Pro-face

Pro-face Memory Link(SIO) Driver

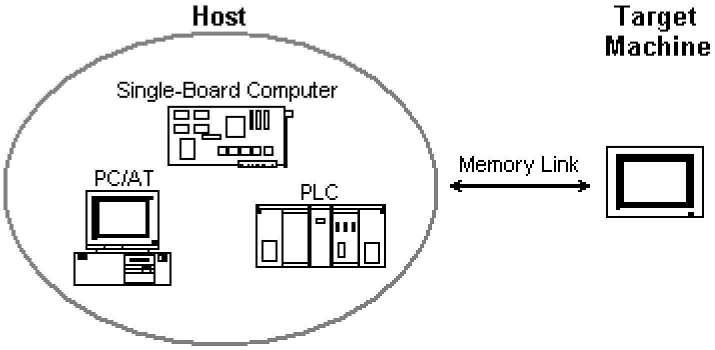
Appendix

- This manual explains how to connect the target machine with other manufacturer devices. For information about how to use the Pro-Designer software, please refer to the Pro-Designer Online Help.
- The types of target machines that are compatible with Pro-Designer depends on the version of Pro-Designer. For information about the compatibility of target machines, please refer to the Pro-Designer Online Help.

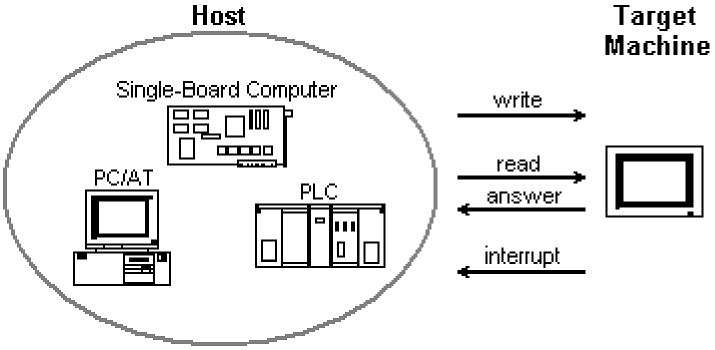
1 About Memory Link

The Memory Link (SIO) driver is a generic protocol that provides a means for users to develop a target machine for their applications without having to implement a complex protocol.

User applications that run on a single-board computer, PC/AT machine, PLC, and so on, can use the Memory Link protocol to display information on the target machine and to receive information from the target machine.



Communication between the host and target machine begins at the host. In other words, the host is the only one that can initiate communication. Even though an application exists on the target machine, and data values can change either through user inputs, scripts, or mapping to system variables, the target machine cannot initiate communications with the host.



Memory Link uses just two commands: Read and Write.

MEMO

Although the target machine cannot initiate communication, the target machine can send an interrupt that triggers the host to send a read or write command. See Section 2.3, *Interrupts*.

2 Memory Link Commands

The communication system for Memory Link uses just two basic commands.
For the host application to retrieve information that resides on the target machine, the host requests information from the target machine using the *read* command. The target machine responds to the request with a *reply* packet.
For the host application to change information on the target machine, the host sends information to the target machine using the *write* command.

MEMO

- In Pro-Designer, addresses are defined in decimal format. In Memory Link commands, addresses are defined in hexadecimal format.
  The following example, which is also used in the Read / Reply Example, uses the address 00C8h, which corresponds to address LS200 in Pro-Designer.

0	0	C	8

Pro-Designer Address = LS200

- Memory Link is an ASCII based protocol. In a memory link command, to convey a single word address requires 4 bytes of ASCII code.

Hexadecimal	0	0	C	8	Word Address
	↓	↓	↓	↓	
ASCII Code	30h	30h	43h	38h	4 bytes

2.1 Read

Read Command

By using the read command, the single-board computer, PLC, or PC/AT can request data from consecutive Memory Link device addresses on the target machine.

ESC	R	Start Address (4 bytes)	Number (4 bytes)	CR

ESC (1Bh) All communication packets begin with the ESC character.
R (52h) Identifies the packet as a read command.
Start Address Defines the first of consecutive device addresses requested by the read command.
Number Defines the number of data to return. Values in up to 255 consecutive device addresses, beginning at the Start Address, can be requested and returned.
CR (0Dh) Defines the end of the packet.

MEMO Use ASCII code in all the commands.

Reply Command

After receiving a read command, the target machine responds with a reply packet that contains the requested data.

ESC	A	Data 1 (4 bytes)	Data 2 (4 bytes)	…	Data n (4 bytes)	CR

ESC (1Bh) All communication packets begin with the ESC character.
A (41h) Identifies the packet as a reply to a read command.
Data Data is divided into 4byte blocks. Each 4byte block corresponds to data in a word address, beginning with the Start Address. The number of 4byte blocks in the reply is determined by the read command. The reply can contain a maximum 255 values.
CR (0Dh) Defines the end of the packet.

MEMO

- All communication from the target machine is in ASCII code.
- After the application sends a read command, it should wait until it receives a reply before sending another read command. Sending too many read commands without waiting for the reply could eventually lead to system errors.

Read / Reply Example

ESC	R	00C8	0003	CR

ESC	A	0049	0010	0F01	CR

In the above example, the first packet is a read command that requests data from 3 words, beginning with address LS200 (00C8h).
In the reply packet, there are three words of data. 0049h is from address LS200 (00C8h), 0010h is from address LS201 (00C9h), and 0F01h is from LS202 (00CAh).

2.2 Write

Write Command

By using the write command, the host application can send data to LS devices on the target machine.

ESC	W	Start Address (4 bytes)	Data 1 (4 bytes)	Data 2 (4 bytes)	…	Data n (4 bytes)	CR

ESC (1Bh) All communication packets begin with the ESC character.
W (57h) Identifies the packet as a write command.
Start Address Defines the device address where the data write operation begins. Each block of data is written to a device address, beginning at the Start Address. A single write command can write data to a maximum 255 consecutive device addresses.
Data Data is divided into 4byte blocks. Each 4byte block corresponds to data in a word address, beginning with the Start Address. The write can contain a maximum 255 values.

MEMO

- Use ASCII code in all the commands.
- There is no reply from a write command.

Write Example

ESC	W	0064	1A2C	145B	0020	ABCD	CR

In the above example, 1A2Ch is written to the Start Address—LS100 (0064h). 145Bh is written to the next address over—LS201 (0065h). 0020h is written to LS202 (0066h), ABCDh to LS203 (0067h).

2.3 Interrupts

Using interrupts, the target machine can notify the host of changes in data for the user application running on the target machine.

Device Address LS13

Device address LS13 on the target machine is reserved for use with interrupts. When the application on the target machine writes an integer value to LS13, the driver sends the bottom 2 bytes (00 to FF) to the host.

3 System Structure

The following table describes the basic setup for using the Memory Link protocol to link the target machine to the host.
To view the cable connection diagram, see Section 4 – *Cable Diagrams*.

Host	Comm. Format	Diagram
Single-Board Computer, Personal Computer, PLC	RS-232C	Cable Diagram 1
	RS-422(4-wire)	Cable Diagram 2

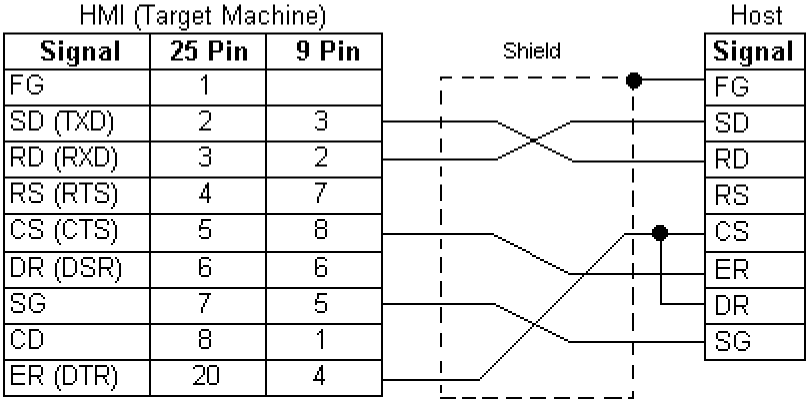
4 Cable Diagrams

- Ground the PLC's FG terminal according to your country's applicable standard. For details, refer to the PLC manual.
- When making your own communication cable, connect the SG signal if there is one.

Diagram 1 RS-232C

To connect the target machine and the host, create your own cable using the following specifications.

Target Machine	Cable / Adapter	Comments
GP, PS-P, PC/AT(PL), PS-G	User-created Cable	Cable length: 15m max.



HMI (Target Machine)

Signal	25 Pin	9 Pin
FG	1	
SD (TXD)	2	3
RD (RXD)	3	2
RS (RTS)	4	7
CS (CTS)	5	8
DR (DSR)	6	6
SG	7	5
CD	8	1
ER (DTR)	20	4

Host

Signal
FG
SD
RD
RS
CS
ER
DR
SG

Diagram 2 RS-422 (4-wire)

To connect the target machine and the host, use a recommended cable (and a pin adapter if necessary), a signal converter, or create your own cable using the following specifications.

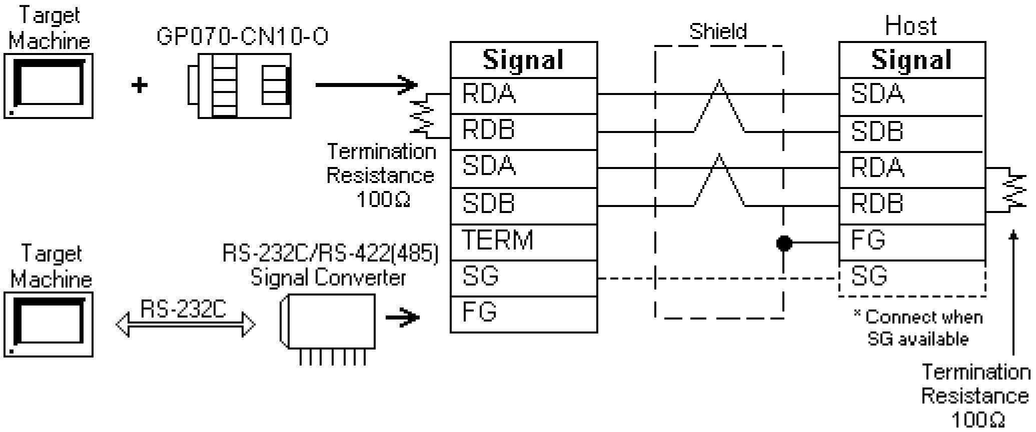
Target Machine	Cable / Adapter		Comments
GP[*1], PS-P[*1]	a	RS-422 Connection Terminal Adapter (Pro-face: GP070-CN10-O)	Cable length: 500m max.
	b	RS-422 Cable (Pro-face: GP230-IS11-O)	
	c	User-created Cable	
PC/AT(PL), PS-G	a	RS-232C / RS-422(485) Signal Converter	

*1 Connect to COM1.

MEMO

Use shielded twisted-pair cables for noise immunity. Connect the cable shields, then connect the cable to a single-point ground.

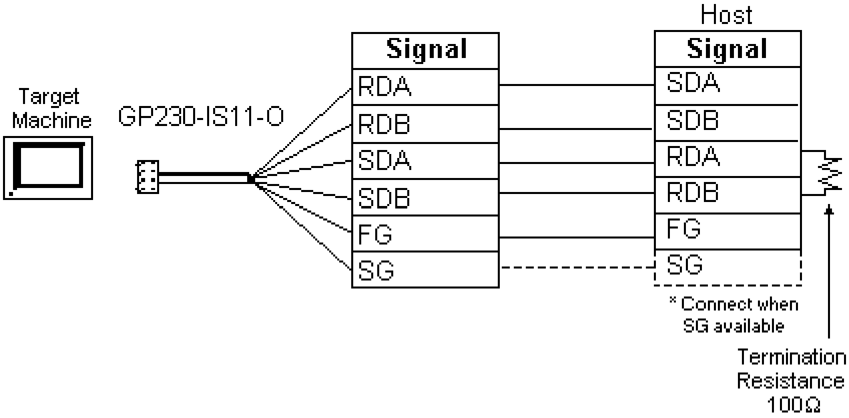
a. RS-422 Connection Terminal Adapter (Pro-face: GP070-CN10-O)
OR: RS-232C/RS-422(485) Signal Converter


Target Machine
GP070-CN10-O
+
Termination Resistance 100Ω
Target Machine
RS-232C
RS-232C/RS-422(485) Signal Converter
Signal
RDA
RDB
SDA
SDB
TERM
SG
FG
Shield
Host
Signal
SDA
SDB
RDA
RDB
FG
SG
* Connect when SG available
Termination Resistance 100Ω

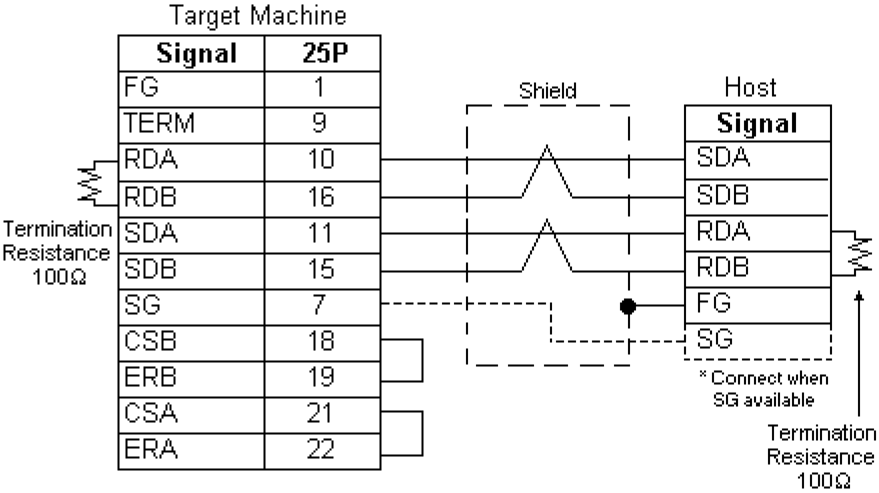
MEMO

- As the RS-422(485) side of the RS-232C/RS-422(485) signal converter is a terminal, use any external line other than the communication line for the power cable.
- Follow the specifications for your RS-232C/RS-422(485) signal converter when connecting termination resistance on the RS-422(485) terminal.
- The RS-232C connection between the RS-232C/RS-422(485) signal converter and target machine depends on the signal converter. Refer to its specifications when connecting the signal converter and target machine.

b. RS-422 Cable (Pro-face: GP230-IS11-O)



c. User-created Cable



Target Machine

Signal	25P
FG	1
TERM	9
RDA	10
RDB	16
SDA	11
SDB	15
SG	7
CSB	18
ERB	19
CSA	21
ERA	22

Host

Signal
SDA
SDB
RDA
RDB
FG
SG

Termination Resistance 100Ω
Shield
* Connect when SG available
Termination Resistance 100Ω

5 Supported Device Addresses

The following table lists the device address ranges you can enter from the Device Address keypad.

Device	Bit Address	Word Address	16-Bit	32-Bit
Local Storage (LS)	LS0:0 – LS8191:15	LS0 – LS8191	L/H[*1]	L/H[*1]

1. 16-bit and 32-bit data, High and Low, refer to data as defined in the following examples.

16 bit

Byte				
0	7	. . .	0	L (Low)
1	15	. . .	8	H (High)

32 bit

Word				
0	15	. . .	0	L (Low)
1	31	. . .	16	H (High)

6 Environment Setup

The following table lists the communication settings, recommended by Pro-face, for the target machine and host.
For details, see Section 8 – *Driver Configuration,* and Section 9 – *Protocol Configuration*.

RS-232C

Target Machine			Host	
Driver Interface	Serial Interface	RS-232C	Conn. Format	RS-232C
	Flow Control	DTR(ER)/CTS	—	
	Transmission Speed	19200 bps	Baud Rate	19200 bps
	Retry Count	2	—	
	Parity Bit	Even	Parity Bit	Even
	Stop Bit	1 bit	Stop Bit	1 bit
	Data Length	8 bits	Data Length	8 bits
Protocol	COM Port	COM1	—	
	TX. Wait Time	0	—	

RS-422(4-wire)

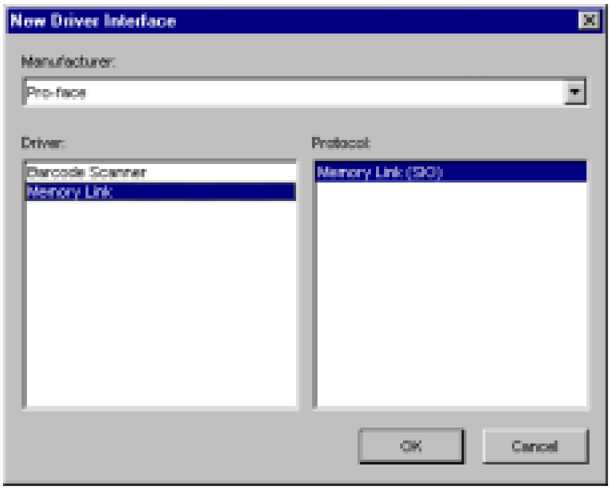
Target Machine				Host	
		GP, PS-P	PC/AT(PL),PS-G		
Driver Interface	Serial Interface	RS-422(4-wire)	RS-232C[*1]	Conn. Format	RS-422(4-wire)
	Flow Control	None	DTR(ER)/CTS	—	
	Transmission Speed	19200 bps		Baud Rate	19200 bps
	Retry Count	2		—	
	Parity Bit	Even		Parity Bit	Even
	Stop Bit	1 bit		Stop Bit	1 bit
	Data Length	8 bits		Data Length	8 bits
Protocol	COM Port	COM1		—	
	TX. Wait Time	0		—	

[*1] By using a RS-232C/RS-422(485) signal converter, you can use RS-422 connections with PC/AT (PL Series) and PS Series Type G target machines. In such a configuration, make sure you set the Driver's Serial Interface to RS-232C.

7 I/O Manager Configuration

Add the Memory Link (SIO) driver to the user application using the I/O Manager.

MEMO For information on how to display the [New Driver Interface] dialog box, or for details about the I/O Manager, see the online help.



New Driver Interface

Manufacturer: Pro-face

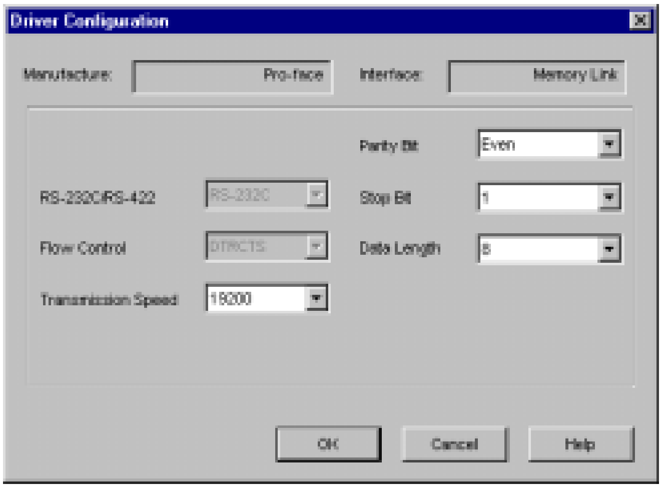
Driver: Barcode Scanner, Memory Link

Protocol: Memory Link (SIO)

8 Driver Configuration

To configure the communication settings for the target machine, use the [Driver Configuration] dialog box. Make sure these communication settings match those in the host.
For an overview of the driver and protocol settings, see Section 6 – *Environment Setup*.

MEMO For information on how to display the [Driver Configuration] dialog box, see the online help.



Driver Configuration

Manufacture: Pro-face

Interface: Memory Link

Parity Bit: Even

RS-232C/RS-422: RS-232C

Stop Bit: 1

Flow Control: DTR/CTS

Data Length: 8

Transmission Speed: 19200

Manufacturer

Displays Pro-face.

Interface

Displays the type of serial connection between the target machine and host.

Serial Interface

Defines the serial connection: RS-232C or RS-422 (4-wire).
For details about the supported connections, see Section 4 – *Cable Diagrams*.

Flow Control

Defines the signals that control the data flow.

Transmission Speed

Sets the communication speed in bits per second. This setting must match the baud rate on the host.

Parity Bit

Sets either a parity bit [Even or Odd] for use in detecting communication errors, or [None] at all.

Stop Bit

Defines the stop bit: 1 bit, 1.5 bits, or 2 bits.

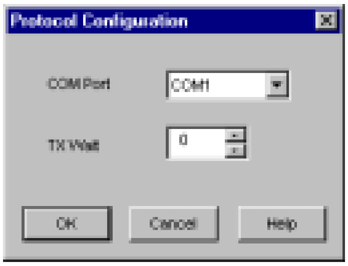
Data Length

Defines the length of each unit of data: 7 bit or 8 bit.

9 Protocol Configuration

To set up details about the communication process between the target machine and the host, use the [Protocol Configuration] dialog box.
For an overview of the driver and protocol settings, see Section 6 – *Environment Setup*.

MEMO For information on how to display the [Protocol Configuration] dialog box, see the online help.



Protocol Configuration

COM Port COM1

TX Wait 0

COM Port

Defines which COM port to use on the target machine for communications with the host.

MEMO Select COM1 for PS Series Type G target machines. Connection is not possible using COM2.

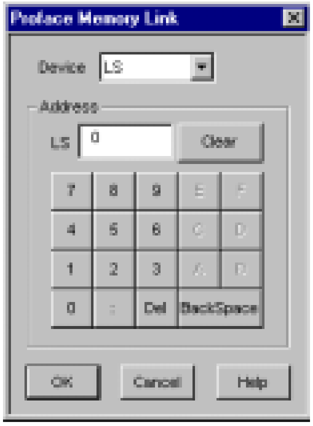
TX Wait Time

Defines the number of milliseconds the target machine waits, after receiving a communication packet, before sending a response.

10 Device Address Configuration

To set up a variable for an LS device in the Variable List, use the Device Address Keypad from the variable properties.
See Section 5 – *Supported Device Addresses*.

MEMO For information on how to display the Device Address Keypad, see the online help.


Proface Memory Link

Device LS

LS 0

Device

Lists the device types.

Address

Enter the device address for the I/O variable. The keypad ensures that you enter the correct format for bit and word devices.

Appendix

1 GP-Pro/PBIII Support

This appendix discusses features of Memory Link that were available in GP-Pro/PBIII, but are not available in Pro-Designer.

Pro-Designer does not support the following features.

- 1:n Cable Connections
- System Data Area
- Special Relays
- Extended Mode

Although the System Data Area and Special Relays is not designed specifically into the driver, some of their features can be supported indirectly using scripts and variables.

1.1 1:n Cable Connections

In Pro-Designer, the Memory Link protocol supports a 1:1 connection using RS-232C or RS-422. Pro-Designer's Memory Link protocol does not support 1:n connections.

1.2 System Data Area

The System Data Area is not available in Pro-Designer.

However, some of the functionality provided by the System Data Area is available through the use of Pro-Designer scripts and variables.

See Appendix 2, *Example: How to Implement the System Data Area* for an example on how to implement the following features in the System Data Area.

- Change Panel
- Clock
- Keypad Status

1.3 Special Relays

Special Relays are not available in Pro-Designer. In GP-Pro/PBIII, devices LS2032 to LS2047 were used as counters and special status addresses. For example, 2036 served as a 1 second binary counter, and 2032 stored various relay states.

You can emulate some Special Relays. For example, to emulate the 1 second binary counter, create an application script that runs once every 1000 milliseconds and turns ON a discrete variable during one scan, then turns OFF the variable on the next.

1.4 Extended Mode

Pro-Designer does not support the Extended Mode as used in GP-Pro/PBIII.

2 Example: How to Implement the System Data Area

In the GP-Pro/PBIII Series of Memory Link, the System Data Area was a reserved block of memory that stored the HMI status, hardware specific functions, and miscellaneous information. Pro-Designer does not specifically support the System Data Area.
However, you can implement some of the functionality provided by the System Data Area using Pro-Designer scripts and variables.

The following table lists System Data Area features available in GP-Pro/PBIII that can be supported in Pro-Designer.

Address*1	Contents	Description
LS1	Status	Bit 3: K-tag status
LS4	Clock: Year	0 - 99 (2 digit year)
LS5	Clock: Month	1 - 12 (Binary)
LS6	Clock: Day	0 - 31 (Binary)
LS7	Clock: Hour	0 - 23 (Binary)
LS8	Clock: Minute	0 - 59 (Binary)
LS13	Interrupt	After value is written to this device address, bottom 2 bytes are sent to the host computer.
LS15	Panel ID	1 - 8999 (Binary) Panel ID of current screen.

*1 Address as used in GP-Pro/PBIII.

Each of the following sections describes how to implement a feature, with the setup required on the Host and target machine.

Change Panel

Set up the following variable and script in Pro-Designer to duplicate the System Data Area's Change Panel feature, address LS15.

Variable

In the Pro-Designer application, create the following variable.

Name	Data Type	Source	Address
PanelID	Integer	PLC	LS15

Script: Change Panel from the Host

In the Pro-Designer application, set up the following script.

Script Type	Application Script
Trigger	Conditional
Trigger Variable	PanelID
Trigger Condition	On Data Change

Script	int a = PanelID.getIntValue( );
	int b = _CurPanelID.getIntValue( );
	if (a != b)
	  _CurPanelID.write(a); // change panel

The host causes the script to run by sending a Write command to device address LS15, which is mapped to the variable PanelID in Pro-Designer.

The script compares the values in PanelID and _CurPanelID. If the values are different, the script copies the value from PanelID to _CurPanelID, which changes the current panel.

MEMO The host machine cannot write directly to system variables. _CurPanelID is a Pro-Designer system variable that stores the ID number of the current panel. As a result, the variable PanelID is required to act as an intermediary between the host application and _CurPanelID system variable.

Change Date/Time

Set up the following variables and scripts in Pro-Designer to duplicate the System Data Area's Clock feature, addresses LS4 to LS8.

Variables

In the Pro-Designer application, create the following variables.

Name	Data Type	Source	Address
YearVar	Integer	PLC	LS4
MonthVar	Integer	PLC	LS5
DayVar	Integer	PLC	LS6
HoursVar	Integer	PLC	LS7
MinutesVar	Integer	PLC	LS8
DateStatus	Discrete	PLC	LS99:00
TimeStatus	Discrete	PLC	LS99:01

Script: Change Date from the Host

In the Pro-Designer application, set up the following script.

Script Type	Application Script
Trigger	Conditional
Trigger Variable	DateStatus
Trigger Condition	When True

Script	int tempYear, tempMonth, tempDay;
	tempYear = YearVar.getIntValue( );
	tempMonth = MonthVar.getIntValue( );
	tempDay = DayVar.getIntValue( );
	Sys.setDate(tempYear,tempMonth,tempDay);
	DateStatus.write(0);

For the host to change the date on the target machine, the host sends a Write command to device addresses LS4 to LS6 with the new year, month, and day values.
To cause the script to run, the host sends a Write command to device address LS99 with the value 1. The script first retrieves the values from device addresses LS4 to LS6 using the variables YearVar, MonthVar, and DayVar. Second, the script sets the new date on the target machine. Third, the script resets the DateStatus variable.

MEMO Use 4 digits for the year.

Script: Change Time from the Host

Set up the following script in Pro-Designer.

Script Type	Application Script
Trigger	Conditional
Trigger Variable	TimeStatus
Trigger Condition	When True

Script	int tempHours, tempMinutes;
	tempHours = HoursVar.getIntValue( );
	tempMinutes = MinutesVar.getIntValue( );
	Sys.setTime(tempHours,tempMinutes,-1);
	TimeStatus.write(0);

For the host to change the time on the target machine, the host sends a Write command to device addresses LS7 and LS8 with the new hour and minute values.
To cause the script to run, the host sends a Write command to device address LS99 with the value 2. The script first retrieves the values from device addresses LS7 and LS8 using the variables HoursVar and MinutesVar. Second, the script sets the new time on the target machine. Third, the script resets the TimeStatus variable.

Signal the Host of Keypad Data Input

Set up the following variables and script to notify the host whenever data is being input on the target machine.

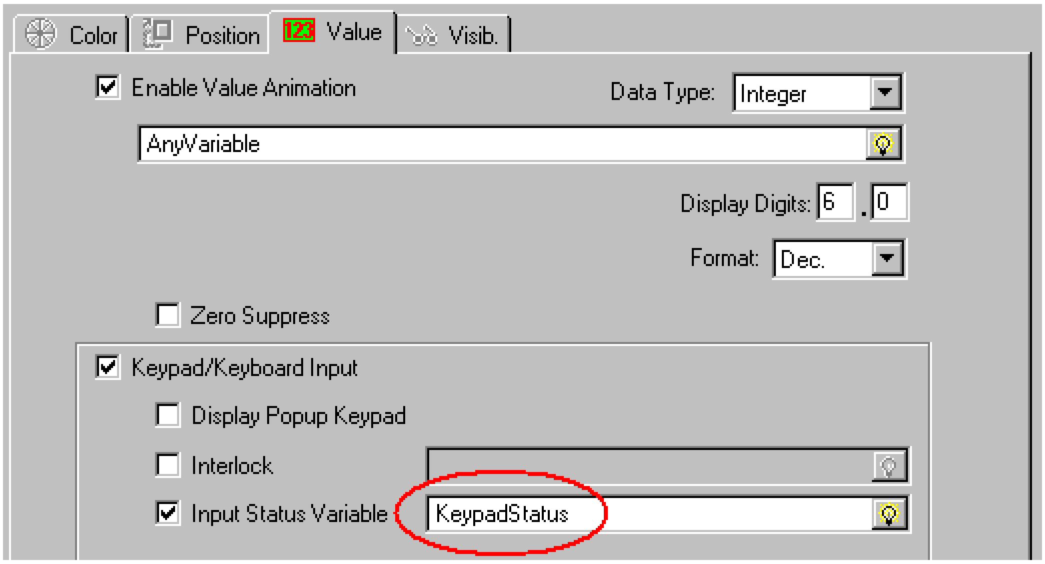
Variables

In the Pro-Designer application, create the following variables.

Name	Data Type	Source	Address
InterruptVar	Integer	PLC	LS13
KeypadStatus	Discrete	PLC	LS1:03

Script: Input Data on the Target Machine

In the Pro-Designer application, set up Value animation as shown.



☑ Enable Value Animation

Data Type: Integer

AnyVariable

Display Digits: 6 . 0

Format: Dec.

☐ Zero Suppress

☑ Keypad/Keyboard Input

☐ Display Popup Keypad

☐ Interlock

☑ Input Status Variable KeypadStatus

In the Pro-Designer application, set up the following script.

Script Type	Application Script
Trigger	Conditional
Trigger Variable	KeypadStatus
Trigger Condition	When True

Script	InterruptVar.write(#);
	// Replace # with value you want to send to
	// the host

On the target machine, when data is input into AnyVariable (see Value Animation), the Input Status Variable (KeypadStatus) is turned ON, which causes the script to run.
The script writes the defined value to InterruptVar. Because InterruptVar is mapped to the interrupt device address (LS13), the target machine sends the bottom two bytes of LS13 to the host.

MEMO

Set up Interrupt Service Routines (ISR) on the host. When the host receives a value from LS13, the host runs the ISR identified by the value in LS13.