Note the resolution, size, and color while selecting.

Programming Type: Selects **Macro** or **Standard**.

Controller/PLC: Specifies the type of controller the operator terminal will communicate with.

Printer: Specifies the type of printer the operator terminal will print to.

Multi-lingual Support: Check this option to support multi-lingual use and specify the startup language. Supports up to 5 different languages including Arabic, Chinese Simplified, Chinese Traditional, Cyrillic, English, Greek, Japanese, Korean, Thai, Turkish and Western European. Please see section [Language 1-5](#) for setup. The **Multi-lingual Support** function allows that only one application file for a machine that can support up to 5 languages has to be maintained.

Control Block: Specifies the controller address to control, and size. The minimum size is 2 words, the maximum size is 32 words (the maximum size is 6 for recipe). The control block enables the controller to control actions in the operator terminal such as change screen, print, send recipes etc. Please see the chapter [Control and Status Block](#) for complete details.

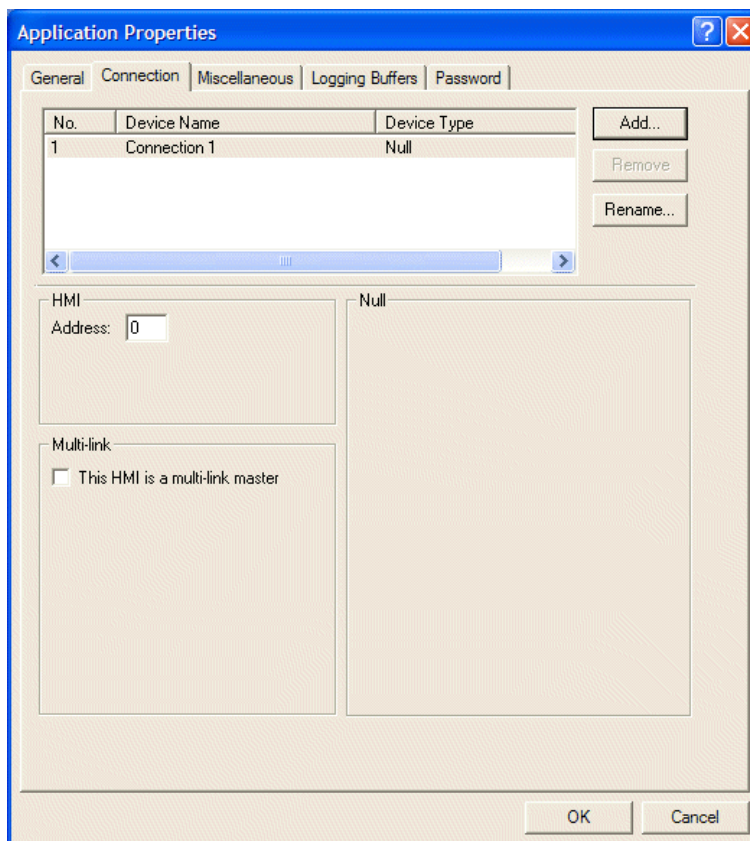
Status Block: Specifies the starting address for the **Status Block**; the fixed size is 10 words. The **Status Block** provides communication between the operator terminal and the controller. The operator terminal will write a continuous block of data. Please see the chapter [Control and Status Block](#) for complete details.

Data Format: Specifies the data format to be read.

Start-up Screen: Specifies the screen to display when the operator terminal starts up.

Connection Tab

On the **Connection Tab**, you can add or delete devices for connection and set up the parameters such as address, connection method and IP address.



The Connection tab of the Application Properties dialog box; Multi-link is selected

Add: Click to add new devices to connect with, including Muti-link, and select controller type to connect with. For the steps or methods related to setup, please see the chapter [Multi-Link: Normal Connection Port](#).

Remove: Removes the connected device. The No. 1 device cannot be removed.

Rename: Modifies the device name and type, but the device type cannot be modified for the No. 1 device. This change has to be made on the **General** tab.

HMI:

Address: Sets up the operator terminal. Once the multi-link is made, the address can be repeated and the range is 0-255.

Port/method used for the connection: Specifies the port and method to connect with the controller or other operator terminal models including **COM1**, **COM2**, **Ethernet (Cross-link)**, **COM1 (Multi-link slave)**, **COM2 (Multi-link slave)** and **Ethernet (Multi-link slave)**.

Multi-link:

- **This HMI is a multi-link master:** Check this option to specify the operator as a master.

Master Port: Specifies the port which connects master with slaves; **COM1**, **COM2** or **Ethernet**.

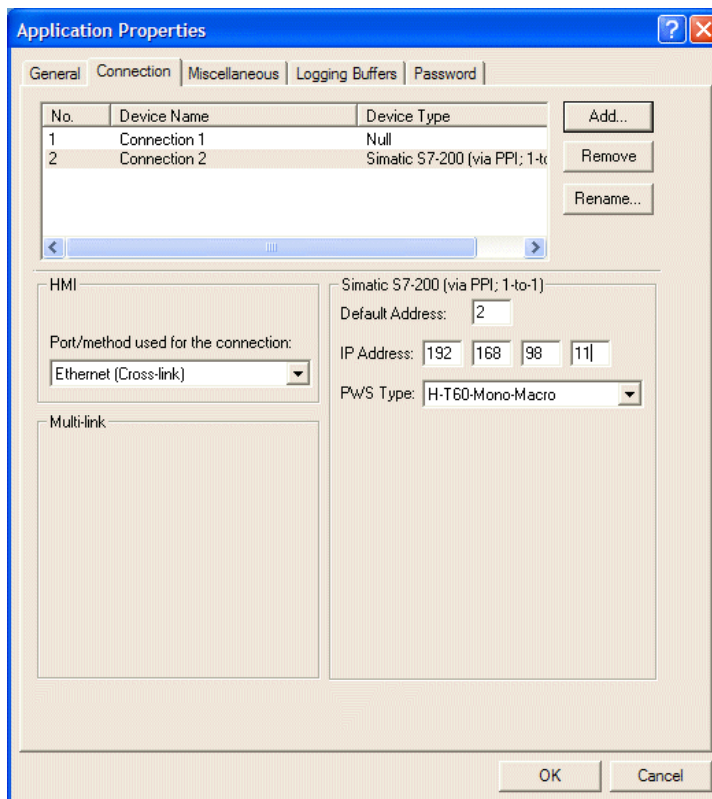
Common Register Block: Specifies the starting location for the Common Register Block (CRB), which master and slaves use.

CRB Size: Specifies the size of Common Register Block.

Common On/Off Block: Specifies the starting location for the Common On/Off Block (COB).

COB Size: Specifies the size of the Common On/Off Block.

Operator terminals arranged into a multi-link (one master; multi-slave) network is not available for all operator terminal models; please see [Appendix A - H-Designer Features and Operator Terminal Models](#) for complete details.



Cross-link and connection to Simatic S7-200 (via PPI; 1-to-1) is selected

IP Address: Specifies the IP addresses of the other operator terminal. The specified operator terminal connects with the controller through the other operator terminal.

HMI Type: Specifies the model to connect with the controller (other operator terminals).

For the setup of Multi-link and Cross-link, please see the chapter [Ethernet Communication](#).

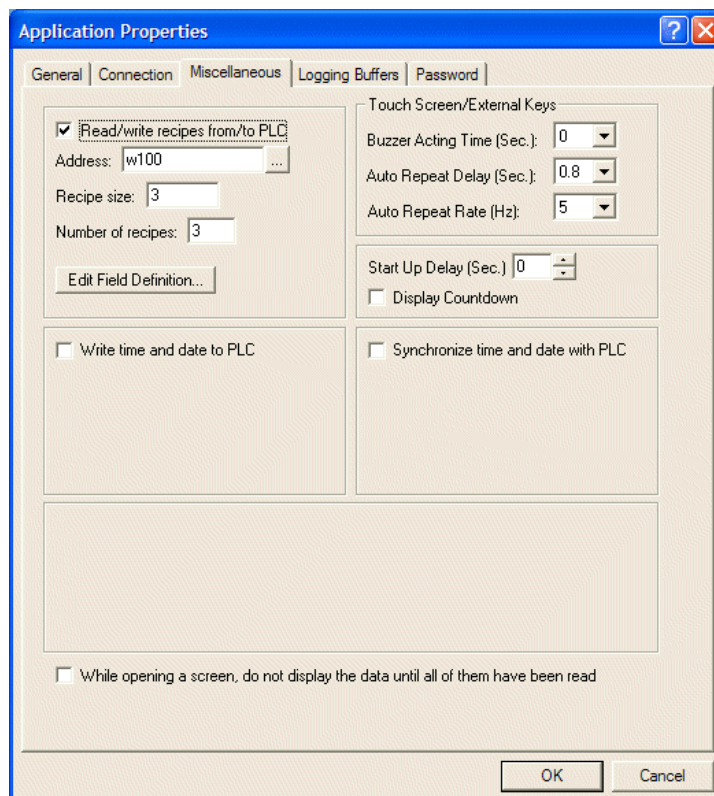
Note:

The transmission parameters for the operator terminal and the controller must be identical as they are linked together. When the controller model is specified, H-Designer will set it up as the controller default, but you must ensure that this setup is identical in the operator terminal.

For the setup of each controller, please see the relevant controller manual or the chapter [Communication between Operator Terminal and Controller](#).

Miscellaneous Tab

On the **Miscellaneous** tab, recipe functions and write time and data to the controller can be set up.



The Miscellaneous tab of the Application Properties dialog box

For setup of recipes, please see chapter [Recipes](#) and section [Recipe Register Block](#).

The recipe function is not available for all operator terminal models; please see [Appendix A - H-Designer Features and Operator Terminal Models](#) for complete details.

Write time and date to PLC: Select this option to enable the operator terminal to write time and date to the real time clock in the controller. Please see section [Time Block](#) for details.

Touch Screen/External Keys: Specifies the format of buffer.

Start up Delay (Sec.): Specifies the length of time before the screen start up.

Logging Buffers Tab

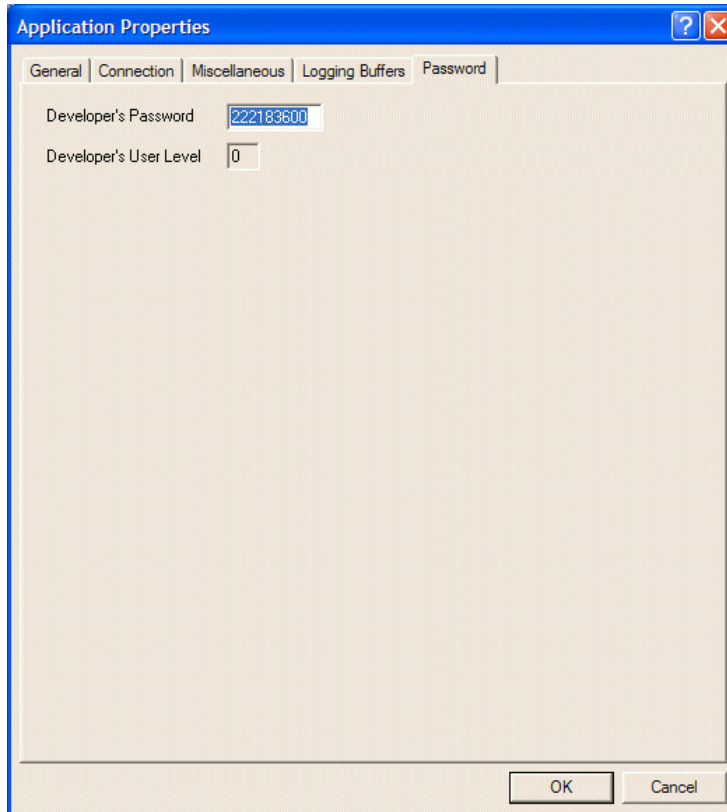
Logging Buffers are used to collect data from the battery backup RAM. It is a continuous data block and there are 12 buffers available.

The object is not available on all operator terminal models; please see [Appendix A - H-Designer Features and Operator Terminal Models](#) for complete details.

When creating a **Historical Display** object, the logging buffer's area and size has to be set up first. Please see section [Historical Display](#) for information.

Password Tab

When you want to copy the screen or upload the application to H-Designer, the operator terminal will ask for the password.

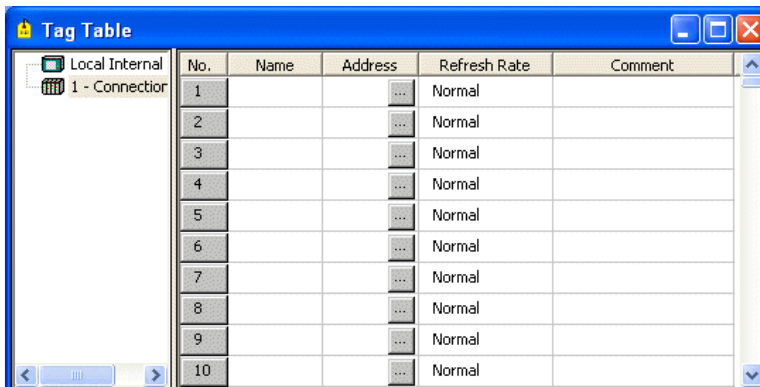


The Password tab of the Application Properties dialog box

2.9.2 Tag Table

The **Tag Table** enables you to provide a name for the controller address and specify the refresh rate.

Select **Application/Tag Table**.



The Tag Table window

2.9.3 Alarm Setup

To use the **Historical Display** object, you must set up its address and parameters first. Then the operator terminal will display the corresponding messages after reading the controller value. Up to 512 messages can be specified.

Select **Application/Alarm Setup**.

No.	Message	ACK	Screen
0		No	None
1		No	None
2		No	None
3		No	None
4		No	None
5		No	None
6		No	None
7		No	None
8		No	None
9		No	None
10		No	None

The Alarm Setup dialog box

Address of Alarm Block: To use a bit (LSB) as a corresponding alarm address. If W130 is the starting position and the number of the alarm is set 160, the operator terminal will monitor 160 bits = 10 words, and this corresponds to W130, W131, W132.....W139. When bit W130 turns on, the operator terminal will sample and record an alarm message.

Number of Alarms: Specifies the number of alarms.

Scan Time: Specifies the sampling time for monitoring the controller data, 1-10 seconds.

Number of Records in Alarm History: Specifies the maximum number of events stored in the alarm buffer. For example, 100 means that when the 101st alarm event occurs, the first alarm message will be overwritten.

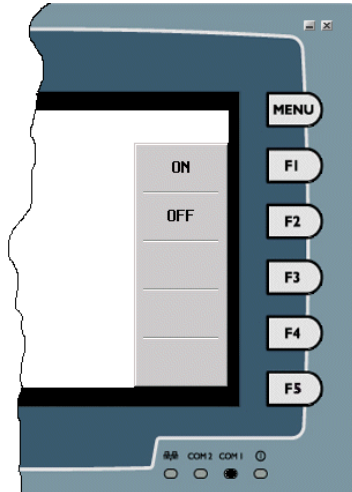
Table:

- **Message:** Enter the text to the alarm message. The format can be modified in its dialog box. Up to 512 alarm messages can be set.
- **ACK:** Acknowledge the message which has been received to conceal the alarm.
- **Screen:** Specifies the screen to display when the alarm occurs.

2.9.4 Slide-out Menu

The function of the **Slide-out Menu** is to operate the functional keys, such as the **Set Button**, **Reset Button** and **Momentary Button**, in a convenient way in the operator terminal. The number of functional keys depends on the selected operator terminal model.

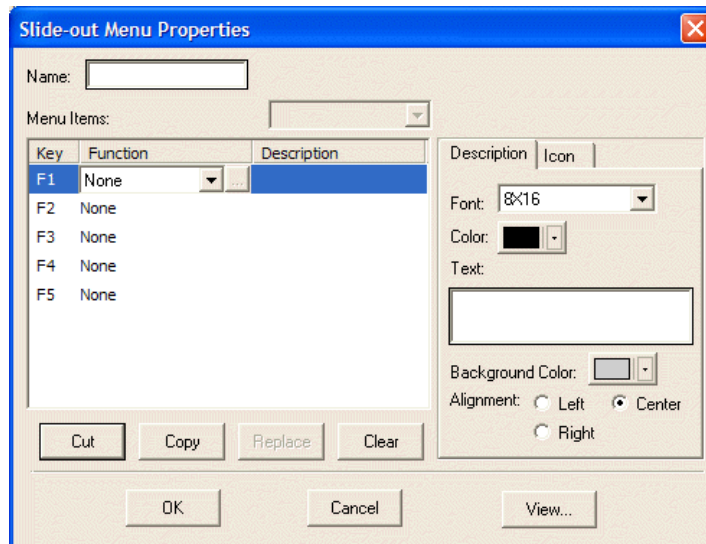
When the **Menu** button is clicked, the operator terminal will display the slide-out menu with its specified functional keys.



The Slide-out Menu

The object is not available on all operator terminal models; please see [Appendix A - H-Designer Features and Operator Terminal Models](#) for complete details.

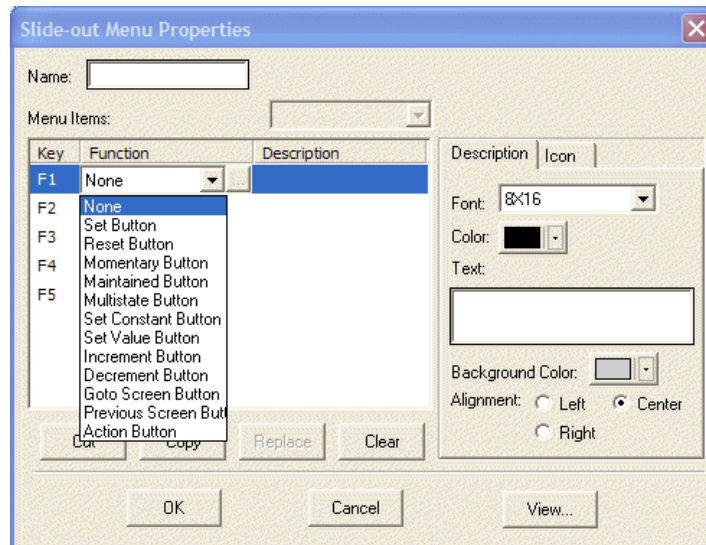
Select **Application/Slide-out Menu**. Click **New** to display the following dialog:



Setup of the Slide-out Menu

Name: Enter the name for the slide-out menu.

Function: Select a function from the drop-down list.



Selecting a function for a button on the Slide-out Menu

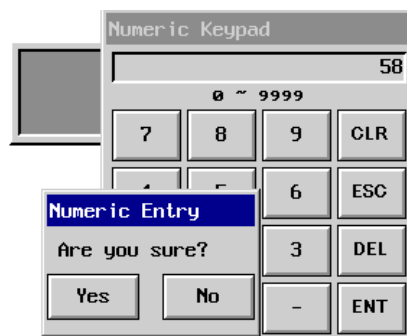
Cut, Copy, Replace, Clear: Use the buttons to cut, copy, replace and clear the button's content.

Description: Enter the name for the button in the Text block.

View: Click the View button to view the created slide-out menu.

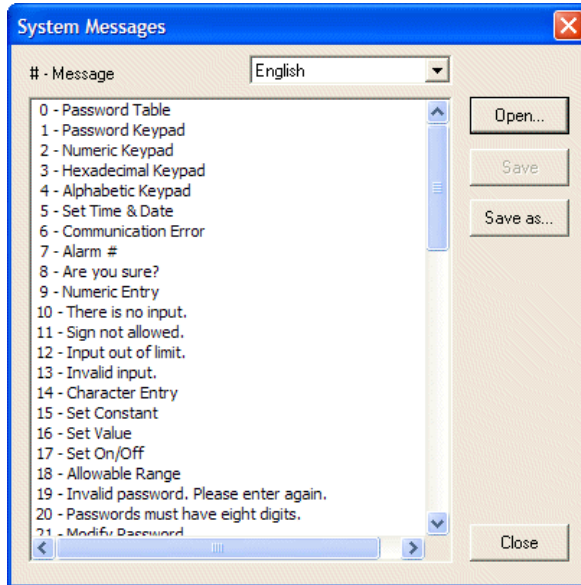
2.9.5 System Message

The **System Message** is used to edit messages for the operator terminal system. When the **Operator Confirmation** option is selected, executing the object will display its system message. For example, the system message **Are you sure?** can be displayed after the numeric entry.



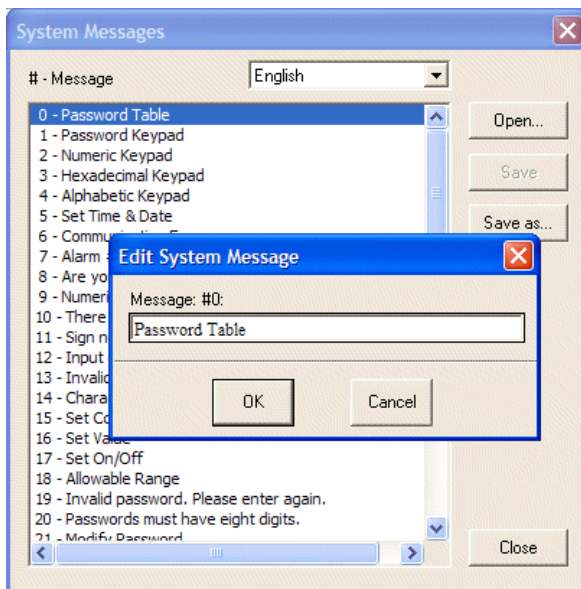
An example of a System Message

Select **Tool/System Messages**.



The System Messages edit window

Double-click on the message block; the message can be modified in its dialog box as shown below.



Editing System Message

The **Save as Default** button is to save the system messages as a default file (*.PSM).

2.9.6 Macros

Macros enable the operator terminal to execute a number of tasks including flow control, data transfer, conversion, counter, system service instructions, etc. Using macros can not only help you communicate to the controller but also connect to other devices. This feature provides an efficient integration system as well as an economical structure for hardware application. In addition, using macros can also significantly reduce program size and optimize controller efficiency.

Please see the chapter [Macros](#) for complete details.

There are three macro options in the **Application** menu; **INITIAL Macro**, **BACKGROUND Macro** and **CLOCK Macro**.

INITIAL Macro

When the operator terminal runs the application for the first time (this means the first time the application is executed after power off), this macro is executed once. The purpose of **INITIAL Macro** is data initialization, communication parameters declaration etc.

BACKGROUND Macro

When the operator terminal runs the application, the command will be executed cyclically. A maximum 30 lines of macro commands can be executed at once. Whatever the screen is, the macro commands will be executed. The purposes of the **BACKGROUND Macro** include communication control, data conversion etc.

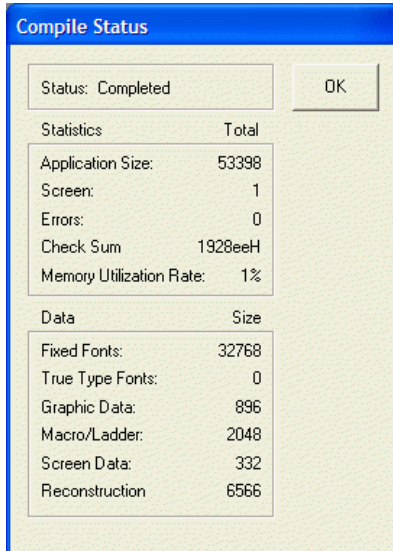
CLOCK Macro

When the operator terminal runs the application, the entire macro will be executed once every 500 ms. The purpose of the **CLOCK Macro** is screen control, bit setting, command control, data transfer etc.

2.9.7 Compile

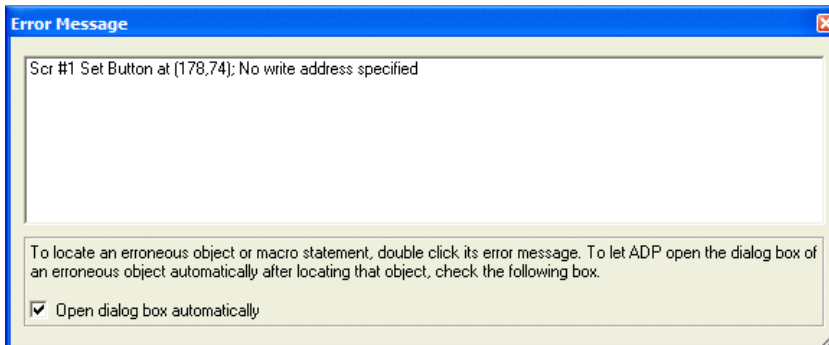
The **Compile** button is used to test the application to see if any errors happen before the application is executed. After correcting the errors, the application can be executed.

Select **Application/Compile**.



The Compile Status dialog box

After clicking OK the **Error Message** dialog box will be displayed on the screen.

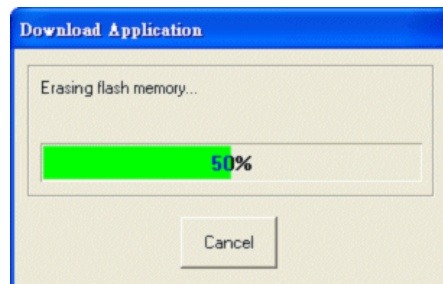


The Error Message dialog box

Double-click the error message to display the incorrect object or macro on the screen. Alternatively, check **Open dialog box automatically**, to automatically display the incorrect object.

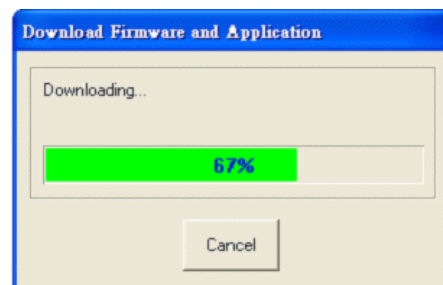
2.9.8 Download Application and Download Firmware and Application

Download Application is used to download the update application and screen to the operator terminal.



Downloading application

Download Firmware and Application is used to download the firmware and application to the operator terminal. This option has to be selected the first time you download the application.



Downloading firmware and application

If the connected operator terminal is not the same as the model specified in the project, a message box will appear on the screen.

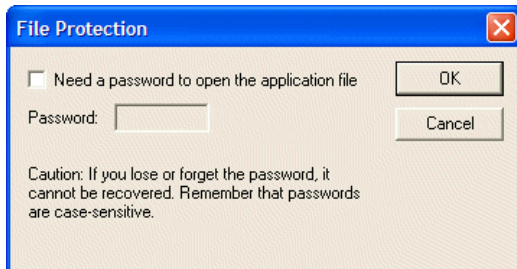
Note:

Remember to execute **Compile** before downloading.

2.9.9 File Protection

File Protection is used to protect the application; you must enter the password to open the application file.

Select **Application/File Protection**.



Setting a password for file protection

Note:

This password is used to protect the application file from being modified by unauthorized users, and is totally different from the password set on the [Password Tab](#). That password provides security against copying and uploading.

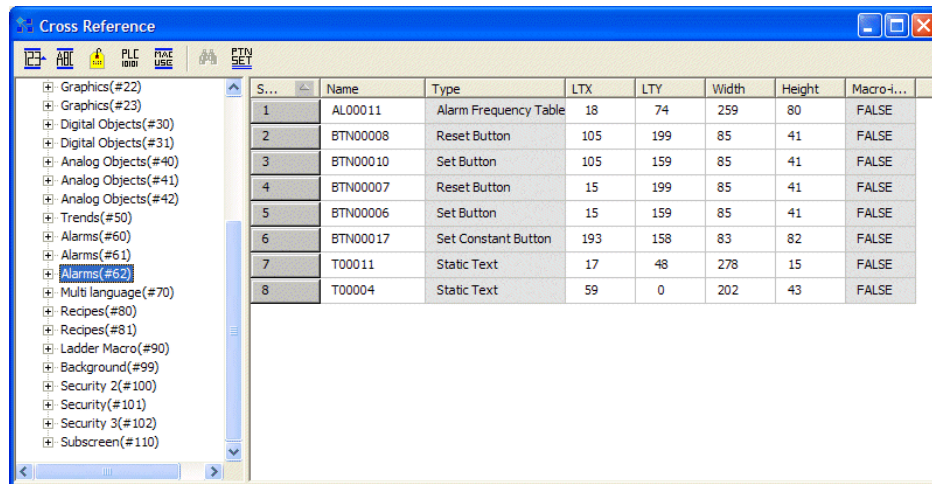
2.10 Tool Menu

The **Tool** menu is used to manage and simulate the application, and to edit recipes.

There are four options in the **Tool** menu: **Cross Reference**, **Off-line Simulation**, **On-line Simulation** and **View/Edit Recipe**.








2.10.1 Cross Reference

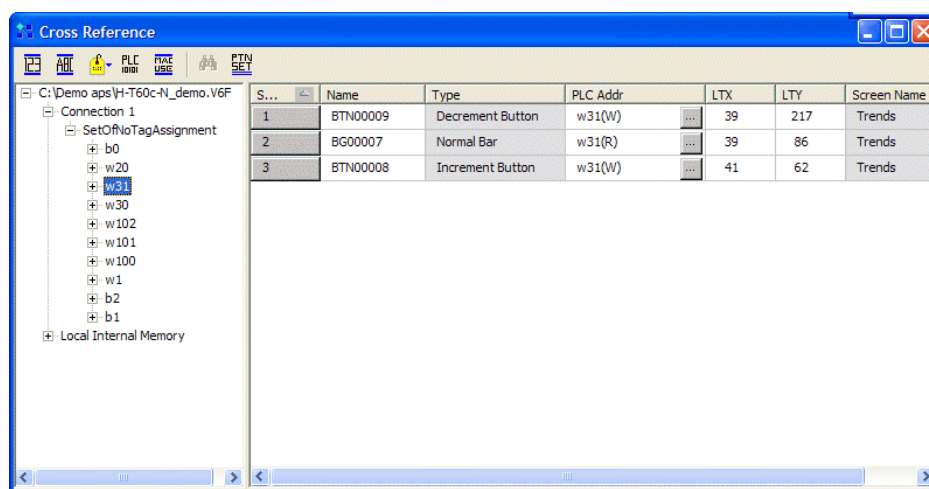
The **Cross Reference** tool helps you identify the screen name, screen number, controller address, tag address or macro-in use quickly.



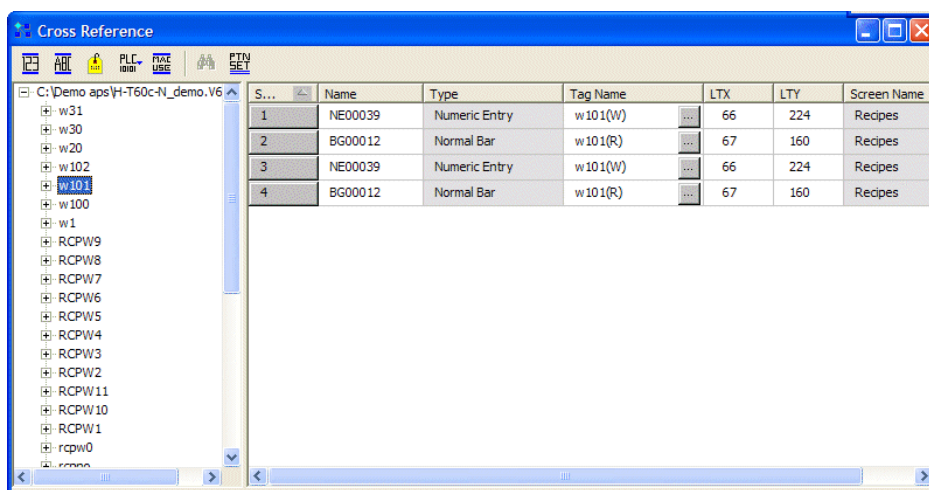
The Cross Reference window

The icons can be used to sort according to screen name, screen number, controller address, tag address or macro-in use. The right-hand table will list its objects and properties. The preview window below will display the selected object. Click on a heading of a column in the table (such as **Serial No**, **Name** or **PLC Addr**) to sort the properties in ascending or descending order.

Icon	Description
	Select this icon to sort by screen number.
	Select this icon to sort by screen name.
	Select this icon to sort by tag name.
	Select this icon to sort by controller address in ascending order; click the address to list the object.
	Select this icon to sort by Macro-in use objects.
	Select this icon to sort by search pattern. This feature is only available when searching controller addresses or tag names.
	Set the desired pattern for search. Select by PLC Addr or by Tag Name , Exact match or Partial match .

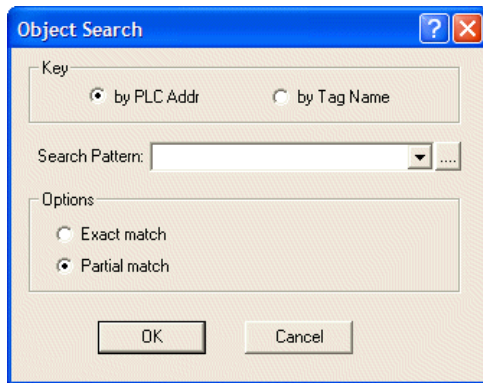


Sorting by Tag Name



Sorting by Address; clicking the address displays the object

When searching controller addresses or tag names, the search pattern can be specified. If the search pattern is **W**, and **Partial match** is selected, the cross reference will display objects with the controller addresses **W** - **W0**, **W1**, **W20**, **W60**.



Specifying the search pattern

2.10.2 Off-line and On-line Simulation

H-Designer supports two types of simulation; **Off-line Simulation** and **On-line Simulation**. They both offer simulation of the operator terminal in the PC.

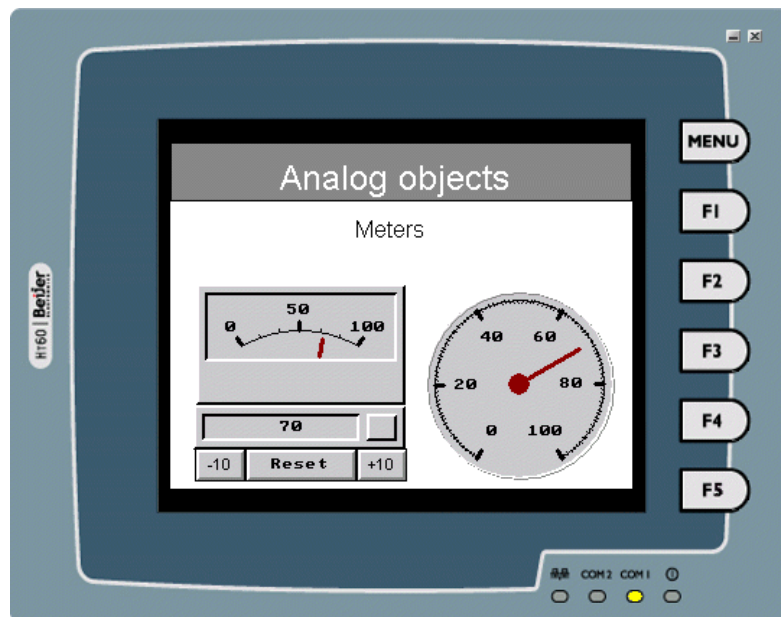
Note:

The application must compile before simulation.

Off-line simulation is available with all controllers which H-Designer offers, but on-line simulation is only available on some controllers.

Off-line Simulation

Off-line Simulation can be used to present results on a PC with the same operation mode as between the operator terminal and the controller.



H-T60 in off-line simulation

Off-line simulation is without communication with the controller, and provides the following benefits:

1. Before purchasing, you can simulate operation and recognize the operator terminal functions sufficiently.
2. Before downloading, you can simulate on the PC to test the application, including screen change, button functions and display etc.
3. Before the completion of the controller program, the operator terminal application can be presented to the customer.

On-line Simulation

H-Designer offers user **On-line Simulation** for connection between the PC and the controller communication ports. If there is only one RS232C serial port in the PC, the user needs to add an adapter for transferring the signal from RS232C to RS422 or RS485 in order to connect with the RS422 or RS485 port in the controller. Note that the communication time between H-Designer and controller is 60 minutes. If you need to connect again, please close the H-Designer and restart it.

2.10.3 View/Edit Recipes

Select **Tool/View/Edit Recipes** to display the recipe editor window.



The Recipe Editor window

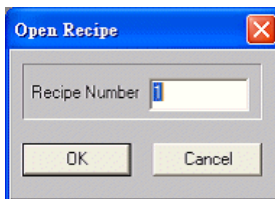
Note that the recipe must upload the file from the operator terminal to a PC and save the file. The data size and total number of of recipes cannot be modified.

The object is not available on all operator terminal models; please see [Appendix A - H-Designer Features and Operator Terminal Models](#) for complete details.

For setup procedures, please see section [Application Menu](#) and chapter [Recipes](#).

Select **File/Open** to open the selected recipe file. Note that the recipe file is stated as *.RCP.

Open the recipe files, then select **Recipe/Open Recipe**.



Enter the recipe number for the recipe.

The recipe is displayed on the screen for you to edit.

No.	BCD	Word Bin	UBin	Double Word Bin	UBin
0	1	1	1		
1	0	0	0		
2	0	0	0		
3	0	0	0		
4	3	3	3		
5	0	0	0		
6	1	1	1		
7	0	0	0		
8	0	0	0		
9	2	2	2		

Editing Recipe #1

The function of the recipe editor is the same as a common edit tool; it includes open, save, print, view the recipe file, and window arrangement.

2.11 Options Menu

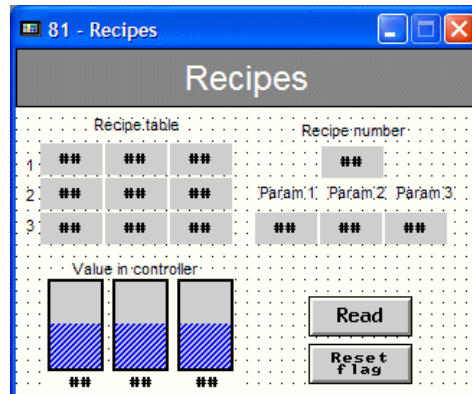
The **Options** menu provides options for editing and transmission.

2.11.1 Snap to Grid

If you select **Snap to Grid** on the edit screen, the edited objects will align to the nearby grid (see section [Display Grid](#)). This command is convenient for aligning objects.

2.11.2 Display Grid

When selecting this option, the edit screen will display a grid to allow easy alignment.



A screen with a visible grid

2.11.3 Grid Attributes

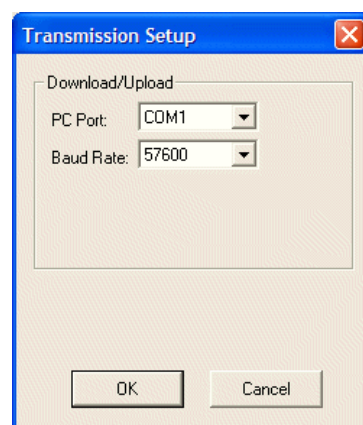
Select this option to specify the grid size. The bigger the grid size is, the longer the distance between points will be.

2.11.4 Transmission Setup

Select **Options/ Transmission Setup** to specify the download/upload port and baud rate between the PC and the operator terminal. The PC port options are **Ethernet**, **COM1.... COM16** and **USB**. A baud rate of **115200** is recommended.

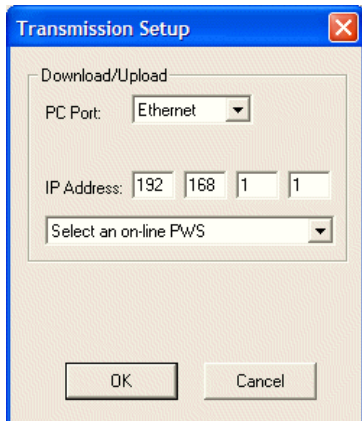
Note:

These settings are not the same as the ones set on the **Connection** tab in **Application/ Workstation Setup**. The former is the setting between the PC and the operator terminal, the latter is the setting between the operator terminal and the controller.



The Transmission Setup dialog box

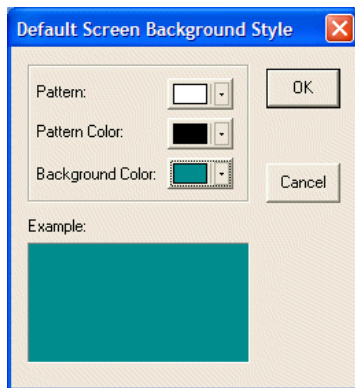
If **Ethernet** is selected for **PC Port**, you need to input the PC address or select from the drop-down list.



Selecting Ethernet for PC Port

2.11.5 Default Screen Background Style

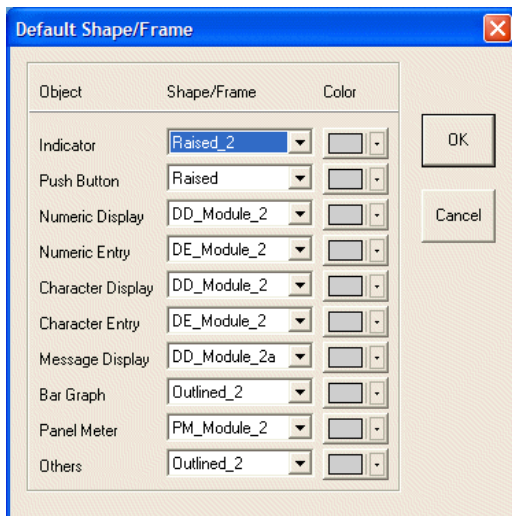
Select **Options/Default Screen Background Style** to specify the pattern, pattern color, and background color for all screens. The default setting will be displayed in all screens except for the special edit screen.



The Default Screen Background Style dialog box

2.11.6 Default Frame Styles

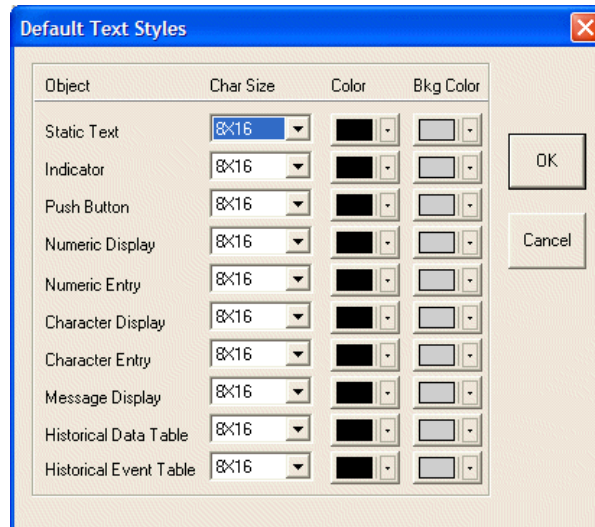
Select **Options/Default Frame Styles** to specify the different object types with their own shape/frame. Then those objects will be shown as specified on the screen.



The Default Frame Styles dialog box

2.11.7 Default Text Styles

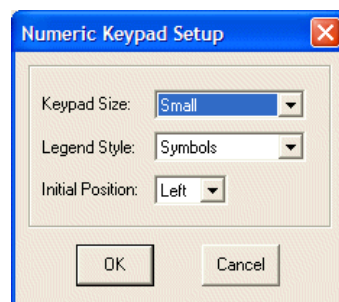
Select **Options/Default Text Styles** to specify character size, color and background color for each type of object. The objects will be displayed with their settings on the screen.



The Default Text Styles dialog box

2.11.8 Numeric Keypad Setup

Select **Options/Numeric Keypad Setup** to set up the numeric keypad on the screen (displayed for example when clicking the **Numeric Entry** object).



The Numeric Keypad Setup dialog box

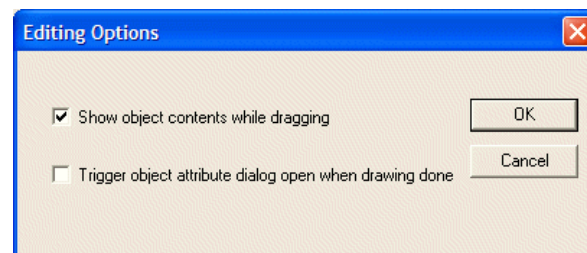
Keypad Size: Small or Large can be selected.

Legend Style: Symbols or Chinese Characters can be selected.

Initial Position: Left or Right can be selected.

2.11.9 Editing Options

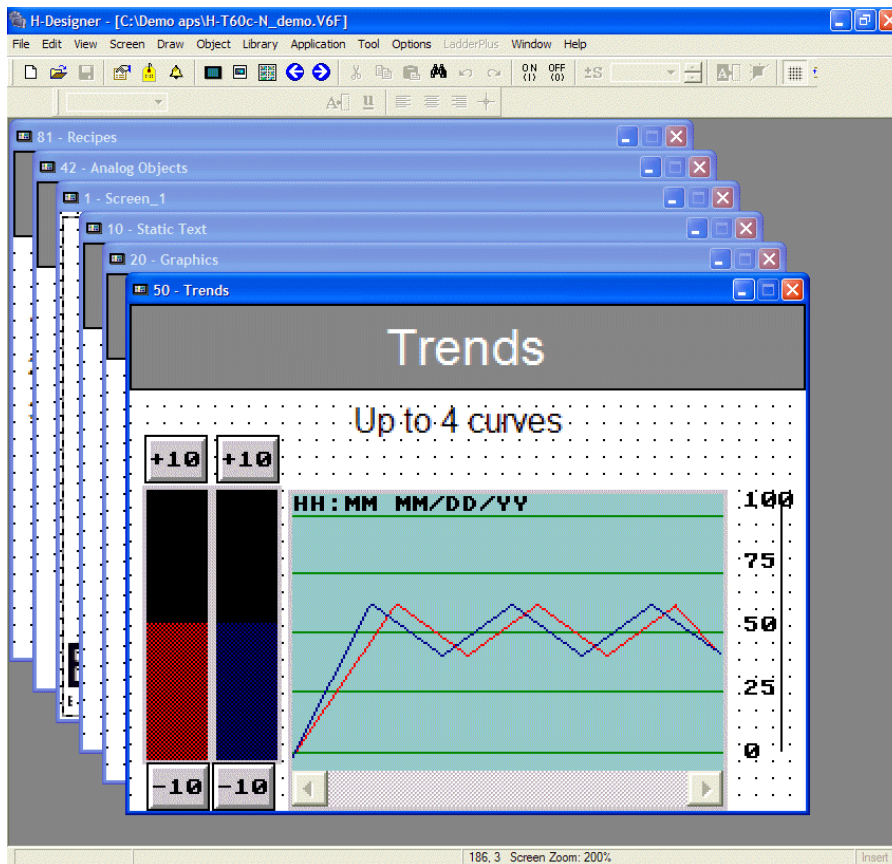
Select **Options/Editing Options** to set up the edit environment here.



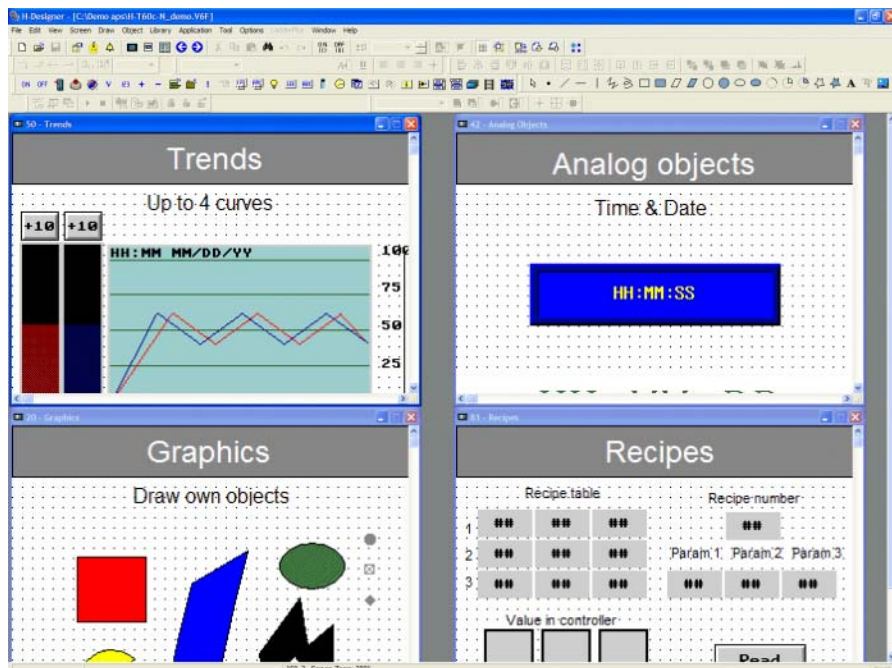
The Editing Options dialog box

2.12 Window Menu

The **Cascade**, **Tile** and **Close All** options available in the **Window** menu and all open screens are listed.



Cascade is selected for screen display



Tile can be used when copying between screens and for an overview of screens

Close All is used to close all open screens at once; the screens will not be saved. The application is not closed.

2.13 Help Menu

The **Help** menu offers complete details and instructions about **Macros** and **Ladder-Plus**.

Selecting **Help/About**, displays the version number and copyright information of **H-Designer**.

3 Recipes

Recipes include blocks of similar systematic data. Because of the similarities, users can edit them as a group of recipes for convenient transmission. In this way, users can send data efficiently and accurately.

Note:

Recipes are not available for all operator terminal models: please refer to [Appendix A - H-Designer Features and Operator Terminal Models](#) for complete details.

3.1 Example

The following coating equipment is used to spray paint on different parts. The paint colors are limited to white, red, blue, dark, and mixed color (1 = spray, 0 = no spray). Recipes can be used to present and save data to simplify matters.

Coating Equipment

Color	White	Red	Blue	Black	Time
Top	1	0	0	0	3
Bottom	0	1	0	0	2
Left	0	1	1	0	1
Right	0	0	0	1	1

This recipe data has five variables: **White**, **Red**, **Blue**, **Black** and **Time**.

The following recipes are to be created:

Recipe 1 paints the top in white for 3 minutes

Recipe 2 paints the bottom in red for 2 minutes

Recipe 3 paints the left side in purple (red+blue) for 1 minute

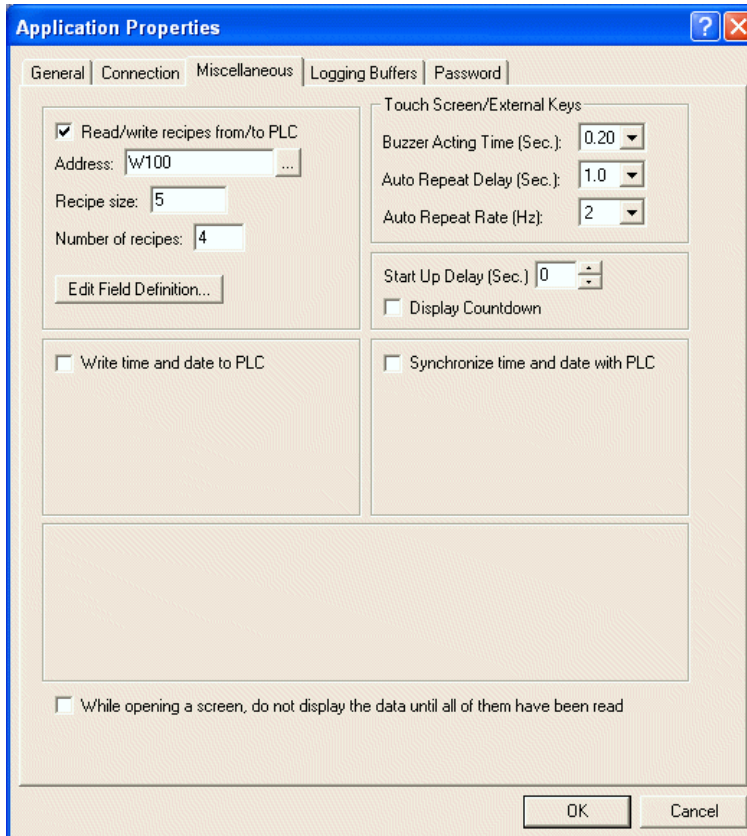
Recipe 4 paints the right side in black for 1 minute

A variable represents a word, the recipe size is 5, and the number of recipes is 4.

3.2 Recipe Operation Steps

This section will illustrate the operation and application of recipes. We will continue to use the coating equipment from section 3.2 as an example.

1. First, a **Recipe Register** has to be defined. Select **Application/Workstation Setup** and enter the PLC address, Recipe size, and Number of recipes on the **Miscellaneous** tab.



Setting Recipe size = 5 and Number of recipes = 4 in this example

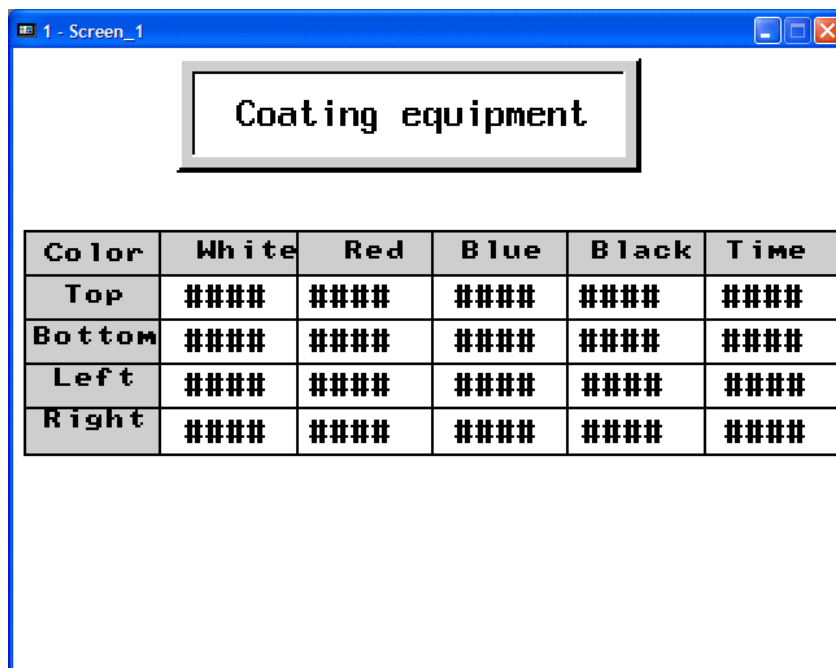
According to the setup above, the starting address of the controller recipe register is **W100** and its size is 5 words. The starting address of the current recipe in the operator terminal is **RCPW0** and its size is 5 words. The starting address of the operator terminal RAM is **RCPW5** and its size is 5×4 words.

If the recipe write flag is ON, the operator terminal will write the current recipe from **RCPW0-RCPW4** in the operator terminal to **W100-W104** in the controller. If the recipe read flag is ON, the operator terminal will write the recipe from controller **W100-W104** to **RCPW0-RCPW4** in the operator terminal. If the user wants the operator terminal to read/write the recipe data from/to the controller, **RNR (Wn+5)** must be defined.

For instructions about operator terminal recipe registers, please see section [Recipe Register Block](#).

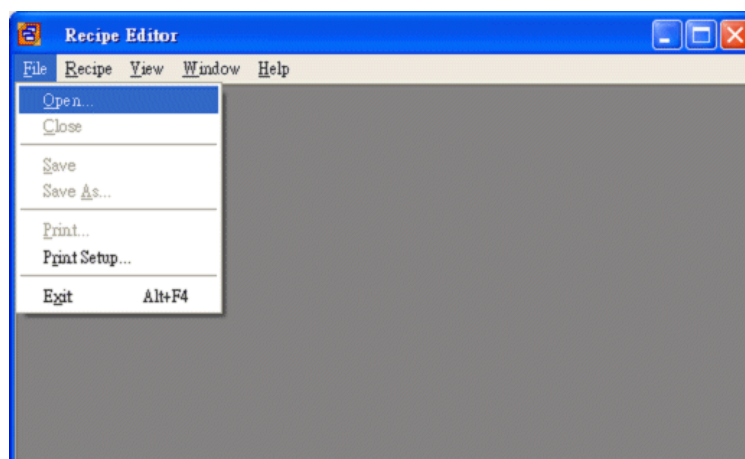
- The application can be edited and saved as a *.V6F file. In the example, a **Numeric Entry** object is used to display the coating equipment data in the operator terminal.

RCPW5-RCPW9 represent top recipe data
 RCPW10-RCPW14 represent button recipe data
 RCPW15-RCPW19 represent left side recipe data
 RCPW20-RCPW24 represent right side recipe data



The Coating Equipment screen

- Download the H-Designer file to the operator terminal. First, select **Download Application** in the operator terminal, then select **Application/Download Firmware and Application** in H-Designer.
- Then upload the recipe from the operator terminal to H-Designer. Select **Upload Recipes** in the operator terminal, and then select **File/Upload Recipes** in H-Designer. The recipe will be uploaded to H-Designer and saved as *.RCP.
- Once the above steps have been completed, the recipe file can be opened to edit. Select **Tool/View/Edit** to display the H-Designer **Recipe Editor** dialog box.



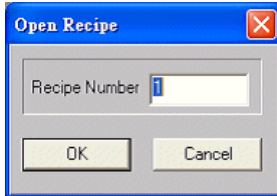
The Recipe Editor dialog box

6. Select **File/Open** and select the desired recipe file (e.g. **painting.RCP**).

Note:

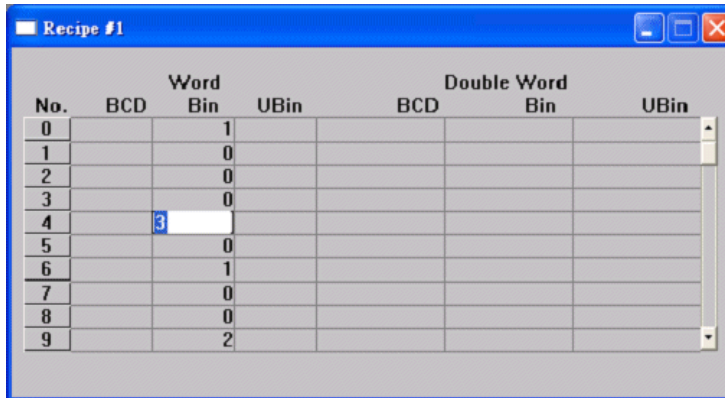
The *.RCP recipe files must be uploaded and saved in the operator terminal first; the recipe size and number of recipes cannot be modified

7. To edit the recipe select **Recipe/Open** and enter the recipe number to open. The coating equipment examples are recipe numbers 1 - 4.



Specifying recipe number

8. The dialog box appears on the screen. The data can be edited in the dialog box. Note that the count for editable data is contained in data size. After editing, select **File/Save** to save the data. For example, Recipe 1 is (1,0,0,0,3), Recipe 2 is (0,1,0,0,2), etc.



Editing the recipe data

9. Select **Download Recipes** in the operator terminal, and then select **File/Download Recipes** to download the recipe file.

Finally, select **Run** in the operator terminal. The operator terminal will display the recipe filled with the same data as displayed on the screen in H-Designer. The coating equipment example includes painting methods and time.

Coating Equipment

Color	White	Red	Blue	Black	Time
Top	1	0	0	0	3
Bottom	0	1	0	0	2
Left	0	1	1	0	1
Right	0	0	0	1	1

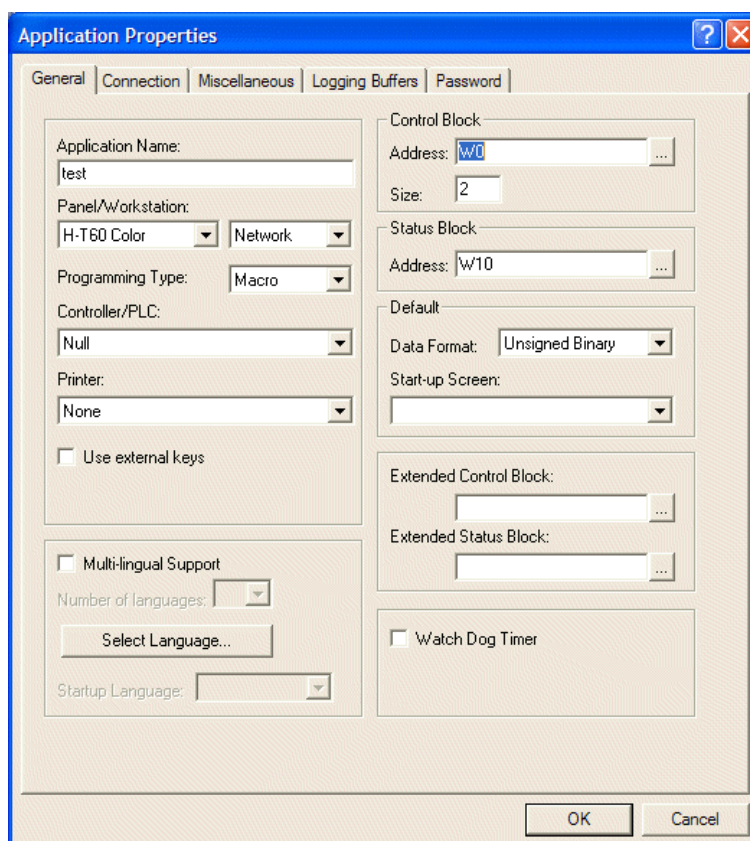
The recipe data in the operator terminal

3.3 Recipe Controlled by Controller

Once the recipe is completed, the controller can be assigned to control the recipe through the communication link between the controller and the operator terminal. The controller can read/write the recipe from/to the operator terminal. This section will introduce the controller setup and corresponding execution related to the operator terminal. For an introduction to communication between the controller and the operator terminal, please see the chapter [Control and Status Block](#) for complete details.

The main steps to read a recipe from the controller to the operator terminal:

1. Set up two continuous blocks; one is the control block, the other is the status block. Select **Application/Workstation Setup** in H-Designer, enter the controller address and its size for **Control Block** and **Status Block** on the **General** tab. The control block size must be set to 6 words.
2. Taking the coating equipment as an example, the control block for the NULL controller W0-W5, the size is 6, and the status block is W10-W15.



Setting up control block and status block

For the properties not explained in this section, please see the sections [Control Block](#) and [Status Block](#).

The recipe register is defined as in section [Recipe Operation Steps](#), the starting address is W100 and the size is 5.

The following are the words and their names in the controller. The words related to read/write recipes are highlighted with a light blue background. Please see the chapter [Control and Status Block](#) for complete details.

Controller Internal Data Block		
Word	Bit	Member
W0	0 1 2 13 14 15	SNR
W1 4 5 6	CFR
W2	.	LBCR#1
W3	.	LBCR#2
W4	.	LBCR#3
W5	0 1 2 13 14 15	RNR
	.	
	.	
W10	0 1 2 13 14 15	SSR
W11 4 5 6	GSR
W12	.	LBCR#1
W13	.	LBCR#2
W14	.	LBCR#3
W15	0 1 2 13 14 15	RIR
	.	
	.	
	.	
	.	
W100	1	1st recipe, 1st word
W101	0	1st recipe, 2nd word
W102	0	1st recipe, 3rd word
W103	0	1st recipe, 4th word
W104	3	1st recipe, 5th word
	.	
	.	
	.	

CFR bit # 4 is the Recipe Write Flag; bit # 5 is the RCPNO Change Flag;
bit # 6 is the Recipe Read Flag.

GSR bit # 4 is the Recipe Write Status; bit # 5 is the RCPNO Change Status;
bit # 6 is the Recipe Read Status.

- Set up RNR to read recipe # N. W5 in the controller is assigned to read recipe # N from the operator terminal. For example, 1st recipe N = 1.
- Then set RCPNO Change Flag to ON for about 1 second. The operator terminal's internal RCPNO and Current Recipe will be changed to read recipe # N. Remember to set the RCPNO Change Flag to OFF before re-triggering.

Status Block: When the value of RCPNO is changed, the value of RIR (W15) will also be changed. The current recipe # N can be checked on the controller. Moreover, if the RCPNO Change Flag is set to ON, the GSR bit (W11 bit # 5) will also be set to ON. The status bit will turn OFF automatically after RCPNO is changed.

For details on addressing recipe data, please see the chapter [Addressing Recipe Data - Enhanced Operator Terminals](#).

Operator Terminal Data Register

Word	Recipe Data	Member
RCPW0	1	Current Recipe
RCPW1	0	
RCPW2	0	
RCPW3	0	
RCPW4	3	
RCPW5	1	Recipe # 1
RCPW6	0	
.	0	
.	0	
RCPW9	3	
RCPW10	0	Recipe # 2
RCPW11	1	
.	0	
.	0	
RCPW14	2	
.	.	.
.	.	.
.	.	.
.	.	.
RCPW20	0	Recipe # 4
RCPW21	0	
.	0	
.	1	
RCPW24	1	
.	.	.
.	.	.
RCPNO	1	Specified Recipe # N
.	.	.
.	.	.

- Finally, set the Recipe Write Flag to ON. The operator terminal will then write the Current Recipe to the controller. The recipe data will be saved in the designated **Recipe Register Block**. Remember to set the Recipe Write Flag to OFF before re-triggering. In this example, set W1 bit # 4 to ON for about 1 second. The operator terminal will then write the Current Recipe to controller W100-W104.

Status Block: When the operator terminal has written a recipe, the GSR bit (W11 bit # 4) will be set to ON automatically. Likewise, if the Recipe Write Flag is set to OFF, the GSR bit will also be set to OFF.

Once the steps above have been completed, the controller can read one cycle of a recipe from the operator terminal. Remember to reset the flag to OFF each time in order to trigger the flag.

Follow the steps above to set the RNR, RCPNO Change Flag and Recipe Read Flag values and to read a recipe from the controller to the operator terminal.

3.4 Recipe Controlled by Operator Terminal

Using objects on the operator terminal display makes it convenient to control read and write actions on recipe data, performed in the controller. Please see the chapter [Control and Status Block](#).

This section uses coating equipment as an example of controlling recipe data in the controller. The following is an illustration of the coating equipment setup on the operator terminal.

Coating equipment					
Color	White	Red	Blue	Black	Time
Top	0	0	0	0	0
Bottom	0	0	0	0	0
Left	0	0	0	0	0
Right	0	0	0	0	0

Recipe no:

1

Confirm

Recipe Write

Recipe Read

An illustration of a coating equipment screen in the operator terminal

Object Design steps:

1. Design an object for the user to enter as recipe N and write recipe N to controller RNR.

In H-Designer, select **Object/Numeric Entry** and enter the address of RNR in the **Write** box.

Using the coating equipment as an example, the address of RNR is W5; so the controller recipe N will write to W5. It is also possible to enter RCPNO directly into a Numeric Entry object. This can only be changed locally in the operator terminal.

Numeric Entry

ID: NE00068

Frame: DE_Module_2

Color: [Select...]

Bkg. Color: [Select...]

Variable Write: RCPNO

Format: Unsigned Binary

Read: [Select...]

Notification: [Select...]

Before Writing [] After Writing [X]

Enabled By: [Select...]

Display Format: Font: 8x8

Character Color: [Black]

Left [] Center [X] Right []

Fill Leading Zeros []

Decimal Pt. Position: 0

Scaling []

Integral Digits: 2

Fractional Digits: 0

Validation and Security: Variable input limits []

Min.: 0

Max.: 99

User Level: 9

Operator Confirmation [X]

Waiting Time (Sec.): 5

External Key: [Select...]

OK Cancel

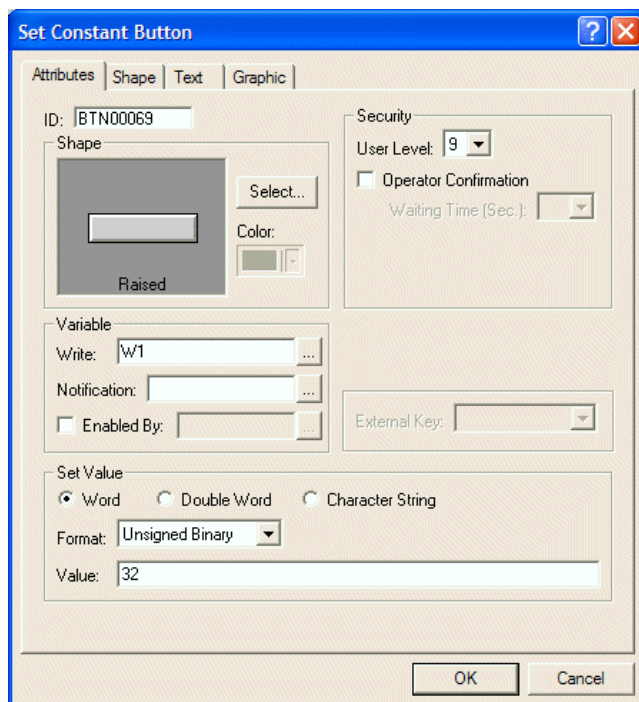
Setting up the address to write recipe N to controller RCPNO

- Design an object that can confirm the designated recipe and write the commands to controller CFR bit # 5 RCPNO Change Flag; set the flag to ON.

In H-Designer, select **Object/Push Button/Set Constant**. First, enter the address of CFR in the **Write** box. Then enter the constant value in the **Value** box to set its register bit.

Using the coating equipment as an example, the address of CFR is W1. The RCPNO Change Flag is located in CFR bit # 5.

The constant is set to 32 ($2^5 = 32$), so the RCPNO Change Flag will be set to ON.

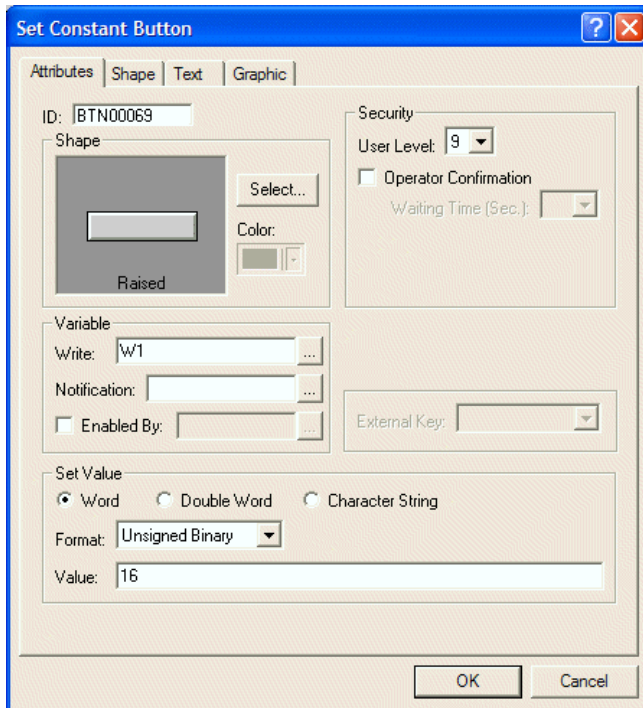


Setting RCPNO Change Flag to ON

- Finally, design an object which can set the CFR bit # 4 Recipe Write Flag to ON and write the Current Recipe to the controller.

Using the coating equipment as an example, the address of CFR is W1.

The Recipe Write Flag is located in CFR bit # 4, so enter the value 16 ($2^4 = 16$). The Recipe Write Flag located in W1 bit # 4 will then be set to ON.

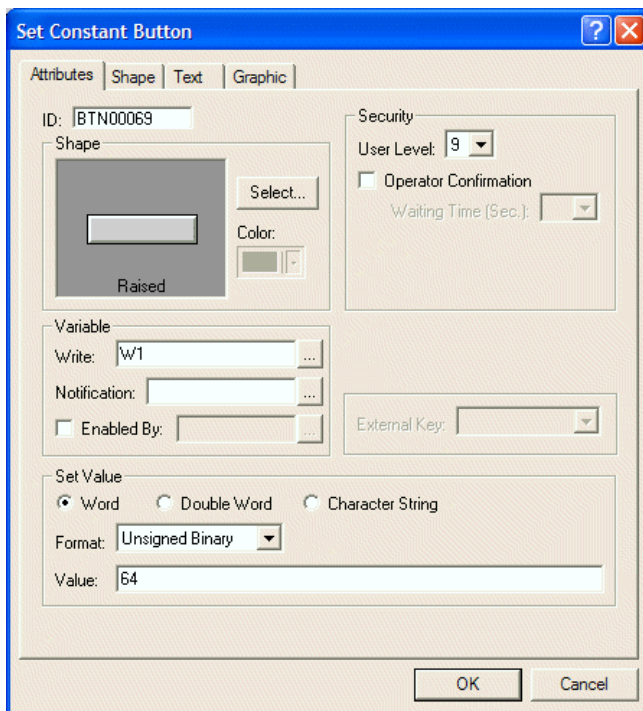


Setting up Recipe Write Flag ON

4. If the user wants to read a recipe from the controller to the operator terminal, the Recipe Read Flag located in CFR bit # 6 must be set to ON.

Using the coating equipment as an example, the address of CFR is W1.

The Recipe Flag is located in CFR bit # 6 ($2^6 = 64$).



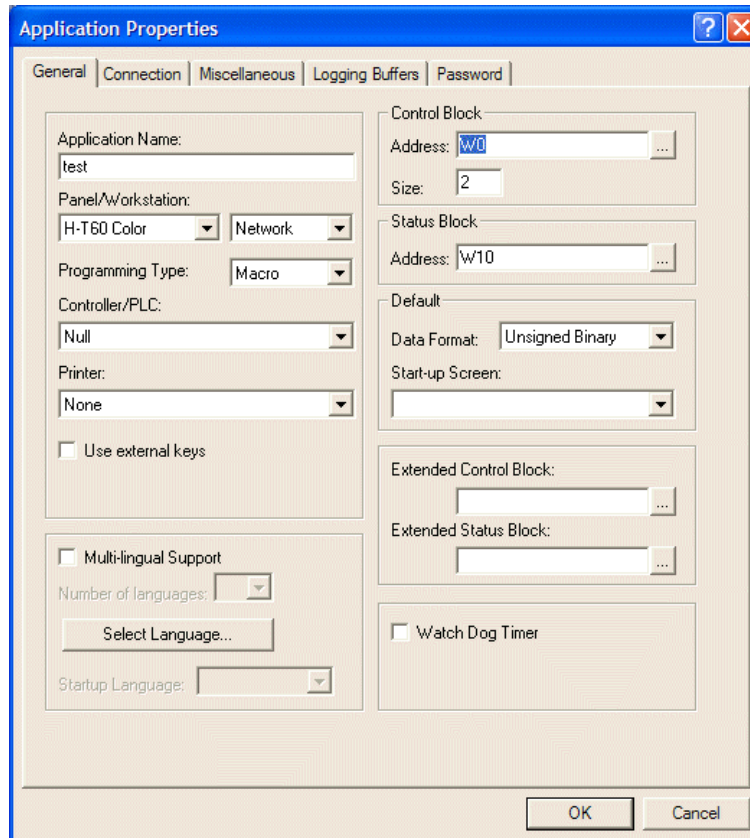
Setting up Recipe Read Flag ON

Once the above object design steps are completed, the user can execute the actions in the operator terminal conveniently.

4 Control and Status Block

This chapter describes the general information that you need for programming the controller to communicate with the operator terminal. Detailed information about connecting specific controllers to the operator terminal is presented.

To set up the **Control Block Address**, **Size** and **Status Block Address**, select **Application/Workstation Setup**.



Setting up control and status block parameters

4.1 Control Block

The **Control Block** is a block of continuous registers in the controller. The most important function in the operator terminal is the control block.

The control block enables the controller to control actions in the operator terminal through the controller program. The minimum number of words used in the control block is 2. The maximum number of words used is 32. The size of the control block varies according to the functionality required (if recipe functionality is used, then the minimum length is 6 words). The members of the Control Block are shown in the following table:

Word #	Member	Example: S7-200	Example: FX2
Wn	Screen Number Register (SNR)	VW0	D0
Wn+1	Command Flag Register (CFR)	VW2	D1
Wn+2	Logging Buffer Control Register #1	VW4	D2
Wn+3	Logging Buffer Control Register #2	VW6	D3
Wn+4	Logging Buffer Control Register #3	VW8	D4
Wn+5	RCPNO Number Register (RNR)	VW10	D5
Wn+6 and above	General User Area Register (GUAR) User's application registers CBn, n must not exceed 31.	VW12=cb6 VW14=cb7 ... VW18=cb9 ...	D6=cb6 D7=cb7 ... D9=cb9 ...

Say, for example, that the starting address is W0 (the starting address can be specified; the members in the above table will shift according to the starting address). Size 10 means that the operator terminal can read data from the W0-W9 (10 words) controller registers and store data in the CB0-CB9 internal control block.

The functions of the words Wn through Wn+m (word n+m) in the control block will be discussed in the following sections.

4.1.1 Screen Number Register

A controller can request a operator terminal to display a specific screen by setting its **Screen Number Register (SNR)** to the number of that screen.

SNR (Wn) enables the controller to control the operator terminal screen or print the screen. For example, a controller can request a operator terminal to display a specific screen by setting its SNR to the number of that screen.

The operator terminal cannot reset the SNR (Wn) to 0 automatically. However, the operator terminal does reset the SNR (Wn) to zero before changing a screen. If the screen specified by the SNR does not exist, the operator terminal does nothing but resetting the SNR (Wn=0).

The value in the SNR can be BCD or binary.

Example



A PLC can control the switch of screen 001.

The value of the SNR data register (Wn) and the functions (bit 0-bit 05) are as follows:

Wn 16-bit #(00-15)	Function
Bit 9-..bit 0	The first 10 bits store the screen number to be changed to.
Bit 10	Reserved
Bit 13=off, 12=off, 11=off	No language was selected
Bit 13=off, 12=off, 11=on	Language 1
Bit 13=off, 12=on, 11=off	Language 2
Bit 13=off, 12=on, 11=on	Language 3
Bit 13=on, 12=off, 11=off	Language 4
Bit 13=on, 12=off, 11=on	Language 5
Bit 13=on, 12=on, 11=off	Reserved
Bit 13=on, 12=on, 11=on	Reserved
Bit 14	Backlight turned off when set to 1
Bit 15	Backlight turned on when set to 1

The register (bit 0-9) is used to control the screen change and the other bits (bit 10-15) are not related. In other words, it is not necessary to control the backlight or language when changing the screen. Likewise, it is not necessary to assign the screen number when setting up the backlight or selecting a language.

4.1.2 Command Flag Register

The functions of the bits in the CFR are summarized in the following table:

Wn+1 16-bit # (00-15)	Function
Bit 0	Alarm History Buffer Clear Flag
1	Alarm Frequency Buffer Clear Flag
2	Print Change Paper Flag/Form Feed Flag
3	Hardcopy Flag
4	Recipe Write Flag - Data sent from operator terminal to controller
5	RCPNO Change Flag
6	Recipe Read Flag - Data sent from controller to operator terminal
7	Buzzer action control
8	Clear Flag #1
9	Clear Flag #2
10	Clear Flag #3
11	Clear Flag #4
12	Trigger Flag #1
13	Trigger Flag #2
14	Trigger Flag #3
15	Trigger Flag #4

The bits of the CFR (Wn+1) are described in more detail below.

Bit 0: Alarm History Buffer Clear Flag

This controller bit is used to clear the data in the alarm history buffer.

If bit 1 is set to clear the data in the alarm history buffer, the operator terminal will clear its data when bit 0 is set to 1.

The controller requires that the bit be reset if the operator terminal is re-assigned to clear the data and it needs enough time for operator terminal detection. The “handshake” function can be used to reset the bit as well. For more about the “handshake” function, please see the section [General Status Register](#).

Bit 1: Alarm Frequency Buffer Clear Flag

This controller bit is used to clear the Alarm Frequency Buffer.

If bit 1 represents to clear the data of alarm frequency buffer, the operator terminal will clear its data when bit 0 sets to bit 1.

The controller requires that the bit be reset if the operator terminal is re-assigned to clear the data and it needs enough time for operator terminal detection. The “handshake” function can be used to reset the bit as well. For more about the “handshake” function, please see the section [General Status Register](#).

Bit 2: Print Change Paper Flag

This controller bit is used to control form feed on the printer connected to the operator terminal.

Set the bit ON, and the printer will change paper.

The controller requires that the bit be reset if the operator terminal is re-assigned to form feed and it needs enough time for operator terminal detection.

Bit 3: Hard Copy Flag

This controller bit is used to control the hard copy function for the printer connected to the operator terminal.

Set the bit to ON and the printer will print the current screen.

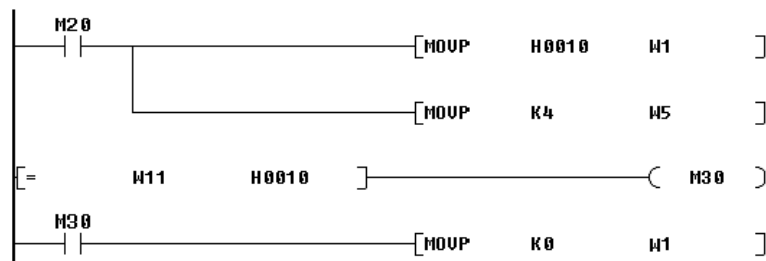
The controller requires that the bit be reset if the operator terminal is re-assigned to print a hard copy and it needs enough time for operator terminal detection.

Bit 4: Recipe Write Flag - from operator terminal to controller

This bit is used to write the recipe from RAM to the controller. This is only supported by operator terminals with a recipe function.

Set the RNR (W_n+5) to write the recipe, set the bit to ON, and the recipe will be written to the controller.

The controller requires that the bit be reset if the operator terminal is re-assigned to write another recipe and it needs enough time for operator terminal detection.



PLC M20 writes the data from terminal 4th recipe to PLC.
Will bit 4 is Recipe Write status bit.

Bit 5: RCPNO Change Flag

This controller bit is used to change the content value of RCPNO. RCPNO is an internal operator terminal register used to control the recipe data. This is only supported by operator terminals with a recipe function.

Set the RNR (W_n+5) to write the recipe, set the bit to ON, RCPNO can be modified.

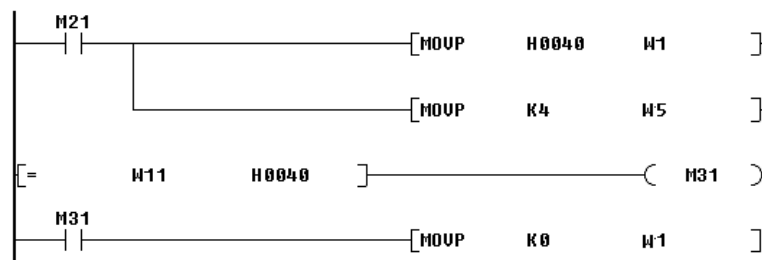
The controller requires that the bit be reset if the operator terminal is re-assigned to modify RCPNO and it needs enough time for operator terminal detection.

Bit 6: Recipe Read Flag - from controller to operator terminal

This controller bit is used to read the recipe data from the controller to the operator terminal and save it in the RAM block.

Set the RNR (D_n+5) to the recipe number to be updated. Then set the bit to ON and the operator terminal will update the corresponding recipe.

The controller requires that the bit be reset if the operator terminal is re-assigned to update and it needs enough time for operator terminal detection.



PLC M21 reads the data from PLC to the 4th recipe.
Will bit 6 is Recipe Read status bit.

Bit 7: Buzzer Flag

This controller bit is used to control the operator terminal's buzzer.

Set the bit to ON (about 1 sec.) to start the buzzer.

The controller requires that the bit be reset if the operator terminal is re-assigned to start the buzzer.

Bit 8-11: Clear Flag #1 - #4

This controller bit is used to clear curves in the operator terminal. There are four clear flags, and the user can set the corresponding signal to clear the desired curve.

Set the bit to ON/OFF once to clear the values of a trend graph or X-Y chart.

The controller requires that the bit be reset if the operator terminal is re-assigned to modify RCPNO and it needs enough time for operator terminal detection.

Bit 12-15: Trigger Flag #1 - #4

This controller bit is used to sample the trend graph data. There are four trigger flags in all.

Once the controller bit is set to ON/OFF, the operator terminal will read the continuous data and convert it into a continuous curve which is displayed as a trend graph or X-Y chart objects.

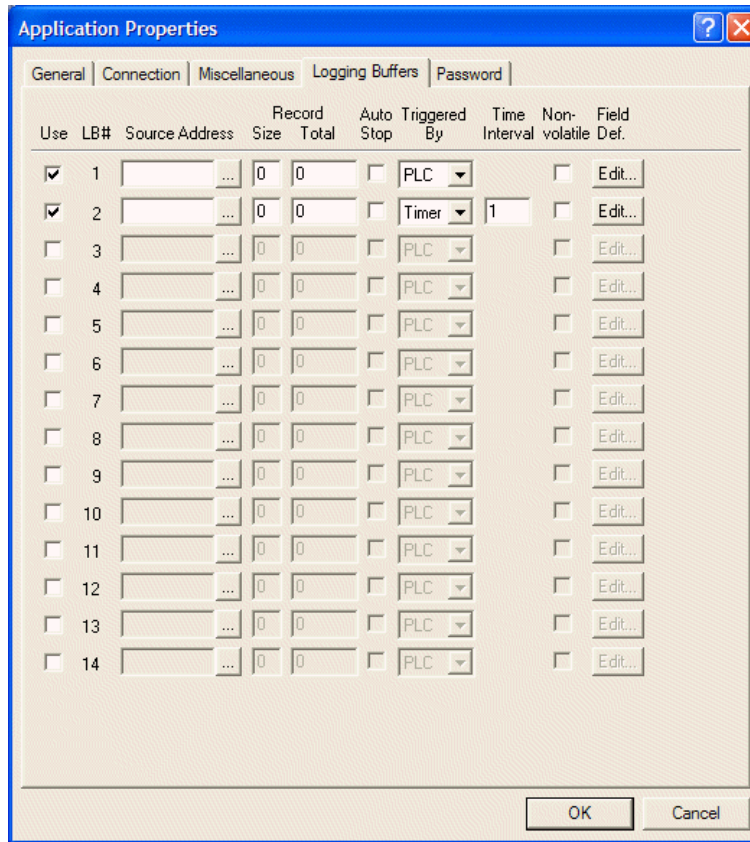
The controller requires that the bit be reset if the operator terminal is re-assigned to sample the data and it needs enough time for operator terminal detection.

4.1.3 Logging Buffer Control Registers: LBCRs

The other type of trend graph in the operator terminal is called the **Historical Trend Graph**. The operator terminal reads the data from the corresponding logging buffer according to the specific signal. The logging buffer is used to save the sample data in battery backup RAM. Remember to specify the logging buffer to read from and its size.

In LBCRs, **Trigger Bits** are used to request logging buffers to sample the data from controllers. **Clear Bits** are used to clear logging buffers and **Size Bits** are used to determine the size of the data to be read. Consequently, users can use the LBCRs to clear logging buffers or to request logging buffers to sample the data from the controllers.

There are twelve logging buffers here, and operator terminals can be set to sample the data automatically at fixed periods, or to sample or clear the historical trend graph controlled by the controllers.



Setting up logging buffers

Note that LBCR1 controls logging buffer #1 through #4. LBCR2 controls logging buffer #5 through #8. LBCR3 controls logging buffer #9 through #12.

The positions of the trigger bit, clear bit and size bit for each logging buffer are shown in the following table:

Bit #	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VW4 LBCR1	0	SB4	CB4	TB4	0	SB3	CB3	TB3	0	SB2	CB2	TB2	0	SB1	CB1	TB1
Bit #	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VW6 LBCR2	0	SB8	CB8	TB8	0	SB7	CB7	TB7	0	SB6	CB6	TB6	0	SB5	CB5	TB5
Bit #	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VW8 LBCR3	0	SB12	CB12	TB12	0	SB11	CB11	TB11	0	SB10	CB10	TB10	0	SB9	CB9	TB9

SB: Size Bit; CB: Clear Bit; TB: Trigger Bit.

LBCR1	Buffer #4	Buffer #3	Buffer #2	Buffer #1
LBCR2	Buffer #8	Buffer #7	Buffer #6	Buffer #5
LBCR3	Buffer #12	Buffer #11	Buffer #10	Buffer #9

Trigger Bit #1- #12: Sampling Control

The operator terminal can not only sample the historical trend graph at a particular time interval, it can also sample the historical trend graph under the control of the trigger bit in the controller. When the trigger bit (TB #1 - TB #12) is set to ON/OFF (about 1 sec.), the operator terminal will execute its sampling function.

Remember to set to OFF before re-triggering.

Clear Bit #1- #12: Clear Control

By triggering the trend graph clear bit (CB #1 - CB #12) ON/OFF once (about 1 sec.), the trend graph can be erased. The flag needs to be set to OFF if it is to be triggered again.

Size Bit #1-#12: Multiple Sampling Control

By triggering the size bit (SB #1 - SB #12) to ON, the operator terminal can sample single or multiple data values. When the trigger bit (TB #1 - TB #12) is set to ON/OFF (about 1 sec.), the operator terminal will execute its sampling function.

Logging Buffer

In setting up the logging buffer, the first step is to specify the **Source Address**, which is used to specify the controller address to read the data from.

After setting up the source address, the size bit is set to OFF and the trigger bit is changed from 0 to 1. Then the logging buffer will read a data value from the controller.

To force the logging buffer to read multiple data values from the controller, the size of the source address to read from must be set. Then set the size bit to ON and change the trigger bit from 0 to 1. Note that the size cannot exceed 1,022 words.

Change the clear bit from 0 to 1 to clear the logging buffer.

To force logging buffer recording, the controller must reset the trigger and clear bits. Sufficient time must be provided for operator terminal detection

Example: FX2 controller

Assumptions:

1. Control block starts from D0 with a size = 6
2. Source address of logging buffer #11 is D200
3. The record size of logging buffer #11 is 3 words

To request logging buffer #11 to read only one data record from the controller, first write the data to be read in D200-D202. Set D4's size bit (10) to OFF and change its trigger bit (8) from 0 to 1. The operator terminal reads D200-D202 into logging buffer #11 after it detects the trigger bit (8) of LBCR3 has changed from 0 to 1.

To request logging buffer #11 to read 50 data records from the controller, set D200 to 150 (=50 x 3). Write the data to be read in D201-D350. Set D4's size bit (10) to ON and change its trigger bit (8) from 0 to 1. The operator terminal first reads D200-D202 to get the actual size of the data to be read after it detects the trigger bit (8) of LBCR3 has changed from 0 to 1. Then the operator terminal reads D200-D350 and stores the data in battery backup RAM.

To request logging buffer #11 to clear the data records, change D4's clear bit from 0 to 1.

4.1.4 RCPNO Number Register: RNR

RCPNO is an internal register of the operator terminal that specifies the current recipe number. To change the RCPNO, the controller first sets the RNR to the recipe number and then sets the RCPNO write flag or recipe read flag.

To change RCPNO by the controller, the controller has to set RNR to the recipe number and set the RCPNO change flag, which is the CFR 5 bit. If the RNR is zero or greater than the maximum recipe number, the operator terminal will ignore the request.

To request the operator terminal to change RCPNO, the controller must reset the RCPNO change flag, or use the RCPNO change status flag, which is the GSR 5 bit.

Be sure to set this flag long enough for the operator terminal to detect it.

4.1.5 General User Area Register

For high-speed display, the operator terminal only reads data from the internal register (cannot write to it) when editing in H-Designer. (The maximum size is 32; the size depends on the length of the control block). The format is shown in the following table:

Format	Description
CB n	n represents the word data of the nth register, where n is a decimal number; $n \geq 0$ but smaller than the specified size.
CBn b	n.b represents the bit data corresponding to the nth word register, where b is in hexadecimal nth b = 0-f.

For example, if the address of the control block is D0, the **Numeric Display** object can be selected to display the recipe number register by configuring it to display CB5 instead of displaying D5.

The internal buffer of the control block is read-only. This means, for example, that you can configure a **Numeric Display** object to show the value of CB2, but you cannot configure the object to allow the operator to change the value of CB2.

Say, for example, that the you want to achieve the effect of the RNR numeric display object using H-Designer. You can specify that D5 be read from (writeable) or that CB5 be read from (non-writeable).

4.1.6 Determine the Control Block Size

As every application needs a **Screen Number Register (SNR)** and a **Control Flag Register (CFR)**, you can refer to the following rules to determine the size of the control block:

1. If the operator terminal reads/writes a recipe from/to a controller, the minimum size is six.
2. If Item 1. is not true and the operator terminal uses LBCR3 to control logging buffer #9 - 12, the minimum size is five.
3. If either Item 1. or Item 2. is not true and the operator terminal uses LBCR2 to control logging buffer #5-8, the minimum size is four.
4. If none of Item 1. through Item 3. is true and the operator terminal uses LBCR1 to control logging buffer #1-4, the minimum size is three.
5. If none of the above is true, the minimum size is two.
6. The size of the control block is the minimum size plus the size of the user area.

4.2 Status Block

The Status Block is a block of contiguous registers in your controller that display status information from the operator terminal. For example, you can get the current screen number from the first word of the Status Block. The members of Status Block are shown in the following table:

Word #	Member	Example: S7-200	Example: FX2
Wn	Screen Status Register (SNR)	VW20	D10
Wn+1	General Status Register (GSR)	VW22	D11
Wn+2	Logging Buffer Status Register #1 (LBSR1)	VW24	D12
Wn+3	Logging Buffer Status Register #2 (LBSR2)	VW26	D13
Wn+4	Logging Buffer Status Register #3 (LBSR3)	VW28	D14
Wn+5	RCPNO Image Register (RIR)	VW30	D15
Wn+6	Reserved	VW32	D16

For example, if the status block is W10 and the size is 6 words, the operator terminal will write the status data of the current screen to W10-W15.

4.2.1 Screen Status Register

When a screen is changed in the operator terminal, the controller sets its **Screen Status Register (SSR)** to the number of the new screen. Consequently, the controller can identify the current screen by reading the SSR.

The value of the SSR can be in BCD or binary format.

4.2.2 General Status Register

The components of the **General Status Register (GSR)** are shown in the following table (bit 0 - bit 15):

Wn11 16-bit #(00-15)	Function
Bit 0	Password Level Status (not available for applications configured to monitor alarms)
1	Password Level Status (not available for applications configured to monitor alarms)
Bit 0	Alarm History Buffer Clear Status
1	Alarm Frequency Buffer Clear Status
2	Form Feed Status
3	Hardcopy Status
4	Recipe Write Status
5	RCPNO Change Status
6	Recipe Read Status
7	Battery Status
8	Clear Status Flag #1
9	Clear Status Flag #2
10	Clear Status Flag #3
11	Clear Status Flag #4
12	Trigger Status Flag #1
13	Trigger Status Flag #2
14	Trigger Status Flag #3
15	Trigger Status Flag #4

Bit 0, 1: Password Level Status (not available for applications configured to monitor alarms)

Once connected to the operator terminal, the password level status bit 0 - bit 3 represent the current user level.

Level 0 ==> Bit 0 = off, Bit 1 = off

Level 1 ==> Bit 0 = on, Bit 1 = off

Level 2 ==> Bit 0 = off, Bit 1 = on

Level 3 ==> Bit 0 = on, Bit 1 = on

Level 4 - 9 ==> Bit 0 = on, Bit 1 = on

Bit 0: Alarm History Buffer Clear Status

The operator terminal will turn ON this status bit when it detects the alarm history buffer clear flag being turned ON. When the operator terminal finishes clearing the alarm history buffer, it will turn OFF this status bit.

Bit 1: Alarm Frequency Buffer Clear Status

The operator terminal will turn ON this status bit when it detects the alarm frequency buffer clear flag being turned ON. When the operator terminal finishes clearing the alarm frequency buffer, it will turn OFF this status bit.

Bit 2: Form Feed Status

The operator terminal will turn ON this status bit when it detects the form feed flag being turned ON. When the operator terminal finishes sending the form feed character to a printer, it will turn OFF this status bit.

Bit 3: Hardcopy Status

The operator terminal will turn ON this status bit when it detects the hardcopy flag being turned ON. When the operator terminal finishes printing the current screen, it will turn OFF this status bit.

Bit 4: Recipe Write Status

The operator terminal will turn ON this status bit when it finishes sending a recipe from the operator terminal's RAM block to the controller. The operator terminal will turn OFF this status bit when it detects the recipe write flag being turned OFF. This bit can be used as a handshake signal to switch the recipe write flag.

Note:

This function is only supported by operator terminals with a recipe function.

Bit 5: RCPNO Change Status

The operator terminal will turn ON this status bit when it detects the RCPNO change flag being turned ON. When the operator terminal finishes changing the RCPNO, it will turn OFF this status bit.

Bit 6: Recipe Read Status

The operator terminal will turn ON this status bit when it finishes reading a recipe from the controller. The operator terminal will turn OFF this status bit when it detects the recipe read flag being turned OFF. One can use this bit as a handshake signal to switch the recipe read flag..

Note:

This function is only supported by operator terminals with a recipe function.

Bit 7: Battery Status

The operator terminal will turn ON the battery status bit if it detects a low battery before running an application.

Bit 8-11: Clear Status Flag #1 - #4

The operator terminal will turn ON one of the clear status bits when it finishes the clearing task requested by the corresponding clear flag controlled by the controller. The operator terminal will turn OFF the same status bit when it detects the corresponding clear flag being turned OFF. One can use clear status bits as handshake signals to switch the clear flags.

Bit 12-15: Trigger Status Flag #1 - #4

The operator terminal will turn ON one of the trigger status bits when it finishes the task triggered by the corresponding trigger flag. The operator terminal will turn OFF the same status bit as it detects the corresponding trigger flag being turned OFF. One can use trigger status bits as handshake signals to switch the trigger flags.

4.2.3 Logging Buffer Status Registers (LBSRs)

LBSR1 saves the status of logging buffer #1 - #4. LBSR2 saves the status of logging buffer #5 - #8. LBSR3 saves the status of logging buffer #9 - #12.

The status bit's position for each of the logging buffers is shown in the table below:

Bit #	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VW4 LBSR1	AB4	FB4	CB4	TB4	FB3	AB3	CB3	TB3	AB2	FB2	CB2	TB2	AB1	FB1	CB1	TB1
Bit #	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VW6 LBSR2	AB8	FB8	CB8	TB8	AB7	FB7	CB7	TB7	AB6	FB6	CB6	TB6	AB5	FB5	CB5	TB5
Bit #	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VW8 LBSR3	AB12	FB12	CB12	TB12	AB11	FB11	CB11	TB11	AB10	FB10	CB10	TB10	AB9	FB9	CB9	TB9

AB: Almost Full Bit - indicates that the buffer is 90% or more full.

FB: Full Bit - indicates that the buffer is full.

CB: Clear Status Bit indicates the clear command was received.

TB: Trigger Status Bit indicates the trigger command was received.

LBSR1	Buffer #4	Buffer #3	Buffer #2	Buffer #1
LBSR2	Buffer #8	Buffer #7	Buffer #6	Buffer #5
LBSR3	Buffer #12	Buffer #11	Buffer #10	Buffer #9

The operator terminal will turn ON one of the trigger status bits when it finishes collecting one data record for the logging buffer. The operator terminal will turn OFF the same status bit as it detects the corresponding trigger flag being turned OFF. One can use the trigger status bits as handshake signals to switch the trigger flag.

4.2.4 RCPNO Image Register

The operator terminal sets the RCPNO Image Register (RIR) to the new value of the RCPNO as this internal register is changed by the user or a controller. Consequently, the controller is able to identify the current value of the RCPNO. The operator terminal reports the value of the RCPNO to the controller by writing the value to the RCPNO Image Register. The RCPNO Image Register is word #5 of the status block and one can keep track of the current recipe with this register.

4.3 Recipe Register Block

The recipe block is located in the controller register. To make the operator terminal read/write the recipe data from/to the controller, a recipe block needs to be defined for the application. Please see the chapter [Recipes](#) for complete details.

The maximum recipe memory block is 524,288 16-bit (word) for the operator terminal with recipe function. For the applied operator terminal models, please see [Appendix A - H-Designer Features and Operator Terminal Models](#) for complete details.

4.3.1 Recipe Register Number - Enhanced Operator Terminals

H-Designer provides internal recipe register number for use in the operator terminal application using the formats shown below:

Format	Description
RCPNO	Recipe Register Number (1-N) RCPNO is an internal register of the operator terminal that specifies the current recipe number; $N \geq 1$.
RCPWnnnnn	Recipe Register #nnnnn is current recipe where nnnnn is a decimal number and $n \geq 0$.
RCPWnnnnn.b	Recipe Register Bit nnnnn is decimal number, $n \geq 0$; b is a hexadecimal number, $b=0-F$.

RCPNO is an internal register of the operator terminal used to display the specified recipe on the screen. The operator terminal changes the RCPNO number to display its corresponding recipe data.

There are two methods to change the RCPNO number:

One way is for the user to change the RCPNO number directly through the numeric entry object.

The other way is for the controller to change the RCPNO constant. To change the RCPNO constant, the user must write the specified number N to RCPNO number register $Dn+5$, then set the RCPNO change flag $Dn+1$ bit 5 as ON (about 1 sec.). The operator terminal will change the RCPNO constant to N and display the recipe data RCPW0 - RCPWm corresponding to the Nth recipe.

4.3.2 Addressing Recipe Data - Enhanced Operator Terminals

Suppose that the number of recipe $N=20$, a recipe size $m=100$ words.

To edit an address, one needs to set up the current recipe $N = \text{RCPNO}$. The operator terminal will display the corresponding recipe data.

1. Enter the recipe number N in RCPNO or change RCPNO using the controller. The operator terminal will display the corresponding recipe data.

For example, if $\text{RCONO } N=5$, RCPW0-RCPW99 displays the data corresponding to the fifth recipe; if $\text{RCPNO } N=7$, RCPW0-RCPW99 displays the data corresponding to the seventh recipe.

2. Another way to edit the corresponding recipe register data is to use the absolute address.

Suppose that an address is greater than RCPW100 , the corresponding recipe address will display the N th recipe data.

RCPW100-RCPW199 represents the first recipe data.

RCPW200-RCPW299 represents the second recipe data.

.

.

RCPW2000-RCPW2099 represents the twentieth recipe data.

Addresses greater than RCPW2099 are invalid.

So, RCP234 represents the second recipe data, 35 words and $\text{RPCW } 34$ in $\text{RCPNO} = 2$.

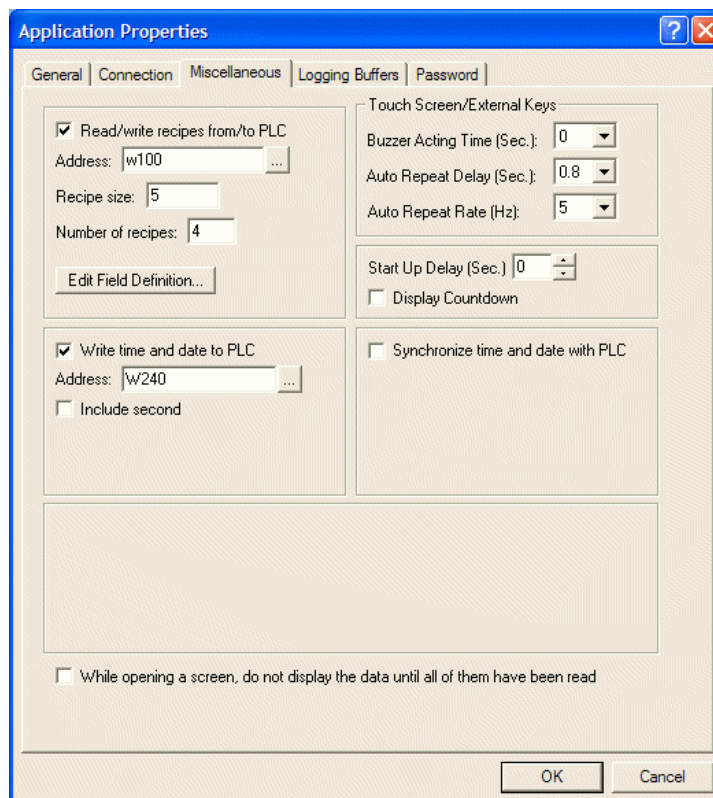
4.4 Time Block

4.4.1 The Operator Terminal Writes to the Controller

To make the operator terminal write the current time and date to the controller, the **Time Block** has to be defined for the application. The time block is a block of three words in the controller and its format is BCD. The operator terminal updates the time block every minute with the time data. The format of time block is shown in the following:

Low byte of word 0 (07-00)	Minute BCD 00-59
High byte of word 0 (15-08)	Hour BCD 00-23
Low byte of word 1 (07-00)	Day BCD 00-31
High byte of word 1 (15-08)	Month BCD 01-12
Low byte of word 2 (07-00)	00-99
High byte of word 2 (15-08)	Day of week 1 = Sunday 2 = Monday 3 = Tuesday 4 = Wednesday 5 = Thursday 6 = Friday 7 = Saturday

The steps to set up the time block follow. Select **Application/Workstation Setup** in H-Designer and you can set up the time block on the **Miscellaneous** tab. The starting address is W240 and the size is 3 words, so the data will be saved in the W240, W241, and W242 16-bit registers. The operator terminal updates the time block every minute with the time data.



Setting up the Time Block

4.4.2 The Controller Writes to the Operator Terminal

The operator terminal can read time and date from the internal real time clock of the controller. Then the operator terminal can modify the corresponding data for the time/date/week read from the real time clock and display the content in the operator terminal. The operator terminal updates the time block every minute with the time data.

4.5 Read Cycle

The operator terminal does the following steps to accomplish one read cycle and it will repeat these steps continuously. The user needs to know this read cycle in order to configure an operator terminal to communicate with the controller efficiently.

Steps of the cycle:

1. Reads control block of the controller.
2. Reads specified register blocks for the current screen.
3. Reads specified On/Off blocks for the current screen.
4. Reads specified alarm register regularly (3-10 sec.).
5. Reads a number of controller locations which: (1) are shown on the current screen; and (2) do not appear in the current screen's register blocks or On/Off blocks and have not been read recently.

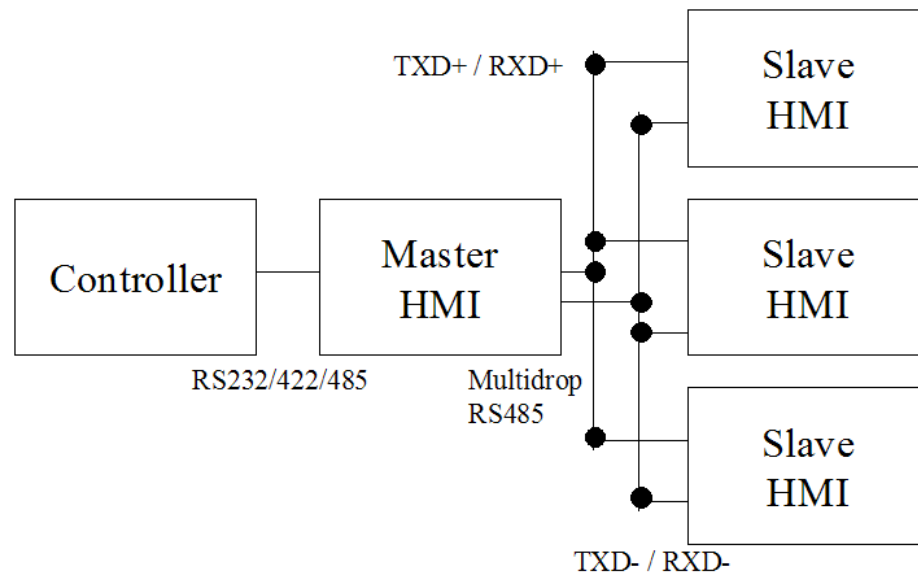
The number of controller locations to be read in this step is specified by the "number of individual reads" per read cycle of the current screen.

This read cycle is repeated continuously from Step 1. to Step 5.

5 Multi-Link: Normal Connection Port

The **Multi-Link** function provides an economical and convenient way to link several operator terminals and communicate with a single controller connection port. One operator terminal is master and the others are slaves. The master is the only operator terminal that is physically connected to a controller and this operator terminal is responsible for data exchange between the controller and the slaves. Each of the slave operator terminals must be assigned a unique address so that the master operator terminal is able to identify which slave to send the data to.

The following illustration shows the setup for four operator terminals with one controller. Note that the RS485 cable must be used for the connection between the master and the slaves and each of the slaves must be assigned a unique address.



A Multi-Link network

The cable and the connection between the master and the controller is the same as for the normal 1-to-1 application. The RS485 cable must be used for the connection between the master and the slaves. Additionally, each of the slaves must be assigned a unique address. All the controller models in H-Designer support this function.

Multi-link can also be connected through Ethernet. Please see the chapter [Ethernet Communication](#) for complete details.

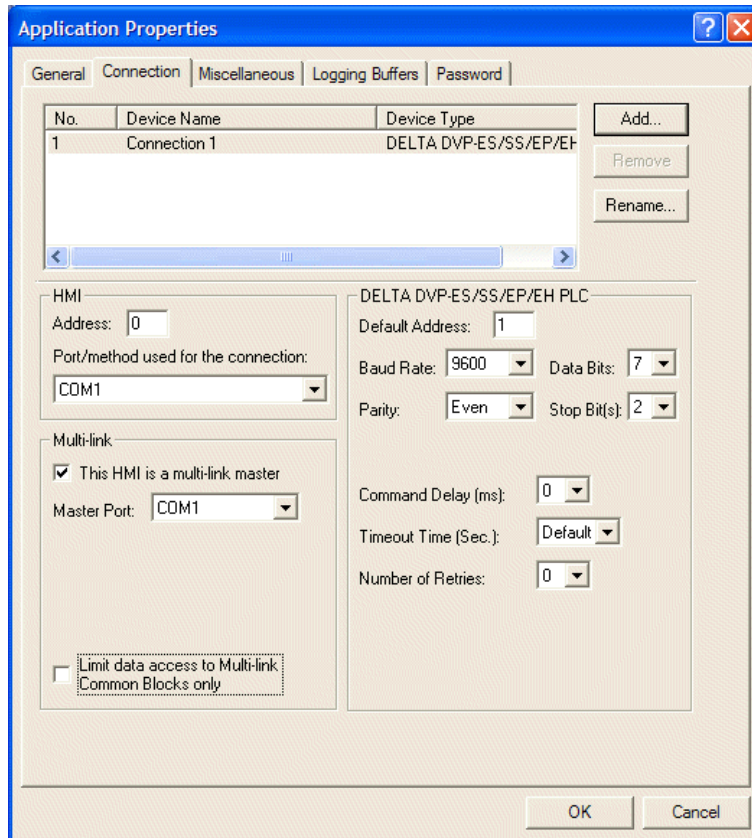
5.1 Communication Parameters

Perform the steps to set up the communication parameters.

Setting up the master

The master is the operator terminal that communicates with the controller.

1. Select **Application/Workstation Setup** and check the **This HMI is a multi-link master** box.



Setting up the multi-link master

2. Download the edited application to the master operator terminal.

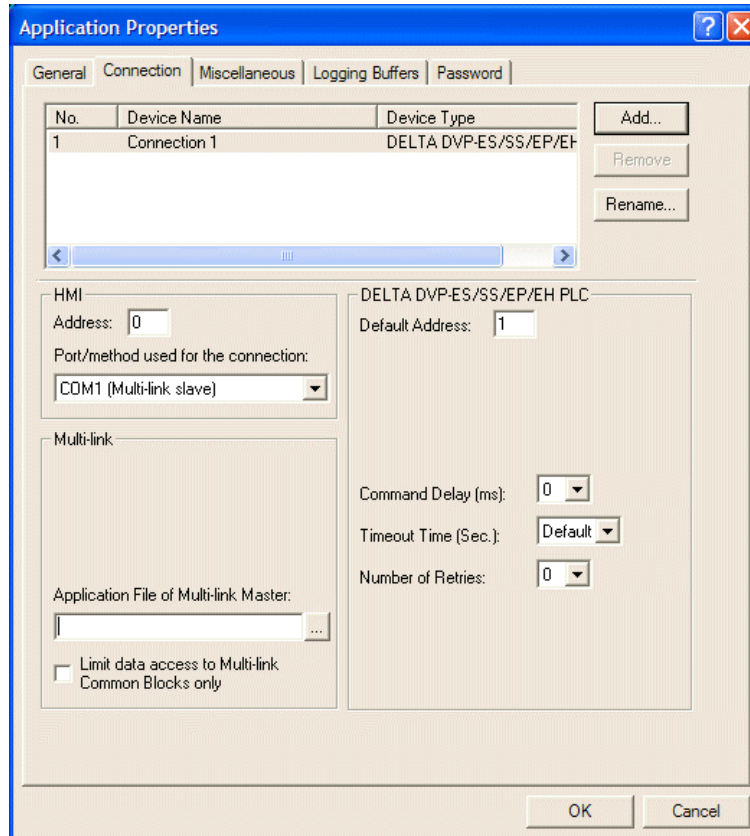
Setting up the slaves

The slave operator terminals do not communicate with the controller

3. Select **Application/Workstation Setup**. On the **Connection** tab, select COM1 port for the connection.

4. Setting up the **Default Address**:

If the slave operator terminal dip switch #5 is set to OFF, the operator terminal will read the communication parameters from H-Designer. The unique address (2-10) must be entered in **Default Address**.



Setting up slave operator terminal parameters

Remember to compile and download applications each time after making any changes to the address.

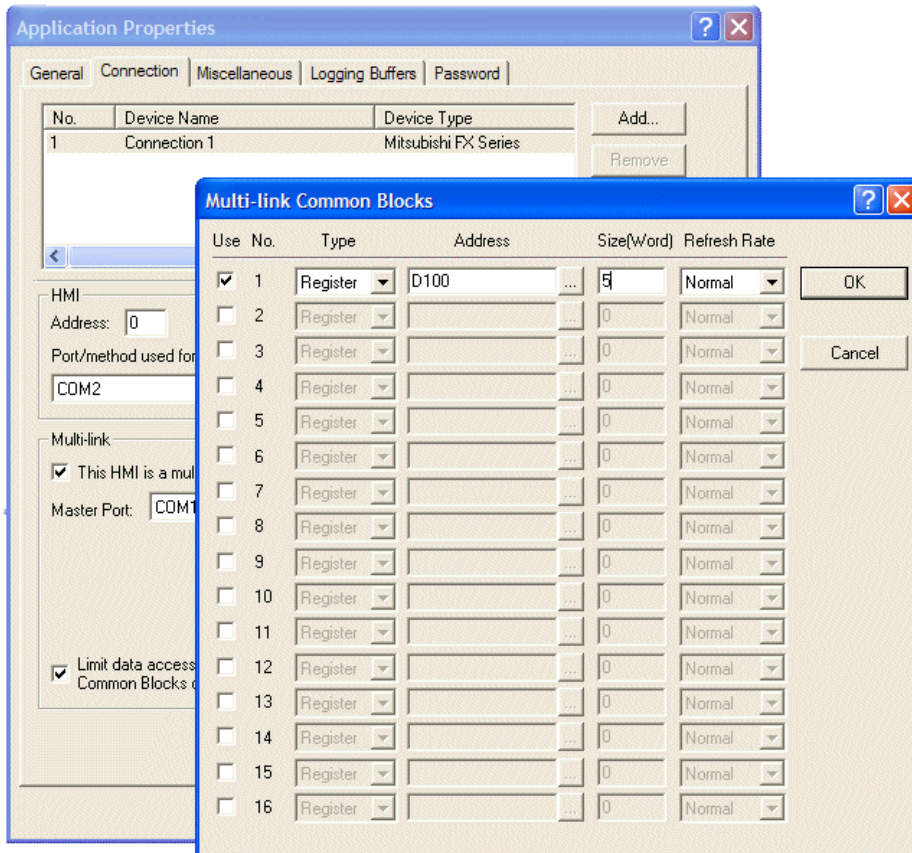
If the slave operator terminal dip switch #5 is set to ON, the operator terminal will read the parameters from the hardware. The user must enter the address (1-15) in **HMI Node Address**.

5. Download the edited application to the slave operator terminals.

Please note that the address number of the master operator terminal will not affect communication with the slave operator terminals. Moreover, it is not necessary to set up the baud rate or data type for the slave operator terminals. The purpose of setting up the slave operator terminals is to make sure the connection port for the master operator terminal is correctly set.

5.2 Communication Efficiency

H-Designer allows you to specify a **Common Register Block (CRB)** and a **Common On/Off Block (COB)** for the operator terminals. Select **Limit data access to Multi-link Common Blocks only**.



The CRB is a block of registers and the COB is a block of On/Off locations in the controller. In every read cycle, the master operator terminal reads the data from both the CRB and the COB. Then the master sends the CRB and COB data to all the slaves.

The CRB and the COB allows a maximum of 128 words and 256 words respectively. In multi-link, the CRB and the COB has to be specified with the same size and format for each of the operator terminals. The slaves are not requested to read the data from the CRB or the COB directly. The slaves read the data from the buffer containing the data sent by the master. The CRB and the COB play important roles in terms of communication efficiency since they can reduce traffic in the multi-link as well as in the link between the master and the controller

For instance, arranging control blocks for the operator terminal in the CRB and the COB is one of the most effective ways to improve performance. Arranging the variables common to some of the operator terminals in the CRB or the COB will also improve performance. Doing so will result in a high refresh rate for the variables held in the CRB and the COB since the variables are refreshed every read cycle.

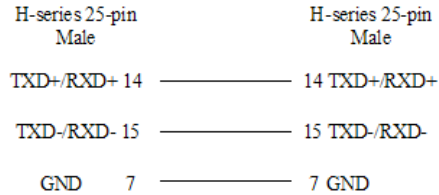
In addition to the CRB and the COB, remember to make use of the register blocks and on/off blocks for screens, since these too lessen the burden of the operator terminal by reducing the number of read commands.

It is recommended to specify the CRB and the COB with continuous locations when designing screens.

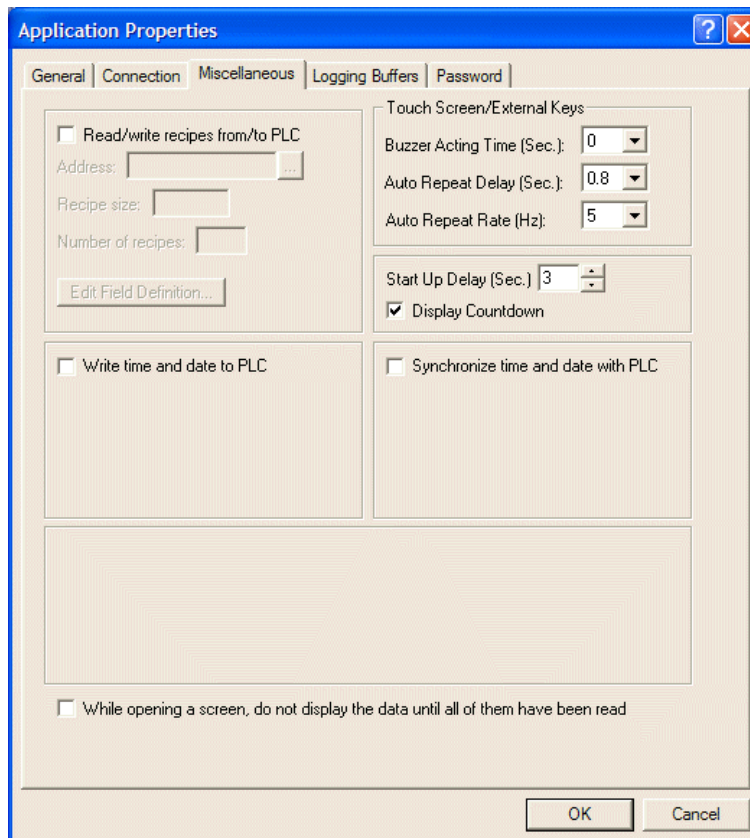
5.3 Important Notes

When using multi-link, please note the following points:

1. The RS485 connection method is suggested for the multi-link. The RS232 connection method is suggested for a single slave operator terminal.



2. Each slave must have its own unique address.
3. The operator terminals on the same multi-link must have the same CRB and COB.
4. The master operator terminal should only start after all the slaves have displayed their first screens. To delay the start-up of the master, select the **Miscellaneous** tab in the **Application Properties** dialog box. Then set the time for the **Start Up Delay**.



Setting up delay of start-up

6 Ethernet Communication

The following chapter will introduce communication setup, application upload/download and Ethernet communication with Ethernet-enabled controllers and operator terminals. There are two methods for Ethernet-enabled operator terminals to communicate with multi-operator terminal/controllers over Ethernet; multi-link and cross-link.

Note:

Ethernet communication is not available for all operator terminal models; please refer to [Appendix A - H-Designer Features and Operator Terminal Models](#) for details.

6.1 Connection

There are two methods to set up the link; using a RJ45 straight through cable or using a RJ45 crossover cable. The RJ45 crossover cable requires a HUB for connection.

The choice between these two methods depends on one's needs and available equipment. The following table describes differences between the two methods.

RJ45 crossover cable	RJ45 straight through cable
Requires no HUB; links to operator terminal directly	HUB required
1-to-1 only	Multi-link

6.2 IP Address Setup

To read or send data from a operator terminal over Ethernet, correct IP addresses have to be set up.

The IP address can be set under **Configure** in the operator terminal system menu.

Date (mm-dd-yy)	08-28-03	CTS handshaking	Disabled
Day of the week	Thu	Battery check	Enabled
Time (hh:mm:ss)	18:34:23	Screen saver time (Min.)	00
Printer	Disabled	PLC model code	0
PLC communication port	COM2	Workstation node address	000
Baud rate	9600	Download/Upload/Copy port	COM1
Data bits	7 bit	RTC adjust	00
Parity	Even	IP address	100.101.102.001
Stop bits	1 bit	Gateway address	100.101.102.254
Command delay (x 10 ms)	000	Sub-network mask	255.255.255. 0
<div> <div>+</div><div>↑</div><div>-</div> <div>←</div><div>↓</div><div>→</div> <div>More</div><div>Quit</div><div>Save & Quit</div> </div>		<div> <div>+</div><div>↑</div><div>-</div> <div>←</div><div>↓</div><div>→</div> <div>More</div><div>Quit</div><div>Save & Quit</div> </div>	

Configuration table of operator terminal with support for networking

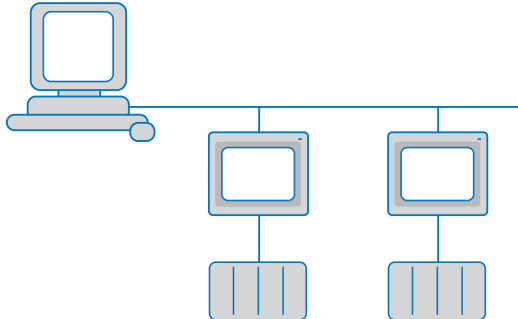
Note:

If uploading/downloading over Ethernet, the first three segments of the PC IP address must be the same as the first three segments of the operator terminal IP address.

Example: PC IP address=192.168.1.10 and operator terminal IP address=192.168.1.XXX. The IP address of the operator terminal should not be shared with other units in the network.

6.3 Application Upload/Download over Ethernet

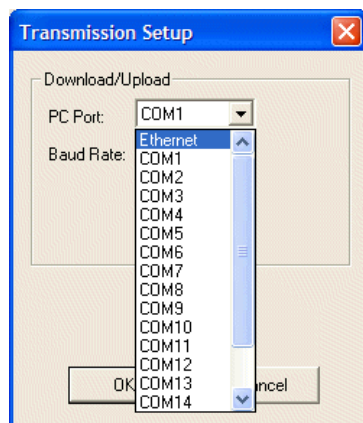
The Ethernet-enabled operator terminals together with H-Designer provide upload/download over Ethernet for application, firmware, recipes and source code.



Uploading/downloading over Ethernet

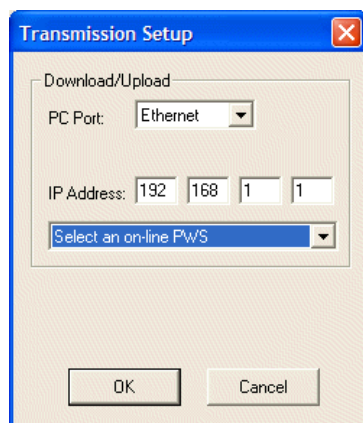
Perform the following steps to upload/download an H-Designer application over Ethernet:

1. Set the IP address, gateway address etc in the **Configuration Table**. Please see the section [IP Address Setup](#) for details.
2. In H-Designer, select **Options/Transmission Setup** and select **Ethernet** from the **PC Port** list.



Selecting PC Port

3. Enter the address under **IP Address** or select from the drop-down list.

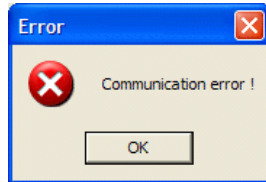


Setting the IP address

4. Select **Application/Download Application** or **Download Firmware and Application** to download the application.

Follow the same steps to **Upload Application**, **Upload Recipes**, **Download Recipes** or **Reconstruct Source over Ethernet**. For **Upload Application**, the steps above must be changed to select **File/Upload Application**.

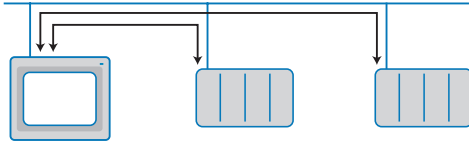
If the link is not set up properly, H-Designer will display an error message.



Communication error message

6.4 Communication with Ethernet-enabled controllers

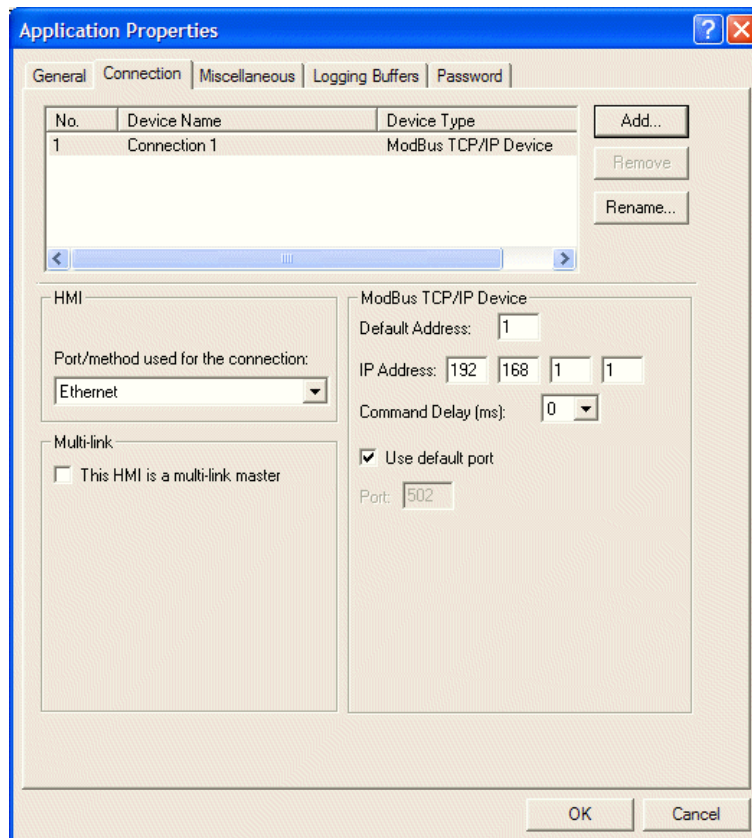
H-Designer supports operator terminals communicating with Ethernet-enabled controllers using Modbus TCP/IP. Consequently, the operator terminal can control or read data from the controllers.



Connection to Ethernet-enabled controllers

Perform the following steps to set up the connection:

1. Select **Application/Workstation Setup**. On the **General** tab, select the type of controller or **Modbus TCP/IP Device** from the **Controller/PLC** list.
2. On the **Connection** tab, select **Ethernet** for **Port/method used for the connection**. Enter the address in the **Default Address** and **IP Address** boxes.

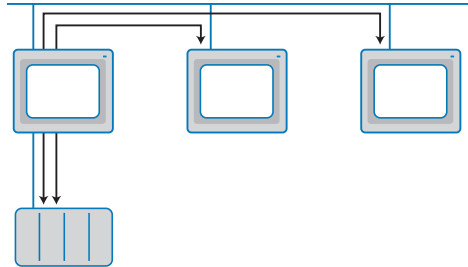


Selecting communication method and setting the IP address

3. Download the H-Designer application file to the operator terminal and set up the link to connect.

6.5 Multi-Link - One Master and Multiple Slaves

The Multi-Link over Ethernet function allows linking several operator terminals (one master and multiple slaves). This speeds up communication between the operator terminals.



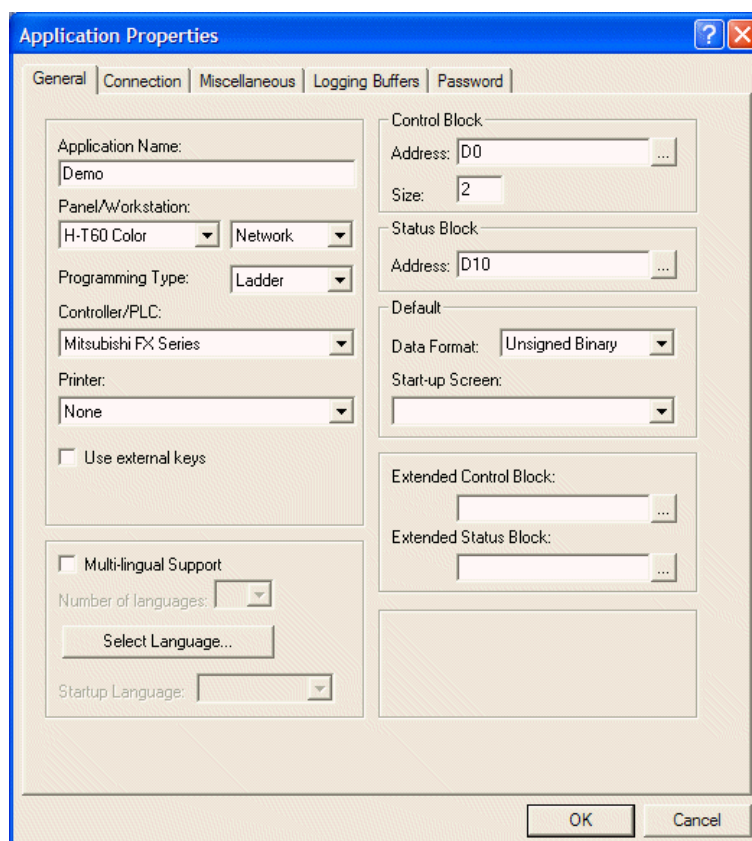
Multi-Link over Ethernet

Perform the following steps to set up the communication:

Setting up the master

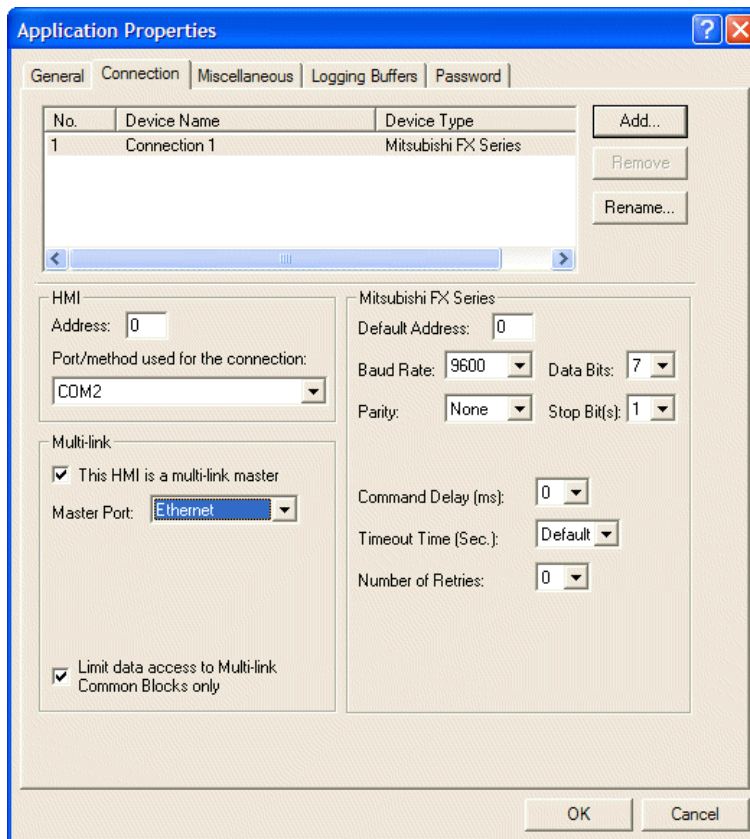
The master is the operator terminal that communicates with the controller.

1. Select **Application/Workstation Setup**. On the **General** tab, select the operator terminal model from the **Panel/Workstation** list and the type of controller from the **Controller/PLC** list.



Selecting operator terminal model and controller model

2. On the **Connection** tab, check the **This HMI is a multi-link master** box and select **Ethernet** from the **Master Port** list. Next, check **Limit data access to Multi-link Common Blocks only**.



Setting up the multi-link master

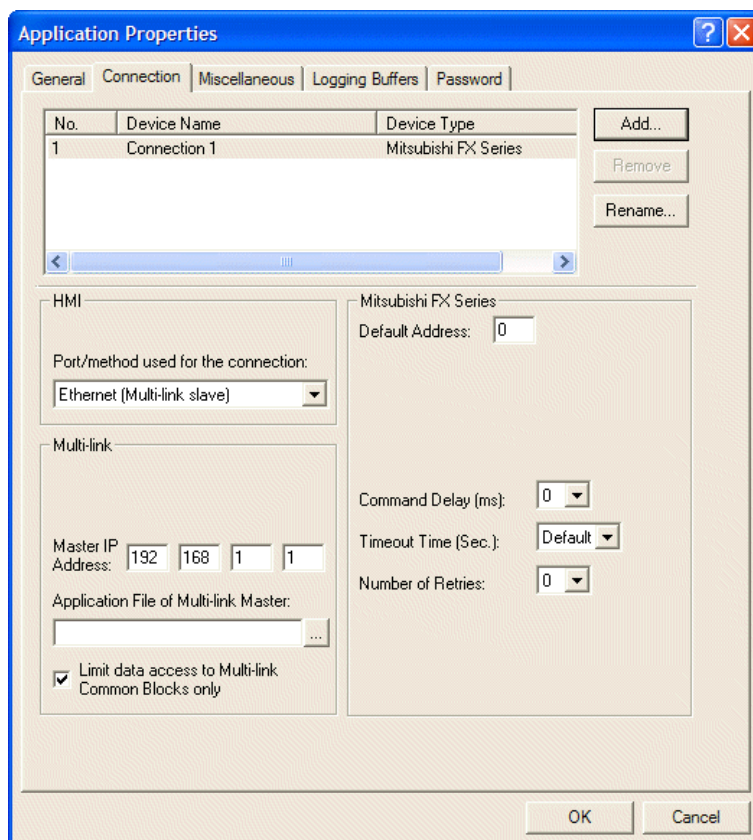
3. Download the edited application to the master operator terminal.

Setting up the slaves

The slave operator terminals do not communicate with the controller

4. Select **Application/Workstation Setup**. On the **Connection** tab, select **Ethernet (Multi-link slave)** port for the connection.

5. Set up Master IP Address, Common Register Block, Common On/Off Block, CRB Size and COB Size.

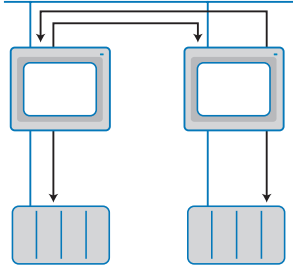


Setting up connection method, master IP address etc.

6. Download the edited application to the slave operator terminals.

6.6 Cross-Link over Ethernet (Data Sharing)

The Cross-Link over Ethernet function allows users to link several operator terminals and controllers and share data between them.

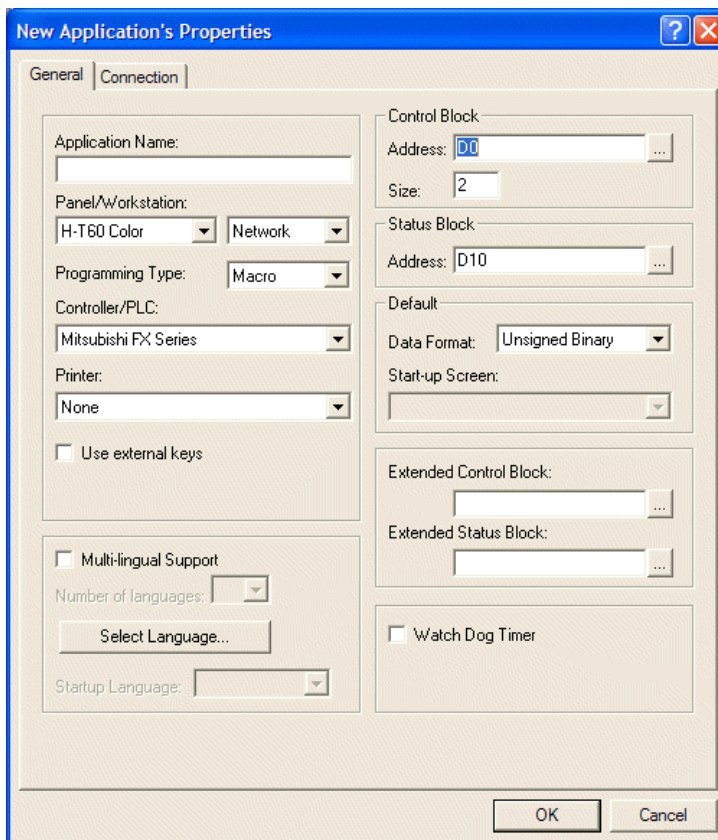


Cross-Link over Ethernet function

The following steps are performed to set up the connection:

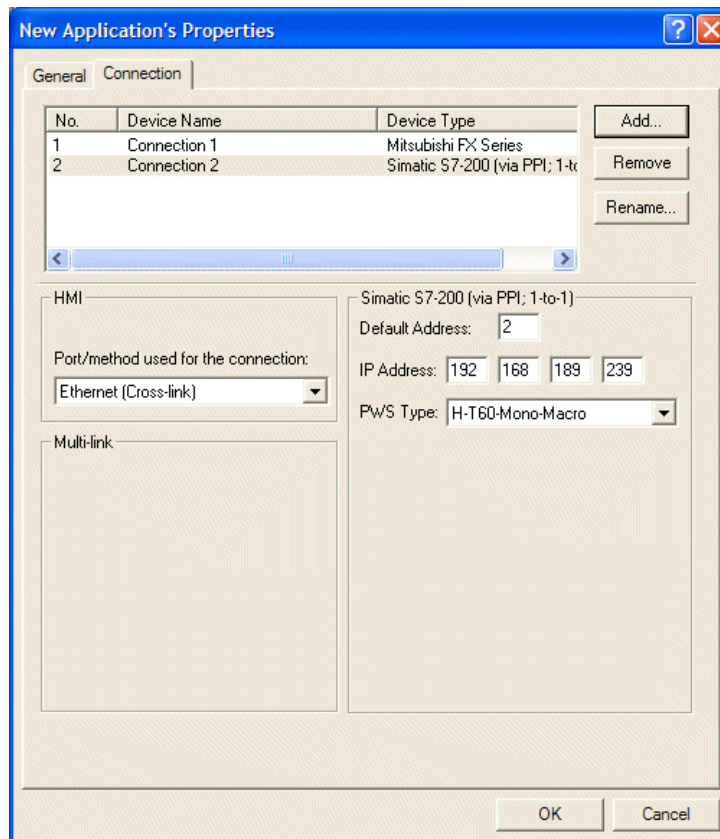
Setup in the first operator terminal

1. Select **Application/Workstation Setup**. On the **General** tab, select the operator terminal model from the **Panel/Workstation** list and the controller type from the **Controller/PLC** list. .



Selecting operator terminal model and controller model

2. On the **Connection** tab, click **Add** to add Connection 2 for the Cross-link.
 Select **Ethernet (Cross-link)** from the **Port/method used for the connection** list.
 Enter the **Default Address**, **IP Address** and **HMI Type** for the desired device.



Adding the Cross-link device

Note that **Connection 1** is linked to the controller by COM Port while **Connection 2** is linked to the controller by Cross-Link Ethernet.

If the operator terminal using Connection 1 is to access data from the controller linked to Connection 2, follow these steps:

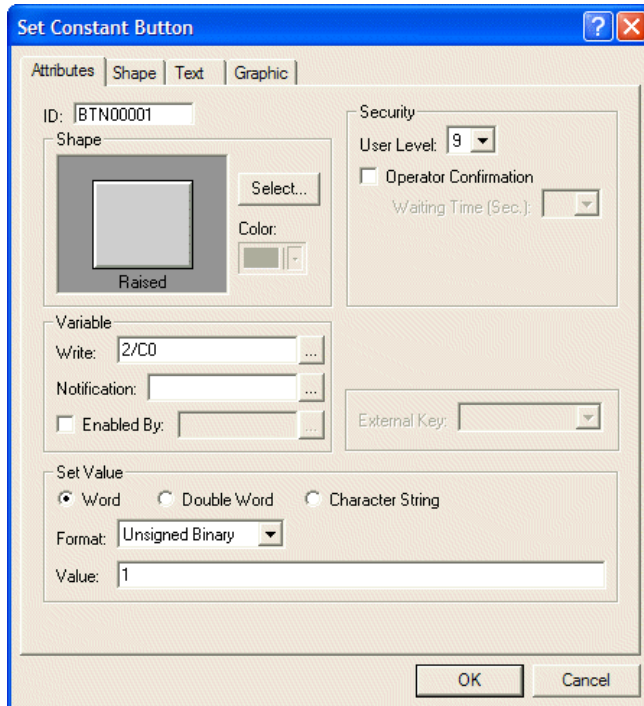
Controller register address setup:

Example: Specify the address of the controller register, **2/C0** for Siemens S7-200.
 Note that controller register **2/C0** refers to **Connection 2** on the **Connection** tab.
 / denotes the separation from the register address.

3. There are three ways to set this up:

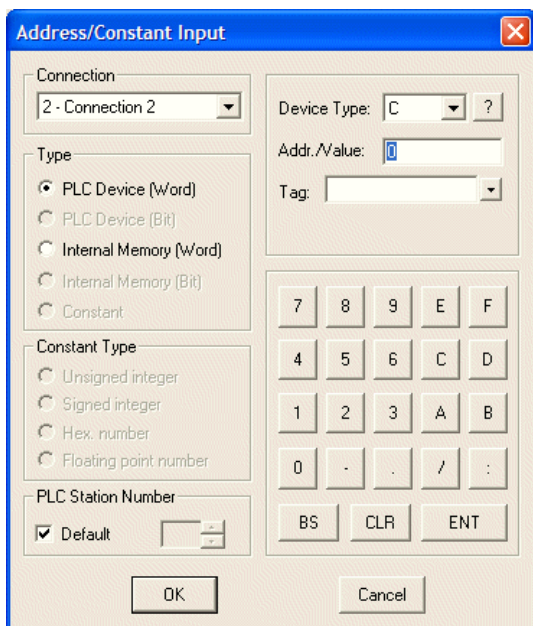
- a) Double-click on the object. Select the **Attributes** tab. For **Write** under **Variable** enter the location **2/Q0.0**.
- b) Enter the address of the controller register in the object attributes dialog box.

In the example the address is 2/C0.



c) Click ... to display the **Address/Constant Input** dialog box.

Select **2-Connection 2** from the **Connection** list. Enter the address in the **Device Type** and **Addr./Value** boxes. In the example the address is C0.

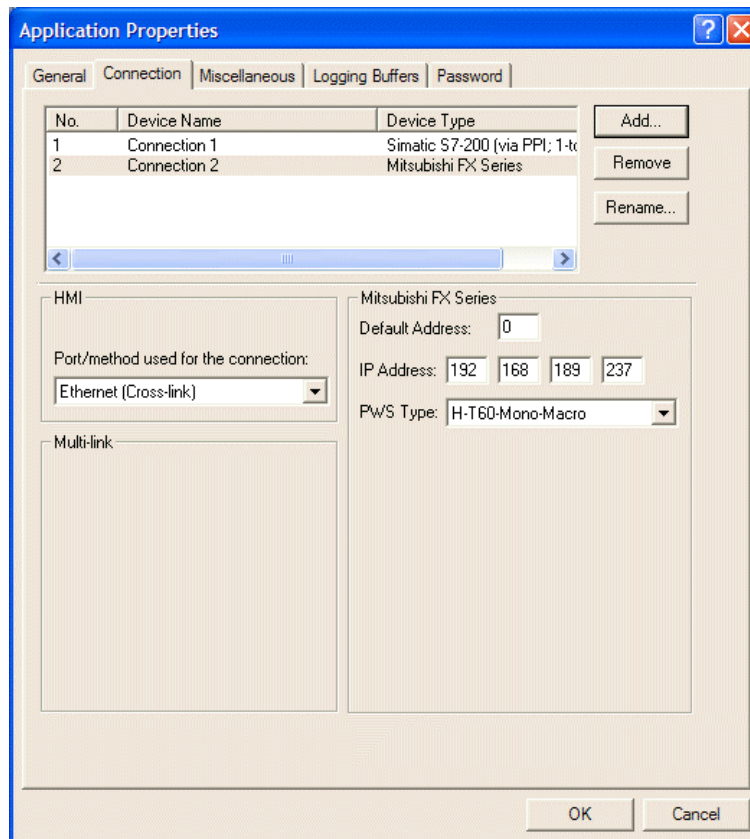


Click **OK** and 2/C0 will appear in the object attribute dialog box.

4. Download the edited application to the first operator terminal.

Setup in the second operator terminal

- The setup steps are the same as for the first operator terminal, with the difference that the controller device in **Connection 1** of the second operator terminal is the controller device in **Connection 2** of the first operator terminal. Note that the **Addr./Value** of the controller device **Connection 2** must be changed.



The connections in operator terminal 2

- Download the H-Designer application file to operator terminal 1 and operator terminal 2. Connect the link to the controllers and network.

7 Multi-Channel Communication

Since there are many different vendors and models of all kinds of equipment on the market, users often have a difficult time with data collection and integration. The operator terminal model with 2 COM Ports and an Ethernet connection can be used to connect controllers or other equipment (such as temperature controllers, servers, inverters etc.) from different vendors in order to integrate and collect data using an operator terminal or PC.

Note:

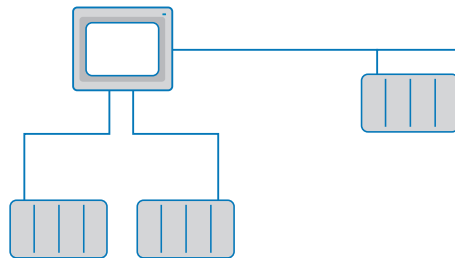
This feature is not available for all operator terminal models: please refer to [Appendix A - H-Designer Features and Operator Terminal Models](#) for details.

7.1 Connection

COM1, COM2 or the Ethernet Port can be used to link the Multi-Channel connection.

The link can be set up via RS232, RS422 or RS485, with the connection method based on needs and available equipment. For Ethernet, RJ45 has to be used to set up the link. The controller must also be Ethernet-enabled.

For connection method and setup, please see the chapter [Communication between Operator Terminal and Controller](#).



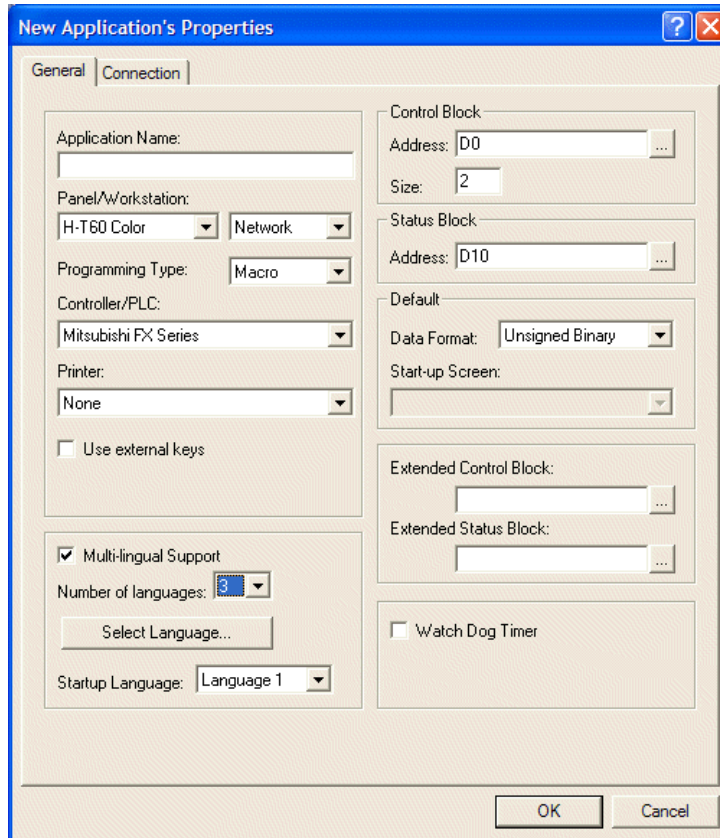
Multi-Channel Connection

7.2 Connection Setup

The Multi-Channel connection setup includes the controller type and its parameters.

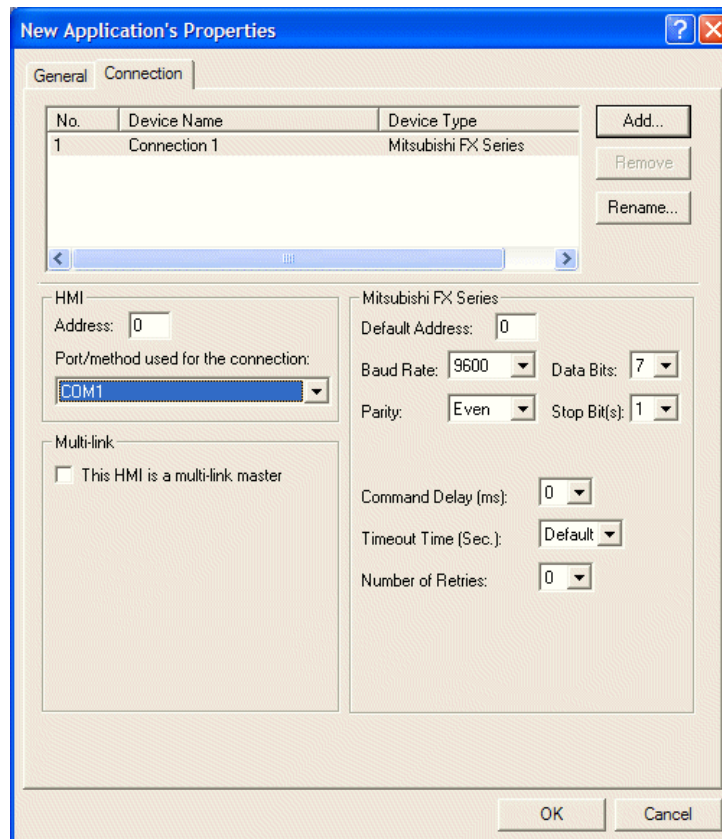
Follow the steps below to set up the connection:

1. In H-Designer, select **File/New**. The **Applications Properties** dialog box is displayed. On the **General** tab, select the type of the first controller from the **Controller/PLC** list, for example, **Mitsubishi FX Series**.



Selecting the type of the first controller

2. On the **Connection** tab, select the method of connection for the first controller from the **Port/method used for the connection** list. Enter the addresses in the **HMI Address** box and controller's **Default Address** box and make the appropriate selections for **Baud Rate**, **Data Bits**, **Parity** and **Stop Bits**.

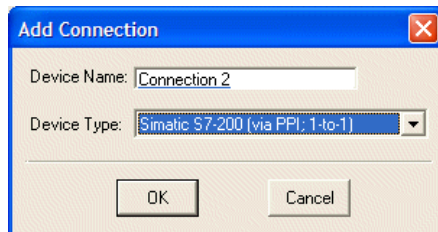


Setup of the connection with the first controller

Note:

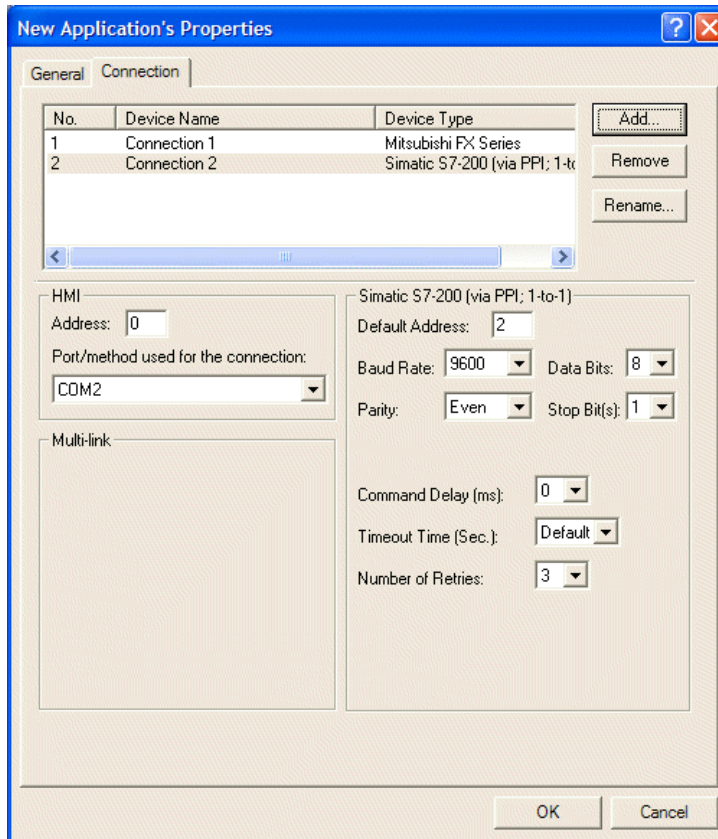
If **SW5 = ON** on the operator terminal, the parameters of the first linked controller must be set up according to the **Configuration Table** in the operator terminal's **System Menu**. If **SW5 = OFF**, the parameters of the first linked controller must be set up according to the **Connection tab** in **Application/Workstation Setup** in H-Designer. The switches are described in the Installation and Operation Manual for the operator terminal.

3. To add a second controller, click **Add** on the **Connection** tab, and select, for example, **Simatic S7-200 via PPI; 1-to-1**.



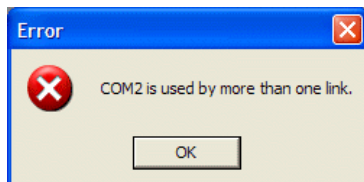
Adding a second linked controller

4. Enter the addresses in the **HMI Address** box and controller's **Default Address** box and make the appropriate selections for **Baud Rate**, **Data Bits**, **Parity** and **Stop Bits**.



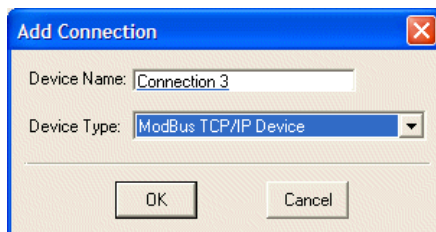
Setup up of the connection with the second controller

5. If the communication port is already being used, the following error message will be displayed.



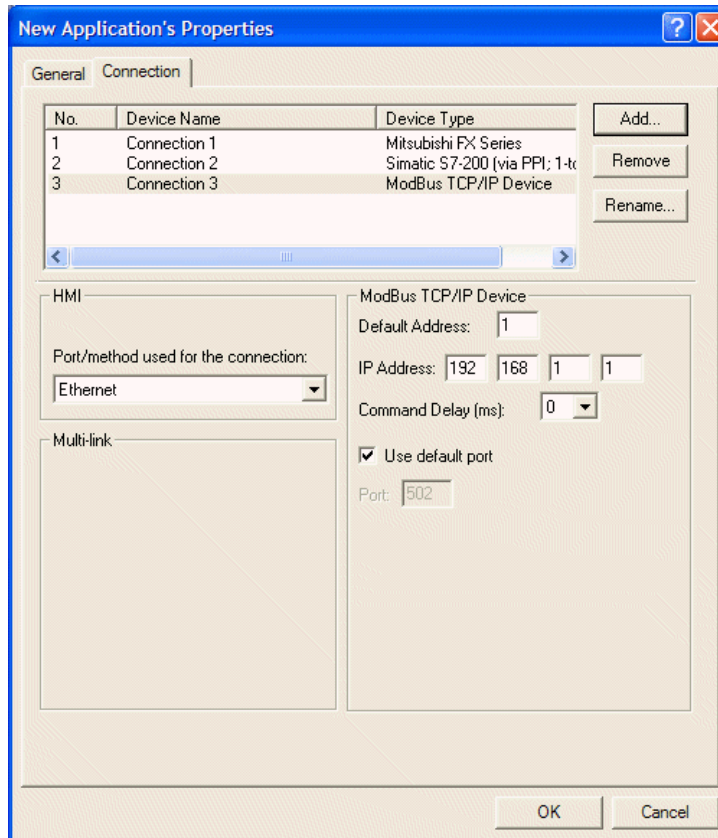
Error message when the port is already being used

6. To add an Ethernet-enabled controller, click **Add** again, to add a third controller, for example, **ModBus TCP/IP Device**.



Adding a third, Ethernet-enabled controller

7. Enter the addresses in the **Default Address** and **IP Address** boxes.



Setup of the connection with the third, Ethernet-enabled controller

8. Click **OK** to finish the setup. To change the setup later, simply select **Application/Workstation Setup**.

Description of the Devices block on the Connection tab

No.	Device Name	Device Type
1	Connection 1	Mitsubishi FX Series
2	Connection 2	Simatic S7-200 (via PPI; 1-to-1)
3	Connection 3	ModBus TCP/IP Device

Item	Description
Add	Click to add a new connection device/controller.
Remove	Click to delete a connection device. The first connection (Connection 1) cannot be deleted.
Rename	Click to change Device Name or Device Type . For Connection 1 , only the Device Name can be changed. The Device Type can be changed on the General tab.
No. column	Numbered according to the order in which devices are added and cannot be changed.

7.3 Read/Write Address Setup

Since there is more than one type of controller, a read/write address for each controller has to be assigned. The symbol / denotes the separation of the connection number from the register address.

Note:

This function is not available for all operator terminal models: please refer to [Appendix A - H-Designer Features and Operator Terminal Models](#) for details.

The connection in section [7.2 Connection Setup](#) is used in this example.

No.	Device Name	Device Type
1	Connection 1	Mitsubishi FX Series
2	Connection 2	Simatic S7-200 (via PPI;1 to 1)
3	Connection 3	Modbus TCP/IP Device

Perform the following steps:

1. For **Connection 1**, in the **Write** and **Read** boxes, enter **1/Y0** for the bit address and **1/D100** for the register address. **1** is for **No. column 1**, and can be omitted, so enter **Y0**.

Variable

Write: Y0

Read: Y0

☐ Enabled By:

2. For **Connection 2**, in the **Write** and **Read** boxes, enter **2/Q0.0** for the bit address. Note that **2** refers to **Connection No. 2** and **/** denotes the separation from its register address.

Variable

Write: 2/Q0.0

Read: 2/Q0.0

☐ Enabled By:

3. For **Connection 3**, in the **Write** and **Read** boxes, enter **3/1** for the bit address. Note that **3** refers to **Connection No. 3** and **/** denotes the separation from its register address.

Variable

Write: 3/1

Read: 3/1

☐ Enabled By:

8 Macros

8.1 Macro Function

H-Designer offers user a convenient and powerful macro application. It enables the operator terminal to execute a number of tasks: Arithmetic, Logic, Flow Control, Data Transfer, Comparison, Conversion and system service instructions, for example. Using macros can also significantly reduce program size and optimize controller efficiency. Macros not only allow the operator terminal to communicate with the controller, but can also connect it to other devices. Macros provide an efficient integration system as well as an economical hardware application structure.

8.2 Macro Classifications

Macros offer a number of functions for different situations and applications. The relevant macro window can be used to define an application according to needs. The operator terminal will execute the macro commands in accordance with different modes.

Macros can be categorized into **Application Macros**, **Screen Macros**, **ON/OFF Macros** and **Sub-Macros**.

8.2.1 Application Macros

There are three types of macro commands in the **Application** menu.

1. **INITIAL Macro**: The **INITIAL Macro** is used for data initialization and communication parameter declarations. This command is executed only once when an application is started, and the start-up screen does not appear until this command is executed. There is one **INITIAL Macro** in an application.
2. **BACKGROUND Macro**: When the operator terminal runs the application, these macros will be executed cyclically. The maximum size of macro commands are 30 rows. The macro commands will execute whatever the current screen. Common uses for the **BACKGROUND Macro** are communication control and controller sample data conversion.
3. **CLOCK Macro**: When the operator terminal runs this application, these macros will be executed every 500 ms. Common uses for the **CLOCK Macro** are display control, controller bit monitor, timer control and data timer conversion.

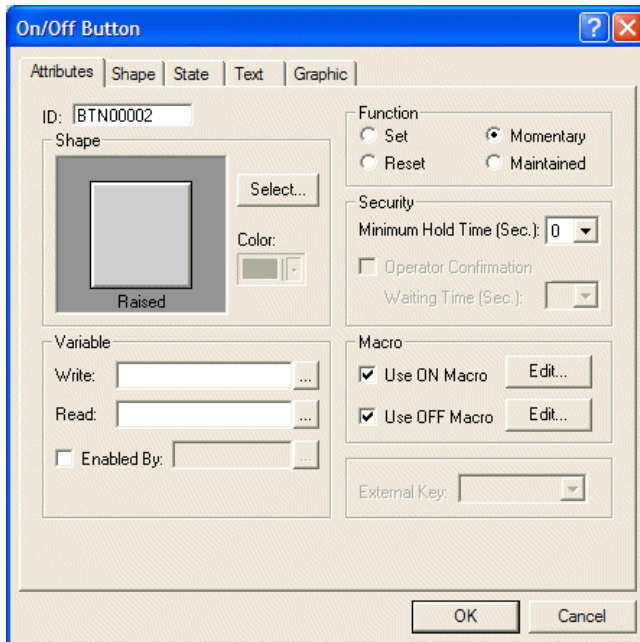
8.2.2 Screen Macros

There are three types of macro commands in the **Screen** menu.

1. **OPEN Macro**: The **OPEN Macro** is executed when the screen is commanded to be opened. Common uses for the **OPEN Macro** are screen initialization, display control, internal register or bit initialization.
2. **CLOSE Macro**: The **CLOSE Macro** is executed when the screen is commanded to be closed. The **CLOSE Macro** will execute its command once.
3. **CYCLIC Macro**: The **CYCLIC Macro** is executed cyclically when the screen is displayed. The operator terminal will execute the **BACKGROUND Macro** and **CLOCK Macro** periodically.

8.2.3 ON/OFF Macros

There are two ON/OFF Macro commands for push-button objects.



ON/OFF macros are available for the button object

ON Macro: The ON Macro is executed when the button is clicked and sets a bit to ON. Common uses for the ON Macro are push-button actions, chain process control, initial screen display and controller register and bit initialization.

OFF Macro: The OFF Macro is executed when the button is clicked and released sets a bit to OFF. The operator terminal will execute the OFF Macro commands once. Common uses for the OFF Macro are push-button actions, sequence process control and displaying the close screen.

8.2.4 Sub-Macros

The Sub-Macro can be selected from the Object menu.

Sub-Macro: The Sub-Macro is a sub-command of Macro. The Sub-Macro is executed by the operator terminal with the CALL command. Common uses for the Sub-Macro are to edit and save some basic functions and macro arithmetic commands.

8.3 Macro Commands

The following chart details the macro commands and their formats. For the set up procedure, please see the next section.

Operation	Format	A1*	A2*	A3*	Data Format	Controller Data
ADD	A1 = ADD (A2, A3)	2	2, 4	2, 4	DW/Signed	-
SUB	A1 = SUB (A2, A3)	2	2, 4	2, 4	DW/Signed	-
MUL	A1 = MUL (A2, A3)	2	2, 4	2, 4	DW/Signed	-
DIV	A1 = DIV (A2, A3)	2	2, 4	2, 4	DW/Signed	-
MOD	A1 = MOD (A2, A3)	2	2, 4	2, 4	DW/Signed	-
OR	A1 = A2 A3	2	2, 4	2, 4	DW	-
AND	A1 = A2 & A3	2	2, 4	2, 4	DW	-
XOR	A1 = A2 ^ A3	2	2, 4	2, 4	DW	-
SHL	A1 = A2 << A3	2	2, 4	2, 4	DW	-
SHR	A1 = A2 >> A3	2	2, 4	2, 4	DW	-
MOV	A1 = A2	0, 2	0, 2, 4	-	DW	Yes
BMOV	BMOV (A1, A2, A3)	0, 2	0, 2	2, 4		Yes
FILL	FILL (A1, A2, A3)	2	2, 4	2, 4		-
CHR	CHR (A1, "A2")	2	5	-	-	-
IF==	IF A1 == A2 GOTO A3	2, 4	2, 4	4	DW/Signed	-
IF!=	IF A1 != A2 GOTO A3	2, 4	2, 4	4	DW/Signed	-
IF>	IF A1 > A2 GOTO A3	2, 4	2, 4	4	DW/Signed	-
IF>=	IF A1 >= A2 GOTO A3	2, 4	2, 4	4	DW/Signed	-
IF<	IF A1 < A2 GOTO A3	2, 4	2, 4	4	DW/Signed	-
IF<=	IF A1 <= A2 GOTO A3	2, 4	2, 4	4	DW/Signed	-
IF AND ==0	IF A1 AND A2 == 0 THEN GOTO A3	2, 4	2, 4	4	DW	-
IF AND !=0	IF A1 AND A2 != 0 THEN GOTO A3	2, 4	2, 4	4	DW	-
IF==ON	IF A1 = ON GOTO	3	4	-	Bit	-
IF==OFF	IF A1 = OFF GOTO	3	4	-	Bit	-
IF-THEN	IF condition ** THEN DO ENDIF	2, 4	2, 4	-	Condition **	-
IF-THEN-ELSE	IF condition ** THEN DO ELSE DO ENDIF	2, 4	2, 4	-	Condition **	-
Nest IF-THEN-ELSE	IF condition ** THEN DO IF-THEN-ELSE ELSE DO IF-THEN-ELSE ENDIF	2, 4	2, 4	-	Condition **	-
ELIF	IF condition 1** THEN DO ELIF condition 2** THEN DO ELIF condition 3** THEN DO ENDIF	2, 4	2, 4	-	Condition **	-

** Condition includes A1==A2, A1!=A2, A1>A2, A1>=A2, A1<A2, A1<=A2, (A1&A2)==0, (A1&A2)!=0, A1==ON and A1==OFF. A1 and A2 are only for internal memory and constant.

* The usable range of memory will be identified according to the commands. The numbers in the table represent: 0 = Controller Device (word), 1 = Controller Device (bit), 2 = Internal Memory (word), 3 = Internal Memory (bit), 4 = Constant, 5 = ASCII Character

Operation	Format	A1*	A2*	A3*	Data Format	Controller Data
GOTO	Goto label A1	4	-	-		-
LABEL	Label A1	4	-	-		-
CALL	Call A1	2, 4	-	-		-
RET	Return	-	-	-		-
FOR	For A1	2, 4	-	-		-
NEXT	Next	-	-	-		-
SETB	Bit setting	1, 3	-	-	Bit	Yes
CLRB	Bit resetting	1, 3	-	-	Bit	Yes
INVB	Bit inversion	1, 3	-	-	Bit	Yes
BCD	A1 = BCD (A2)	2	2	-	DW	-
BIN	A1 = BIN (A2)	2	2	-	DW	-
W2D	A1 = W2D (A2)	2	2	-	Signed	-
B2W	A1 = B2W (A2, A3)	2	2	2, 4		-
W2B	A1 = W2B (A2, A3)	2	2	2, 4		-
SWAP	SWAP (A1, A2)	2	2, 4	-		-
MAX	A1 = MAX (A2, A3)	2	2, 4	2, 4	DW/Signed	-
MIN	A1 = MIN (A2, A3)	2	2, 4	2, 4	DW/Signed	-
A2H	A1 = A2H	2	2			-
H2A	A1 = H2A (A2)	2	2			-
TIMETICK	A1 = TIMETICK	2	-	-	DW	-
COMMENT	#A1 = "Chars"	5	-	-		-
SYS	SYS (A1, A2)					-
	SYS (SET_TIMER,N)		4			-
	SYS (STOP_TIMER,N)		4			-
	SYS (STOP_COUNTER,N)		4			-
	SYS (WAIT_TIMER,N)		4			-
	SYS (INI_COM,N)		4			-
	SYS (GET_CHAR,N)		4			-
	SYS (GET_CHARS,N)		4			-
	SYS (PUT_CHAR,N)		4			-
	SYS (PUT_CHARS,N)		4			-
	SYS (READ_WORDS,N)		4			-
	SYS (READ_BITS,N)		4			-
	SYS (WRITE_WORDS,N)		4			-
	SYS (WRITE_BIT,N)		4			-
	SYS (SUM_ADD,N)		4			-
	SYS (SUM_XOR,N)		4			-

* The usable range of memory will be identified according to the commands. The numbers in the table represent: 0 = Controller Device (word), 1 = Controller Device (bit), 2 = Internal Memory (word), 3 = Internal Memory (bit), 4 = Constant, 5 = ASCII Character

8.3.1 Arithmetic

Note:

Only internal memory can be used in these commands. The internal memory includes @, RCPW, CB, RCPNO and *@ (indirect internal memory). The data format is word, double-word, signed binary and unsigned binary.

Format: $A1 = A2 + A3$

Description: Adds A2 and A3 and saves the result in A1.

ADD → Format: $A1 = A2 + A3$. Adds A2 and A3 and saves the result in A1.

SUB → Format: $A1 = A2 - A3$. Subtracts A3 from A2 and saves the result in A1.

MUL → Format: $A1 = A2 \times A3$.

DIV → Format: $A1 = A2 / A3$. A1 is the quotient and A3 cannot be zero.

MOD → Format: $A1 = A2 \% A3$. A1 is the remainder and A3 cannot be zero.

8.3.2 Logical

Note:

Only internal memory can be used in these commands. The internal memory includes @, RCPW, CB, RCPNO and *@ (indirect internal memory). The data format is word, double-word etc. (no signed binary, floating point number arithmetic).

A	B	F
0	0	0
0	1	1
1	0	1
1	1	1

A	B	F
0	0	0
0	1	0
1	0	0
1	1	1

Truth tables for OR (to the left) and AND (to the right) logical operations

OR → Format: $A1 = A2 \mid A3$.

Performs the bit-wise **OR** operation on A2 (word) and A3 (word) and saves the result in A1 (word), or on A2 (double-word) and A3 (double-word) and saves the result in A1 (double-word).

AND → Format: $A1 = A2 \& A3$. Performs the bit-wise **AND** operation on A2 (word) and A3 (word) and saves the result in A1 (word) or on A2 (double-word) and A3 (double-word) and saves the result in A1 (double-word).

XOR → Format: $A1 = A2 \wedge A3$. Performs the bit-wise exclusive **OR** operation on A2 (word) and A3 (word) and saves the result in A1 (word) or on A2 (double-word) and A3 (double-word) and saves the result in A1 (double-word).

SHL → Format: $A1 = A2 \ll A3$. Shifts A2 (word) left by A3 bits and saves the result in A1 (word). The left shift command puts 0 into bit 0 and the last bit will shift out. If the displacement (A3) is greater than 16, then 16 will be the most shiftable amount.

Alternatively, shifts A2 (double-word) left by A3 bits and saves the result in A1 (double-word). The left shift command puts 0 into bit 0 and the last bit will shift out. If the displacement (A3) is greater than 32, then 32 will be the most shiftable amount.

SHR → Format: $A1 = A2 \gg A3$. Shifts A2 (word) right by A3 bits and saves the result in A1 (word). The right shift command puts 0 into bit 15 and the first bit will shift out. If the displacement (A3) is greater than 16, then 16 will be the most shiftable amount.

Alternatively, shifts A2 (double-word) right by A3 bits and saves the result in A1 (double-word). The right shift command puts 0 into bit 31 and the first bit will shift out. If the displacement (A3) is greater than 32, then 32 will be the most shiftable amount.

8.3.3 Data transfer

Note:

Both the MOV and BMOV commands can be located in the controller memory or internal memory. These include @, RCPW, CB, RCPNO and *@ (indirect internal memory). The data format for this command is word.

MOV → Format: $A1 \text{ (word)} = A2 \text{ (word)}$, $A1 \text{ (double-word)} = A2 \text{ (double-word)}$. The MOV command copies the value of A2 to A1 and the value of A2 is unchanged. If A1 is located in the controller, it represents the A2 data in the operator terminal internal register in the controller. If A2 is located in the controller, it will represent the data read and copied from the operator terminal internal register A1.

BMOV → Format: **BMOV** (A1, A2, A3). Copies a block of data starting at A2 to the memory block starting at A1. A3 specifies the number of words to be copied. The data format is word. The BMOV command copies a block of length A3 starting at A2 to the A3 long block starting at A1. The A2 data is unchanged. A3 must be between 2 and 524. Format: BMOV (A1, A2, A3).

FILL → Format: **FILL** (A1, A2, A3). Fill a block of memory starting at A1 with the value of A2. A3 specifies the number of words to be filled. The data format is word. The FILL command fills a block of A3 words starting at A1 with the A2 data. The A2 data is unchanged. A3 must be between 2 and 524.

CHR → Format: **CHR** (A1, "A2"). Copies the character string A2 to A1. The A1 data is in ASCII format.

8.3.4 Comparison

Note:

Only internal memory can be used in these commands. The internal memory includes @, RCPW, CB, RCPNO and *@ (indirect internal memory).

IF == → Format: **IF** A1 == A2 **THEN GOTO LABEL A3**. Goes to LABEL A3 if A1 is equal to A2.

IF != → Format: **IF** A1 != A2 **THEN GOTO LABEL A3**. Goes to LABEL A3 if A1 is not equal to A2.

IF > → Format: **IF** A1 > A2 **THEN GOTO LABEL A3**. Goes to LABEL A3 if A1 is greater than A2.

IF >= → Format: **IF** A1 >= A2 **THEN GOTO LABEL A3**. Goes to LABEL A3 if A1 is greater than or equal to A2.

IF < → Format: **IF** A1 < A2 **THEN GOTO LABEL A3**. Goes to LABEL A3 if A1 is less than A2.

IF <= → Format: **IF** A1 <= A2 **THEN GOTO LABEL A3**. Goes to LABEL A3 if A1 is less than or equal to A2.

IF AND == 0 → Format: IF (A1&A2) == 0 THEN GOTO LABEL A3. Goes to LABEL A3 if the result of an AND operation on A1 and A2 is 0.

IF AND! = 0 → Format: IF (A1&A2)! = 0 THEN GOTO LABEL A3. Goes to LABEL A3 if the result of an AND operation on A1 and A2 is not 0.

IF == ON → Format: IF A1==ON THEN GOTO LABEL A2. If bit A1 is ON (1), goes to LABEL A2.

IF ==OFF → Format: IF A1==OFF THEN GOTO LABEL A2. If bit A1 is OFF (0), goes to LABEL A2.

Example

Command: IF . . . DO; ELIF . . . DO; ELSE DO; ENDIF

Description: Use an IF statement when you want your macro to choose between two or more options. An IF statement consists of the keyword IF, a condition to be evaluated, the keyword THEN, the keyword DO, and the keyword ENDIF, as shown below:

```
IF condition THEN DO
    # statements to be executed if condition is true
ENDIF
```

The condition can be one of the following:

```
A1 == A2
A1 != A2
A1 > A2
A1 >= A2
A1 < A2
A1 <= A2
(A1 & A2) == 0
(A1 & A2) != 0
A1 == ON
A1 == OFF
```

The following IF statement structures are provided:

IF-THEN structure

The simplest IF statement evaluates a condition and performs a specified action if the condition is true. If the condition is not true, the entire statement is ignored. For example:

```
IF @100 == 50 THEN DO
    CALL 50
ENDIF
```

If @100 equals 50, sub-macro 50 is called. If @100 contains anything else, the entire statement is ignored.

IF-THEN-ELSE structure

An IF statement can also specify one or more statements to be executed if the condition is false. This option is indicated with the keyword ELSE, as shown below.

```
IF condition THEN DO
    # statements to be executed if condition is true
ELSE DO
```



```
# statements to be executed if condition is false
```

```
ENDIF;
```

In the example below, if @100 equals 50, the sub-macro 50 is called. If @100 is not equal to 50, sub-macro 1 is called following the else statement.

```
IF @100 == 50 THEN DO
```

```
    CALL 50
```

```
ELSE DO
```

```
    CALL 1
```

```
ENDIF
```

Nested IF-THEN-ELSE structure

You can create nested IF statements, in which one IF statement is embedded in another:

```
IF first condition THEN DO
```

```
    IF second condition THEN DO
```

```
        # statements to be executed if second condition is true
```

```
    ELSE DO
```

```
        # statements to be executed if second condition is false
```

```
    ENDIF
```

```
ELSE DO
```

```
    IF third condition THEN DO
```

```
        # statements to be executed if third condition is true
```

```
    ELSE DO
```

```
        # statements to be executed if third condition is false
```

```
    ENDIF
```

```
ENDIF
```

In the following example, if the value of @100 is 50, sub-macro 50 is called. If the value of @100 is 100, sub-macro 100 is called. If @100 is not equal to either of those values, the sub-macro 1 is called.

```
IF @100 == 50 THEN DO
```

```
    CALL 50
```

```
ELSE DO
```

```
    IF @100 == 100 THEN DO
```

```
        CALL 100
```

```
    ELSE DO
```

```
        CALL 1
```

```
    ENDIF
```

```
ENDIF
```

ELIF structure

The ELIF statement is provided to create IF structures in which the ELIF branch of one IF statement leads to another option:

```
IF first condition THEN DO
    # statements to be executed if condition is true
ELIF second condition THEN DO
    # statements to be executed if condition is false
ELIF third condition THEN DO
    # statements to be executed if third condition is false
ENDIF
```

In the following example, if @100 equals 50, sub-macro 50 is called. If @100 is not equal to 50, the program continues to the ELIF statement to test if @100 is equal to 100. If @100 equals 100, sub-macro 100 is called. If @100 is not equal to 100, the program moves to the next ELIF, and so on.

```
IF @100 == 50 THEN DO
    CALL 50
ELIF @100 == 100 THEN DO
    CALL 100
ELIF @100 == 150 THEN DO
    CALL 150
ENDIF
```

You cannot define a label inside an IF statement.

8.3.5 Flow Control

Note:

Only internal memory can be used in these commands.

GOTO → Format: GOTO LABEL A1. Goes to LABEL A1 unconditionally. The GOTO command will cause a branch to the specified label (Label A1). LABEL A1 must be in the macro.

LABEL → Format: LABEL A1. Note that no two labels are allowed to have the same number in one macro but the same number in different macros is acceptable.

CALL → Call Sub-macro. Format: CALL A1. The **CALL** command can assign control to a sub-macro. Common uses of sub-macros are to execute some specific functions, to pass the parameter table and complex instruction sets. Note that the specified sub-macro must exist and return via the **RET** command when the end of the sub-macro is reached. Then the next macro will be executed. The number of sub-macros is from 001 to 512, and they can be named. A sub-macro can also be assigned to **CALL** another sub-macro.

RET → Return to macro. The **RET** command only exists in sub-macro, although **CALL** exists in macro. Each **RET** command must have a corresponding **CALL** command.

FOR..NEXT → Loop, **FOR** is the start of a loop and **NEXT** is the end of a loop. Note that the maximum number of **FOR** loops is 3, for example, the FOR A1..NEXT. **FOR** loop is formed by the set of **FOR** and **NEXT** commands and executes the macro instructions within the **FOR** loop A1 times. A1 can be a variable or a constant. When A1 is 0, the macro will skip the **FOR** loop and execute the line of code following the **NEXT** command. When A1 is greater than 0, the macro will execute the loop continuously until the end of the **FOR** loop. The value of A1 can be changed within the **FOR** commands. Note that if A1 is too great, the CPU will overload and malfunction.

The **FOR/NEXT** loop command can execute the program repeatedly. Each **FOR** command must have one corresponding **NEXT** command. One is allowed to have up 3 nested **FOR** loops, such as FOR @1..., FOR @2..., FOR@3... NEXT, NEXT, NEXT.

END → End the macro. The **END** command represents the end of the macro. The macro will not execute the next line of code after the **END** command but will start at the first line of code next time the program is run.

Note:

The **END** command represents the end of the macro and is invalid in sub-macro. Sub-macro must use the **RET** command, otherwise, the program will cause errors.

8.3.6 Data Conversion

Note:

Only internal memory can be used in these commands. The internal memory includes @, RCPW, CB, RCPNO and *@ (indirect internal memory).

BCD → Convert BIN to BCD. Format: A1 = BCD (A2). This command is used to convert A2 (integer, word or double-word) from a binary number to a BCD number and saves the result in A1. Valid integer values of A2 are between 0 and 9999 (word) or 0 and 99999999 (double-word).

BIN → Convert BCD to BIN. Format: A1 = BIN (A2). This command is used to convert A2 from a BCD number (word or double-word) to a binary number and saves the result in A1 (integer, word or double-word). Valid BCD numbers are between 0 and 9999 (word) or 0 and 99999999 (double-word).

W2D → Convert WORD to DOUBLE WORD. Format: A1 = W2D (A2). The W2D command is used to convert A2 from a WORD number (integer) to a DOUBLE WORD (integer) and saves the result in A1 (double-word, signed, or unsigned). Valid integer values of A2 are between 0 and 65535 (word, unsigned) or -32768 and 32767 (word, signed). This function can extend the size of a 16-bit signed integer (word) to a 32-bit integer (double-word).

B2W → Convert BYTE to WORD. Format: A1 = B2W (A2, A3). The byte array starting at A2 with the size A3 and the result is saved in the memory starting at A1 (word). The high bytes of the word array are set to 0.

W2B → Convert WORD to BYTE. Format: A1 = W2B (A2, A3). The word array starting at A2 with the size A3. The result is saved in memory starting at A1. The conversion will discard the high bytes of the A2 word array.

SWAP → Swap the Bytes, Format: SWAP (A1, A2). The SWAP command is used to swap the low byte and high byte of each word of a memory block starting at A1. A2 specifies the size of the memory block in words. After execution, the A1 data will be changed.

MAX → Maximum. Format: A1 = MAX (A2, A3). Sets A1 to the larger of A2 and A3. (The data format can be word, dword, signed binary, or unsigned binary.)

MIN → Minimum. Format: A1 = MAX (A2, A3). Sets A1 to the smaller one of A2 and A3. (The data format can be word, dword, signed binary, or unsigned binary.)

A2H → Convert 4-digit hex number in ASCII character form into a binary number. Format: A1 = A2H (A2). The character of the fourth digit is in word A2 and the characters of the other digits are in the words following A2 in sequence. The result will be saved in A1. For example, suppose A2 is @200 and the data in @210=9538H. After the conversion, the result will be saved in A1=@210 and will be @200=0039H, @201=0033H, @202=0035H, and @203=0038H. (The data format is word only.)

H2A → Convert a 16-bit binary number into a 4-digit hex number in ASCII character form. Format: A1 = A2H (A2). The number to be converted is in A2. The character of the fourth digit will be saved in A1 and the characters of the other digits will be saved in the words following A1 in sequence. For example, suppose A2 is @100 and the data in @100=1234H. After the conversion, the result will be saved in A1=@110 and will be @110=0031H, @111=0032H, @112=0033H, and @113=0034H. (The data format is word only.)

8.3.7 Bit Setting

Note:

Both internal memory and controller bit can be used in these commands, including @nnn.b and RCPWnnn.b.

SETB → Set bit to ON. Format: SETB A1.

CLRB → Set bit to OFF. Format: CLRB A1.

INVB → Inverse bit state. Format: INVB A1.

8.3.8 Others

There are three special commands to use.

TIMETICK → Get the current system time tick (CPU internal clock time). Format: A1= TIMETICK (). The system time tick is increased by 1 in every 100 ms.

COMMENT → This is a non-executable instruction and it is used to comment macros.

SYS → There are a number of system services which can be used in the **SYS** command. Please see below for full details:

1. **SET_TIMER** → Specify the internal timer.
Format: SYS (SET_TIMER,N).

@N: Time number. N is between 0 and 7.

@N+1: Current Timer Value.

@N+2: Timer Limit.

@N+3: Time-up Flag.

@N+4: Type of Operation as below:

0 Timer will stop when it reaches the default setting, the flag will be set to 1.

1 Timer resets to 0 automatically when the flag is changed to 0 or 1. When the flag is 1, the timer resets to 0 automatically. When the flag is 0, the timer resets to 0 automatically.

STOP_TIMER → Stops the internal timer.

Format: SYS (STOP_TIMER,N).

2. **STOP_COUNTER** → Stop the internal counter.
Format: SYS(STOP_COUNTER,N).

3. **WAIT_TIMER** → Wait for the time-up event in the internal timer.
Format: SYS (WAIT_TIMER,N).

The macro instruction following this command will not be executed until the timer reaches the Timer Limit. Remember that the corresponding timer must be activated by the **SET_TIMER** service before requesting this service.

4. **INIT_COM** → Select and initialize a COM port.

Format: SYS (INIT_COM,N).

The word @n specifies the communication setting of the COM port. The format of the setting is shown below:

Bit 1, Bit 0 → DATA Bit S 10:7 Bit S, 11:8 Bit S.

Bit 2 → STOP Bit S 0:1 Bit,1:2 Bit S.

Bit 4, Bit 3 → PARITY.> 00:NONE, 01:ODD, 11:EVEN.

Bit 6, Bit 5 → COM PORT > 00: COM1, 01: COM2, 10: COM3, 11: COM4.

Bit 7 → Not used.

Bit11, Bit 10, Bit 9, Bit 8 → 0001: 115200, 0010: 57600, 0011: 38400, 0110: 19200, 1100: 9600, Others: 4800.

Bit 15 → Computer Protocol Driver; 0: Disable, 1: Enable

If this service is successful, the word @n+1 will be set to 1; otherwise, it will be set to 0.

Some models provide a Computer Protocol slave driver for the second COM port. This function provides communication between PC/another operator terminal on the second COM port. The operator terminal can communicate with the controller over the first COM port and the connection steps are the same as the normal steps. The P PC can read from the internal registers @0-@8191 (the data in W0-W8191 corresponds to @0-@8191).

When using the Computer Protocol driver, this function is unavailable to GET_CHAR, GET_CHARS, PUT_CHAR, and PUT_CHARS.

5. **GET_CHAR** → Gets a character from the COM port.

Format: SYS (GETCHAR,N).

The character will be saved in the low byte of the word @n. If there is no input, the word @n will be set to be -1(ffffH).

6. **GET_CHARS** → Gets a number of characters from the COM port.

Format: SYS (GETCHARS,N).

The word @n specifies the maximum number of characters to receive. The actual number of characters received is saved in word @n+1. The characters received will be saved in the low bytes of the words @n+2, @n+3, @n+4, and so on.

7. **PUT_CHAR** → Sends a character in the low byte of the word @n to the COM port.

Format: SYS (PUTCHAR,N).

If this service is successful, the word @n+1 will be set to 1; otherwise, it will be set to another value.

8. **PUT_CHARS** → Sends the characters in the low bytes of the words starting from @n+2 to the COM port.

Format: SYS (PUTCHARS,N).

The word @n specifies the number of characters to be sent and the actual number of characters sent is saved in the word @n+1.

9. **SUM_ADD** → Calculates the sum of a block of words by normal arithmetic addition.

Format: SYS (SUM_ADD,N).

The output data is saved in “@N+3”. This feature offers a more convenient application for macros. For example, the command SYS(SUM_ADD,30) (here N = 30) will calculate the sum of the @30, @31, @32, @33 internal registers.

@N=30 represents the pointer parameter, and the internal value of @30 must be 0.

@N+1(@31) represents the starting address of the block.

@N+2(@32) represents the size of the WORDS block.

@N+3(@33) represents the initial value of the summand and the sum will be saved in this address automatically. The command must be set before execution. Most communication protocols regulate the initial value of the summand = 00H or FFH, so please refer to initial value assigned by the vendor.

10. **SUM_XOR** → Calculate the sum of a block of words by the bit-wise logical exclusive-or operation and save the result in the specified address.

Format: SYS(SUM_XOR,N).

The output data will be saved in “@N+3”. This function is convenient for macro communication applications. For example, SYS(SUM_XOR,50) (here N = 50) will calculate the sum of the @50, @51, @52, @53 internal registers. Execution of this command requires the internal values of @50, @51, @52 and @53.

@N=50 represents controller station number and the internal value of @50 must be 0 if no controller station is required.

@N+1(@51) represents the starting address of the block.

@N+2(@52) represents the size of the WORDS block.

@N+3(@53) represents the initial value of the summand and the sum will be saved in this address automatically. The command must be set before execution. Most communication protocols regulate the initial value of the summand = 00H or FFH, so please refer to initial value assigned by the vendor.

11. **READ_WORDS** → Read a number of words from controller word devices or internal memory and save the result in the specified address.

Format: SYS (READ_WORDS,N).

The data will be saved in “@+5”. This command is powerful for use in communication with any controller registers and can be used for setting and monitoring controller data. Take, for example, SYS(READ_WORDS,80) (here N = 80). Execution of this command requires the internal values of @80, @81, @82, @83, @84, @85 and @86.

@N (@80) represents the controller station number, and the internal value of @80 must be 0 if no controller station is required.

@N+1(@81) represents the device type setting. For the device type of controller, please see the chapter *Communication between Operator Terminal and Controller* for full details.

@N+2(@82) represents the low word of the device address.

@N+3(@83) represents the high word of the device address.

@N+4(@84) represents the auxiliary address if required else set to 0.

@N+5(@85) represents the address of the internal memory to receive the data and the size of data is specified by N+6(@86).

@N+6(@86) represents the number of words to be read.

12. **READ_Bit** → Read a controller bit device or internal bit and save the data in the specified address.

Format: SYS (READ_Bit,N).

The data will be saved in “@+5”. This command is powerful for use in communication with any controller bit-state and can be used to set and monitor controller data. For example, SYS(READ_bit,80) (here N = 80). Execution of this command requires the internal values of @80, @81, @82, @83, @84 and @85.

@N(@80) represents the controller station number, and the internal value of @80 must be 0 if no controller station is required.

@N+1(@81) represents the device type. For controller device types, please see the chapter *Communication between Operator Terminal and Controller* for full details.

@N+2(@82) represents the low word of the device address.

@N+3(@83) represents the high word of the device address.

@N+4(@84) represents the auxiliary address if required else set to 0.

@N+5(@85) represents the address of the internal memory to receive the data.

N+5(@85) DATA = 1 if the bit is ON; DATA = 0 if the bit is OFF.

13. **WRITE_WORDS** → Writes a block of data in internal memory to controller word devices or internal memory.

Format: SYS(WRITE_WORDS,N).

The data will be saved in “@N+5”. This command is powerful for the random modification of controller data and can be used to set and monitor controller data. For example, SYS(WRITE_WORDS,90) (here N = 90). Execution of this command requires the internal values of @90, @91, @92, @93, @94, @95 and @96.

@N(@90) represents the controller station number, and the internal value of @90 must be 0 if no controller station is required.

@N+1(@91) represents the device type. For controller device types, please see the chapter *Communication between Operator Terminal and Controller* for full details.

@N+2(@92) represents the low word of the device address.

@N+3(@93) represents the high word of the device address.

@N+4(@94) represents the auxiliary address if required else set to 0.

@N+5(@95) represents the source address while the size of the continuous block of data is assigned by N+6 (@96).

@N+6(@96) represents the number of words in the data block.

14. **WRITE_Bit** → Set a controller bit device or internal bit to the state of an internal word.

Format: SYS(WRITE_Bit,N).

The source address is “@+5”. This command is powerful for the random modification of controller data and can be used to set and monitor controller data. For example, SYS(WRITE_Bit,90) (here N = 90). Execution of this command requires the internal values of @90, @91, @92, @93, @94 and @95.

@N(@90) represents the controller station number, and the internal value of @90 must be 0 if no controller station is required.

@N+1(@91) represents the device type. For controller device types, please see the chapter *Communication between Operator Terminal and Controller* for full details.

@N+2(@92) represents the low word of the device address.

@N+3(@93) represents the high word of the device address.

@N+4(@94) represents the auxiliary address if required else set to 0.

@N+5(@95) represents the address of the internal memory to receive the data.

N+5(@95) DATA = 1 if the bit is ON; DATA = 0 if the bit is OFF.

8.4 Cautions

The last line of code must be the **RET** command, otherwise an error will occur when you compile.

Except in sub-macro, the **END** command marks the end of the macro.

The CPU will execute other programs after the execution of INITIAL Macro, CLOCK Macro, ON/OFF Macro, OPEN Macro and CLOSE Macro.

For BACKGROUND Macro, CYCLIC Macro and sub-macro, the CPU executes the 30 command lines once. The CPU will then execute other programs. The CPU will execute the 30 command lines following the last executed command until the next cycle.

To use the macro communication function, the user must define the related communication format for INICOM. This command is only used once, so it is usually entered in INITIAL Macro.

8.5 Internal Memory

The operator terminal provides some internal registers that can be read from/written to. By using these internal registers, users can make more efficient and convenient use of the macro function. The internal registers can not only enhance macro with infinite functions, but can also store a great deal of arithmetic source data and results. Note that this system provides the internal registers divided into RAM and ROM.

Details of the four types of internal memory follow:

Words	Device Type	Size	Address	Aux. Address	R/W
RCPNO	0x81	W	0 (only one word)	0	R/W
RCPWn	0x82	W	0 - 8191	0	R/W
CBn	0x83	W	0 - 31	0	R
@n	0x85	W	8191	0	R/W

Bits	Device Type	Address	Aux. Address	R/W
CBn.b (b=0-f)	0x83	0 - 31	0 - 15	R/W
RCPWn.b (b=0-f)	0x84	0 - 8191	0 - 15	R
@n.b (b=0-f)	0x86	8191	0 - 15	R/W

1. RCPNO.
2. The n value of RCPWn is based on the size of the recipe and the maximum number. The data register can be used as bit.
3. The n value of CBn is based on the size of the control block. The current size is 2-32. This data register can be used as bit.
4. @n: Internal Register. The size is 8191 WORDS (n=0-8191). This data register can be used by bit.

9 Communication between Operator Terminal and Controller

Note:

The **Register and Range of Relay Numbers** in this document refer to the setup range available in H-Designer. In practice, please make sure not to exceed the maximum range defined by the controller CPU in use. Otherwise communication errors will occur.

This chapter provides current information on the settings of the supported controllers at the time of writing. As controller vendors continue to release new models, controller data capacity, data range and drivers are also often updated. Please visit www.beijerelectronics.com for the latest information.

For correct communication between the controller and operator terminal, the **Communication Format**, **Station**, **Baud Rate** and **Data Format** settings must be consistent.

Before connection, please setup the operator terminal and controller communication parameter and DIP switch settings.

9.1 Allen Bradley Micrologic 1000/1500

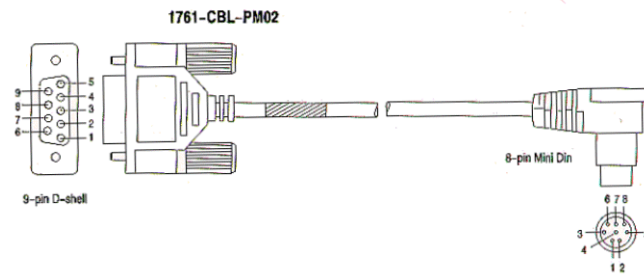
The controller data settings and ranges which H-Designer can access: Word Device.

Register Type	Format	Register Range	Device Type/ Aux. Address		Data Size R/W
Output file	O:0.n	n=0-3	0	0	Word
Input file	I:1.n	n=0-3	1	0	Word
Status file	S2:nn	nn=0-65	2	2	Word
Bit file	B3:nnn	nnn=0-254	3	3	Word
Timer flag	T4:nnn	nnn=0-254	4	4	Word
Timer Preset Value	T4:nnn.pre	nnn=0-254	5	4	Word
Timer Accumulator Value	T4:nnn.acc	nnn=0-254	6	4	Word
Counter flag	C5:nnn	nnn=0-254	7	5	Word
Counter Preset Value	C5:nnn.pre	nnn=0-254	8	5	Word
Counter Accumulator Value	C5:nnn.acc	nnn=0-254	9	5	Word
Control file	R6:nnn	nnn=0-254	10	6	Word
Control Size of Bit Array	R6:nnn.len	nnn=0-254	11	6	Word
Control Reserved file	R6:nnn.pos	nnn=0-254	12	6	Word
Integer file	N7:nnn	nnn=0-254	13	7	Word
Floating point number	F8:nnn	nnn=0-254	14	8	Word
The operator terminal does not support block read for the registers in TIMER, COUNTER and CONTROL FILES. It is necessary to open the files which the operator terminal will access in the Allen Bradley controller.					

Data format and range of the controller ON/OFF location which H-Designer can access: Dip Device.

Relay Type	Format	Relay Range	Device Type/ Aux. Address		Data Size R/W
Output	O:0.n/bb	n=0-3; bb=0-15	0xC0	0-15	Word
Input	I:1.n/bb	n=0-3; bb=0-15	0xC1	0-15	Word
Status	S2:nn/bb	nn=0-65; bb=0-15	0xC2	0-15	Word
Bit	B3:nnn/bb	nnn=0-254; bb=0-15	0xC3	0-15	Word
Timer	T4:nnn/bb	nnn=0-254; bb=0-15	0xC4	0-15	Word
Timer Preset Value	T4:nnn.pre/bb	nnn=0-254; bb=0-15	0xC5	0-15	Word
Timer Accumulator Value	T4:nnn.acc/bb	nnn=0-254; bb=0-15	0xC6	0-15	Word
Counter flag	C5:nnn/bb	nnn=0-254; bb=0-15	0xC7	0-15	Word
Counter Preset Value	C5:nnn.pre/bb	nnn=0-254; bb=0-15	0xC8	0-15	Word
Counter Accumulator Value	C5:nnn.acc/bb	nnn=0-254; bb=0-15	0xC9	0-15	Word
Control	R6:nnn/bb	nnn=0-254; bb=0-15	0xCA	0-15	Word
Control Size of Bit Array	R6:nnn.len/bb	nnn=0-254; bb=0-15	0xCB	0-15	Word
Control Reserved	R6:nnn.pos/bb	nnn=0-254; bb=0-15	0xCC	0-15	Word
Integer	N7:nnn/bb	nnn=0-254; bb=0-15	0xCD	0-15	Word
The operator terminal does not support block read for the registers in TIMER, COUNTER and CONTROL FILES.					

Cable Drawings



HMI-COM port		PLC-port RS232C
25-pin	-----CABLE-----	9-pin male
RXD 3		2 SD
TXD 2		3 RD
GND 7		5 SG
RTX 4		
CTX 5		

HMI-COM port		1761-CBL-PM02
9-pin	-----CABLE-----	9-pin male
RXD 2		2 SD
TXD 3		3 RD
GND 5		5 SG
RTX 7		
CTX 8		

Operator terminal to RS232C connection on controller 1761-CBL-PM02

Communication Format

Before connection, please set up the communication parameters and the DIP switches as follows:

Format	Controller Setting	Operator Terminal Setting
Communication Format	RS232C	COM2 = RS232/422/485 When using RS422, set DIP switch SW10=OFF. When using RS485, set DIP switch SW10=ON.
Node Address		Set PLC Station No. in H-Designer according to the controller setting.
Transmission Speed	19200 bps	Set the SW5=OFF if parameters are set in H-Designer.
Transmission Format	Size: 8-bit	Set the SW5=ON if parameters are set in the operator terminal.
	Parity: NONE	
	Stop bit: 1-bit	
COM Port	FULL DUPLEX	
CRC Error Check	YES	

9.2 Allen Bradley PLC-5

Data format and range of the controller registers which H-Designer can access: Word Device.

Register Type	Format	Register Range	Data Size R/W
Output file	O:nnn	nnn= 0-277	Word
Input file	I:nnn	nnn= 0-277	Word
Status file	S:nnn	nnn= 0-127	Word
Bit file	Bfff:nnn B:nnn	fff= 3 or 9-999; default file is 3 if fff omitted; nnn=0-999	Word
Timer file	Tfff:nnn T:nnn Tfff:nnn.PRE T:nnn.PRE Tfff:nnn.ACC T:nnn.ACC	fff= 4 or 9-999; default file is 4 if fff omitted; nnn=0-999	Word
Counter file	Cfff:nnn C:nnn Cfff:nnn.PRE C:nnn.PRE Cfff:nnn.ACC C:nnn.ACC	fff=5 or 9-999; default file is 5 if fff omitted; nnn=0-999	Word
Control file	Rfff:nnn R:nnn Rfff:nnn.LEN R:nnn.LEN Rfff:nnn.POS R:nnn.POS	fff=6 or 9-999; default file is 6 if fff omitted; nnn=0-999	Word
Integer file	Nfff:nnn N:nnn	fff=7 or 9-999; default file is 7 if fff omitted; nnn=0-999	Word

The operator terminal can read up to 30 words in one read command. The operator terminal does not support block read for the registers in TIMER, COUNTER and CONTROL FILES.

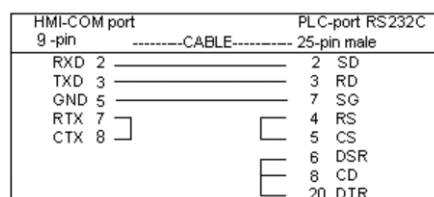
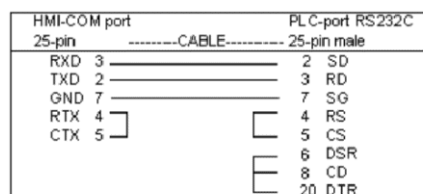
It is necessary to open the files which the operator terminal will access in the Allen Bradley controller.

Data format and range of the controller ON/OFF location which H-Designer can access: Bit Device.

Relay Type	Format	Relay Range
Output file	O:nnn/bb	nnn= 0-277; bb= 0-17
Input file	I:nnn/bb	nnn= 0-277; bb= 0-17
Status file	S:nnn/bb	nnn= 0-127; bb= 0-15
Bit file	Bfff:nnn/bb	fff= 3 or 9-999; default file is 3 if fff omitted; nnn=0-999; bb= 0-15
Timer file	Tfff:nnn/bb Tfff:nnn.PRE/bb Tfff:nnn.ACC/bb T:nnn/EN T:nnn/TT T:nnn/DN	fff= 4 or 9-999; default file is 4 if fff omitted; nnn=0-999; bb= 0-15
Counter file	Cfff:nnn/bb Cfff:nnn.PRE/bb Cfff:nnn.ACC/bb Cfff:nnn/CC Cfff:nnn/CD Cfff:nnn/DN Cfff:nnn/OV Cfff:nnn/UN Cfff:nnn/UA	fff=5 or 9-999; default file is 5 if fff omitted; nnn=0-999; bb= 0-15
Control file	Rfff:nnn/bb Rfff:nnn.LEN/bb Rfff:nnn.POS/bb Rfff:nnn/EN Rfff:nnn/EU Rfff:nnn/DN Rfff:nnn/EM Rfff:nnn/ER Rfff:nnn/UL Rfff:nnn/IN Rfff:nnn/FD	fff=6 or 9-999; default file is 6 if fff omitted; nnn=0-999; bb= 0-15
Integer file	Nfff:nnn/bb	fff=7 or 9-999; default file is 7 if fff omitted; nnn=0-999; bb= 0-15

The operator terminal can read up to 480 bits in one read command. The operator terminal does not support block read for the bits in TIMER, COUNTER and CONTROL FILES.

Cable Drawings



Operator terminal to RS232C on PLC-5

Communication Format

Before connection, please set up the communication parameters and the DIP switch as follows:

Format	Controller Setting	Operator Terminal Setting
Communication Format	RS232C PLC-5	COM2 = RS232/422/485 When using RS422, set DIP switch SW10=OFF. When using RS485, set DIP switch SW10=ON.
Node Address		Set PLC Station No. in H-Designer according to the controller setting.
Transmission Speed	9600/19200 bps	Set the SW5=OFF if parameters are set in H-Designer.
Transmission Format	Size: 8-bit	Set the SW5=ON if parameters are set in the operator terminal.
	Parity: NONE	
	Stop bit: 1-bit	
COM Port	FULL DUPLEX	
BCC Error Check	YES	

9.3 Allen Bradley SLC-503/504

Data format and range of the controller registers which H-Designer can access: Word Device.

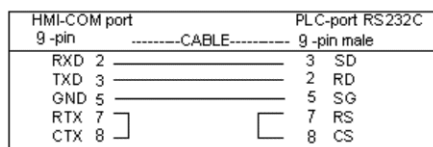
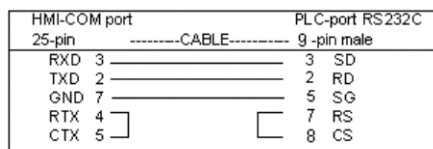
Register Type	Format	Register Range	Data Size
Output file	O:nn	nn= 0-30	Word
Input file	I:nn	nn= 0-30	Word
Status file	S:nn	nn= 0-127	Word
Bit file	Bfff:nnn B:nnn	fff= 3 or 10-255; default file is 3 if fff omitted; nnn=0-254	Word
Timer file	Tfff:nnn T:nnn Tfff:nnn.PRE T:nnn.PRE Tfff:nnn.ACC T:nnn.ACC	fff= 4 or 10-255; default file is 4 if fff omitted; nnn=0-254	Word
Counter file	Cfff:nnn C:nnn Cfff:nnn.PRE C:nnn.PRE Cfff:nnn.ACC C:nnn.ACC	fff=5 or 10-255; default file is 5 if fff omitted; nnn=0-254	Word
Control file	Rfff:nnn R:nnn Rfff:nnn.LEN R:nnn.LEN Rfff:nnn.POS R:nnn.POS	fff=6 or 10-255; default file is 6 if fff omitted; nnn=0-254	Word
Integer file	Nfff:nnn N:nnn	fff=7 or 10-255; default file is 7 if fff omitted; nnn=0-254	Word
<p>The operator terminal can read up to 30 words in one read command. The operator terminal does not support block read for the registers in TIMER, COUNTER and CONTROL FILES.</p> <p>It is necessary to open the files which the operator terminal will access in the Allen Bradley controller.</p>			

Data format and range of the controller ON/OFF location which H-Designer can access: Bit Device.

Relay Type	Format	Relay Range
Output file	O:nn/bb	nn= 0-30; bb= 0-15
Input file	I:nn/bb	nn= 0-30; bb= 0-15
Status file	S:nn/bb	nn= 0-31; bb= 0-15
Bit file	Bfff:nnn/bb	fff= 3 or 10-255; default file is 3 if fff omitted; nnn=0-254; bb= 0-15
Timer file	Tfff:nnn/bb Tfff:nnn.PRE/bb Tfff:nnn.ACC/bb T:nnn/EN T:nnn/TT T:nnn/DN	fff= 4 or 10-255; default file is 4 if fff omitted; nnn=0-254; bb= 0-15
Counter file	Cfff:nnn/bb Cfff:nnn.PRE/bb Cfff:nnn.ACC/bb Cfff:nnn/CU Cfff:nnn/CD Cfff:nnn/DN Cfff:nnn/OV Cfff:nnn/UN	fff=5 or 10-255; default file is 5 if fff omitted; nnn=0-254; bb= 0-15
Control file	Rfff:nnn/bb Rfff:nnn.LEN/bb Rfff:nnn.POS/bb Rfff:nnn/EN Rfff:nnn/DN Rfff:nnn/ER Rfff:nnn/UL Rfff:nnn/IN Rfff:nnn/FD	fff=6 or 10-255; default file is 6 if fff omitted; nnn=0-254; bb= 0-15
Integer file	Nfff:nnn/bb	fff=7 or 10-255; default file is 7 if fff omitted; nnn=0-254; bb= 0-15

The operator terminal can read up to 480 bits in one read command. The operator terminal does not support block read for the bits in TIMER, COUNTER and CONTROL FILES.

Cable Drawings



Operator terminal to RS232C on SLC-503/504

Communication Format

Before connection, please set up the communication parameters and the DIP switch as follows:

Format	Controller Setting	Operator Terminal Setting
Communication Format	RS232C SLC-503-504	COM2 = RS232/422/485 When using RS422, set DIP switch SW10=OFF. When using RS485, set DIP switch SW10=ON.
Node Address		Set PLC Station No. in H-Designer according to the controller setting.
Transmission Speed	9600/19200 bps	Set the SW5=OFF if parameters are set in H-Designer.
Transmission Format	Size: 8-bit	Set the SW5=ON if parameters are set in the operator terminal.
	Parity: NONE	
	Stop bit: 1-bit	
COM Port	FULL DUPLEX	
BCC Error Check	YES	

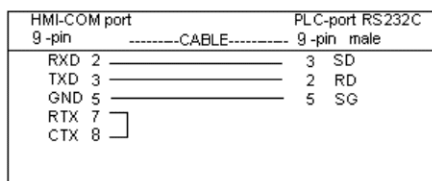
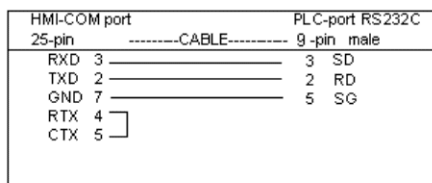
9.4 Allen Bradley IQ Master Servo Controller

Data format and range of the controller registers which H-Designer can access: Word Device and Bit Device.

Register Type	Format	Register Range	Data Size
G type	Gnn	nn= 1-64	Double Word (32 bits)
V type	Vnn	nn= 1-64	Double Word (32 bits)
G type	WGnn	nn= 1-64	Word (16 bits)
V type	WVnn	nn= 1-64	Word (16 bits)

Relay Type	Format	Relay Range
I type	Inn	nn= 1-48
O type	Onn	nn= 1-24
B type	Bn	nn= 1-8
F type	Fnn	nn= 1-64

Cable Drawings



Operator terminal to RS232 port (9-pin male) on controller

Communication Format

Before connection, please set up the communication parameters and the DIP switch as follows:

Format	Controller Setting	Operator Terminal Setting
Communication Format	RS232C	COM1 or COM2 = RS232
Station No	0	
Transmission Speed	9600 bps	Set the SW5=OFF if parameters are set in H-Designer. Set the SW5=ON if parameters are set in the operator terminal.
Transmission Format	Size: 8-bit	
	Parity: NONE	
	Stop bit: 1-bit	

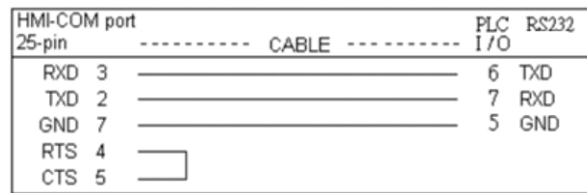
9.5 ABB Comli (Slave Mode)

Data format and range of the controller registers which H-Designer can access: Word Device and Bit Device.

Register Type	Format	Register Range	Data Size
Word IO	MWnnnnn	nnnnn= 0-37760 (must be a multiple of 8)	Word (16 bits)
V type	RWnnnnn	nnnn= 0-3071	Word (16 bits)

Relay Type	Format	Realy Range
Bit IO	Mnnnnn	nnnnn= 0-37777 (8 bits)

Cable Drawing



Operator terminal to RS232 port on controller

Communication Format

Before connection, please set up the communication parameters and the DIP switch as follows:

Format	Controller Setting	Operator Terminal Setting
Communication Format	RS232C	RS232C
Station No	01 (controller sets 2-197)	
Transmission Speed	9600 bps	Set the SW5=OFF if parameters are set in H-Designer. Set the SW5=ON if parameters are set in the operator terminal.
Transmission Format	Size: 8-bit	
	Parity: ODD	
	Stop bit: 1-bit	

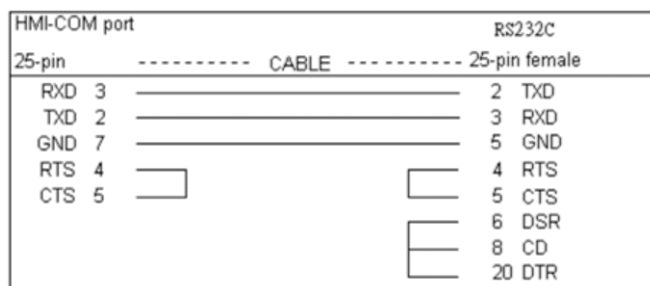
9.6 Computer (as Master/Slave/V2/Null) / Modbus Master

Data format and range of the controller registers which H-Designer can access: Word Device and Bit Device.

Register Type	Format	Register Range	Data Size
Data Register	Wnnnn	nnnn= 0-2047	Word (16 bits)

Relay Type	Format	Realy Range
Bit Realy	Bnnnn	nnnn= 0-1023

Cable Drawing



Operator terminal to RS232 port

Note:

This figure is a PC simulation, and the hook up method depends on the actual controller pin position.

Communication Format

Before connection, please set up the communication parameters and the DIP switch as follows:

Format	Controller Setting	Operator Terminal Setting
Communication Format	RS232C (RS422/RS485)	RS232C (RS422/RS485)
Station No	NONE	
Transmission Speed	9600 bps	Set the SW5=OFF if parameters are set in H-Designer. Set the SW5=ON if parameters are set in the operator terminal.
Transmission Format	Size: 8-bit	
	Parity: NONE	
	Stop bit: 1-bit	

Note:

COMPUTER (AS MASTER) V2 does not only include the function of COMPUTER (AS MASTER) - it also can inform the slaves when the data is changed (e.g. value input).
The Null function includes macro and ladder but no communication with any controller.

9.7 Delta DVP

Data format and range of the controller registers which H-Designer can access: Word Device and Bit Device.

Register Type	Format	Register Range	Data Size
S_Data	Snnnn	nnnn=0-1008 (must be a multiple of 16)	Word (16 bits)
X_Data	Xnnn	nnn= 0-360 (X0-X7, X10-X17, must be a multiple of 20)	Word (16 bits)
Y_Data	Ynnn	nnn= 0-360 (Y0-Y7, Y10-Y17, must be a multiple of 20)	Word (16 bits)
M_Data	Mnnnn	Nnnn=0-1264, must be a multiple of 16	Word (16 bits)
T_Register	Tnnn	nnn= 0-255	Word (16 bits)
C_Register	Cnnn	nnnn= 0-127	Word (16 bits)
D_Register	Dnnnn	nnn= 0-1279	Word (16 bits)
C_Register	Cnnn	nnn= 232-255	Double-Word

Relay Type	Format	Relay Range	Block
S_Data	Snnnn	nnn=0-1023	Multiple of 16
X_Data	Xnnn	nnn=0-377 (Oct. code)	Multiple of 16
Y_Data	Ynnn	nnn=0-377 (Oct. code)	Multiple of 16
M_Data	Mnnnn	nnnn=0-1279	Multiple of 16
T_Coil	Tnnn	nnn=0-254	Multiple of 16
C_Coil	Cnnn	nnn=0-254	Multiple of 16

Note:

X_Data and Y_Data are Oct. code, such as X0-X7, X10-X17, X20-X27 or X30-X37.

Connection

The operator terminal can be connected to the controller RS232C port using a DELTA 8-pin male connector (PC ↔ DELTA DVP PLC).

Communication Format

Before connection, please set up the communication parameters and the DIP switch as follows:

Format	Controller Setting	Operator Terminal Setting
Communication Format	RS232C	COM1 or COM2 = RS232C
Station No	01	Set PLC Station No.= 01 in H-Designer
Transmission Speed	9600 bps	Set the SW5=OFF if parameters are set in H-Designer. Set the SW5=ON if parameters are set in the operator terminal.
Transmission Format	Size: 7-bit	
	Parity: EVEN	
	Stop bit: 2-bit	

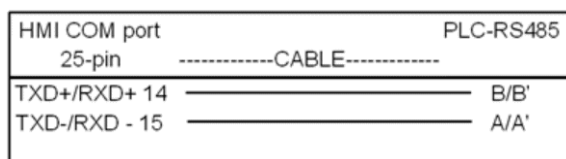
9.8 Ero TFS/THS/LFS

Data format and range of the controller registers which H-Designer can access:

Register Type	Format	Register Range	Data Size
Word Register	mmm:Wnnn	mmm=0-255 ; nnn=0-529	Word (16 bits)

Relay Type	Format	Realy Range
Realy	mmm:Bnnn	mmm=0-255 ; nnn=1-342

Cable Drawing



Operator terminal to RS485 port on controller

Communication Format

Before connection, please set up the communication parameters and the DIP switch as follows:

Format	Controller Setting	Operator Terminal Setting
Communication Format	RS485	RS485
Station No	None	Set the SW5=OFF if parameters are set in H-Designer. Set the SW5=ON if parameters are set in the operator terminal.
Transmission Speed	9600 bps	
Transmission Format	Size: 8-bit	
	Parity: EVEN	
	Stop bit: 1-bit	

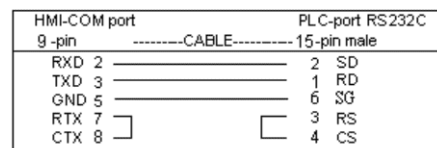
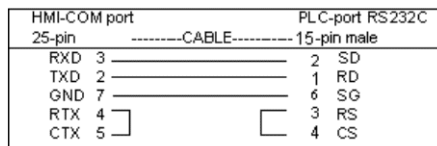
9.9 Facon FB

Data format and range of the controller registers which H-Designer can access: Word Device and Bit Device.

Register Type	Format	Register Range	Data Size
Input Relay	WXnnn	nnn=0-9984; (0 or multiple of 8)	Word
Output Relay	WYnnn	nnn=0-9984; (0 or multiple of 8)	Word
Internal Relay	WMnnnn	nnnn=0-9984; (0 or multiple of 8)	Word
Special Relay	WMnnnn	nnnn=0-9984; (0 or multiple of 8)	Word
Step Relay	WSnnn	nnn=0-9984; (0 or multiple of 8)	Word
Timer Present Value	RTnnnn	nnnn=0-9999	Word
Counter Present Value	RCnnnn	nnnn=0-9999	Word
Data Register	Rnnnnn	nnnn=0-65534	Word
32-bit Counter Present Value	DRCnnn	nnn=2000-255	Word
Data Register	Dnnnnn	nnnn=0-65534	Word
The operator terminal can read up to 32 words in one read command.			

Relay Type	Format	Relay Range
Input Relay	Xnnnn	nnn=0-9999
Output Relay	Ynnnn	nnn=0-9999
Internal Relay	Mnnnn	nnnn=0-9999
Special Relay	Mnnnn	nnnn=0-9999
Step Relay	Snnnn	nnn=0-9999
Timer Flag	Tnnnn	nnn=0-9999
Counter Flag	Cnnnn	nnn=0-9999

Cable Drawings



Operator terminal to RS232C on FB-MC controller

Communication Format

Before connection, please set up the communication parameters and the DIP switch as follows:

Format	Controller Setting	Operator Terminal Setting
Communication Format	RS232C	COM1 or COM2 = RS232C
Station No	1	Set controller station 01 in H-Designer.
Transmission Speed	9600/19200 bps	Set the SW5=OFF if parameters are set in H-Designer. Set the SW5=ON if parameters are set in the operator terminal.
Transmission Format	Size: 7-bit	
	Parity: EVEN	
	Stop bit: 1-bit	

Note:

Facon FB Series (RS232/RS485) for RS232 w/o RTS control (3-PIN cable) or RS485 Facon FB Series (RS232-RTS) for RS232 with RTS control (5-PIN cable, RTS, CTS with connect), not for RS485.

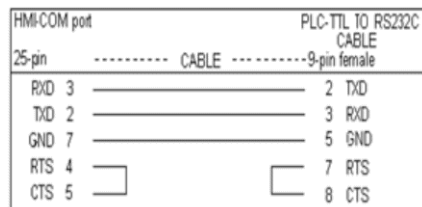
9.10 Festo FPC

Data format and range of the controller registers which H-Designer can access: Word Device and Bit Device.

Register Type	Format	Register Range	Data Size
Input	IWnnn	nnn=0-255	Word (16 bits)
Output	QWnnn	nnn=0-255	Word (16 bits)
Flag	FWnnnn	nnnn=0-9999	Word (16 bits)
Timer	TWnnn	nnn=0-255	Word (16 bits)
Counter	CWnnn	nnn=0-255	Word (16 bits)
Register	Rnnn	nnn=0-255	Word (16 bits)
Timer_Preset	TPnnn	nnn=255	Word (16 bits)
Counter_Preset	CPnnn	nnn=0-255	Word (16 bits)

Relay Type	Format	Relay Range	Block
Input	Innn.bb	nnn=0-255; bb=0-15	b=0, e.g., I20.0
Output	Qnnn.bb	nnn=0-255; bb=0-15	b=0, e.g., Q20.0
Flag	Fnnnn.bb	nnnn=0-9999; bb=0-15	b=0, e.g., F20.0
Timer	Tnnn	nnn=0-255	None
Counter	Cnnn	nnn=0-255	None
Timer_on	TONnnn	nnn=0-255	None
Timer_off	TOFFnnn	nnn=255	None

Cable Drawing



Operator terminal to controller (TTL to RS232C cable)

The operator terminal needs to use a FESTO TTL to RS232C cable connection, a 6-pin telephone connector with a controller port and a 9-pin male with 9-pin female.

Communication Format

Before connection, please set up the communication parameters and the DIP switch as follows:

Format	Controller Setting	Operator Terminal Setting
Communication Format	RS232C	COM1 or COM2 = RS232C
Station No	None	
Transmission Speed	9600 bps	Set the SW5=OFF if parameters are set in H-Designer. Set the SW5=ON if parameters are set in the operator terminal.
Transmission Format	Size: 8-bit	
	Parity: NONE	
	Stop bit: 1-bit	

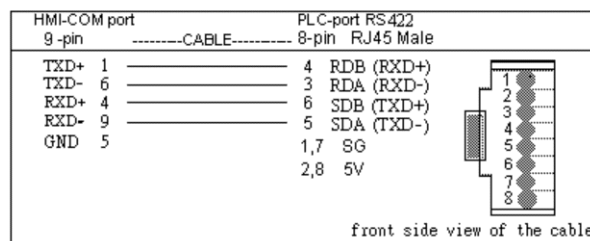
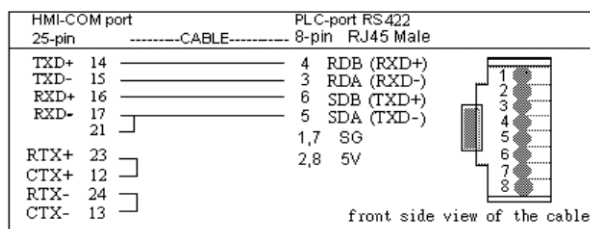
9.11 Fuji NB

Data format and range of the controller registers which H-Designer can access: Word Device and Bit Device.

Register Type	Format	Register Range	Data Size
Data Register	Dnnn	nnn=hex number 0-3FF	Word
Special Register	Dnnnn	nnnn=hex number 8000-80FF	Word
Timer Current Value	TNnnn	nnn=hex number 0-1FF	Word
Counter Current Value	CNnn	nn=hex number 0-FF	Word
Input Relay	WXnn	nn=hex number 0-1F	Word
Output Relay	WYnn	nn=hex number 0-1F	Word
Internal Relay	WMnn	nn=hex number 0-3F	Word
Latch Relay	WLnn	nn=hex number 0-3F	Word
Special Relay	WMnnn	nnn=hex number 800-81F	Word
Step Relay	WSnnn	nnn=hex number 0-3F	Word

Relay Type	Format	Relay Range	Block
Timer output	Tnnn	nnn=hex number 0-1FF	End with 0
Counter output	Cnn	nn=hex number 0-FF	End with 0
Input Relay	Xnnn	nnn=hex number 0-1FF	End with 0
Output Relay	Ynnn	nnn=hex number 0-1FF	End with 0
Internal Relay	Mnnn	nnn=hex number 0-3FF	End with 0
Latch Relay	Lnnn	nnn=hex number 0-3FF	End with 0
Special Relay	Mnnnn	nnnn=hex number 8000-81FF	End with 0
Step Relay	Snnn	nnnn=hex number 0-3FF	End with 0

Cable Drawings



Operator terminal to RS422 on NB/NS/NJ-CPU port

Communication Format

Before connection, please set up the communication parameters and the DIP switch as follows:

Format	Controller Setting	Operator Terminal Setting
Communication Format	RS422	COM = DIP switch RS422
Station No	None	
Transmission Speed	19200 bps	Set the SW5=OFF if parameters are set in H-Designer. Set the SW5=ON if parameters are set in the operator terminal.
Transmission Format	Size: 8-bit	
	Parity: ODD	
	Stop bit: 1-bit	
Controller Password	None or 0000-9999	The password needs to be set in the controller.

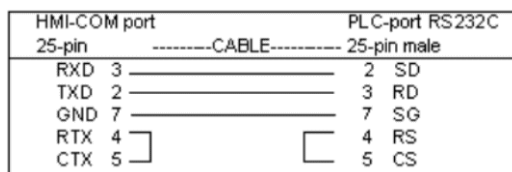
9.12 GE Series 90 CCM

Data format and range of the controller registers which H-Designer can access: Word Device and Bit Device.

Register Type	Format	Register Range	Data Size
Discrete Input	%Innnnn	nnnnn=1-12288	Word (16 bits)
Discrete Output	%Qnnnnn	nnnnn=1-12288	Word (16 bits)
Register	%Rnnnnn	nnnnn=1-16384	Word (16 bits)
Discrete Input (%Innnnn), Discrete Output (%Qnnnnn): Address must be 1 or multiple of 16 + 1.			

Relay Type	Format	Relay Range	Block
Discrete Input	%Innnnn	nnnnn=1-12288	1 or multiple of 16 + 1
Discrete Output	%Qnnnnn	nnnnn=1-12288	1 or multiple of 16 + 1

Cable Drawing



Operator terminal to RS232C port on controller

Communication Format

Before connection, please set up the communication parameters and the DIP switch as follows:

Format	Controller Setting	Operator Terminal Setting
Communication Format	RS232C	COM1 or COM2 = RS232C
Station No	0	
Transmission Speed	19200 bps	Set the SW5=OFF if parameters are set in H-Designer. Set the SW5=ON if parameters are set in the operator terminal.
Transmission Format	Size: 8-bit	
	Parity: NONE	
	Stop bit: 1-bit	

9.13 GE-Fanuc 90-SNP

Data format and range of the controller registers which H-Designer can access: Word Device and Bit Device.

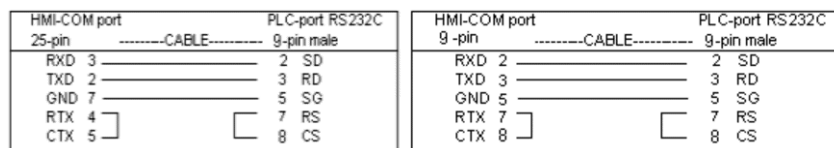
Register Type	Format	Register Range	Data Size
Discrete Inputs	%Innnnn	nnnn=1-12288;1 or multiple of 16 + 1	Word
Discrete Outputs	%Qnnnnn	nnnn=1-12288;1 or multiple of 16 + 1	Word
Discrete Temporaries	%Tnnn	nnn=1-256; 1 or multiple of 16 + 1	Word
Discrete Internals	%Mnnnnn	nnnn=1-12288;1 or multiple of 16 + 1	Word
%SA Discrete	%SAnnn	nnn=1-128; 1 or multiple of 16 + 1	Word
%SB Discrete	%SBnnn	nnn=1-128; 1 or multiple of 16 + 1	Word
%SC Discrete	%SCnnn	nnn=1-128; 1 or multiple of 16 + 1	Word
%S Discrete	%Snnn	nnn=1-128; 1 or multiple of 16 + 1	Word
Genius Global Data	%Gnnnn	nnnn=1-7680; 1 or multiple of 16 + 1	Word
Registers	%Rnnnnn	nnnn=2-16384	Word
Analog Inputs	%Alnnnn	nnnn=1-8192	Word
Analog Outputs	%AQnnnn	nnnn=1-8192	Word

The operator terminal can read up to 50 words in one read command.

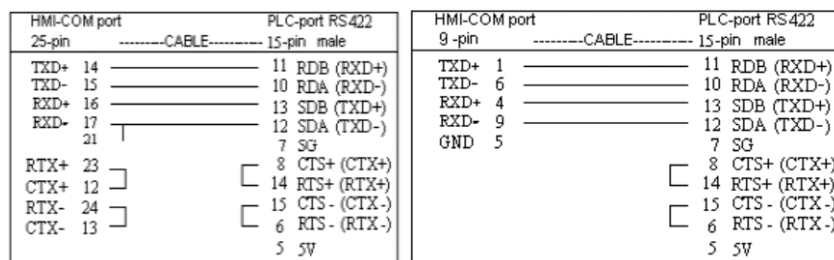
Relay Type	Format	Relay Range	Block
Discrete Inputs	%Innnnn	nnnn=1-12288	1 or multiple of 16 + 1
Discrete Outputs	%Qnnnnn	nnnn=1-12288	1 or multiple of 16 + 1
Discrete Temporaries	%Tnnn	nnn=1-256	1 or multiple of 16 + 1
Discrete Internals	%Mnnnnn	nnnn=1-12288	1 or multiple of 16 + 1
%SA Discrete	%SAnnn	nnn=1-128	1 or multiple of 16 + 1
%SB Discrete	%SBnnn	nnn=1-128	1 or multiple of 16 + 1
%SC Discrete	%SCnnn	nnn=1-128	1 or multiple of 16 + 1
%S Discrete	%Snnn	nnn=1-128	1 or multiple of 16 + 1
Counter Flag	%Gnnnn	nnnn=1-7680	1 or multiple of 16 + 1

The operator terminal can read up to 800 bits in one read command.

Cable Drawings



Operator terminal to RS232C on controller mini-converter kit



Operator terminal to RS422 port on controller 9030 CPU

Communication Format

Before connection, please set up the communication parameters and the DIP switch as follows:

Format	Controller Setting	Operator Terminal Setting
Communication Format	RS422 CPU port or RS232C	COM2 = RS232C/RS422/RS485 RS422: Set SW10=OFF
Station No	None	
Transmission Speed	9600/19200 bps	Set the SW5=OFF if parameters are set in H-Designer. Set the SW5=ON if parameters are set in the operator terminal.
Transmission Format	Size: 8-bit	
	Parity: NONE	
	Stop bit: 1-bit	
Controller ID	Blanks	
Controller Password	None or *****	The password needs to be set in the controller.

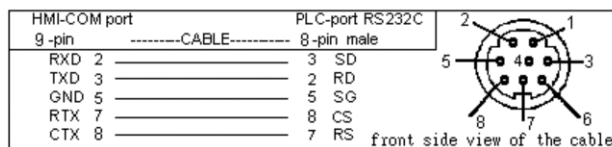
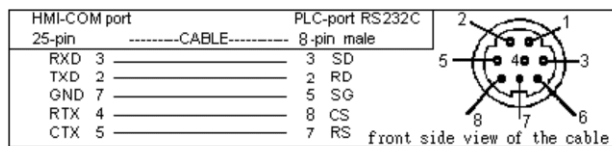
9.14 Hitachi EC

Data format and range of the controller registers which H-Designer can access: Word Device and Bit Device.

Register Type	Format	Register Range	Data Size
Internal Register	WMnnn	nnn=400, 402, 404, ..., 654	Word
Internal Register	WMnnn	nnn=700, 702, 704, ..., 954	Word
Internal Register	WMnnn	nnn=960, 962, 964, ..., 990	Word
Timer/Counter Register	TCnnn	nnn=100-195	Word
Timer/Counter Register	TCnnn	nnn=200-295	Word
The operator terminal can read up to 60 words in one read command.			

Relay Type	Format	Relay Range	Block
Input Relay	Xnnn	nnn=0-15, 20-35, 40-55, 60-75, ..., 180-195	X0, X20, X40...
Output Relay	Ynnn	nnn=200-215, 220-235, 240-255, ..., 380-395	Y200, Y220...
Auxiliary Relay	Mnnn	nnn=400-655, 700-955, 960-991	M400, M420...
Timer/Counter Relay	TCnn	nn=0-95	TC0, TC1, TC80

Cable Drawings



Operator terminal to controller programming port

Communication Format

Before connection, please set up the communication parameters and the DIP switch as follows:

Format	Controller Setting	Operator Terminal Setting
Communication Format	RS232C	COM1 or COM2 = RS232C
Station No	None	
Transmission Speed	9600 bps	Set the SW5=OFF if parameters are set in H-Designer. Set the SW5=ON if parameters are set in the operator terminal.
Transmission Format	Size: 7-bit	
	Parity: EVEN	
	Stop bit: 1-bit	
CTS Handshaking	Enabled	

Note:

When using exclusive protocol, you must set the controller peripheral mode selector to COM2.

9.15 Hitachi H/EH1

Data format and range of the controller registers which H-Designer can access: Word Device and Bit Device.

Register Type	Format	Register Range	Data Size
Ext. Input	WXnnnn	nnnn=0-4FF9	Word (16 bits)
Ext. Output	WYnnnn	nnnn=0-4FF9	Word (16 bits)
Int. Output	WRnnnn	nnnn=0-C3FF	Word (16 bits)
Int. Output	WRnnnn	nnnn=F0000-F1FF	Word (16 bits)
Int. Output	WMnnn	nnn=0-3FF	Word (16 bits)
CPU Link	WLnnn	nnn=0-3FF	Word (16 bits)
CPU Link	WLnnnn	nnnn=1000-13FF	Word (16 bits)
T/C CV	TCnnn	nnn=0-511	Word (16 bits)

Hitachi H/EH1 Series must use procedure 1 protocol to communicate.
 Hitachi EH2 Series must use procedure 2 protocol to communicate.
 Ext. Input (WXnnnn) and Ext. Output (WYnnnn) registers cannot access the "Read Block" function.

Relay Type	Format	Relay Range
Ext. Input	Xmnnnn	m=0-4, nnnn=0-FF95
Ext. Output	Ymnnnn	m=0-4, nnnn=0-FF95
Int. Output	Rnnn	nnn=0-7FF
Int. Output	Mnnnn	nnnn=0-3FFF
CPU Link	Lnnnn	nnnn=0-3FFF
CPU Link	Lnnnnn	nnnnn=10000-13FFF
On-delay timer bit	TDnnn	nnn=0-255
Single-shot timer bit	SSnnn	nnn=0-255
Up counter	CUnnn	nnn=0-511
U/D counter up coil	CTUnnn	nnn=0-511
U/D counter down coil	CTDnnn	nnn=0-511
U/D counter contact	CTnnn	nnn=0-511
T/C CV clear	CLnnn	nnn=0-511
Rising edge	DIFnnn	nnn=0-511
Falling edge	DFNnnn	nnn=0-511

Ext. Input (WXnnnn) and Ext. Output (WYnnnn) cannot access the "Read Block" function.

Cable Drawings

HMI-COM port 25-pin	PLC-port RS232C 15-pin male	HMI-COM port 25-pin	PLC-port RS232C 15-pin male
RXD 3	2 SD	RXD 3	2 SD
TXD 2	3 RD	TXD 2	3 RD
RTX 4	5 CTS	RTX 4	5 CTS
CTX 5	4 RTS	CTX 5	4 RTS
GND 7	9,10 SG	GND 7	9,10 SG
Baud Rate is set by DIP-SW of CPU	7 DSR	Baud Rate is 19200 bps	7 DSR
	14 +12V		14 +12V
			8

Operator terminal to RS232C port on controller

Communication Format

Before connection, please set up the communication parameters and the DIP switch as follows:

Format	Controller Setting	Operator Terminal Setting
Communication Format	RS232C	COM1 or COM2 = RS232C
Station No	None	
Transmission Speed	19200 bps	Set the SW5=OFF if parameters are set in H-Designer. Set the SW5=ON if parameters are set in the operator terminal.
Transmission Format	Size: 7-bit	
	Parity: EVEN	
	Stop bit: 1-bit	

9.16 Hust CNC Controller

Data format and range of the controller registers which H-Designer can access: Word Device and Bit Device.

Register Type	Format	Register Range	Data Size
16-bit variable	Wnnnnn	nnnnn=0-65534	Word (16 bits)
32-bit variable	Dnnnnn	nnnnn=0-65534	Double-word (32 bits)
Up to 28 variables can be read at one time.			

16-bit variable (Wnnnnn):

Only used in word objects.

Only used in Bit 0-Bit 15 of each variable.

If written, Bit 16-Bit 31 will clear as 0.

32-bit variable (Dnnnnn):

Only used in double word objects.

Only used in Bit 0-Bit 31 of each variable.

Relay Type	Format	Relay Range	Block
1-bit variable	Bnnnnn.bb	nnnnn=0-65534, bb=0-31	bb must be 0
I - bit data	Innn	nnn=0-255	nnn must be 0 or multiple of 32.
O - bit data	Onnn	nnn=0-255	nnn must be 0 or multiple of 32.
C - bit data	Cnnn	nnn=0-255	nnn must be 0 or multiple of 32.
S - bit data	Snnn	nnn=0-255	nnn must be 0 or multiple of 32.
A - bit data	Annnn	nnn=0-1023	nnn must be 0 or multiple of 32.
I - Bit Data, O - Bit Data, C - Bit Data, S - Bit Data, A - Bit Data are read-only. If written, the action will be invalid. (No error messages)			

Cable Drawing

HMI-COM port 25-pin	CABLE	PLC-port RS232C 9-pin male
RXD 3	=====	2 TXD
TXD 2	=====	3 RXD
GND 7	=====	5 GND
RTS 4	=====	7 CTS
CTS 5	=====	8 RTS

Operator terminal to RS232C port (9-pin male) on controller

Communication Format

Before connection, please set up the communication parameters and the DIP switch as follows:

Format	Controller Setting	Operator Terminal Setting
Communication Format	RS232C	COM1 or COM2 = RS232C
Station No	None	
Transmission Speed	9600 (19200/38400) bps	Set the SW5=OFF if parameters are set in H-Designer. Set the SW5=ON if parameters are set in the operator terminal.
Transmission Format	Size: 7-bit	
	Parity: EVEN	
	Stop bit: 2-bit	

9.17 Idec Micro-3

Data format and range of the controller registers which H-Designer can access: Word Device and Bit Device.

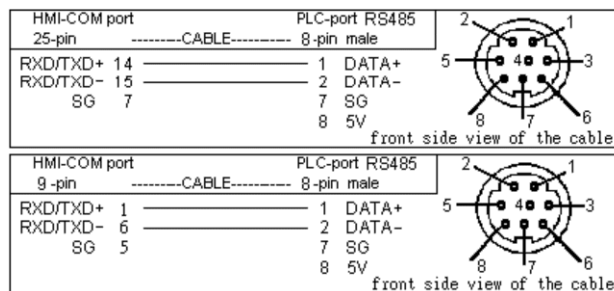
Register Type	Format	Register Range	Data Size
Input Relay	Xn	n=0-3	Byte
Output Relay	Yn	n=0-3	Byte
Internal Relay	Mnn	nn=0-31	Byte
Shift Register	Rnn	nn=0-48; must be 0 or multiple of 8	Bit
Timer Preset	TPnn	nn=0-31 (read only)	Word
Timer Current	Tnn	nn=0-31	Word
Counter Preset	CPnn	nn=0-31 (read only)	Word
Counter Current	Cnn	nn=0-31	Word
Data Register	Dnn	nn=0-99	Word
Calender/CLOCK	Wn	n=0-6	Word

The operator terminal can read up to 50 words in one read command.
Timer Preset and Counter Preset are read-only. If the value of TP or CP exists in Data Register, the value of Data Register will be read.

Relay Type	Format	Relay Range	Block
Input Relay	Xnb	n=0-3; b=0-7	b=0, e.g., X10
Output Relay	Ynb	n=0-3; b=0-7	b=0, e.g., Y00
Internal Relay	Mnnb	nn=0-31; b=0-7	b=0, e.g., M10
Shift Register	Rnn	nn=0-63	Must be 0 or multiple of 8
Timer Status	Tnn	nn=0-31 (read only)	Must be 0 or multiple of 8
Counter Status	Cnn	nn=0-31 (read only)	Must be 0 or multiple of 8

The operator terminal can read up to 800 bits in one read command.

Cable Drawings



Operator terminal to controller Program Loader RS485 port (MICRO3-CPU PORT)

Communication Format

Before connection, please set up the communication parameters and the DIP switch as follows:

Format	Controller Setting	Operator Terminal Setting
Communication Format	RS485	COM2 = RS232C/RS422/RS485 RS485: Set SW10=ON
Station No	0-31; 255	Set PLC Station to 255 in H-Designer
Transmission Speed	9600 bps	Set the SW5=OFF if parameters are set in H-Designer. Set the SW5=ON if parameters are set in the operator terminal.
Transmission Format	Size: 7-bit	
	Parity: EVEN	
	Stop bit: 1-bit	
CTS Handshaking	Disabled	

Note:

If only one controller is used, set the Station No. as 255. In a network, set the Station No. to 0-31.

9.18 Jetter Nano_B

Data format and range of the controller registers which H-Designer can access: Word Device and Bit Device.

Register Type	Format	Register Range	Data Size
User Register	Rnnnnn	nnnnn=0-32767	Double-word (32 bits)
User Register	WRnnnnn	nnnnn=0-32767	Word (16 bits)

Relay Type	Format	Relay Range	Block
Input Relay	Innbb	nn=1-16,bb=01-08	1 or multiple of 16 + 1
Output Relay	Onnbb	nn=1-16,bb=01-08	1 or multiple of 16 + 1
Flag Relay	Fnnnn	nnnn=0-2301	If nnn<256, must be 0 or multiple of 24. If nnn>255 and <2048, must be multiple of 24 +16. If nnn>2047, must be multiple of 24 + 8.

Cable Drawings

HMI-COM port 25-pin	----- CABLE -----	PLC-port RS232C 15-pin male
RXD 3	=====	2 TXD
TXD 2	=====	3 RXD
GND 7	=====	7 GND

Operator terminal to RS232C port on controller (15-pin male)

HMI-COM port 25-pin	----- CABLE -----	PLC-port RS232C 9-pin male
RXD 3	=====	2 TXD
TXD 2	=====	3 RXD
GND 7	=====	5 GND

Operator terminal to RS232C port on controller (9-pin male)

Communication Format

Before connection, please set up the communication parameters and the DIP switch as follows:

Format	Controller Setting	Operator Terminal Setting
Communication Format	RS232C	COM1 or COM2 = RS232C
Station No	None	
Transmission Speed	9600 bps	Set the SW5=OFF if parameters are set in H-Designer.
Transmission Format	Size: 8-bit	Set the SW5=ON if parameters are set in the operator terminal.
	Parity: EVEN	
	Stop bit: 1-bit	

9.19 Jetter Delta

Data format and range of the controller registers which H-Designer can access: Word Device and Bit Device.

Register Type	Format	Register Range	Data Size
User Register	Rnnnnn	nnnn= 0-20479 21000-24999 31000-34999 41000-44999 50200-59999 61440-64999	Double-word (32 bits)
User Register	WRnnnnn	nnnn= 0-20479 21000-24999 31000-34999 41000-44999 50200-59999 61440-64999	Word (16 bits)

Relay Type	Format	Relay Range	Block
Input Relay	Inbb	n=1-8,bb=01-64	bb=01
Output Relay	Onbb	n=1-8,bb=01-64	bb=01
Flag Relay	Fnnnn	nnnn=0-2047	Must be >255 and, and multiple of 24 +16.

Cable Drawings

HMI-COM port 25-pin	----- CABLE -----	PLC-port RS232C 15-pin male
RXD 3	=====	2 TXD
TXD 2	=====	3 RXD
GND 7	=====	7 GND

Operator terminal to RS232C port on controller (15-pin male)

HMI-COM port 25-pin	----- CABLE -----	PLC-port RS232C 9-pin male
RXD 3	=====	2 TXD
TXD 2	=====	3 RXD
GND 7	=====	5 GND

Operator terminal to RS232C port on controller (9-pin male)

Communication Format

Before connection, please set up the communication parameters and the DIP switch as follows:

Format	Controller Setting	Operator Terminal Setting
Communication Format	RS232C	COM1 or COM2 = RS232C
Station No	None	
Transmission Speed	9600 bps	Set the SW5=OFF if parameters are set in H-Designer.
Transmission Format	Size: 8-bit	Set the SW5=ON if parameters are set in the operator terminal.
	Parity: EVEN	
	Stop bit: 1-bit	

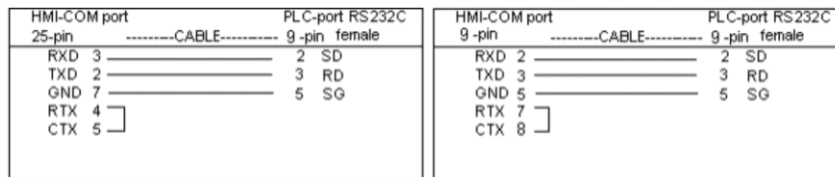
9.20 Klockner Moeller PS

Data format and range of the controller registers which H-Designer can access: Word Device and Bit Device.

Register Type	Format	Register Range	Data Size
Word Marker	MWnnnnn	nnnn=0-32766	Word
The operator terminal can read up to 32 words in one read command.			

Relay Type	Format	Relay Range	Block
Bit Marker0	Mnnnnn.b	nnnn=0-32766; b=0-7	b=0, e.g., M10.0
The operator terminal can read up to 512 bits in one read command.			

Cable Drawings



Operator terminal to RS232C programming port on controller CPU ZB4-303-KB1

Communication Format

Before connection, please set up the communication parameters and the DIP switch as follows:

Format	Controller Setting	Operator Terminal Setting
Communication Format	RS232C	COM1 or COM2 = RS232C
Station No	0=PS4-201 1=PS316	Set PLC Station to 0/1 in H-Designer.
Transmission Speed	9600 bps	Set the SW5=OFF if parameters are set in H-Designer.
Transmission Format	Size: 8-bit	Set the SW5=ON if parameters are set in the operator terminal.
	Parity: NONE	
	Stop bit: 1-bit	

9.21 Koyo SA/TI 325/330

Data format and range of the controller registers which H-Designer can access: Word Device and Bit Device.

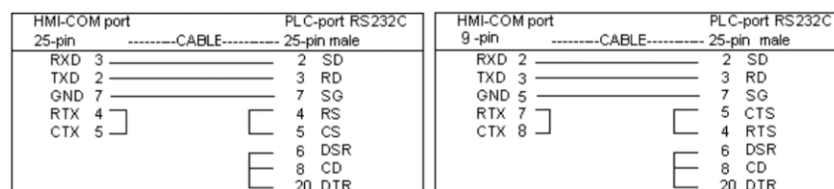
Register Type	Format	Register Range	Data Size
TMR/CTR	nnn	nnn=octal number 600-677	Word
Register Values	mmm	Accumulator mmm=octal number 400-576; mmm must be a multiple of 2	Even Bytes

Relay Type	Format	Relay Range	Block
Input/Output Bits	Bnnn	nnn=octal number 0-157	End with 0
Input/Output Bits	Bnnn	nnn=octal number 700-767	End with 0
Internal Relay Bits	Bnnn	nnn=octal number 160-377	End with 0
Shift Register Bits	Bnnn	nnn=octal number 400-577	End with 0
TMR/CTR Bits	Bnnn	nnn=octal number 600-677	End with 0

When operator terminal changes a relay's state, the operator terminal must read 1 byte (8 relays). After changing the corresponding bit, the operator terminal will write the byte to the controller. These actions will take more than one controller scan. PLC ladder cannot control other bits (relay) of the byte before the operator terminal has completed the **Change the Relay** action, otherwise these bits (relay) will return to their initial values. In other words, the controller's control action will be resumed.

For example, if the operator terminal wants to change B3's state, it will read B0-B7. After the corresponding bit B3 is changed, the operator terminal will write the entire byte to the controller. The PLC ladder command will be cancelled after the operator terminal writes the changes.

Cable Drawings



Operator terminal to RS232C port on controller SA21

Note:

KOYO SA21 series E02-DM and TI305-02DM have the same wiring.

Communication Format

Before connection, please set up the communication parameters and the DIP switch as follows:

Format	Controller Setting	Operator Terminal Setting
Communication Format	RS422 or RS232C	COM2 = RS232C/RS422/RS485
Station No	None	
Transmission Speed	9600/19200 bps	Set the SW5=OFF if parameters are set in H-Designer. Set the SW5=ON if parameters are set in the operator terminal.
Transmission Format	Size: 8-bit	
	Parity: ODD	
	Stop bit: 1-bit	
Operation Mode	RUN Mode	
Mode	ASCII Mode	SW2-8=ON

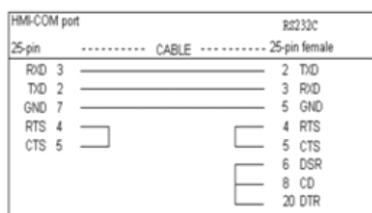
9.22 Koyo Direct DL/Koyo SU Series/TI435

Data format and range of the controller registers which H-Designer can access: Word Device and Bit Device.

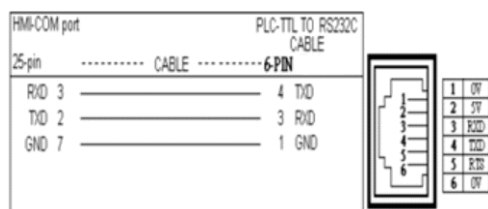
Register Type	Format	Register Range	Data Size
Timer Accumulated	Vnnn	nnn=octal number 0-177	Word
Counter Accumulated	Vnnnn	nnnn=octal number 1000-1177	Word
V Memory	Vnnnn	nnnn=octal number 1400-7777	Word
Link Relays	Vnnnnn	nnnnn=octal number 40000-40037	Word
Input Status	Vnnnnn	nnnnn=octal number 40400-40423	Word
Output Status	Vnnnnn	nnnnn=octal number 40500-40523	Word
Control Relays	Vnnnnn	nnnnn=octal number 40600-40635	Word
Stage	Vnnnnn	nnnnn=octal number 41000-41027	Word
Timer Status	Vnnnnn	nnnnn=octal number 41100-41107	Word
Counter Status	Vnnnnn	nnnnn=octal number 41140-41147	Word
Spec. Relay 1	Vnnnnn	nnnnn=octal number 41200-41205	Word
Spec. Relay 2	Vnnnnn	nnnnn=octal number 41216-41230	Word

Relay Type	Format	Relay Range	Block
Input Status	Xnnn	nnn=octal number 0-477	End with 0
Output Status	Ynnn	nnn=octal number 0-477	End with 0
Control Relays	Cnnn	nnn=octal number 0-737	End with 0
Stage	Snnn	nnn=octal number 0-577	End with 0
Timer Status	Tnnn	nnn=octal number 0-177	End with 0
Counter Status	CTnnn	nnn=octal number 0-177	End with 0
Spec. Relay 1	SPnnn	nnn=octal number 0-137	End with 0
Spec. Relay 2	SPnnn	nnn=octal number 320-617	End with 0
Linker Relays	GXnnn	nnn=octal number 0-777	End with 0

Cable Drawings



Operator terminal to RS232C port on controller



Operator terminal to RS232C port on controller (CPU240)

Communication Format

Before connection, please set up the communication parameters and the DIP switch as follows:

Format	Controller Setting	Operator Terminal Setting
Communication Format	RS232C	COM2 = RS232C/RS422/RS485
Station No	01	
Transmission Speed	9600 bps	Set the SW5=OFF if parameters are set in H-Designer. Set the SW5=ON if parameters are set in the operator terminal.
Transmission Format	Size: 8-bit	
	Parity: ODD	
	Stop bit: 1-bit	
Comm. Protocol Mode	HEX. (TISOFT AUX26)	

9.23 LG Glofa GM6

Data format and range of the controller registers which H-Designer can access: Word Device and Bit Device.

Register Type	Format	Register Range	Data Size
Input Image	%IWn.m.b	n=0-1, m=0-7, b=0-3	Word (16 bits)
Input Image	%IDn.m.b	n=0-1, m=0-7, b=0-1	Double-word (32 bits)
Output Image	%QWn.m.b	n=0-1, m=0-7, b=0-3	Word (16 bits)
Output Image	%QDn.m.b	n=0-1, m=0-7, b=0-1	Double-word (32 bits)
Internal memory	%MWnnnn	nnnn=0-4095	Word (16 bits)
Internal memory	%MDnnnn	nnnn=0-65534	Double-word (32 bits)
The operator terminal can read up to 60 words (30 double-words) in one read/write command and only supports CPU modules.			

Relay Type	Format	Relay Range	Block
Input Image	%IXn.m.bb	n=0-1, m=0-7, bb=0-63	bb must be 0 or multiple of 16
Output Image	%QXn.m.bb	n=0-1, m=0-7, bb=0-63	bb must be 0 or multiple of 16
Internal memory	%MXnnnnn	nnnnn=0-2047	n. must be 0 or multiple of 16

Cable Drawing

HMI-COM port 25-pin	----- CABLE -----	PLC-port RS232C 9-pin male
RXD 3	=====	7 TXD
TXD 2	=====	4 RXD
GND 7	=====	5 GND

Operator terminal to RS232C port on controller (9-pin male)

Communication Format

Before connection, please set up the communication parameters and the DIP switch as follows:

Format	Controller Setting	Operator Terminal Setting
Communication Format	RS232C (RS422/RS485)	COM1 or COM2 = RS232C (RS422/RS485)
Station No	0	
Transmission Speed	19200 (9600/38400) bps	Set the SW5=OFF if parameters are set in H-Designer. Set the SW5=ON if parameters are set in the operator terminal.
Transmission Format	Size: 8-bit	
	Parity: NONE	
	Stop bit: 1-bit	

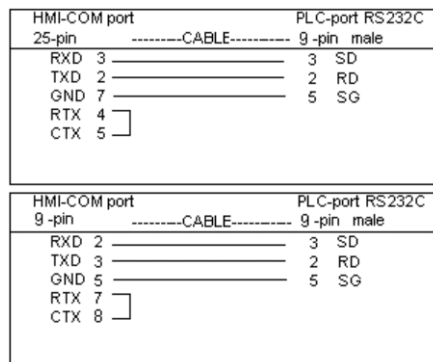
9.24 LG K10/60H/200H

Data format and range of the controller registers which H-Designer can access: Word Device and Bit Device.

Register Type	Format	Register Range	Data Size
Auxiliary Relay	Mnn	nn=0-63	Byte
Input/Output Relay	Pnn	nn=0-15	Byte
Keep Relay	Knn	nn=0-31	Byte
Link Relay	Lnn	nn=0-31	Bit
Special Relay	Fnn	nn=0-15	Word
Timer Current Value	Tnnn	nnn=0-255	Word
Counter Current Value	Cnnn	nnn=0-255	Word
Timer Set Value	TSnnn	nnn=0-255	Word
Counter Set Value	CSnnn	nnn=0-255	Word
Data Register	Dnnnn	nnnn=0-1023	Word
The operator terminal can read up to 60 words in one read command.			

Relay Type	Format	Relay Range	Block
Auxiliary Relay	Mnnb	nn=0-63; b=hex number 0-F	End with b=0
Input/Output Relay	Pnnb	nn=0-63; b=hex number 0-F	End with b=0
Keep Relay	Knnb	nn=0-63; b=hex number 0-F	End with b=0
Link Relay	Lnnb	nn=0-63; b=hex number 0-F	End with b=0
Special Relay	Fnnb	nn=0-63; b=hex number 0-F	End with b=0
Timer Relay	Tnnn	nnn=0-255	Must be 0 or multiple of 16
Counter Relay	Cnnn	nnn=0-255	Must be 0 or multiple of 16

Cable Drawings



Operator terminal to RS232C port on controller K200H CPU

Communication Format

Before connection, please set up the communication parameters and the DIP switch as follows:

Format	Controller Setting	Operator Terminal Setting
Communication Format	RS232C	COM1/COM2 = RS232C
Station No	00	
Transmission Speed	9600 bps	Set the SW5=OFF if parameters are set in H-Designer. Set the SW5=ON if parameters are set in the operator terminal.
Transmission Format	Size: 8-bit	
	Parity: NONE	
	Stop bit: 1-bit	

9.25 LG K200S

Data format and range of the controller registers which H-Designer can access: Word Device and Bit Device.

Register Type	Format	Register Range	Data Size
Input/Output Relay	PWnn	nn=0-63	Word (16 bits)
Auxiliary Relay	MWnnn	nnn=0-191	Word (16 bits)
Keep Relay	KWnn	nn=0-31	Word (16 bits)
Link Relay	LWnn	nn=0-63	Word (16 bits)
Special Relay	FWnn	nn=0-63	Word (16 bits)
Timer	TWnnn	nnn=0-255	Word (16 bits)
Counter	CWnnn	nnn=0-255	Word (16 bits)
Data Register	DWnnnn	nnnn=0-9999	Word (16 bits)

Relay Type	Format	Relay Range	Block
Input/Output Relay	Pnnb	nn=0-15; b=0-f	b must be 0
Auxiliary Relay	Mnnnb	nnn=0-191; b=0-f	b must be 0
Keep Relay	Knnb	nn=0-31; b=0-f	b must be 0
Link Relay	Lnnb	nn=0-63; b=0-f	b must be 0
Special Relay	Fnnb	nn=0-63; b=0-f	b must be 0
Timer	Tnnn	nnn=0-255	None
Counter	Cnnn	nnn=0-255	None

Cable Drawing

HMI-COM port 25-pin	----- CABLE -----	PLC-port RS232C 9-pin male
RXD 3	=====	3 TX
TXD 2	=====	2 RX
GND 7	=====	5 FG

Operator terminal to RS232C port on controller (9-pin male)

Communication Format

Before connection, please set up the communication parameters and the DIP switch as follows:

Format	Controller Setting	Operator Terminal Setting
Communication Format	RS232C	COM1 or COM2 = RS232C
Station No	None	
Transmission Speed	38400 bps	Set the SW5=OFF if parameters are set in H-Designer. Set the SW5=ON if parameters are set in the operator terminal.
Transmission Format	Size: 8-bit	
	Parity: NONE	
	Stop bit: 1-bit	

9.26 LG K300S

Data format and range of the controller registers which H-Designer can access: Word Device and Bit Device.

Register Type	Format	Register Range	Data Size
Input/Output Relay	PWnn	nn=0-31	Word (16 bits)
Auxiliary Relay	MWnnn	nnn=0-191	Word (16 bits)
Keep Relay	KWnn	nn=0-31	Word (16 bits)
Link Relay	LWnn	nn=0-63	Word (16 bits)
Special Relay	FWnn	nn=0-63	Word (16 bits)
Timer	TWnnn	nnn=0-255	Word (16 bits)
Counter	CWnnn	nnn=0-255	Word (16 bits)
Step Controller	SWnnnn	nnnn=0-9999	Word (16 bits)
Data Register	DWnnnn	nnnn=0-9999	Word (16 bits)

Relay Type	Format	Relay Range	Block
Input/Output Relay	PWnnb	nn=0-31; b=0-f	
Auxiliary Relay	MWnnnb	nnn=0-191; b=0-f	
Keep Relay	KWnnb	nn=0-31; b=0-f	
Link Relay	LWnnb	nn=0-63; b=0-f	
Special Relay	FWnnb	nn=0-63; b=0-f	

Cable Drawing

HMI-COM port 25-pin	----- CABLE -----	RS232C PORT 9-pin male
RXD 3	-----	7 TXD
TXD 2	-----	4 RXD
GND 7	-----	5 GND

Operator terminal to RS232C port on controller

Communication Format

Before connection, please set up the communication parameters and the DIP switch as follows:

Format	Controller Setting	Operator Terminal Setting
Communication Format	RS232C	
Station No	0	
Transmission Speed	9600 bps	Set the SW5=OFF if parameters are set in H-Designer. Set the SW5=ON if parameters are set in the operator terminal.
Transmission Format	Size: 8-bit	
	Parity: NONE	
	Stop bit: 1-bit	

9.27 LG Master-K10S/K30S/60S/100S

Data format and range of the controller registers which H-Designer can access: Word Device and Bit Device.

Register Type	Format	Register Range	Data Size
Input/Output Relay	PWn	nn=0-5	Word (16 bits)
Auxiliary Relay	MWnn	nnn=0-31	Word (16 bits)
Keep Relay	KWnn	nn=0-15	Word (16 bits)
Link Relay	LWnn	nn=0-15	Word (16 bits)
Special Relay	FWnn	nn=0-15	Word (16 bits)
Timer	TWnnn	nnn=0-127	Word (16 bits)
Counter	CWnnn	nnn=0-127	Word (16 bits)
Data Register	DWnnn	nnn=0-255	Word (16 bits)

Relay Type	Format	Relay Range	Block
Input/Output Relay	Pnb	n=0-5; b=0-f	b must be 0
Auxiliary Relay	Mnnb	nn=0-31; b=0-f	b must be 0
Keep Relay	Knnb	nn=0-15; b=0-f	b must be 0
Link Relay	Lnnb	nn=0-15; b=0-f	b must be 0
Special Relay	Fnnb	nn=0-15; b=0-f	b must be 0
Timer	Tnnn	nnn=0-127	n.. must be 0 or multiple of 16
Counter	Cnnn	nnn=0-127	n.. must be 0 or multiple of 16

Cable Drawing

HMI-COM port 25-pin	----- CABLE -----	PLC-port RS232C 9-pin male
RXD 3	=====	3 TX
TXD 2	=====	2 RX
GND 7	=====	5 FG

Operator terminal to RS232C port on controller (9-pin male)

Communication Format

Before connection, please set up the communication parameters and the DIP switch as follows:

Format	Controller Setting	Operator Terminal Setting
Communication Format	RS232C	COM1 or COM2 = RS232C
Station No	None	
Transmission Speed	9600 bps	Set the SW5=OFF if parameters are set in H-Designer.
Transmission Format	Size: 8-bit	Set the SW5=ON if parameters are set in the operator terminal.
	Parity: NONE	
	Stop bit: 1-bit	

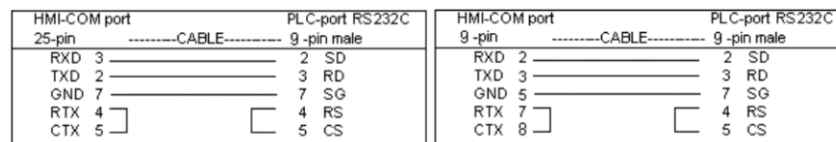
9.28 Matsushita FP

Data format and range of the controller registers which H-Designer can access: Word Device and Bit Device.

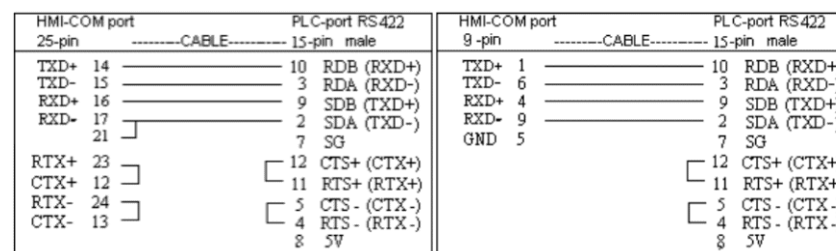
Register Type	Format	Register Range	Device Type/ Aux. Address		Block Read R/W
Internal Relay	WRnn	nn=0-97 (875)	0	0	Word
Special Internal Relay	WRnnn	nnn=900-910	1	0	Word
Link Relay	WLnnn	nnn=0-127 (639)	2	0	Word
External Input Relay	WXnnn	nnn=0-127 (255)	3	0	Word
External Output Relay	WYnnn	nnn=0-127 (255)	4	0	Word
Timer/Counter PV	EVnnn	nnn=0-254 (2047)	5	0	Word
Timer/Counter SV	SVnnn	nnn=0-254 (2047)	6	0	Word
Data Register	DTnnnn	nnnn=0-2047 (32764)	7	0	Word
Special Data Register	DTnnnn	nnnn=9000-9255	8	0	Word
Link Data Register	LDnnn	nnn=0-127 (8447)	9	0	Word
File Register	FLnnnnn	nnnn=0-8191 (32764)	10	0	Word
The operator terminal can read up to 27 words in one read command. The register setting range of FP10SH is nnnnn = 0-32764.					

Relay Type	Format	Relay Range	Block
Internal Relay	Rnnnb	nn=0-97 (875); b= 0-f	b=0, e.g., R1230
Special Internal Relay	Rnnnb	nnn=900-910; b= 0-f	b=0, e.g., R9100
Link Relay	Lnnnb	nnn=0-127 (639); b=0-f	b=0, e.g., L110
External Input Relay	Xnnnb	nnn=0-127 (255); b=0-f	b=0, e.g., X00
External Output Relay	Ynnnb	nnn=0-127 (255); b=0-f	b=0, e.g., Y00
Timer Flag Content	Tnnn	nnn=0-254 (2047)	Must be 0 or multiple of 16
Counter Flag Content	Cnnn	nnn=0-254 (2047)	Must be 0 or multiple of 16
The operator terminal can read up to 432 bits in one read command. The relay setting range of FP10SH is nnnn = 0-2047.			

Cable Drawings



Operator terminal to RS232C LINK on FP3 CCU or LINK port on FP1



Operator terminal to RS422 programming port (FP3 CPU port)

Communication Format

Before connection, please set up the communication parameters and the DIP switch as follows:

Format	Controller Setting	Operator Terminal Setting
Communication Format	RS422 or RS232C	COM2 = RS232C/RS422/RS485
Station No	02-27 FP CPU PORT=238 FP1: Set computer link	Set PLC Station to 01 in H-Designer, CPU PORT 238
Transmission Speed	19200/9600 bps FP10SH -CPU: Set 115.2K bps	Set the SW5=OFF if parameters are set in H-Designer. Set the SW5=ON if parameters are set in the operator terminal.
Transmission Format	Size: 8-bit	
	Parity: ODD	
	Stop bit: 1-bit	

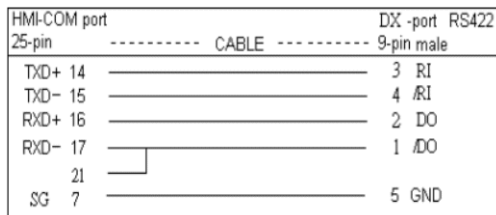
9.29 Mirle DX

Data format and range of the controller registers which H-Designer can access: Word Device and Bit Device.

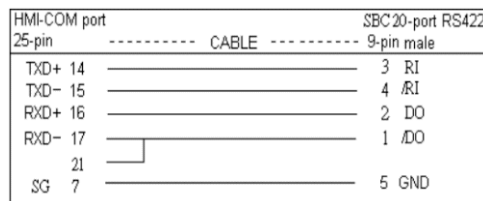
Register Type	Format	Register Range	Data Size
IR Area	IRnnn	nnn=0-111	Word
DM Area	DMnnnn	nnnn=0-2367	Word

Relay Type	Format	Relay Range	Block
IR Area	IRnnnbb	nnn=0-111, bb=00-15	bb=00

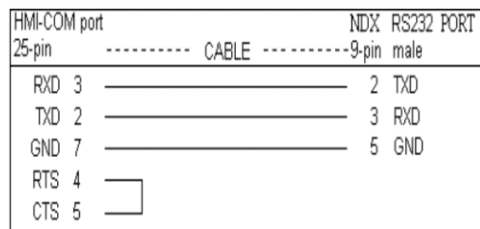
Cable Drawings



Operator terminal to RS422 port on MIRLE DX



Operator terminal to RS422 port on MIRLE SBC 20



Operator terminal to RS232C port on MIRLE nNDX

Communication Format

Before connection, please set up the communication parameters and the DIP switch as follows:

Format	Controller Setting			Operator Terminal Setting
Communication Format	RS232C/RS422			COM2=RS232C/RS422
Station No	0			Set PLC Station to 00 in H-Designer.
Transmission Speed	9600 bps			Set the SW5=OFF if parameters are set in H-Designer.
Transmission Format	Size	DX, SBC20: 8-bit	NDX: 8-bit	Set the SW5=ON if parameters are set in the operator terminal.
	Parity	DX, SBC20:ODD	NDX: NONE	
	Stop bit	DX, SBC20: 1-bit	NDX: 1-bit	

9.30 Mitsubishi FX

Data format and range of the controller registers which H-Designer can access: Word Device and Bit Device..

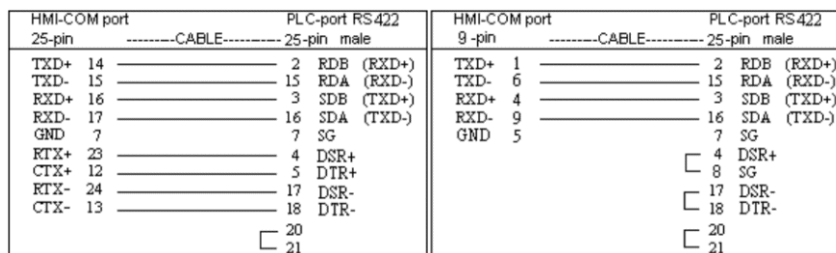
Register Type	Format	Register Range	Data Size
Auxiliary Relay	Mnnnn	nnnn=0-3071; must be 0 or multiple of 8	Byte
Special Auxiliary Relay	Mnnnn	nnnn=8000-8255; must be 0 or multiple of 8	Byte
Status Relay	Snnn	nnn=0-999; must be 0 or multiple of 8	Byte
Input Relay	Xnnn	nnn=octal number 0-377; end with 0	Byte
Output Relay	Ynnn	nnn=octal number 0-377; end with 0	Byte
Timer PV	Tnnn	nnn=0-254	Word
16-bit Counter PV	Cnnn	nnn=0-199	Word
32-bit Counter PV	Cnnn	nnn=200-255	Double-word
Data Register	Dnnn	nnn=0-1023(7999) D1000=FILE REGISTER	Word
Special Data Register	Dnnnn	nnnn=8000-8255	Word

The operator terminal can read up to 32 words in one read command.

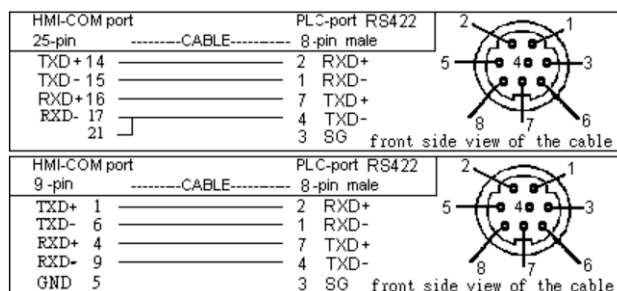
Relay Type	Format	Relay Range	Block
Auxiliary Relay	Mnnnn	0-3071	Must be 0 or multiple of 8
Special Auxiliary Relay	Mnnnn	8000-8255	Must be 0 or multiple of 8
Status Relay	Snnn	0-999	Must be 0 or multiple of 8
Input Relay	Xnnn	Octal number 0-377	End with 0
Output Relay	Ynnn	Octal number 0-377	End with 0
Timer Flag	Tnnn	0-255	Must be 0 or multiple of 8
Counter Flag	Cnnn	0-255	Must be 0 or multiple of 8

The operator terminal can read up to 512 bits in one read command.

Cable Drawings



Operator terminal to RS422 port of controller FX2 CPU



Operator terminal to Program Loader port of controller (FX2n/FX0n CPU port)

Communication Format

Before connection, please set up the communication parameters and the DIP switch as follows:

Format	Controller Setting	Operator Terminal Setting
Communication Format	RS422	
Transmission Speed	9600 bps	Set the SW5=OFF if parameters are set in H-Designer. Set the SW5=ON if parameters are set in the operator terminal.
Transmission Format	Size: 7-bit	
	Parity: EVEN	
	Stop bit: 1-bit	

9.31 Mitsubishi A

Data format and range of the controller registers which H-Designer can access: Word Device and Bit Device.

Register Type	Format	Register Range	Data Size
Input Relay	Xnnn	nnn=hex number 0-7FF; end with 0	Word
Output Relay	Ynnn	nnn=hex number 0-7FF; end with 0	Word
Link Relay	Bnnn	nnn=hex number 0-7FFF; end with 0	Word
Internal Relay	Mnnnn	nnnn=0-8191; must be 0 or multiple of 16	Word
Special Relay	Mnnnn	nnnn=9000-9255; -9000 must be multiple of 16	Word
Latch Relay	Lnnnn	nnnn=0-2047; must be 0 or multiple of 16	Word
Flag	Fnnnn	nnnn=0-2047; must be 0 or multiple of 16	Word
Timer PV	TNnnn	nnn=0-999	Word
Counter PV	CNnnn	nnn=0-999	Word
Data Register	Dnnnn	nnnn=0-8191	Word
Special Register	Dnnnn	nnnn=9000-9255	Word
File Register	Rnnnn	nnnn=0-8191	Word
Link Register	Wnnn	nnn=hex number 0-FFF	Word
Peripheral Input Relay	PXnnn	nnn=0-7FF must be 0 or multiple of 16	Word
The operator terminal can read up to 64 words in one read command.			

Relay Type	Format	Relay Range	Block
Input Relay	Xnnn	nnn=hex number 0-7FF	End with 0
Output Relay	Ynnn	nnn=hex number 0-7FF	End with 0
Link Relay	Bnnn	nnn=hex number 0-FFF	End with 0
Internal Relay	Mnnnn	0-8191	Must be 0 or multiple of 16.
Special Relay	Mnnnn	9000-9255	The last 3 digits must be multiple of 16.
Latch Relay	Lnnnn	0-2047	Must be 0 or multiple of 16.
Flag	Fnnnn	0-2047	Must be 0 or multiple of 16.
Timer Set	TSnnn	0-999	Must be 0 or multiple of 16.
Timer Coil	TCnnn	0-999	Must be 0 or multiple of 16.
Counter Set	CSnnn	0-999	Must be 0 or multiple of 16.
Counter Coil	CCnnn	0-999	Must be 0 or multiple of 16.
Peripheral Input Relay	Xnnnb	0-7ff	Must be 0 or multiple of 16.
The operator terminal can read up to 512 bits in one read command.			

Cable Drawings

HMI-COM port 25-pin	PLC-port RS232C 9-pin male	HMI-COM port 25-pin	PLC-port RS232C 25-pin male
RXD 3	3 SD	RXD 3	2 SD
TXD 2	2 RD	TXD 2	3 RD
GND 7	5 SG	GND 7	7 SG
RTX 4	8 CTS	RTX 4	5 CTS
CTX 5	7 RTS	CTX 5	4 RTS
	6 DSR		6 DSR
	4 DTR		8 CD
	1 DCD		20 DTR

HMI-COM port 9-pin	PLC-port RS232C 9-pin male	HMI-COM port 9-pin	PLC-port RS232C 25-pin male
RXD 2	3 SD	RXD 2	2 SD
TXD 3	2 RD	TXD 3	3 RD
GND 5	5 SG	GND 5	7 SG
RTX 7	8 CTS	RTX 7	5 CTS
CTX 8	7 RTS	CTX 8	4 RTS
	6 DSR		6 DSR
	4 DTR		8 CD
	1 DCD		20 DTR

Operator terminal to AJ71UC24-R2/S8

HMI-COM port 25-pin	PLC-port RS422 8-pin Screw terminals
TXD+ 14	RDA (RXD+)
TXD- 15	RDB (RXD-)
RXD+ 16	SDA (TXD+)
RXD- 17	SDB (TXD-)
21	SG
RTX+ 23	
CTX+ 12	
RTX- 24	
CTX- 13	

HMI-COM port 9-pin	PLC-port RS422 8-pin Screw terminals
TXD+ 1	RDA (RXD+)
TXD- 6	RDB (RXD-)
RXD+ 4	SDA (TXD+)
RXD- 9	SDB (TXD-)
GND 5	SG

Operator terminal to AJ71UC24 RS422

HMI-COM port 25-pin	PLC-port RS422 25-pin male
TXD+ 14	2 RDB (RXD+)
TXD- 15	15 RDA (RXD-)
RXD+ 16	3 SDB (TXD+)
RXD- 17	16 SDA (TXD-)
21	7 SG
RTX+ 23	4 CTX+
CTX+ 12	5 RTX+
RTX- 24	17 CTX-
CTX- 13	18 RTX-
	20
	21

Operator terminal to AnA and AnA/AnS/AnU CPU port RS422

Communication Format

Before connection, please set up the communication parameters and the DIP switch as follows:

Format	Controller Setting	Operator Terminal Setting
Communication Format	RS422 or RS232C	COM2 = RS232C/RS422/RS485
AJ71UC24 SW1=ON SW1=OFF	RS422	RS422: Set SW10=OFF
	RS232C	
Station No	00 (CPU port)	Set PLC Station to 00 in H-Designer. Set operator terminal station to 255.
	00 (AISJ71C24-S3)	
	00-31 (AJ71UC24)	
Transmission Speed	9600/19200 bps	Set the SW5=OFF if parameters are set in H-Designer. Set the SW5=ON if parameters are set in the operator terminal.
	CPU port=9600 bps	
Transmission Format	Size: 8-bit	
	Parity: ODD	
	Stop bit: 1-bit	
Comm. Protocol	Format 1;5;A	
Check Sum	YES	
Write during Run	Allowed	
For AISJ71C24 or AJ71c24, set the operator terminal station no. to 255 and PLC station no. to 0 and connect with controller CPU port. To communicate via COM2, please set the DIP switch = off and the communication parameters to 9600, 8, ODD, 1.		

9.32 Mitsubishi QnA

Data format and range of the controller registers which H-Designer can access: Word Device and Bit Device.

Register Type	Format	Register Range	Data Size
Link Relay	Bn	n=hex number 0-1FFF; end with 0	Word
Counter Coil	CCn	n=0-1023; must be 0 and multiple of 16	Word
Counter Current Value	CNn	n=0-1023	Word
Counter Contact	CSn	n=0-1023; must be 0 or multiple of 16	Word
Data Register	Dn	n=0-12287	Word
Direct Input	DXn	n=hex number 0-1FFF; end with 0	Word
Direct Output	DYn	n=hex number 0-1FFF; end with 0	Word
Flag	Fn	n=0-2047; must be 0 or multiple of 16	Word
Latch Relay	Ln	n=0-8191; must be 0 or multiple of 16	Word
Internal Relay	Mn	n=0-8191; must be 0 or multiple of 16	Word
File Register	Rn	n=0-32767	Word
Step Relay	Sn	n=0-8191; must be 0 or multiple of 16	Word
Special Link Relay	SBn	n=hex number 0-7FF; end with 0	Word
Retentive Timer Coil	SCn	n=0-2047; must be 0 or multiple of 16	Word
Special Register	SDn	n=0-2047	Word
Special Relay	SMn	n=0-2047; must be 0 or multiple of 16	Word
Retentive Timer Current Value	SNn	n=0-2047	Word
Retentive Timer Contact	SSn	n=0-2047; must be 0 or multiple of 16	Word
Special Link Register	SWn	n=hex number 0-7FF	Word
Timer Coil	TCn	n=0-2047; must be 0 or multiple of 16	Word
Timer Current Value	TNn	n=0-2047	Word
Timer Contact	TSn	n=0-2047; must be 0 or multiple of 16	Word
Edge Relay	Vn	n=0-2047; must be 0 or multiple of 16	Word
Link Register	Wn	n=hex number 0-1FFF	Word
Input Relay	Xn	n=hex number 0-1FFF; end with 0	Word
Output Relay	Yn	n=hex number 0-1FFF; end with 0	Word
Index Register	Zn	n=0-15	Word
File Register	ZRn	n=hex number 0-FE7F	Word

Relay Type	Format	Relay Range
Link Relay	Bn	hex number 0-1FFFF
Counter Coil	CCn	0-1023
Counter Contact	CSn	0-1023
Direct Input	DXn	n=hex number 0-1FFF
Direct Output	DYn	n=hex number 0-1FFF
Flag	Fn	0-2047
Latch Relay	Ln	0-8191
Internal Relay	Mn	0-8191
Step Relay	Sn	0-8191
Special Link Relay	SBn	n=hex number 0-7FF
Retentive Timer Coil	SCn	0-2047
Special Relay	SMn	0-2047
Retentive Timer Contact	SSn	0-2047
Timer Contact	TSn	0-2047

Relay Type	Format	Relay Range
Timer Coil	TCn	0-2047
Edge Relay	Vn	0-2047
Input Relay	Xn	n=hex number 0-1FFF
Output Relay	Yn	n=hex number 0-1FFF

Cable Drawings

HMI-COM port 25-pin	PLC-port RS232C 6-pin
RXD 3	TXD 2
TXD 2	RXD 1
GND 7	GND 3
RTX 4	CTS 5
CTX 5	RTS 6

Operator terminal to QnA CPU port (RS232)

HMI-COM port 25-pin	PLC-port RS232C 9-pin
RXD 3	TXD 3
TXD 2	RXD 2
GND 7	GND 5
RTX 4	CD 1
CTX 5	DTR 4

Operator terminal to Q Series C24 (RS232)

HMI-COM port 25-pin	PLC-port RS422 8-pin Screw terminals
TXD+ 14	RDA (RXD+)
TXD- 15	RDB (RXD-)
RXD+ 16	SDA (TXD+)
RXD- 17	SDB (TXD-)
21	SG
RTX+ 23	
CTX+ 12	
RTX- 24	
CTX- 13	

HMI-COM port 9-pin	PLC-port RS422 8-pin Screw terminals
TXD+ 1	RDA (RXD+)
TXD- 6	RDB (RXD-)
RXD+ 4	SDA (TXD+)
RXD- 9	SDB (TXD-)
GND 5	SG

Operator terminal to Q Series C24 (RS422)

Communication Format

Before connection, please set up the communication parameters and the DIP switch as follows:

Format	Controller Setting			Operator Terminal Setting
Communication Format	RS422 or RS232C			COM2 = RS232C / RS422/RS485
Station No	00 (CPU port/Q Series C24)			Set PLC Station to 00 in H-Designer.
Transmission Speed	19200 bps			Set the SW5=OFF if parameters are set in H-Designer.
Transmission Format	Size	CPU Port: 8-bit	Q Series C24: 7-bit	Set the SW5=ON if parameters are set in the operator terminal.
	Parity	CPU Port: ODD	Q Series C24: EVEN	
	Stop bit	CPU Port: 1-bit	Q Series C24: 2-bit	

9.33 Modbus Slave

Data format and range of the controller registers which H-Designer can access: Word Device and Bit Device.

Register Type	Format	Register Range	Data Size
Register	Wnnnnn	nnnnn=0-65535	Word (16 bits)

Relay Type	Format	Relay Range	Block
Relay	Bn	n=0-65535	

Cable Drawing

HMI-COM port 25-pin	----- CABLE -----	PLC-port RS232C 9-pin male
RXD 3	=====	3 TX
TXD 2	=====	2 RX
GND 7	=====	5 FG

Operator terminal to RS232C port (9-pin male) on controller

Communication Format

Before connection, please set up the communication parameters and the DIP switch as follows:

Format	Controller Setting	Operator Terminal Setting
Communication Format	RS232C (RS422/RS485)	COM1 or COM2 = RS232C (RS422/RS485)
Station No		
Transmission Speed	9600 bps (9600-115200)	Set the SW5=OFF if parameters are set in H-Designer. Set the SW5=ON if parameters are set in the operator terminal.
Transmission Format	Size: 8-bit	
	Parity: NONE	
	Stop bit: 1-bit	
	(7), (EVEN/NONE), (2)	

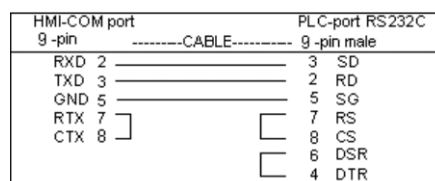
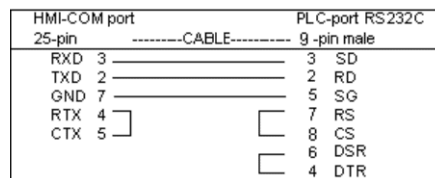
9.34 Modicon PC984 or Modbus (ASCII) or TSX Quantum

Data format and range of the controller registers which H-Designer can access: Word Device and Bit Device.

Register Type	Format	Register Range	Data Size
Input Registers (Modicon PC 984/Modbus (ASCII))	nnnnn	nnnnn= 30001-39999 (slave) 30001-31024 (master)	Word
Output Registers (Modicon PC 984/Modbus (ASCII))	nnnnn	nnnnn= 40001-49999 (slave) 40001-41024 (master)	Word
Input Registers (TSX Quantum)	nnnnnn	nnnnn= 300001-365535	Word
Output Registers (TSX Quantum)	nnnnnn	nnnnn= 400001-465535	Word
The operator terminal can read up to 125 words in one read command.			

Relay Type	Format	Relay Range
Discrete Outputs (Modicon PC 984/Modbus (ASCII))	Nnnnn	nnnnn= 1-4999 (slave) 1-1024 (master)
Discrete Inputs (Modicon PC 984/Modbus (ASCII))	Nnnnn	nnnnn= 10001-19999 (slave) 10001-11024 (master)
Discrete Outputs (TSX Quantum)	Nnnnn	nnnnn= 000001-065535
Discrete Inputs (TSX Quantum)	Nnnnn	nnnnn= 100001-165535
The operator terminal can read up to 2000 bits in one read command. Modbus (ASCII) master - V2 as Modbus (ASCII) master. In functional way, it will inform the connected controller automatically when the operator terminal data is changed.		

Cable Drawings



Operator terminal series to controller RS232C CPU port

Communication Format

Before connection, please set up the communication parameters and the DIP switch as follows:

Format	Controller Setting	Operator Terminal Setting
Communication Format	RS232C (RS422/RS485)	COM1 or COM2 = RS232C
Station No	1	Set PLC Station 01 in H-Designer.
	01-247 (mem setup)	
Transmission Speed	19200/9600 bps	Set the SW5=OFF if parameters are set in H-Designer. Set the SW5=ON if parameters are set in the operator terminal.
Transmission Format	Size: 8-bit	
	Parity: EVEN	
	Stop bit: 1-bit	
	(7,O,1);(7,E,1);(7,E,2)	
RTU Mode	(8,E,1);(8,O,1);(8,N,1)	

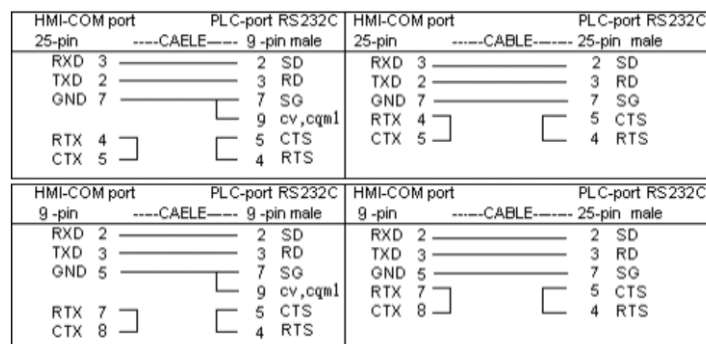
9.35 Omron C

Data format and range of the controller registers which H-Designer can access: Word Device and Bit Device.

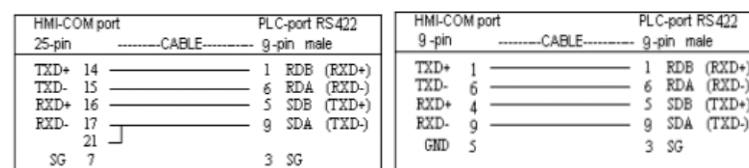
Register Type	Format	Register Range	Block Read	
IR Area	IRnnn	nnn=0-511	Word (16 bits)	Max. 28
HR Area	HRnn	nn=0-99	Word (16 bits)	Max. 28
AR Area	ARnn	nn=0-27	Word (16 bits)	Max. 28
LR Area	LRnn	nn=0-63	Word (16 bits)	Max. 28
TC Area	TCnnn	nnn=0-511	Word (16 bits)	Max. 28
DM Area	DMnnnn	nnnn=0-6655	Word (16 bits)	Max. 28

Relay Type	Format	Relay Range	Block
IR Area	IRnnnbb	nnn=0-511; bb=00-15	bb=00, e.g., IR12300
HR Area	HRnnbb	nn=0-99; bb=00-15	bb=00, e. g., HR2300
AR Area	ARnnbb	nn=0-27; bb=00-15	bb=00, e.g., AR100
LR Area	LRnnbb	nn=0-63; bb=00-15	bb=00, e.g., LR2300
TC Area	TCnnn	nnn=0-511	Multiple of 16, e.g. TC16

Cable Drawings



Operator terminal to RS232C HOST LINK of LK201/C200HS/C28H/C40H/CQM1



Operator terminal to RS422 HOST LINK on C200H-LK202

Communication Format

Before connection, please set up the communication parameters and the DIP switch as follows:

Format	Controller Setting	Operator Terminal Setting
Communication Format	RS232C/RS422/ RS485	COM2 = RS232C/RS422/RS485
Station No	00(00-31)	Set PLC Station to 00 in H-Designer
Transmission Speed	19200/9600 bps	Set the SW5=OFF if parameters are set in H-Designer. Set the SW5=ON if parameters are set in the operator terminal.
Transmission Format, initial value	Size: 7-bit	
	Parity: EVEN	
	Stop bit: 2-bit	
Operation Mode	Monitor Mode	
Protocol	Multiple-Link	

Note:

For CQM1 and CPU21, set the DM6648=0000.

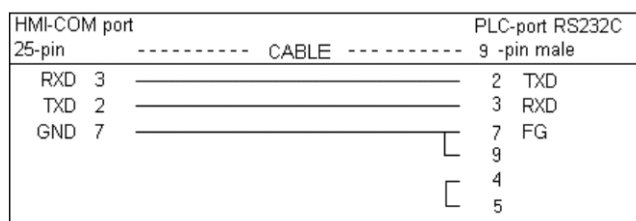
9.36 Omron CS1

Data format and range of the controller registers which H-Designer can access: Word Device and Bit Device.

Register Type	Format	Register Range	Data Size
IR Area (CIO Area)	IRnnnn	nnnn=0-6143	Word (16 bits)
HR Area	HRnnn	nn=0-511	Word (16 bits)
AR Area	ARnnn	nn=0-959	Word (16 bits)
LR Area	LRnnn	nn=0-199	Word (16 bits)
TC Area	TCnnnn	nnn=0-4095	Word (16 bits)
DM Area	DMnnnn	nnnn=0-9999	Word (16 bits)
EM Area	EMm.nnnnn	m=0-c, nnnn=0-9999	Word (16 bits)

Relay Type	Format	Relay Range	Block
IR Area (CIO Area)	IRnnnnbb	nnnn=0-6143; bb=00-15	bb must be 00
HR Area	HRnnnbb	nnn=0-511; bb=00-15	bb must be 00
LR Area	LRnnnbb	nnn=0-199; bb=00-15	bb must be 00
Timer Area	Tnnnn	nnnn=0-2047	
Counter Area	Cnnnn	nnnn=0-2047	

Cable Drawing



Operator terminal to RS232 port (9-pin male) on controller

Communication Format

Before connection, please set up the communication parameters and the DIP switch as follows:

Format	Controller Setting	Operator Terminal Setting
Communication Format	RS232C (RS422/RS485)	COM1 or COM2 = RS232C (RS422/RS485)
Station No	0 (0-31)	
Transmission Speed	9600 bps	Set the SW5=OFF if parameters are set in H-Designer. Set the SW5=ON if parameters are set in the operator terminal.
Transmission Format	Size: 7-bit	
	Parity: EVEN	
	Stop bit: 2-bit	

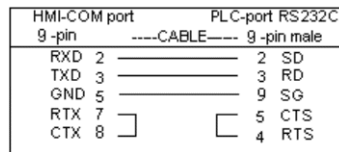
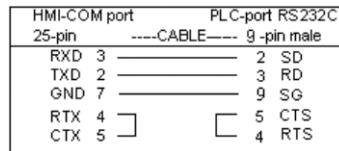
9.37 Omron CV

Data format and range of the controller registers which H-Designer can access: Word Device and Bit Device.

Register Type	Format	Register Range	Data Size
CIO Area	CIOnnnn	nnnn=0-2555	Word (16 bits)
TC Area	TCnnnn	nnnn=0-1023	Word (16 bits)
TC Area	TCnnnn	nnnn=2048-3071	Word (16 bits)
AR Area	ARnnn	nnn=0-511	Word (16 bits)
DM Area	DMnnnn	nnnn=0-9999	Word (16 bits)

Relay Type	Format	Relay Range	Block
CIO Area	CIOnnnnbb	nnnn=0-2555; bb=00-15	bb must be 00
TC Area	TCnnnn	nnnn=0-1023	
TC Area	TCnnnn	nnnn=2048-3071	

Cable Drawings



Operator terminal to RS232 port on controller

Communication Format

Before connection, please set up the communication parameters and the DIP switch as follows:

Format	Controller Setting	Operator Terminal Setting
Communication Format	RS232C	COM1 or COM2 = RS232C
Station No	0	
Transmission Speed	9600 bps	Set the SW5=OFF if parameters are set in H-Designer. Set the SW5=ON if parameters are set in the operator terminal.
Transmission Format	Size: 7-bit	
	Parity: EVEN	
	Stop bit: 2-bit	

9.38 Parker 6K

Data format and range of the controller registers which H-Designer can access: Word Device and Bit Device.

Register Type	Command	Format	Register Range	Data Size	Data Range
Input	!nTIN	In	n=0-8	Double-word	
Output	!nTOUT	On	n=0-8	Double-word	
Alarm Status	!INTHW	Asn	n=1	Double-word	
Axis Status	!nTAS	AXSn	n=1-8	Double-word	
System Status	!TSS	SYSn	n=1	Double-word	
User Status	!TUS	USSn	n=1	Word	
Binary	!VARBnnn	VARBnnn	nnn=001-125	Double-word	
Numeric	!VARnnn	VARnnn	nnn=001-225	Double-word	+ -999,999,999
Integer	!VARInnn	VARInnn	nnn=001-225	Double-word	+ -2,147,483,647
String	!VARSn	VARSn	nn=01-50	10 words	0-20 characters
Motor Position	!nTPC	MOPn	n=1-8	Double-word	+ -2,147,483,647
Motor Velocity	!nTVEL	MOVn	n=1-8	Double-word	
Encoder Position	!nTPE	ENPn	n=1-8	Double-word	
Timer	!nTTIM	Tn	n=1	Double-word	0-999999999
Nnn	Run Program	Nnn	nn=1-50	10 words	0-20 characters

If Parker 6K still executes, it will cause a communication time-out.
Parker 6K needs an internal program in order to accept commands with the initial code "!", otherwise communication errors may arise.

Register Type	Command	Format	Register Range	Data Range
Input	!nTIN	In.bb	n=0-8; bb=1-32	Bit
Output	!nTOUT	On.bb	n=0-8; bb=1-32	Bit
Alarm Status	!INTHW	Asn.bb	n=1; bb=1-32	Bit
Axis Status	!nTAS	AXSn.bb	n=1-8; bb=1-32	Bit
System Status	!TSS	SYSn.bb	n=1; bb=1-32	Bit
User Status	!TUS	USSn.bb	n=1; bb=1-16	Bit
Binary	!VARBnnn	VARBnnn.bb	nnn=001-125; bb=1-32	Bit
Error Status	TERn.bb	TERn.bb	n=1; bb=1-32	Bit
RUN	Run Program	RUNnn	nn=1-50	Bit

Cable Drawing

HMI-COM port 25-pin	CABLE	PLC-port RS232C 9-pin male
RXD 3	_____	3 TXD
TXD 2	_____	2 RXD
GND 7	_____	5 FG

Operator terminal to RS232 port on controller

Communication Format

Before connection, please set up the communication parameters and the DIP switch as follows:

Format	Controller Setting	Operator Terminal Setting
Communication Format	RS232C	COM1 or COM2 = RS232C
Station No	None	
Transmission Speed	9600 bps	Set the SW5=OFF if parameters are set in H-Designer. Set the SW5=ON if parameters are set in the operator terminal.
Transmission Format	Size: 8-bit	
	Parity: NONE	
	Stop bit: 1-bit	

9.39 Side Mida 20/20D

1. Data format and range of the controller registers which H-Designer can access:
Word Device and Bit Device.

Register Type	Format	Register Range	Data Size
Register	Wnnnnn	nnnn=0-10499	Word (16 bits)
Display	Disp n	n=0-4	10 words
TC Area	Daten	n=0-6	Word

Relay Type	Format	Relay Range	Block
I/O Relay	Bnnn	nnn=0-1599	Must be 0 or multiple of 16
Hardware Reset	HardRn	n=0	
Software Reset	SoftRn	n=0	
Clear RAM, EEPROM	Clr0-n	n=0	
Clear database	Clr1-n	n=0	
Clear RAM, EEPROM, database	Clr2-n	n=0	
Clear RAM, EEPROM, database and default setup	Clr3-n	n=0	

Cable Drawings

HMI-COM port 25-pin	CABLE	PLC RS232C 9-pin male
RXD 3	_____	3 TXD
TXD 2	_____	2 RXD
GND 7	_____	5 PG
RTS 4	_____	8 CTS
CTS 5	_____	7 RTS

Operator terminal to RS232 port on controller

HMI-COM port 25-pin	CABLE	PLC RS485
TXD+/RXD+ 14	_____	17 (+)
TXD-/RXD- 15	_____	18 (-)

Operator terminal to RS485 port on controller

Communication Format

Before connection, please set up the communication parameters and the DIP switch as follows:

Format		Controller Setting		Operator Terminal Setting	
		RS232C	RS485	RS232C	RS485
Communication Format		RS232C	RS485	Use SIDE MIDA 20/20D driver	Use MODBUS Slave driver
Station No		153	1	153	1
Transmission Speed		9600 bps	9600 bps	Set the SW5=OFF if parameters are set in H-Designer. Set the SW5=ON if parameters are set in the operator terminal.	
Transmission Format	Size	7-bits	8-bits		
	Parity	NONE	EVEN		
	Stop bit	1-bit	1-bit		
Mode		Stop mode	Running mode		

9.40 Siemens Simatic S5

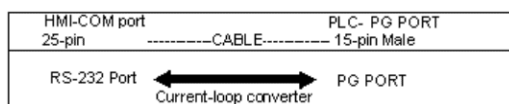
Data format and range of the controller registers which H-Designer can access: Word Device and Bit Device.

Register Type	Format	Register Range	Data Size	
Input Image	IBnnn	nnn=0-127	Byte	Max. 30
Output Image	QBnnn	nnn=0-127	Byte	Max. 30
Extended I/O	OBnnn	nnn=0-8191	Byte	Max. 30
Flag Bits	FBnnn	nnn=0-8191	Byte	Max. 30
Peripheral I/O	PBnnn	nnn=0-8191	Byte	Max. 30
System Data Area	RSnnn	nnn=0-255	Word	Max. 30
System Data Area	RInnn	nnn=0-255	Word	Max. 30
System Data Area	RJnnn	nnn=0-255	Word	Max. 30
System Data Area	RTnnn	nnn=0-255	Word	Max. 30
Timer Current	Tnnn	nnn=0-255	Word	Max. 30
Counter Current	Cnnn	nnn=0-255	Word	Max. 30
Data Block	DBmmm/nnn DBnnn	mmm=0-255; mmm is the block number nnn=0-65535; nnn defines the block. If mmm is not set, mmm defaults to 3 = DB3/nnn	Max. 30	
Data Block	DWmmm/nnn DWnnn	mmm=0-255; mmm is the block number nnn=0-32767; nnn defines the block. If mmm not set, mmm defaults to 3 = DW3/nnn	Max. 30	

The operator terminal can read up to 30 words in one read command.

Relay Type	Format	Relay Range	Block
Input Image	IBnnn.b	nnn=0-127; b=0-7	b=0, e.g. IB30.0
Output Image	QBnnn.b	nnn=0-127; b=0-7	b=0, e.g. QB2.0
Extended I/O	OBnnn.b	nnn=0-255; b=0-7	b=0, e.g. OB0.0
Flag Bits	FBnnn.b	nnn=0-255; b=0-7	b=0, e.g. FB23.0
Peripheral I/O	PBnnn.b	nnn=0-255; b=0-7	b=0, e.g. PB23.0

Cable Drawing



An RS-232/Current-loop converter cable must be used

Communication Format

Before connection, please set up the communication parameters and the DIP switch as follows:

Format	Controller Setting	Operator Terminal Setting
Communication Format	RS-232/Current-loop converter	COM1 or COM2=>RS232
Station No	None	
Transmission Speed	9600 bps	PLC MODE CODE 0: 90U 1: 95U 2:100U 3:102U 4:103U 5:115U 6:135U/921 7:135U/922 8:135U/928
Transmission Format	Size: 8-bit	Set the SW5=OFF if parameters are set in H-Designer. Set the SW5=ON if parameters are set in the operator terminal.
	Parity: EVEN	
	Stop bit: 1-bit	
PLC Mode Code	Controller's Data Block n=3-255 must OPEN	
Command Delay		Operator terminal Command Delay

9.41 Siemens Simatic S5 3964R

Data format and range of the controller registers which H-Designer can access: Word Device and Bit Device.

Register Type	Format	Register Range	Data Size
Register	nnn	nnn=0-255, data block: set 3	Word (16 bits)
Register	mmm/nnn	mmm=0-255, nnn=0-255	Word (16 bits)

Relay Type	Format	Relay Range	Block
Relay	nnn.b	nnn=0-255, b=0-f, data block: set 3	b must be 0
Relay	mmm/nnn.b	mmm=0-255, nnn=0-255, b=0-f	b must be 0

Cable Drawing

HMI-COM port 25-pin	----- CABLE -----	PLC-port RS232C 9-pin male
RXD 3	=====	3 TXD
TXD 2	=====	2 RXD
GND 7	=====	5 FG

Operator terminal to RS232 port on controller

Communication Format

Before connection, please set up the communication parameters and the DIP switch as follows:

Format	Controller Setting	Operator Terminal Setting
Communication Format	RS232C	COM1 or COM2 = RS232C
Station No	None	
Transmission Speed	9600 bps	Set the SW5=OFF if parameters are set in H-Designer. Set the SW5=ON if parameters are set in the operator terminal.
Transmission Format	Size: 8-bit	
	Parity: EVEN	
	Stop bit: 1-bit	

9.42 Siemens Simatic S7-200 PPI

Data format and range of the controller registers which H-Designer can access: Word Device and Bit Device.

Register Type	Format	Register Range	Data Size
Input Image	IWn	n=0-14	Word
Input Image	IDn	n=0-12	Double-word
Output Image	QWn	n=0-14	Word
Output Image	QDn	n=0-12	Double-word
Internal Bits	MWnn	nn=0-99	Word
Internal Bits	MDnn	nn=0-97	Double-word
Timer	Tnnn	nnn=0-255	Word
Counter	Cnnn	nnn=0-255	Word
Special S	SWnn	nn=0-99	Word
Special S	SDnn	nn=0-97	Double-word
Special Bits	SMWnnn	nnn=0-27	Word* read only
Special Bits	SMWnnn	nnn=28-199	Word
Special Bits	SMDnnn	nnn=0-197	Double-word
Analog Input Word	AIWnn	nn=0-30	Word* read only
Analog Output Word	AQWnn	nn=0-30	Word* read only
Data Area	VWnnnn	nnnn=0-9998	Word
Data Area	VDnnnn	nnnn=0-9996	Double-word
Data Area	DBWnnnn	nnnn=0-9998	Word
AQW; SW; SD cannot be used in CPU212/214.			

Relay Type	Format	Relay Range	Block
Input Image	In.b	n=0-15; b=0-7	b=0 e.g. I3.0
Output Image	Qn.b	n=0-15; b=0-7	b=0 e.g. Q2.0
Internal Bits	Mnn.b	n=0-100; b=0-7	b=0 e.g. M0.0
Timer Bit	Tnnn	nnn=0-255	b=0 e.g. T0 *read only
Counter Bit	Cnnn	nnn=0-255	b=0 e.g. C0 *read only
Special Bit	SMnnn.b	nnn=0-200; b=0-7	b=0 e.g. SM23.0
Data Area Bit	Vnnnn.b	nnnn=0-999; b=0-7	b=0 e.g. V2323.0
Special M	Snn.b	nn=0-100; b=0-7	b=0 e.g. S25.0

Cable Drawings

HMI-COM port 25-pin	-----CABLE-----	PLC-port RS485 9-pin male
RXD/TXD+ 14	-----	3 DATA+
RXD/TXD- 15	-----	8 DATA-
SG 7	-----	5 SG
		7 24V
HMI-COM port 9-pin	-----CABLE-----	PLC-port RS485 9-pin male
RXD/TXD+ 1	-----	3 DATA+
RXD/TXD- 6	-----	8 DATA-
GND 5	-----	5 SG
		7 24V

Operator terminal to programming port on controller (RS485 mode)

Communication Format

Before connection, please set up the communication parameters and the DIP switch as follows:

Format	Controller Setting	Operator Terminal Setting
Communication Format	RS485C	COM2 = RS232C/RS422/RS485
Station No	02 (02-27)	RS485: Set SW10=ON
Transmission Speed	9600/19200 bps	Set the SW5=OFF if parameters are set in H-Designer. Set the SW5=ON if parameters are set in the operator terminal.
Transmission Format	Size: 8-bit	
	Parity: EVEN	
	Stop bit: 1-bit	
Command Delay		Operator terminal Command Delay

Note:

The communication mode of the Simatic S7-200 network is a Token Ring structure and can exist in several majors. When on-line, the PC can download from/upload to the controller.

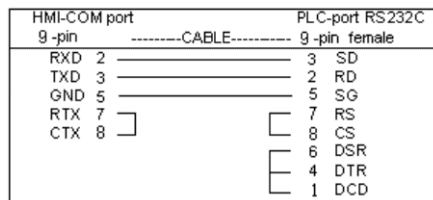
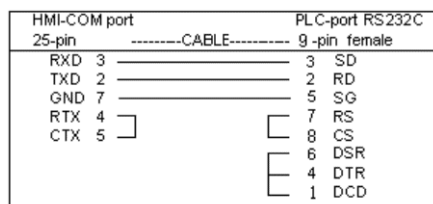
9.43 Siemens Simatic S7-300 CP340

Data format and range of the controller registers which H-Designer can access: Word Device and Bit Device.

Register Type	Format	Register Range	Data Size
DBmmm.DBWnnnn	mmm=1-255 nnnn=0-8190	DBmmm.DBWnnnn is the address of a word located at byte #nnnn and the byte following #nnnn of data block #3	Word
DBmmm.DBDnnnn	mmm=1-255 nnnn=0-8188	DBmmm.DBDnnnn is the address of a double-word located at byte #nnnn and the three bytes following #nnnn of data block #mmm	Double-word
The operator terminal can read up to 32 words in one read command.			

Relay Type	Format	Relay Range	Block
DBmmm.DBXnnnn.b	mmm=1-255 nnnn=0-8191 b=0-7	DBmmm.DBXnnnn.b is the address of bit #b of the word located at byte #nnnn of data block #mmm	b=0

Cable Drawings



Operator terminal to RS232C port on CP340

Communication Format

Before connection, please set up the communication parameters and the DIP switch as follows:

Format	Controller Setting	Operator Terminal Setting
Communication Format	RS232=CP340 RS422=CP340 RS485=CP340	COM2 = RS232C/RS422/RS485 RS422: Set SW10=OFF RS485: Set SW10=ON
Station No	None	Set the SW5=OFF if parameters are set in H-Designer. Set the SW5=ON if parameters are set in the operator terminal.
Transmission Speed	9600/19200 bps	
Transmission Format	Size: 8-bit Parity: EVEN Stop bit: 1-bit	
Function Block	FB40, FB2, FB3, DB2, DB3 for CP	

9.44 Siemens Simatic S7-300 via MPI port

Data format and range of the controller registers which H-Designer can access: Word Device and Bit Device.

Register Type	Format	Register Range	Device Type/ Aux. Address		Data Size R/W
Input Image	IWnnnnn	nnnnn=0-65534	0	0	Word
Input Image	IDnnnnn	nnnnn=0-65532	1	0	Double-word
Output Image	QWnnnnn	nnnnn=0-65534	2	0	Word
Output Image	QDnnnnn	nnnnn=0-65532	3	0	Double-word
Bits	MWnnn	nnnnn=0-254	4	0	Word
Bits	MDnnn	nnnnn=0-252	5	0	Double-word
Data Area (DB10)	VWnnnnn	nnnnn=0-65534	6	0	Word
Data Area (DB10)	VDnnnnn	nnnnn=0-65532	7	0	Double-word
Data Area (DB10)	DBWnnnnn	nnnnn=0-65534; the byte nnnnn & nnnnn+1 make DBWnnnnn	6	0	Word
Data Area (DB10)	DBDnnnnn	nnnnn=0-65532; a double- word address, DBWnnnn & DBWnnnn+1 make DBDnnnn	7	0	Double-word
Data Area	DBmmm. DBWnnnnn	mmm=1-255 nnnnn=0-65534	8	0	Word
Data Area	DBmmm. DBDnnnnn	mmm=1-255 nnnnn=0-65532	9	0	Double-word
Timer	Tnnnnn	nnnnn=0-65534	10	0	Word
Counter	Cnnnnn	nnnnn=0-65534	11	0	Word

Timer and Counter are read-only.

Relay Type	Format	Relay Range	Device Type/ Aux. Address		Block R/W
Input Image	Innnnn.b	nnnnn=0-65535; b=0-7	0xC0	0-7	Bit
Output Image	Qnnnnn.b	nnnnn=0-65535; b=0-7	0xC1	0-7	Bit
Bit	Mnnn.b	n=0-100; b=0-7	0xC2	0-7	Bit
Data Area Bit (=DB10)	Vnnnnn.b	nnnnn=0-65535; b=0-7	0xC3	0-7	Bit
Data Area Bit (=DB10)	DBXnnnn.b	nnnnn=0-65535; b=0-7	0xC4	0-7	Bit
	DB10.DBXnnnn.b is a bit address, it is in #b bit of #nnnn word with DB10				
Data Area Bit	DBmm.DBXnnnn.b	mm=1-31 nnnnn=0-65535; b=0-7	0xC5	0-7	Bit

Cable Drawing

HMI-COM port 25-pin	----- CABLE -----	PLC-port RS232C 9-pin male
TXD+ 14	-----	3 TXD+
TXD- 15	-----	8 TXD-

Operator terminal to MPI port (RS485) on controller

Communication Format

Before connection, please set up the communication parameters and the DIP switch as follows:

Format	Controller Setting	Operator Terminal Setting
Communication Format	RS485	COM1 or COM2 = RS485
Station No	2	
Transmission Speed	19200 bps	Set the SW5=OFF if parameters are set in H-Designer. Set the SW5=ON if parameters are set in the operator terminal.
Transmission Format	Size: 8-bit	
	Parity: EVEN	
	Stop bit: 1-bit	

Notes:

1. The cable is the same as for the Siemens Simatic S7 200.
2. The operator terminal and PLC stations are between 0 and 15. The operator terminal station no. is lower than the PLC station.
3. When using a macro, the size of block move is limited to within 10 words.
4. Due to the Token Ring protocol, no error message is generated when removing the cable.

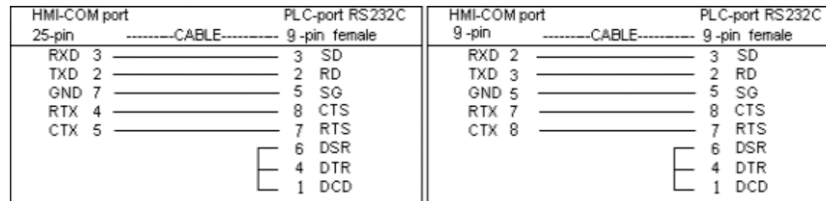
9.45 Siemens Simatic S7-300 MPI-Cable

Data format and range of the controller registers which H-Designer can access: Word Device and Bit Device.

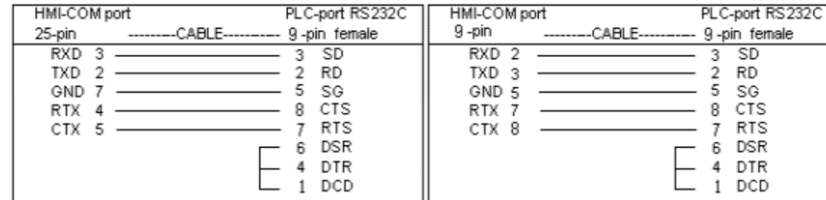
Register Type	Format	Register Range	Device Type/ Aux. Address		Data Size R/W
Input Image	IWnnnnn	nnnnn=0-65534	0	0	Word
Input Image	IDnnnnn	nnnnn=0-65532	1	0	Double-word
Output Image	QWnnnnn	nnnnn=0-65534	2	0	Word
Output Image	QDnnnnn	nnnnn=0-65532	3	0	Double-word
Bits	MWnnn	nnnnn=0-65534	4	0	Word
Bits	MDnnn	nnnnn=0-65532	5	0	Double-word
Data Area (DB10)	VWnnnnn	nnnnn=0-65534	6	0	Word
Data Area (DB10)	VDnnnnn	nnnnn=0-65532	7	0	Double-word
Data Area (DB10)	DBWnnnnn	nnnnn=0-65534; the byte nnnnn & nnnnn+1 make DBWnnnnn	6	0	Word
Data Area (DB10)	DBDnnnnn	nnnnn=0-65532; a double- word address, DBWnnnn & DBWnnnn+1 make DBDnnnn	7	0	Double-word
Data Area	DBmmm. DBWnnnnn	mmm=2-205 nnnnn=0-65534	8	0	Word
Data Area	DBmmm. DBDnnnnn	mmm=2-205 nnnnn=0-65532	9	0	Double-word

Relay Type	Format	Relay Range	Device Type/ Aux. Address		Block R/W
Input Image	Innnnn.b	nnnnn=0-65535; b=0-7	0xC0	0-7	Bit
Output Image	Qnnnnn.b	nnnnn=0-65535; b=0-7	0xC1	0-7	Bit
Bit	Mnnn.b	nnn=0-65535; b=0-7	0xC2	0-7	Bit
Data Area Bit (=DB10)	Vnnnnn.b	nnnnn=0-65535; b=0-7	0xC3	0-7	Bit
Data Area Bit (=DB10)	DBXnnnn.b	nnnnn=0-65535; b=0-7	0xC4	0-7	Bit
	DB10.DBXnnnn.b is a bit address, it is in #b bit of #nnnn word with DB10			0-7	
Data Area Bit	DBmm.DBXnnnn.b	mm=2-26 nnnnn=0-65535; b=0-7	0xC5	0-7	Bit

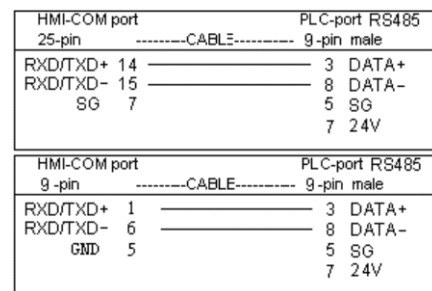
Cable Drawings



Operator terminal to MPI cable RS232C port MPI 6ES7-972-0CA21-0XA0



Operator terminal to MPI cable RS232C port HMI 6ES7-972-0CA10-0XA0



Example of connections between operator terminal and S7-300/400 CPU MPI port

Communication Format

Before connection, please set up the communication parameters and the DIP switch as follows:

Format	Controller Setting	Operator Terminal Setting
Communication Format	MPI cable RS232C	COM1 or COM2 = RS232C
Station No	02	Set PLC Station to 02 in H-Designer.
Transmission Speed	19200/38400 bps	Set the SW5=OFF if parameters are set in H-Designer. Set the SW5=ON if parameters are set in the operator terminal.
Transmission Format	Size: 8-bit	
	Parity: ODD	
	Stop bit: 1-bit	
Command Delay		Set operator terminal Command Delay
Data Block	DB10 for S7-300-CPU	

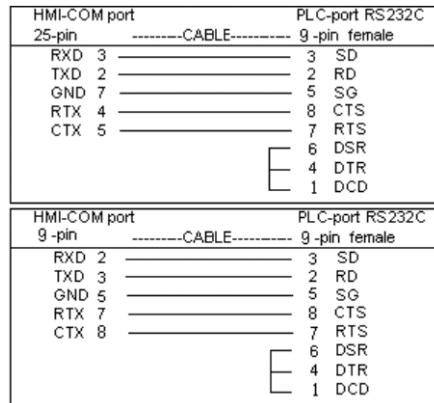
9.46 Siemens Simatic S7-300 HMI-Cable

Data format and range of the controller registers which H-Designer can access: Word Device and Bit Device.

Register Type	Format	Register Range	Data Size
Input Image	IWnnnnn	nnnnn=0-65534 length=2-523	Word
Input Image	IDnnnnn	nnnnn=0-65532 length=2-523	Double-word
Output Image	QWnnnnn	nnnnn=0-65534 length=2-523	Word
Output Image	QDnnnnn	nnnnn=0-65532 length=2-523	Double-word
Bits	MWnnn	nnnnn=0-65534 length=2-78	Word
Bits	MDnnn	nnnnn=0-65532 length=2-78	Double-word
Timer Image	Tnnnnn.10ms	nnnnn=0-65534 length=2-523	Word
Timer Image	Tnnnnn.100ms	nnnnn=0-65534 length=2-523	Word
Timer Image	Tnnnnn.1s	nnnnn=0-65534 length=2-523	Word
Timer Image	Tnnnnn.10s	nnnnn=0-65534 length=2-523	Word
Counter Image	Cnnnnn	nnnnn=0-65534 length=2-523	Word
DBmmm.DBWnnnn	mmm=2-205 nnnn=0-65534	nnnn=0-65534; DBWnnnn is a word address, the byte nnnn & nnnn+1 make DBWnnnn	Word
DBmmm.DBDnnnn	mmm=2-205 nnnn=0-65532	nnnn=0-65532; DBDnnnn is a double-word address, DBWnnnn & DBWnnnn+1 make DBDnnnn	Double-word
Data Area (DB10)	DBWnnnnn	nnnnn=0-65534; DBWnnnnn is a word address, the byte nnnnn & nnnnn+1 make DBWnnnnn	Word
Data Area (DB10)	DBDnnnnn	nnnnn=0-65532; DBDnnnnn is a double-word address, DBWnnnn & DBWnnnn+1 make DBDnnnn	Double-word
Data Area (DB10)	VWnnnnn	nnnnn=0-65534; VWnnnnn is a word address, the byte nnnnn & nnnnn+1 make DBWnnnnn	Word
Data Area (DB10)	VDnnnnn	nnnnn=0-65532; VDnnnnn is a double-word address, DBWnnnn & DBWnnnn+1 make DBDnnnn	Double-word

Relay Type	Format	Relay Range	Block
Input Image	Innnnn.b	nnnnn=0-65535; b=0-7	b=0, e.g. I3.0
Output Image	Qnnnnn.b	nnnnn=0-65535; b=0-7	b=0, e.g. Q2.0
Bit	Mnnnnn.b	nnnnn=0-65535; b=0-7	b=0, e.g. M0.0
Data Area Bit	DBmm.DBXnnnnn.b	mm=2-26 nnnnn=0-65535; b=0-7	b=0, e.g. DB22.DBX20.0
Data Area Bit (=DB10)	DBXnnnnn.b	nnnnn=0-65535; b=0-7	b=0, e.g. DBX23.0 DBX23.0=DB10.DBX23.0
Data Area Bit (=DB10)	Vnnnnn.b	nnnnn=0-65535; b=0-7	b=0, e.g. V23.0
	DB10.DBXnnnnn.b is a bit address, it is in #b bit of #nnnnn word with DB10		V23.0=DB10.DBX23.0

Cable Drawings



Operator terminal and RS232C port on HMI 6ES7-972-0CA10-0XA0

Communication Format

Before connection, please set up the communication parameters and the DIP switch as follows:

Format	Controller Setting	Operator Terminal Setting
Communication Format	HMI Cable RS232C	COM2 = RS232C
Node Address	02	
Transmission Speed	9600/19200/38400 bps	Set the SW5=OFF if parameters are set in H-Designer. Set the SW5=ON if parameters are set in the operator terminal.
Transmission Format	Size: 8-bit	
	Parity: ODD	
	Stop bit: 1-bit	
Open Data Block	DB block for S7-300-CPU	
Command Delay		Set operator terminal Command Delay

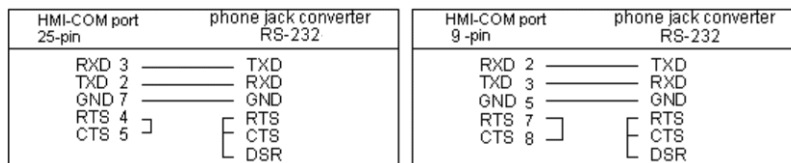
9.47 Taian TP01

Data format and range of the controller registers which H-Designer can access: Word Device and Bit Device.

Register Type	Format	Register Range	Data Size
Input Register	WXnn	nn=1-24	Word (16 bits)
Output Register	WYnn	nn=1-27	Word (16 bits)
Special Register	WSnn	nn=1-40	Word (16 bits)
Constant Register	WCnnn	nnn=1-512	Word (16 bits)
Data Register	Vnnnn	nnnn=1-1024	Word (16 bits)

Relay Type	Format	Relay Range	Block
Input Relay	Xnnn	nnn=1-384	Must be multiple of 16 + 1
Output Relay	Ynnn	nn=1-384	Must be multiple of 16 + 1
Auxiliary Relay	Cnnnn	nn=1-1024	Must be multiple of 16 + 1

Cable Drawings



Operator terminal to RS232 port on controller

Communication Format

Before connection, please set up the communication parameters and the DIP switch as follows:

Format	Controller Setting	Operator Terminal Setting
Communication Format	RS232C	COM1 or COM2 = RS232C
Station No	0	
Transmission Speed	9600 bps	Set the SW5=OFF if parameters are set in H-Designer. Set the SW5=ON if parameters are set in the operator terminal.
Transmission Format	Size: 8-bit	
	Parity: ODD	
	Stop bit: 1-bit	

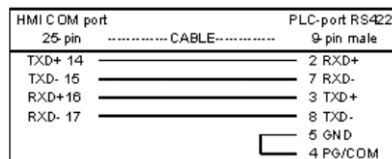
9.48 Taian TP02

Data format and range of the controller registers which H-Designer can access: Word Device and Bit Device..

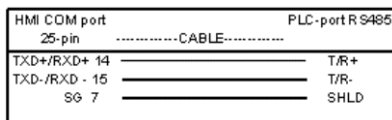
Register Type	Format	Register Range	Data Size
Input	Xnnn	nnn=1-369 (must be 1 or multiple of 16 + 1)	Word
Output Register	Ynnn	nnn=1-369(must be 1 or multiple of 16 + 1)	Word
Auxiliary Register	Vnnnn	nnnn=1-1024	Word
Auxiliary Register	Dnnnn	nnnn=1-1024	Word
System Register	WSnnn	nnn=1-128	Word
Auxiliary Relay Register	Cnnnn	nnnn=1-2048 (must be 1 or multiple of 16 + 1)	Word
Constant Register	WCnnn	nnn=1-912	Word

Relay Type	Format	Relay Range
Input	Xnnn	nnn=1-384
Output	Ynnn	nnn=1-384
Auxiliary Relay	Cnnnn	nnnn=1-2048
Special Relay	SCnnn	nnn=1-128

Cable Drawings



Operator terminal series to RS422 port on TP02



Operator terminal to RS485 port on TP02

Communication Format

Before connection, please set up the communication parameters and the DIP switch as follows:

Format	Controller Setting	Operator Terminal Setting
Communication Format	RS422/RS485	
Station No	01	PLC Station = 01
Transmission Speed	19200 bps	Set the SW5=OFF if parameters are set in H-Designer. Set the SW5=ON if parameters are set in the operator terminal.
Transmission Format	Size: 7-bit	
	Parity: EVEN	
	Stop bit: 2-bit	

Note:

The above **Controller Setting** is the H-Designer default - please refer to the controller manual.

To set up WS041, WS042, WS044, WS045:

WS041---SET RS422, baud rate, data bit, parity, stop bit

WS042---SET RS422 station number

WS044---SET RS485 baud rate, data bit, parity, stop bit

WS045---SET RS485 station number

If TP02 controller RS422 port station no.=01 → WS042 set to 01 (decimal), transmission speed (19200 bps) and transmission format (7 bits, Even, 2 bits) → WS041 set to 0120 (decimal).

9.49 Taian N2

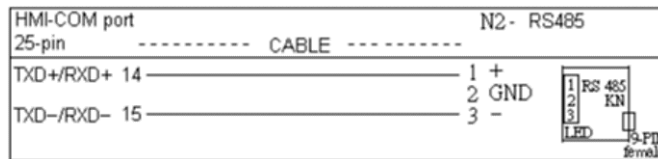
Data format and range of the controller registers which H-Designer can access: Word Device and Bit Device.

Register Type	Format	Register Range	Data Size
Function	Fnnn	nnn=0-155	Word (16 bits)

Cable Drawings

For connection to the RS232C port, the TAIAN's FA-RS-232-N2 cable must be used.

For connection to the RS485 port, the TAIAN's FA-RS-485-KN cable must be used, connected as below:



Communication Format

Before connection, please set up the communication parameters and the DIP switch as follows:

Format	Controller Setting	Operator Terminal Setting
Communication Format	RS232C RS485	
Station No	RS232C=1 RS485=1	
Transmission Speed	RS232C: 9600 bps	Set the SW5=OFF if parameters are set in H-Designer.
Transmission Format	Size RS232C: 7-bit Parity RS232C: ODD Stop bit RS232C: 1-bit	Set the SW5=ON if parameters are set in the operator terminal.

Notes:

1. Unsigned binary object must be used.
2. Even though the F125 can be chosen, it cannot be used. (Display 33333)
3. F0, F21, F42, F63, F84, F105 are quick read/write start positions. (Block sizes are 21, 21, 21, 21, 21, 23.)
4. 33333 represents the **Function** is reserved.
5. In the **Function** table, **Format** and **Function** values marked * cannot be changed.
6. If editing a double-word object, its value is composed of nearby two values. (Don't use.)
7. Please adjust the value of the operator terminal: Command Delay (block read displays 0020 error message).
8. Object, integer, decimal, digit must correspond with reality (unsigned binary object)(refer to the user manual).

9.50 Telemecanique TSX Micro

Data format and range of the controller registers which H-Designer can access: Word Device and Bit Device.

Register Type	Format	Register Range	Data Size
Internal Word	Wnnnn	nnnn=0-9999	Word
The operator terminal can read up to 60 words in one read command. Controller denotation: %MWnnnn.			

Relay Type	Format	Relay Range	Block
Bit of Internal Word	Wnnnn:bb	nnnn=0-9999; bb=0-15	bb=00, e.g. W0:0

Notes:

1. The writing unit for a single point bit is 16 bits.
2. When operator terminal changes a relay's state, the operator terminal must read 1 word (16 bits). After changing the corresponding bit, the operator terminal will write the word to the controller. These actions will take more than one controller scan. The controller ladder cannot control other bits (word) before the operator terminal has completed the "Change the Relay" action, otherwise these bits (word) will return to their initial values. In other words, the controller's control action will be resumed.

For example: When the operator terminal changes W1234:7, the operator terminal must read the word from W1234:0 to W1234:15. After changing bit 7, the operator terminal will write the word to the controller. If the operator terminal does not write to the controller, the controller ladder has changed W1234:0-W1234:6 or W1234:8-W1234:15. The operator terminal's write action will cause the controller control action to resume.

Cable Drawings

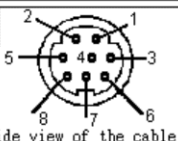
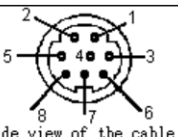
HMI-COM port 25-pin	-----CABLE-----	PLC-port RS232C g-pin male
RXD 3		2 SD
TXD 2		3 RD
GND 7		5 SG
RTX 4		7 CTS
CTX 5		8 RTS

HMI-COM port 9-pin	-----CABLE-----	PLC-port RS232C g-pin male
RXD 2		2 SD
TXD 3		3 RD
GND 5		5 SG
RTX 7		7 CTS
CTX 8		8 RTS

Example: Operator terminal COM port to RS232C port

HMI-COM port 25-pin	-----CABLE-----	PLC-port RS485 8-pin male
RXD/TXD+ 14		1 DATA+
RXD/TXD- 15		2 DATA-
SG 7		7 SG
		8 5V

HMI-COM port 9-pin	-----CABLE-----	PLC-port RS485 8-pin male
RXD/TXD+ 1		1 DATA+
RXD/TXD- 6		2 DATA-
SG 5		7 SG
		8 5V



Example: Operator terminal COM port to TER RS485 port

Communication Format

Before connection, please set up the communication parameters and the DIP switch as follows:

Format	Controller Setting	Operator Terminal Setting
Communication Format	RSRS232C/RS485	COM2 = RS232C/RS422/RS485 RS485: Set SW10=ON
Station No	0-8; Master	Set operator terminal station to 1-8 SLAVE in H-Designer
Transmission Speed	9600/19200 bps	Set the SW5=OFF if parameters are set in H-Designer. Set the SW5=ON if parameters are set in the operator terminal.
Transmission Format	Size: 8-bit	
	Parity: ODD	
	Stop bit: 1-bit	

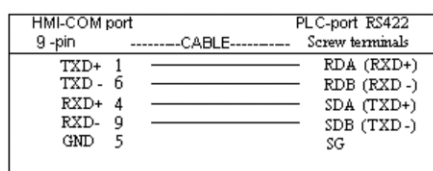
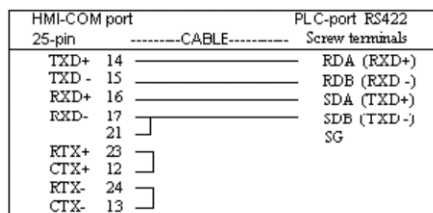
9.51 Toshiba M20/M40

Data format and range of the controller registers which H-Designer can access: Word Device and Bit Device.

Register Type	Format	Register Range	Data Size
Input Relay Register	XWnn	nn=0-63	Word
Output Relay Register	YWnn	nn=0-63	Word
Internal Relay Register	RWnn	nn=0-63	Word
Link Register	ZWnn	nn=0-31	Word
Timer Register	Tnnn	nnn=0-127	Word
Counter Register	Cnn	nn=0-95	Word
Data Register	Dnnnn	nnnn=0-1535	Word
The operator terminal can read up to 32 words in one read command.			

Relay Type	Format	Relay Range	Block
Input Relay	Xnnb	nn=0-31; b=0-f	b=0 e.g. X10
Output Relay	Ynnb	nn=0-31; b=0-f	b=0 e.g. Y00
Internal Relay	Rnnb	nn=0-63; b=0-f	b=0 e.g. R100
Link Relay	Znnb	nn=0-31; b=0-f	b=0 e.g. Z310

Cable Drawings



Operator terminal to RS422 Computer Link port on controller

Communication Format

Before connection, please set up the communication parameters and the DIP switch as follows:

Format	Controller Setting	Operator Terminal Setting
Communication Format Programmer/computer link	RS422 Computer link can be used	COM2 = RS232C/RS422/RS485 RS422: Set SW10=OFF RS485: Set SW10=ON
Station No	0	
Transmission Speed	9600 bps	Set the SW5=OFF if parameters are set in H-Designer.
Transmission Format	Size: 8-bit	Set the SW5=ON if parameters are set in the operator terminal.
	Parity: EVEN/ODD/NONE	
	Stop bit: 1-bit	

9.52 Toshiba T1/T2

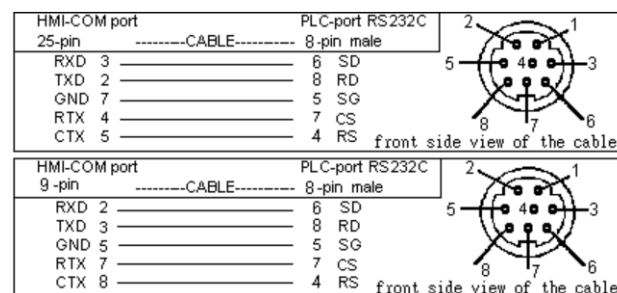
Data format and range of the controller registers which H-Designer can access: Word Device and Bit Device.

Register Type	Format	Register Range	Data Size
External Input Register	XWnn	nn=0-63	Word
Output Relay Register	YWnn	nn=0-63	Word
Direct Input Register	IWnn	nn=0-63	Word
Direct Output Register	OWnn	nn=0-63	Word
Auxiliary Relay Register	RWnnn	nnn=0-127	Word
Special Register	SWnnn	nnn=0-255	Word
Timer Register	Tnnn	nnn=0-255	Word
Counter Register	Cnnn	nnn=0-255	Word
Data Register	Dnnnn	nnnn=0-4095	Word
Link Register	Wnnnn	nnnn=0-1023	Word
Link Relay Register	LWnnn	nnn=0-255	Word
File Register	Fnnnn	nnnn=0-1023	Word

The operator terminal can read up to 32 words in one read command.

Relay Type	Format	Relay Range	Block
External Input Device	Xnnb	nn=0-63; b=hex number 0-F	b=0 e.g. X10
Output Relay Device	Ynnb	nn=0-63; b=0-f	b=0 e.g. Y00
Direct Input Device	Innb	nn=0-63; b=0-f	b=0
Direct Output Device	Onnb	nn=0-63; b=0-f	b=0
Auxiliary Relay Device	Rnnnb	nnn=0-127; b= 0-f	b=0 e.g. R100
Special Device	Snnnb	nnn=0-255; b= 0-f	b=0 e.g. S230
Timer Device	T.nnn	nnn=0-255	
Counter Device	C.nnn	nnn=0-255	
Link Device	Znnnb	nnn=0-511; b= 0-f	b=0 e.g. Z30
Link Relay	Lnnnb	nnn=0-255; b= 0-f	b=0 e.g. L2550

Cable Drawings



Operator terminal COM port to T1 CPU serial port

Communication Format

Before connection, please set up the communication parameters and the DIP switch as follows:

Format	Controller Setting	Operator Terminal Setting
Communication Format	RS232C	COM1 or COM2 = RS232C
Station No	0 =T1	Set PLC Station to 00 in H-Designer.
Transmission Speed	9600 bps	Set the SW5=OFF if parameters are set in H-Designer. Set the SW5=ON if parameters are set in the operator terminal.
Transmission Format	Size: 8-bit	
	Parity: ODD	
	Stop bit: 1-bit	

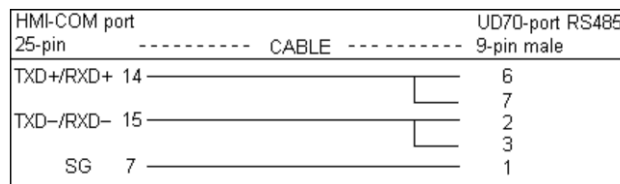
9.53 Unidriver UD70

Data format and range of the controller registers which H-Designer can access: Word Device and Bit Device.

Register Type	Format	Register Range	Data Size
Data Register	#70.00-#70.99	W 0 - W 198	Word (32 bits)
	#71.00-#71.99	W 200 - W 398	Word (32 bits)
	#72.00-#72.99	W 400 - W 598	Word (32 bits)
	#73.00-#73.99	W 600 - W 798	Word (32 bits)
	#18.01-#18.30	W 800 - W 858	Word (32 bits)
	#19.01-#19.30	W 860 - W 918	Word (32 bits)
	#20.01-#20.50	W 920 - W1018	Word (32 bits)
	#91.01-#91.10	W1020 - W1029	Word (16 bits)

Relay Type	Format	Relay Range	Block
Bit Relay	#18.31-#18.50	B0 - B19	
	#19.31-#19.50	B20 - B39	

Cable Drawing



Operator terminal to RS232C port on controller

Communication Format

Before connection, please set up the communication parameters and the DIP switch as follows:

Format	Controller Setting	Operator Terminal Setting
Communication Format	RS485	RS485
Station No	Set the parameter of UD70 address #14.01 to 11	Set PLC Station to 11 in H-Designer.
Transmission Speed	9600 bps	Set the SW5=OFF if parameters are set in H-Designer. Set the SW5=ON if parameters are set in the operator terminal.
Transmission Format	Size: 7-bit	
	Parity: EVEN	
	Stop bit: 1-bit	
Mode	Set the parameter of UD70 address#14.02 to 06	

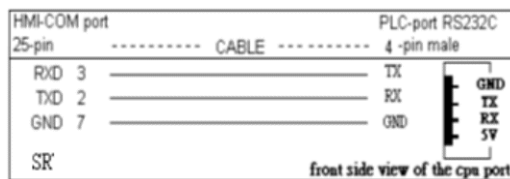
9.54 Vigor M

Data format and range of the controller registers which H-Designer can access: Word Device and Bit Device.

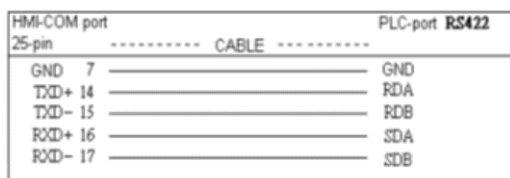
Register Type	Format	Register Range	Data Size
Input Relay	Xnnn	nnn=0-777, oct., 0 or multiple of 8	Byte
Output Relay	Ynnn	nnn=0-777, oct., 0 or multiple of 8	Byte
Auxiliary Relay	Mnnnn	nnnn=0-5119 0 or multiple of 8	Byte
Step Relay	Snnn	nnn=0-999 0 or multiple of 8	Byte
Special Relay	Mnnnn	Nnnn=9000-9255 0 or multiple of 8	Byte
Data	Dnnnn	nnn= 0-8191	Word (16 bits)
Special Data	Dnnnn	nnnn= 9000-9255	Word (16 bits)
Timer Register	Tnnn	nnn= 0-255	Word (16 bits)
Counter Register	Cnnn	nnn= 0-199	Word (16 bits)
Counter Register	Cnnn	nnn= 200-255	Word (32 bits)

Relay Type	Format	Relay Range	Block
Input Relay	Xnnn	nnn=0-777, oct.	Must be 0 or multiple of 8
Output Relay	Ynnn	nnn=0-777, oct.	Must be 0 or multiple of 8
Auxiliary Relay	Mnnnn	nnnn=0-5119	Must be 0 or multiple of 8
ST1 Status	Snnn	nnn=0-999	Must be 0 or multiple of 8
Special Relay	Mnnnn	Nnnn=9000-9255	Must be 0 or multiple of 8
Timer Relay	Tnnn	nnn= 0-255	Must be 0 or multiple of 8
Counter Relay	Cnnn	nnnn= 0-255	Must be 0 or multiple of 8
Timer Coil	TCnnn	nnn= 0-255	Must be 0 or multiple of 8
Counter Coil	CCnnn	nnn= 0-199	Must be 0 or multiple of 8

Cable Drawings



Operator terminal to RS232C port on controller with 4-pin male connector



Operator terminal to RS422 port on controller

Communication Format

Before connection, please set up the communication parameters and the DIP switch as follows:

Format	Controller Setting	Operator Terminal Setting
Communication Format	RS232C, RS422	
Station No	RS232C=0 RS422=1	RS232C=0 RS422=1
Transmission Speed	19200 bps	Set the SW5=OFF if parameters are set in H-Designer. Set the SW5=ON if parameters are set in the operator terminal.
Transmission Format	Size: 7-bit	
	Parity: EVEN	
	Stop bit: 1-bit	

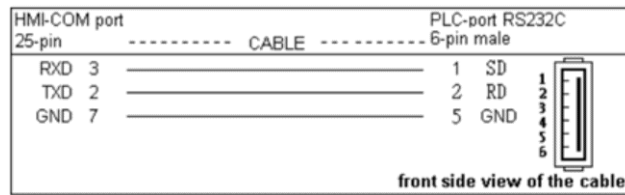
9.55 Yokogawa FA-M3

Data format and range of the controller registers which H-Designer can access: Word Device and Bit Device.

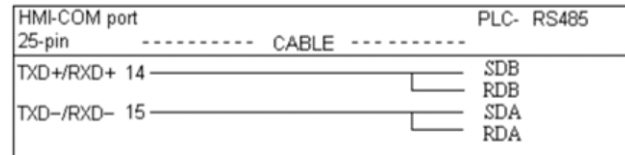
Register Type	Format	Register Range	Data Size	Example
Input Relay	Xnnnnn	nnnnn=201-65499 (not continue)	Word (16 bits)	X00201
Output Relay	Ynnnnn	nnnnn=201-65499 (not continue)	Word (16 bits)	Y00201
Intern Relay	Innnnn	nnnn=1-16384	Word (16 bits)	I00001
Common Relay	Ennnn	nnnn=1-4096	Word (16 bits)	E0001
Link Relay	Lnnnnn	nnnnn=1-65499 (not continue)	Word (16 bits)	L00001
Special Relay	Mnnnn	nnnn=1-9984	Word (16 bits)	M0001
Preset Timer	TPnnnn	nnnn=1-3072	Word (16 bits)	TP0001
Set Timer	TSnnnn	nnnn=1-3072	Word (16 bits)	TS0001
Preset Counter	CPnnnn	nnnn=1-3072	Word (16 bits)	CP0001
Set Counter	CSnnnn	nnnn=1-3072	Word (16 bits)	CS0001
Data Register	Dnnnn	nnnn=1-8192	Word (16 bits)	D0001
File Register	Bnnnnn	nnnnn=1-32768	Word (16 bits)	B00001
Link Register	Wnnnn	nnnnn=1-65499 (not continue)	Word (16 bits)	W00001
Special Register	Znnn	nnn=1-512	Word (16 bits)	Z001
Index Register	Vnn	nn= 1-64	Word (16 bits)	V01
Common Register	Rnnnn	nnnn= 1-4096	Word (16 bits)	R0001
The last two digits of the X and Y addresses must be multiples of 16 + 1. The I, E, L, and M addresses must be multiples of 16 + 1. One communication can READ/WRITE up to 60 words.				

Relay Type	Format	Relay Range	Data Size
Input Relay	Xnnnnn	nnnnn=201-65499 (not continue)	Bit
Output Relay	Ynnnnn	nnnnn=201-65499 (not continue)	Bit
Intern Relay	Innnnn	nnnn=1-16384	Bit
Common Relay	Ennnn	nnnn=1-4096	Bit
Link Relay	Lnnnnn	nnnn=1-65499	Bit
Special Relay	Mnnnn	nnnn=1-9984	Bit
Timer	TUnnnn	nnnn=1-3072	Bit
Counter	CUnnnn	nnnn=1-3072	Bit
The last two digits of the X, Y, and L addresses must be multiples of 16 + 1. The I, E, and M addresses must be multiples of 16 + 1.			

Cable Drawings



Operator terminal to RS232 port on controller - YOKOGAWA's CABLE must be used



Operator terminal to RS485 port on controller

Communication Format

Before connection, please set up the communication parameters and the DIP switch as follows:

Format	Controller Setting	Operator Terminal Setting
Communication Format	RS232C/RS485	
Station No	RS232C=1 RS485=1	RS232C=1 (based on CPU slot 1-4) RS485=1
Transmission Speed	RS232C=9600 bps	Set the SW5=OFF if parameters are set in H-Designer. Set the SW5=ON if parameters are set in the operator terminal.
Transmission Format	Size: 8-bit	
	Parity: EVEN	
	Stop bit: 1-bit	

Note:

Operator terminal station no. must match YOKOGAWA controller CPU slot no. If CPU plugs in the first slot, the operator terminal station no. must be set to 1.

10 Appendix A - H-Designer Features and Operator Terminal Models

The following table summarizes the H-Designer features and operator terminal models.

H-Designer Feature	H-K30	H-T50	H-T60b			H-T60c		
			S	P	N	S	P	N
Ethernet	No	No	No	No	Yes	No	No	Yes
Printer	No	No	No	Yes	Yes	No	Yes	Yes
Upload Application	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Upload/Download Recipes	No	No	No	Yes	Yes	No	Yes	Yes
Reconstruct Source	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Auxiliary Keys	No	No	No	Yes	Yes	No	Yes	Yes
Contrast Adjustment	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Turn off Backlight	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Set Time & Date	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Character Entry	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
List and Drop-down List	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Numeric Display	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Time, Date, and Day of Week Display	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Historical Display	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Alarm Display	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Macro	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Ladder	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Multi-Link (One master, Slaves)	Slave	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Cross-Link (Mutual Read)	No	No	No	No	Yes	No	No	Yes
Multi-Channel Communication	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Logging Buffer	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Common Key	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes
Slide-out Menu	No	No	Yes	Yes	Yes	Yes	Yes	Yes
System Messages	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Off-line/On-line Simulation	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
View/Edit Recipes	No	No	No	Yes	Yes	No	Yes	Yes

