20 | Using Scripts

(Programming without Parts)

This chapter explains the basics of creating and using programming scripts in GP-Pro EX. Please start by reading "20.1 Settings Menu" (page 20-2) and then turn to the corresponding page.

20.1   Settings Menu

You can use D-Scripts to create simple programs. Using this feature, you can perform operations in the GP or communicate between the GP and unsupported peripheral devices.

⚠ WARNING

Be sure to not use D-Scripts/Global D-Scripts to control systems that can cause life-threatening or serious injury.

NOTE
- D-Scripts are set up on a Base Screen. That Base Screen looks at the conditions while it is displayed and executes the script.
- When the GP is running, a Global D-Script runs based on the trigger, regardless of the screen displayed.
- Extended Scripts should be used for high-level communication programs.
- In addition to scripts, you can use logic programs for control applications.
☞ "28.1 Setup Menu" (page 28-2)

Conditional Operations

Create a script which changes screens to screen number 7 after three seconds.

☞ Setup Procedure (page 20-6)
☞ Introduction (page 20-5)

| After 1 second | After 2 seconds | After 3 seconds |

Time

Process Script ──⟨Process⟩──⟨Process⟩──⟨Process⟩──▶

| D100=1 | D100=2 | D100=3 |

D100 is not 3 so portion after "if" does not execute.

D100 is not 3 so portion after "if" does not execute.

D100 = 3 so condition is true and [w:LS0008]=7 executes.

Copying Data in Blocks

Create a script which detects the rising edge (0 to 1) of bit address M0100 and copies data stored in the connected device into another address.

☞ Setup Procedure| (page 20-13)
☞ Introduction (page 20-12)

| D0099 | C |
	·
	·
	·
	B
D0000	A

D0200	C
	·
	·
	·
	B
D0101	A

Displaying an Alarm When an Error Occurs

The temperature management system detects an error bit from the connected device and displays alarm messages when the temperature information storage address (D200) rises to 70 degrees C or higher, or falls to 30 degrees C or lower. Also, this script counts the number of detected errors.
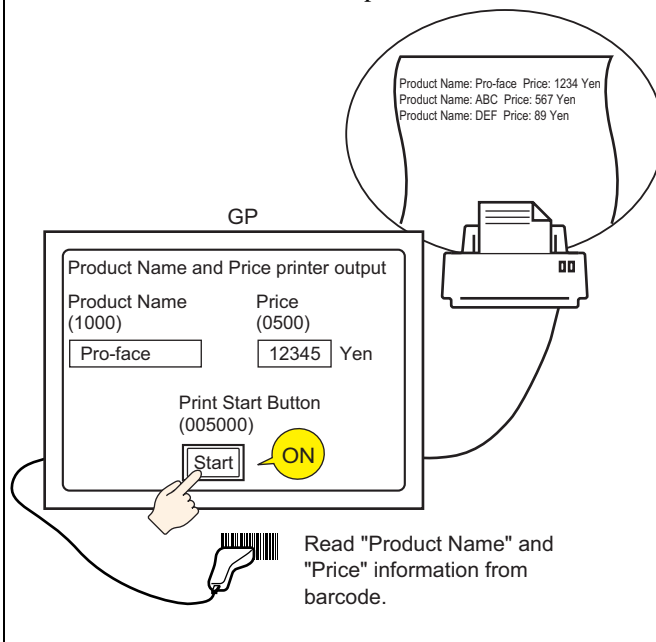
☞ Setup Procedure (page 20-18)
☞ Introduction (page 20-17)
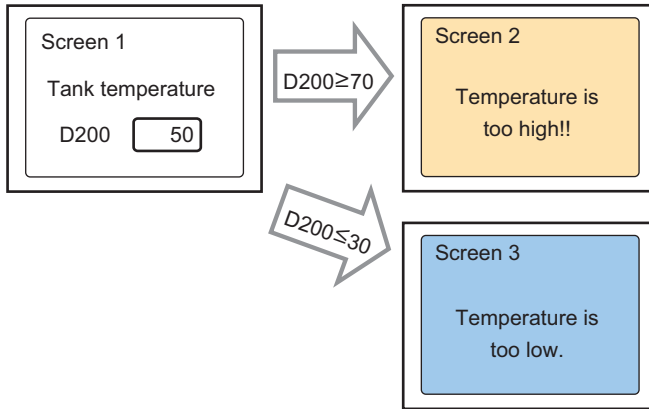
Communicating with Unsupported Peripheral Devices

Create an extended script to output data read from a barcode connected to the USB to a serial printer connected to COM1.
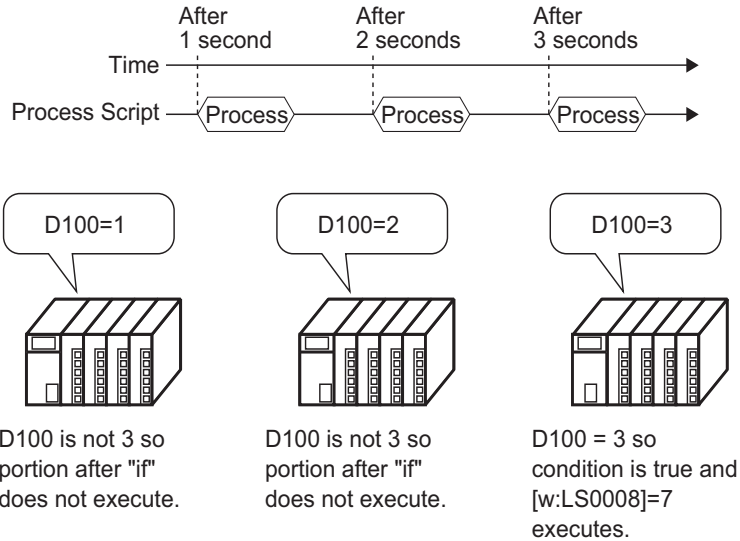
☞ Setup Procedure (page 20-37)
☞ Introduction (page 20-21)

20.2 Conditional Operations

NOTE
• Please refer to the settings guide for details.
☞ "20.8.1 D-Script/Common [Global D-Script] Settings Guide" (page 20-53)

Action

Create a script which automatically changes screens to screen number 7 after 3 seconds.

| D100=1 | D100=2 | D100=3 |

D100 is not 3 so portion after "if" does not execute.

D100 is not 3 so portion after "if" does not execute.

D100 = 3 so condition is true and [w:LS0008]=7 executes.

Commands Used

Command	Function Summary
Assignment (=)	Assigns the right side value to the left side.
Addition (+)	Adds a constant to a Word device's data.
if ( )	When a condition becomes true, the process following the "if ( )" statement is executed.
Equivalent (==)	Compares the value on the right and left sides. Becomes true if the left side equals the right side.
LS0008	Changes to the screen number stored in this value. ☞ "A.1.4.2 System Data Area" (page A-11)

Triggered Condition

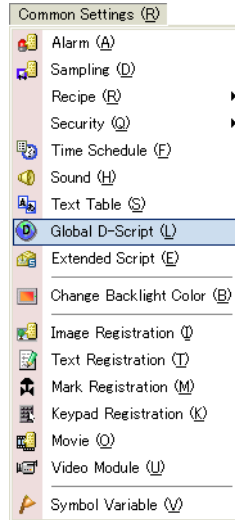Select the timer as below and set the [Timer Settings] to 1 second.
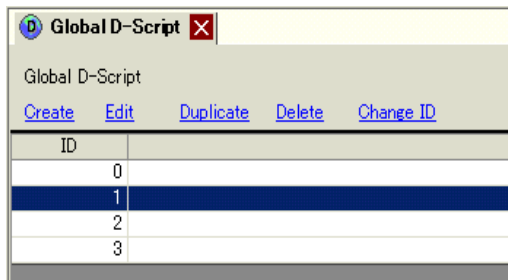

Completed Script

```
Script Expression Area    Enlarge Script Expression Area    [img]  Input Address
0001 [w:[PLC1]D00100]=[w:[PLC1]D00100]+1
0002 if([w:[PLC1]D00100]==3)
0003 {
0004   [w:[#INTERNAL]LS0008]=7
0005 }
0006 endif
0007
```
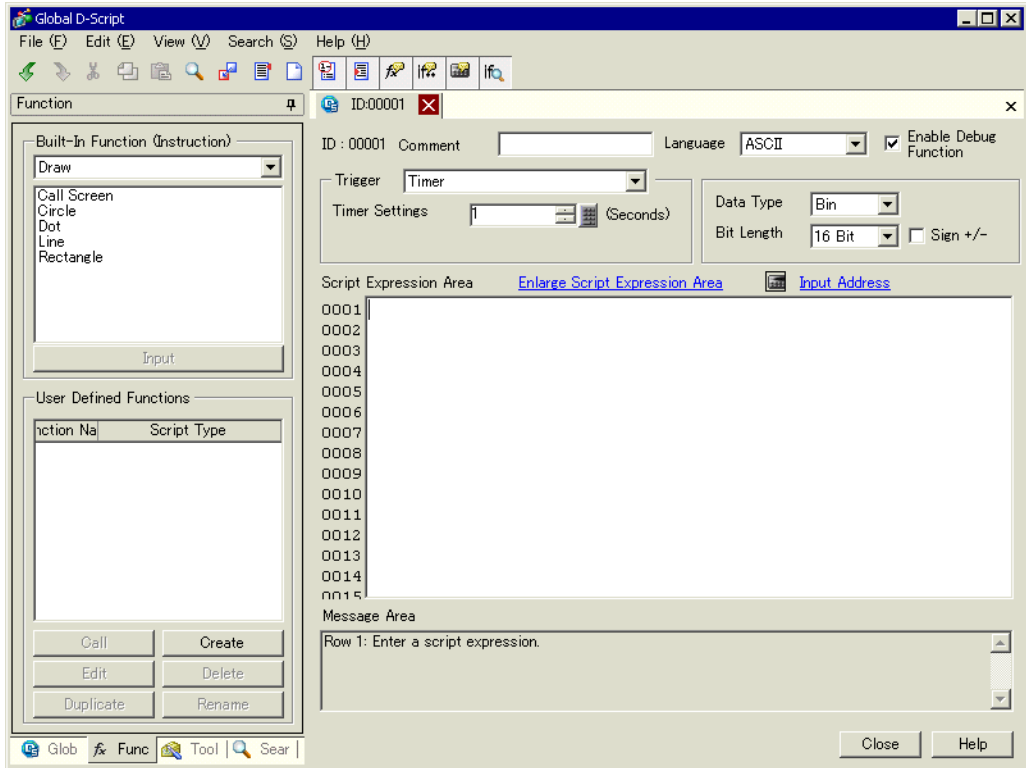
Creation Procedure

1 From the [Common Settings (R)] menu, select [Global D-Script Settings (L)].

```
Common Settings (R)
  Alarm (A)
  Sampling (D)
  Recipe (R)              ▶
  Security (Q)            ▶
  Time Schedule (F)
  Sound (H)
  Text Table (S)
  Global D-Script (L)
  Extended Script (E)
  Change Backlight Color (B)
  Image Registration (I)
  Text Registration (T)
  Mark Registration (M)
  Keypad Registration (K)
  Movie (O)
  Video Module (U)
  Symbol Variable (V)
```

2 Click [Create]. To view an existing script, select the ID number and click [Edit], or double-click the ID number row.

```
Global D-Script  X

Global D-Script

Create    Edit    Duplicate    Delete    Change ID

  ID
      0
      1
      2
      3
```
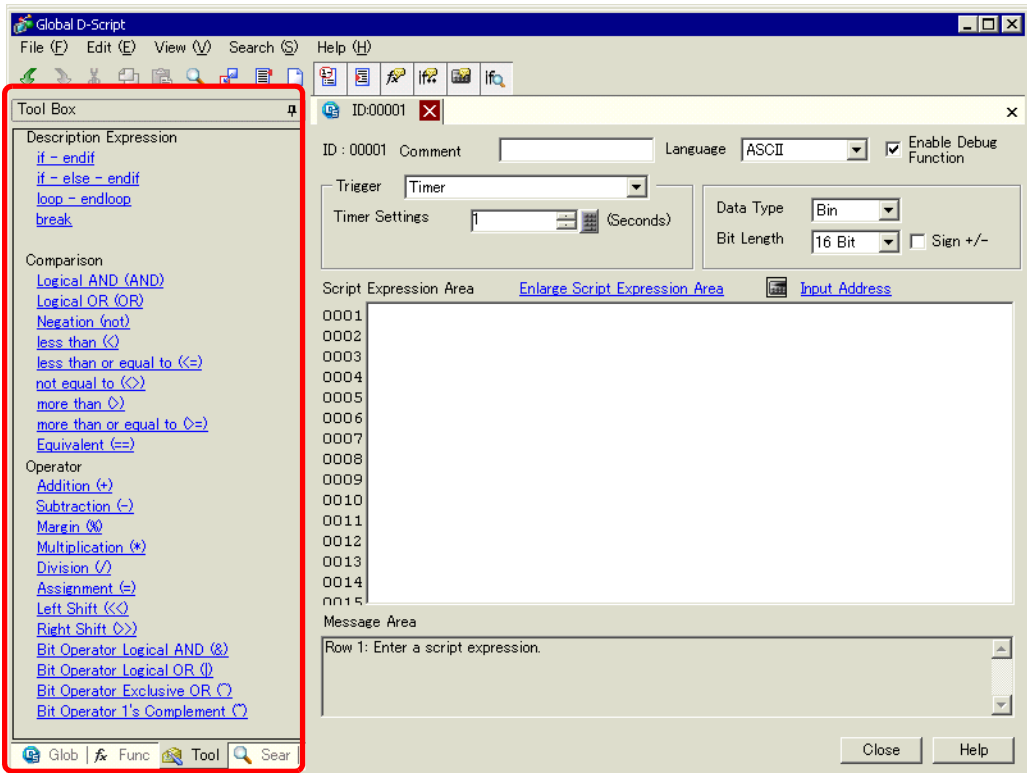
3 The [Global D-Script] dialog box is displayed.

4 Select [Timer] in the Script Trigger Conditions (trigger), select [Timer] and specify the [Timer Settings] as 1 second.
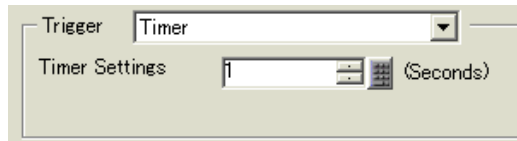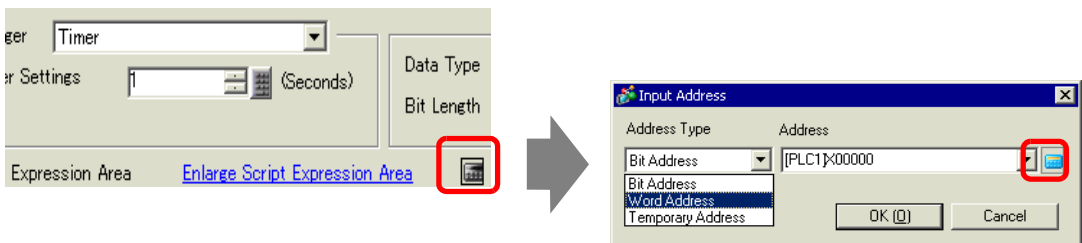
5 Click the [Tool Box] tab. The toolbox allows you to easily place a command to use in the script.
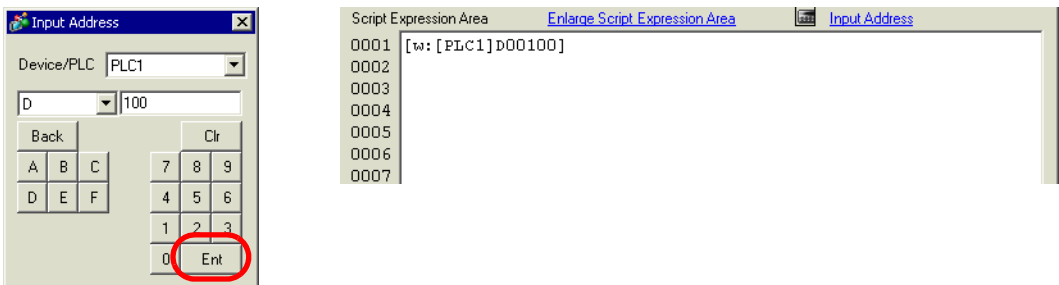

6 Create the first line of script. If you specify the D00100 default value as 0, the first line operation is a count operation that increases and stores the count every time a process completes.
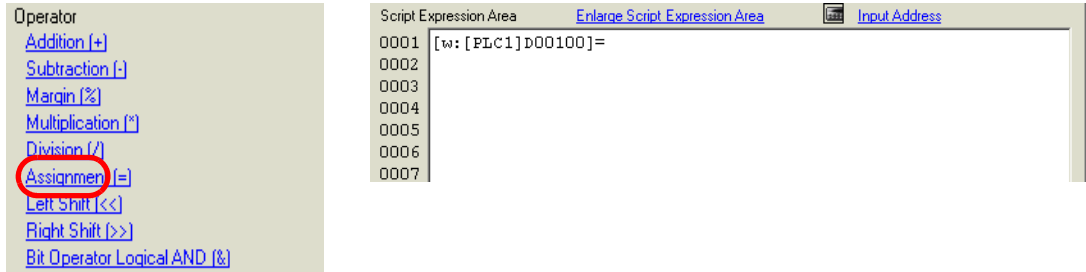
Click and select [Word Address], click .

7 Input D00100, and click [ENT].

8 Click [Assignment (=)] in the Toolbox.

9 Place D00100 in the same way as steps 6-7.
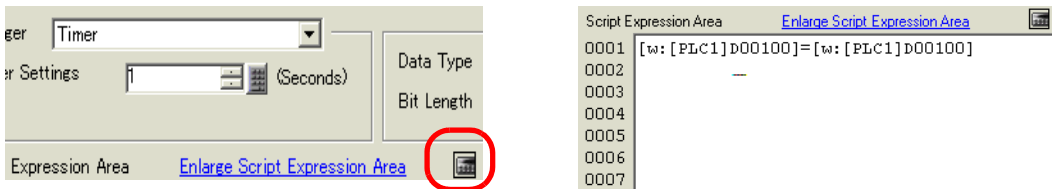
10 Click [Addition (+)] and type "1". The first row is now complete.
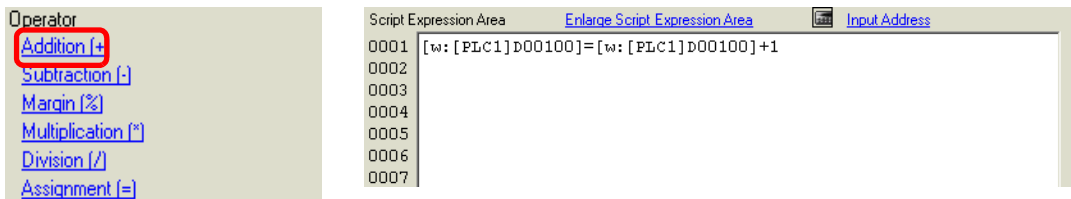
11 Create the second row of the script. In the second row, when a condition becomes true, the process following the "if ( )" statement is executed. Click [if - endif].
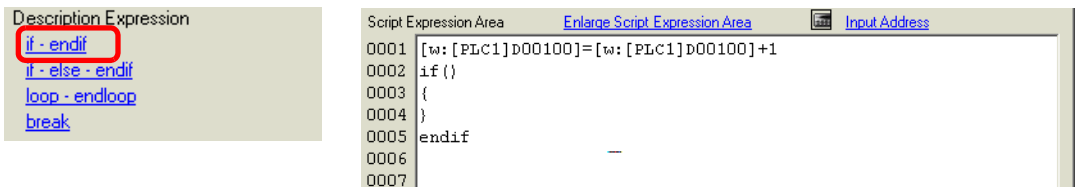
12 Create the condition expression inside the brackets "( )" following "if". The condition expression compares the value stored in D00100 to "3", and turns true if they are equal. Place the cursor inside the brackets "( )" and repeat steps 6 to 7 to place another D00100.

13 Click [Equivalent (==)] and input "3". The second row is now complete.

14 Place the cursor inside the "{ }" brackets and press Enter. Repeat steps 6 to 7 to place another LS0008.

15 Click [Assignment (=)] and input "7".

16 The script is now complete.

NOTE	• When selecting text, press the [Ctrl] Key + [Shift] Key + [->] Key or [<-] Key to select the entire text block.

• Press the [Ctrl] Key + [F4] Key to close the currently selected display.
• Press the [Esc] Key to overwrite the script or to delete and exit.

20.3   Copying Data in Blocks

Action

Create a script which detects the rising edge (0 to 1) of bit address M0100 and copies data stored in the connected device into another address.

Commands Used

Command	Function Summary
Copy Memory memcpy ( )	Copies a stored value into a device in one operation. Data for the number of Addresses will be copied to the copy destination Word Addresses beginning from the source data's first Word Address. [Format] memcpy ([Copy To Address], [Copy From Address], Words)

Triggered Condition

In [Trigger Condition], select [Bit ON], and set the [Bit Address] to M000100.
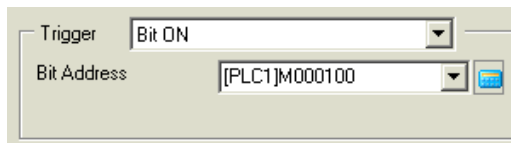
Completed Script

Creation Procedure

1 From the [Parts (P)] menu, select [D-Script (R)] or click ![icon] from the toolbar.
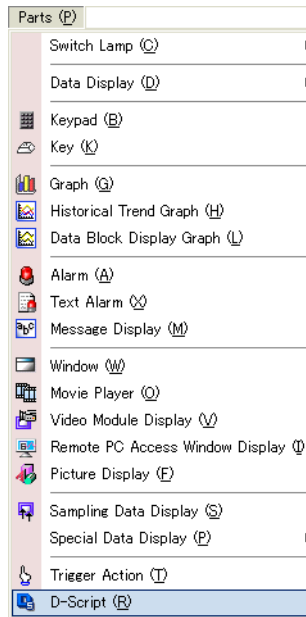
2 Click [Create]. The IDs for existing scripts are displayed in the [D-Script List].
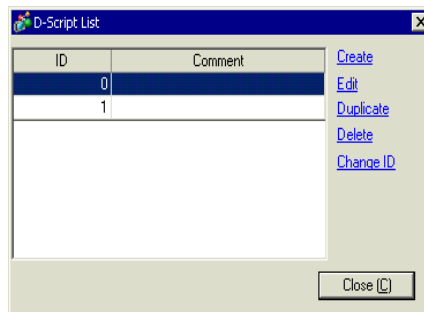
3 The [D-Script] dialog box is displayed.
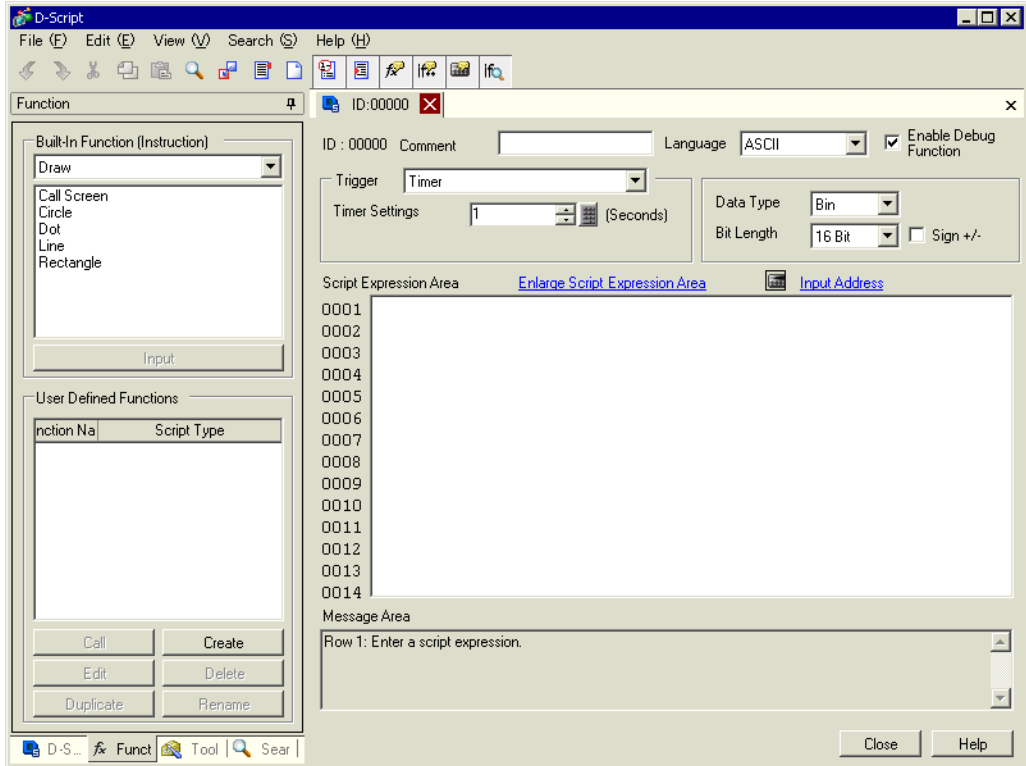
4 Select [Bit ON] in the Script Trigger Conditions (trigger) and specify M000100 as [Bit Address].
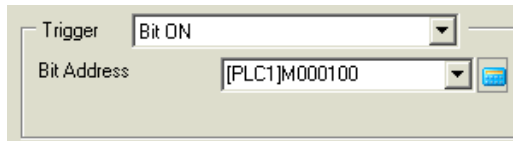
5 Click the [Function] tab. The built-in functions allow you to easily place a command to use in the script.
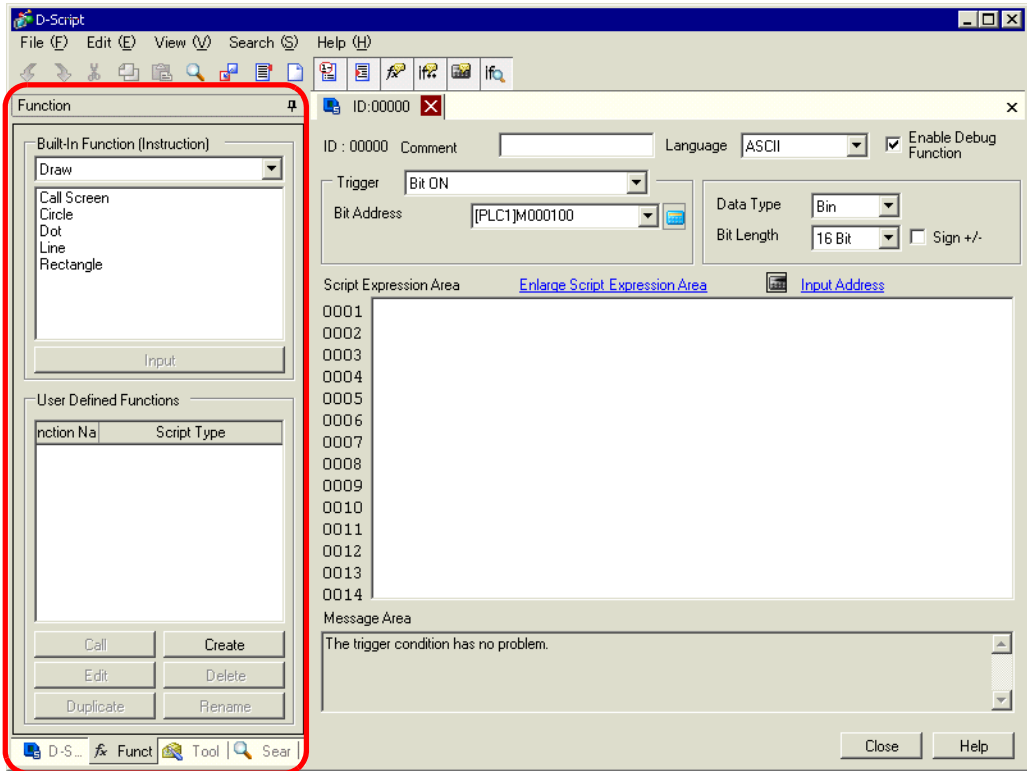
6 From [Built-in Function (Instruction)], select [Memory Operation].
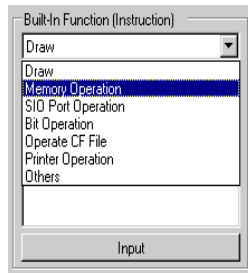
7 Double-click [Copy Memory]. Click 　 .
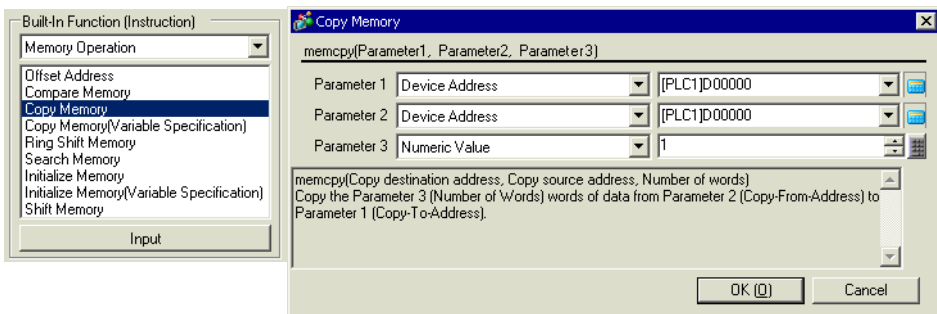
8 For [Parameter 1], enter D00101, and click [ENT].
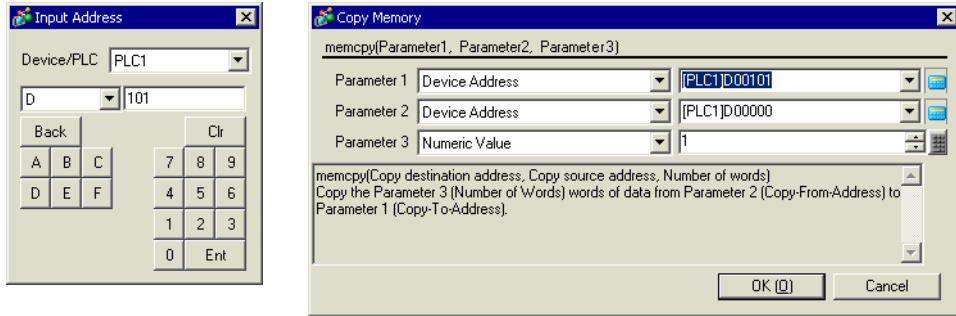
9 For [Parameter 2] enter D00000, and click [OK].
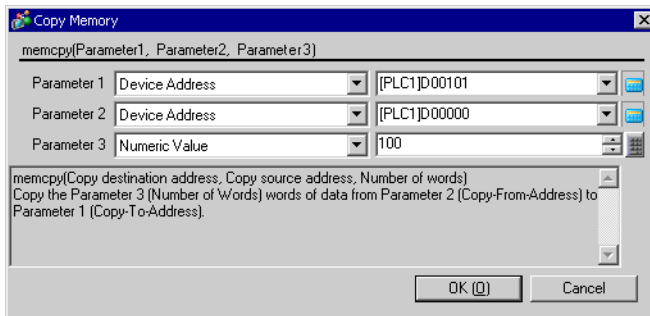
10 The script is now complete.
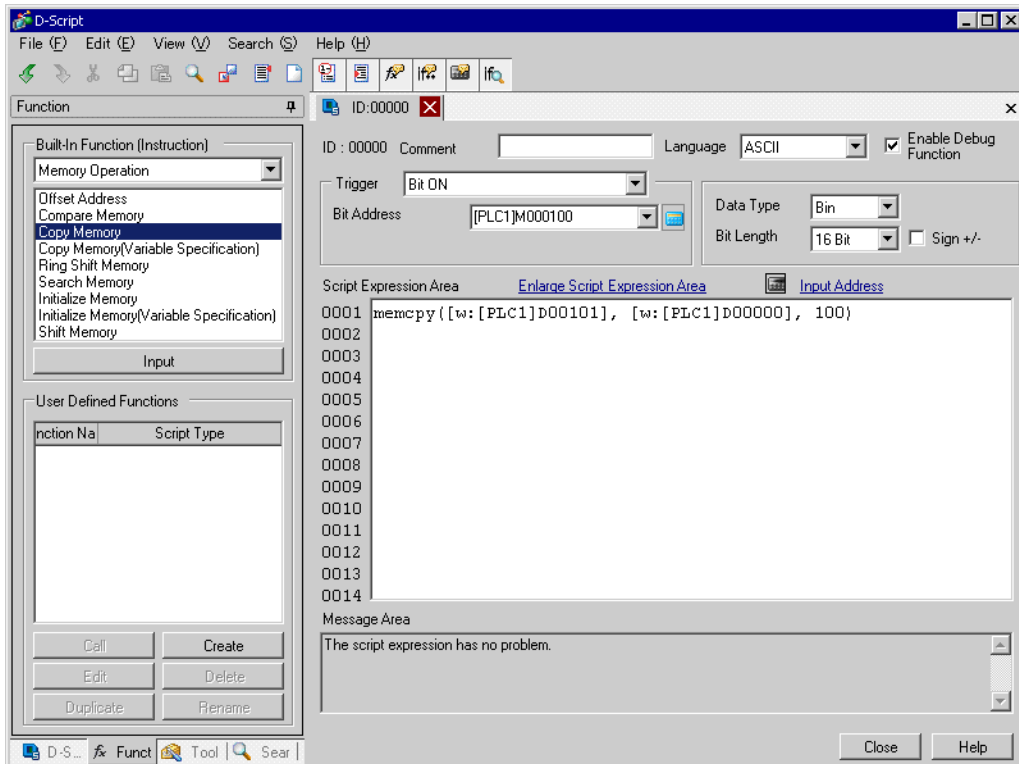
20.4 Displaying an Alarm When an Error Occurs

| NOTE | • Please refer to the settings guide for details.
☞ "20.8.1 D-Script/Common [Global D-Script] Settings Guide" (page 20-53) |

Action

The temperature management system detects an error bit from the connected device and displays alarm messages when the temperature information storage address (D200) rises to 70 degrees C or higher, or falls to 30 degrees C or lower. Also, this script counts the number of detected errors.
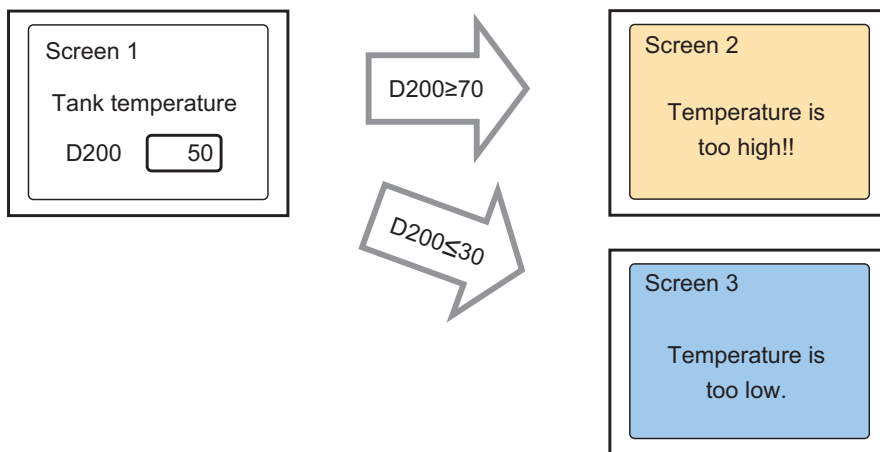
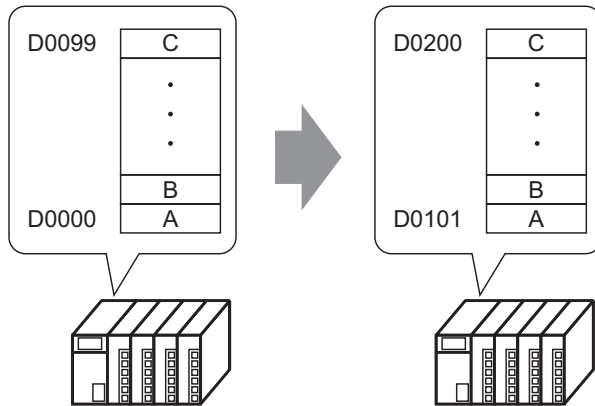The address that counts each time D200 rises to 70 degrees C or higher and stores the number of times:LS0300
The address that counts each time D200 falls to 30 degrees C or lower and stores the number of times:LS0301
The address that stores the screen number to display an alarm screen:LS0302

Commands Used

Command	Function Summary
if ( )	When a condition enclosed becomes true, the process following the "if ( )" statement is executed.
greater than or equal to (>=)	True if N1 is more than or equal to N2 (N1 >= N2).
Assignment (=)	Assigns the right side value to the left side.
Addition (+)	Adds a constant to a Word device's data.
less than or equal to (<=)	True if N1 is less than or equal to N2 (N1 <= N2).

Triggered Condition

In [Trigger Condition], select [Bit ON], and set the [Bit Address] to M000100.

Completed Script

```
Script Expression Area     Enlarge Script Expression Area      Input Address
0001 if([w:[PLC1]D00200]>=70)                             //When temp is greater than 70 degrees
0002 {
0003     [w:[#INTERNAL]LS0302]=100                        //Greater than 70 degrees alarm screen number
0004     [w:[#INTERNAL]LS0300]=[w:[#INTERNAL]LS0300]+1    //Increase error count
0005 }
0006 endif
0007
0008 if([w:[PLC1]D00200]>=30)                             //When temp is greater than 30 degrees
0009 {
0010     [w:[#INTERNAL]LS0302]=101                        //Greater than 30 degrees alarm screen number
0011     [w:[#INTERNAL]LS0301]=[w:[#INTERNAL]LS0301]+1    //Increase error count
0012 }
0013 endif
0014
```

Creation Procedure

1 From the [Parts] menu click [D-Script (R)] or click .
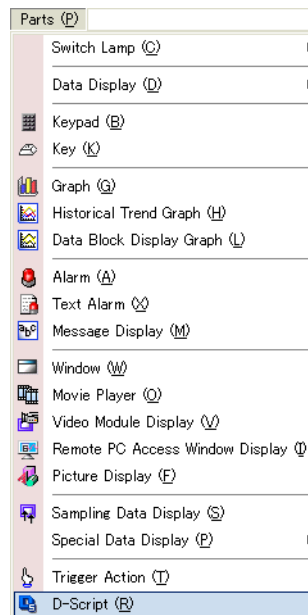
2 Click [Create]. The IDs for existing scripts are displayed in the [D-Script List].
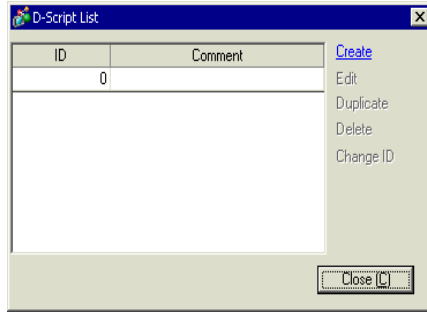
3 The [D-Script] dialog box is displayed.
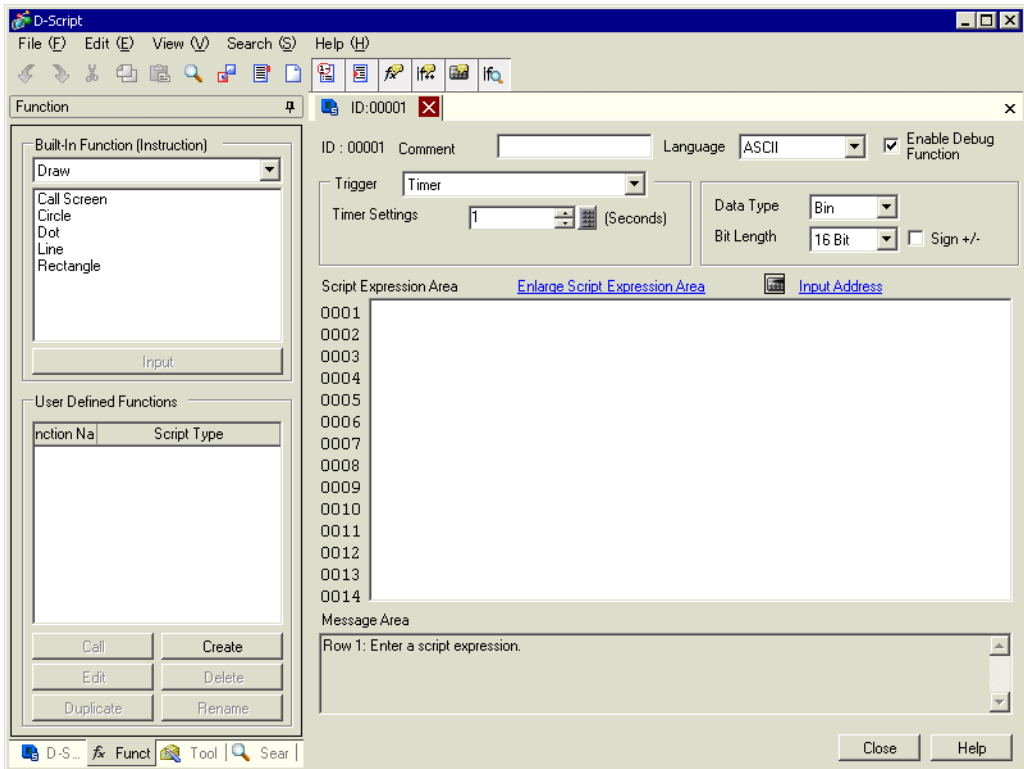
4 In [Comment], enter "Alarm Display".
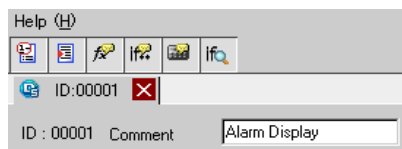
5 In [Trigger Conditions], select [Bit ON], and specify the [Bit Address] as M00100.

6 Create a program in the Action area by entering Functions, Statements, and Expressions. The setting is complete.

```
Script Expression Area        Enlarge Script Expression Area        Input Address
0001  if([w:[PLC1]D00200]>=70)                              //When temp is greater than 70 degrees
0002  {
0003      [w:[#INTERNAL]LS0302]=100                         //Greater than 70 degrees alarm screen number
0004      [w:[#INTERNAL]LS0300]=[w:[#INTERNAL]LS0300]+1     //Increase error count
0005  }
0006  endif
0007
0008  if([w:[PLC1]D00200]>=30)                              //When temp is greater than 30 degrees
0009  {
0010      [w:[#INTERNAL]LS0302]=101                         //Greater than 30 degrees alarm screen number
0011      [w:[#INTERNAL]LS0301]=[w:[#INTERNAL]LS0301]+1     //Increase error count
0012  }
0013  endif
0014
```

NOTE	• When selecting text, press the [Ctrl] Key + [Shift] Key + [->] Key or [<-] Key to select the entire text block. • Press the [Ctrl] Key + [F4] Key to close the currently selected display. • Press the [Esc] Key to overwrite the script or to delete and exit.

20.5   Communicating with Unsupported Peripheral Devices

■ Action

This example creates an extended script to output the data read from a barcode connected to the USB to a serial printer connected to COM1.
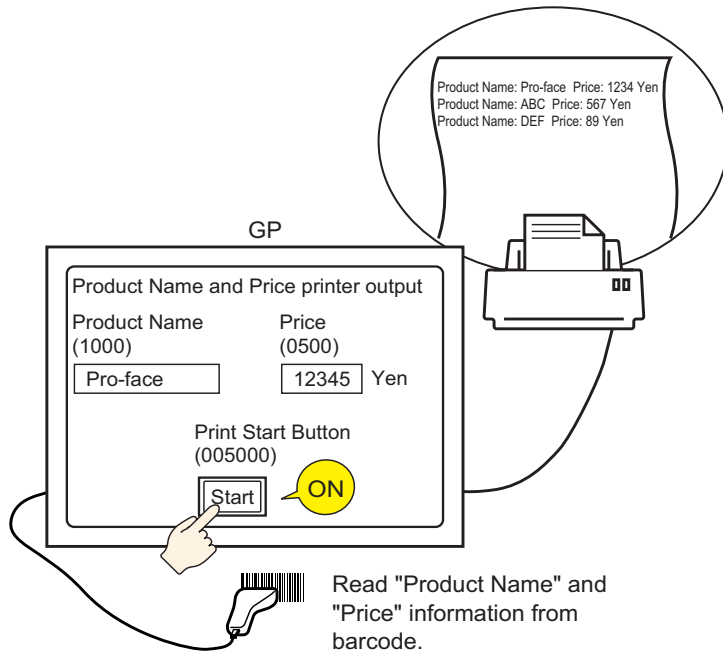
■ **Extended Script Structure**

Extended Scripts are scripts used for communicating between the GP internal Serial Port and connected input/output devices.

For Extended Script data management, as shown in the following picture, data is stored in databuf0 to databuf3 via the Send/Receive Buffer. Databuf is not divided by address, so store the data in internal memory before editing the data on the device/PLC.
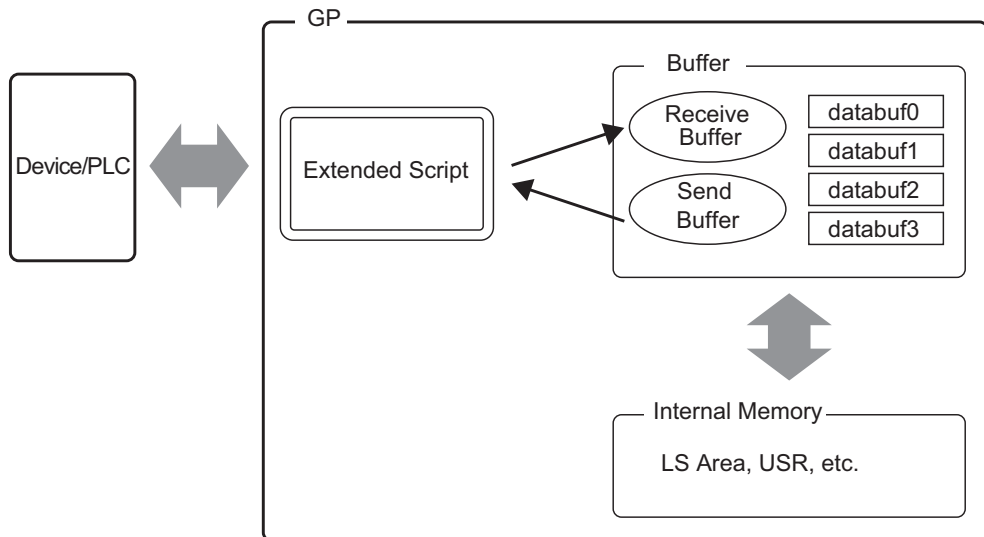
Receive Buffer/Send Buffer

For communication with the device/PLC, this acts as a bit memory space which distinguish sent and received data in real time.

databuf0-databuf3

These are byte (8-bit) memory spaces used for data storage. The buffer size is 1 KB.

■ Procedure to Create Scripts

```
Main function
   Decides whether or not to start the printing when the Printer
   Start Button turns ON.
```

↓

```
INIT (User Defined Function)
   Configures the Send Buffer, Receive Buffer, and Error
   initialization.
```

↓

```
PINIT (User Defined Function)
   Initializes the printer.
```

↓

```
Strset (User Defined Function)
   Changes the data format in order to transfer print data to the
   printer.
```

↓

```
Print (User Defined Function)
   Appends a "line feed" to allow for continuous printer output, and
   transfers print data to the printer.
```

■ **Flow Chart**

(1) Main Process

```
     ╭──────────────────────╮
     │    Main Process      │
     ╰──────────────────────╯
                │
                ▼
          ◇ Communication Start Bit
            ON ◇ ───── No ──────┐
                │               │
               Yes              ▼
                │        ┌───────────────────┐
                ▼        │ Communication Start Bit
          ◇ Receive Data │        OFF        │
            Exist ◇ ── No └───────────────────┘
                │               │
               Yes              │
                ▼               │
        ┌─────────────────┐     │
        │ Extended SIO    │     │
        │ Initialization  │     │
        └─────────────────┘     │
                │               │
                ▼               │
        ┌─────────────────┐     │
        │ Printer         │     │
        │ Initialization  │     │
        └─────────────────┘     │
                │               │
                ▼               │
        ┌─────────────────┐     │
        │ Execution of Data│    │
        │ Transmission     │    │
        └─────────────────┘     │
                │               │
                └───────────────┘
```

(2) Initialization Function (INT)

```
     ╭──────────────────────╮
     │    Initialization    │
     ╰──────────────────────╯
                │
                ▼
        ┌─────────────────┐
        │ Clear Send Buffer│
        └─────────────────┘
                │
                ▼
        ┌─────────────────┐
        │ Clear Receive Buffer│
        └─────────────────┘
                │
                ▼
        ┌─────────────────┐
        │   Clear Error   │
        └─────────────────┘
                │
                ▼
     ╭──────────────────────╮
     │      Finish          │
     ╰──────────────────────╯
```

(3) Printer Initialization Function (PINIT)

```
┌─────────────────────────┐
│  Printer Initialization │
└─────────────────────────┘
            │
            ▼
┌─────────────────────────┐
│  Set the printer        │
│  initialization data to │
│  the data buffer        │
└─────────────────────────┘
            │
            ▼
┌─────────────────────────┐
│  Send the Data          │
└─────────────────────────┘
            │
            ▼
┌─────────────────────────┐
│  Finish                 │
└─────────────────────────┘
```
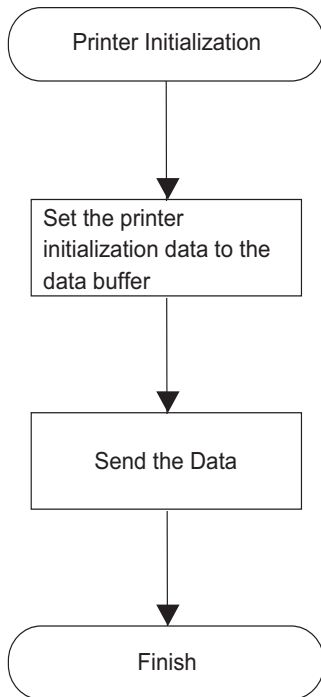
(4) String Function (Strset)

```
┌─────────────────────────┐
│  Send                   │
└─────────────────────────┘
            │
            ▼
┌─────────────────────────┐
│  Set the header string  │
│  to the data buffer     │
└─────────────────────────┘
            │
            ▼
        ╱ Loop 1 ╲
            │
            ▼
┌─────────────────────────┐
│  Convert the data set   │
│  to the address from    │
│  binary values to ASCII │
│  codes                  │
└─────────────────────────┘
            │
            ▼
        30 characters of
        data converted?
           Loop 1
            │
            ▼
┌─────────────────────────┐
│  Set the footer string  │
│  to the data buffer     │
└─────────────────────────┘
            │
            ▼
┌─────────────────────────┐
│  Finish                 │
└─────────────────────────┘
```

(5) Send Function (Print)

```
        ╭─────────────────────╮
        │        Send         │
        ╰─────────────────────╯
                  │
                  ▼
      ┌─┬─────────────────┬─┐
      │ │  Call Send Data │ │
      └─┴─────────────────┴─┘
                  │
                  ▼
      ┌───────────────────────┐
      │  Set new-line data    │
      │  to the data buffer   │
      └───────────────────────┘
                  │
                  ▼
      ┌───────────────────────┐
      │     Send the Data     │
      └───────────────────────┘
                  │
                  ▼
        ╭─────────────────────╮
        │        Finish       │
        ╰─────────────────────╯
```
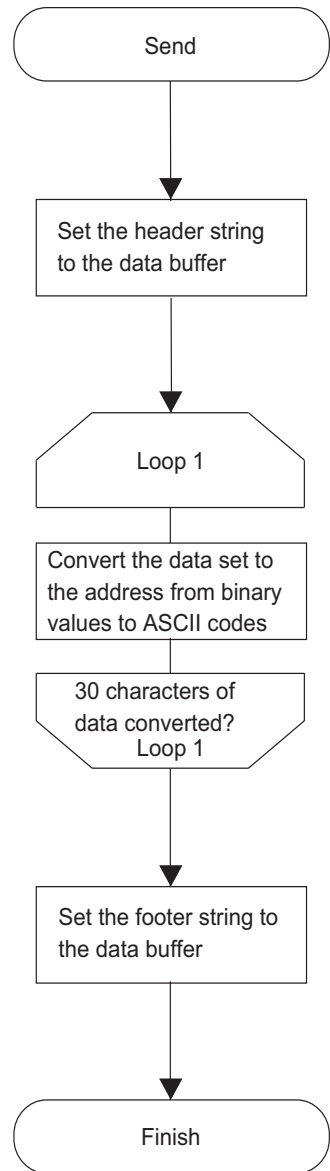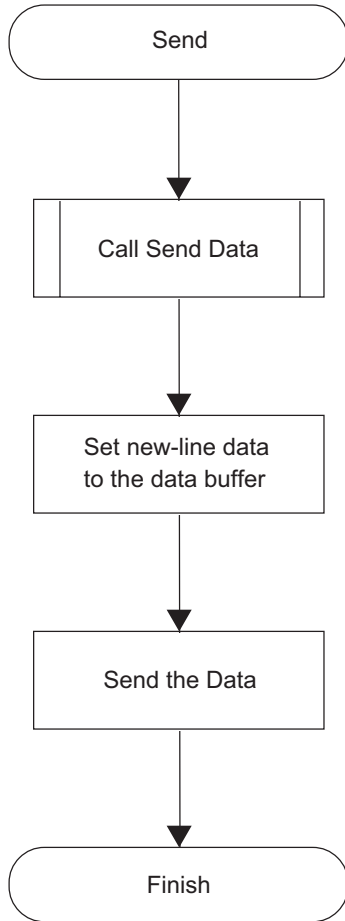
■ Script Function Summary

◆ Main Function

Completed Script
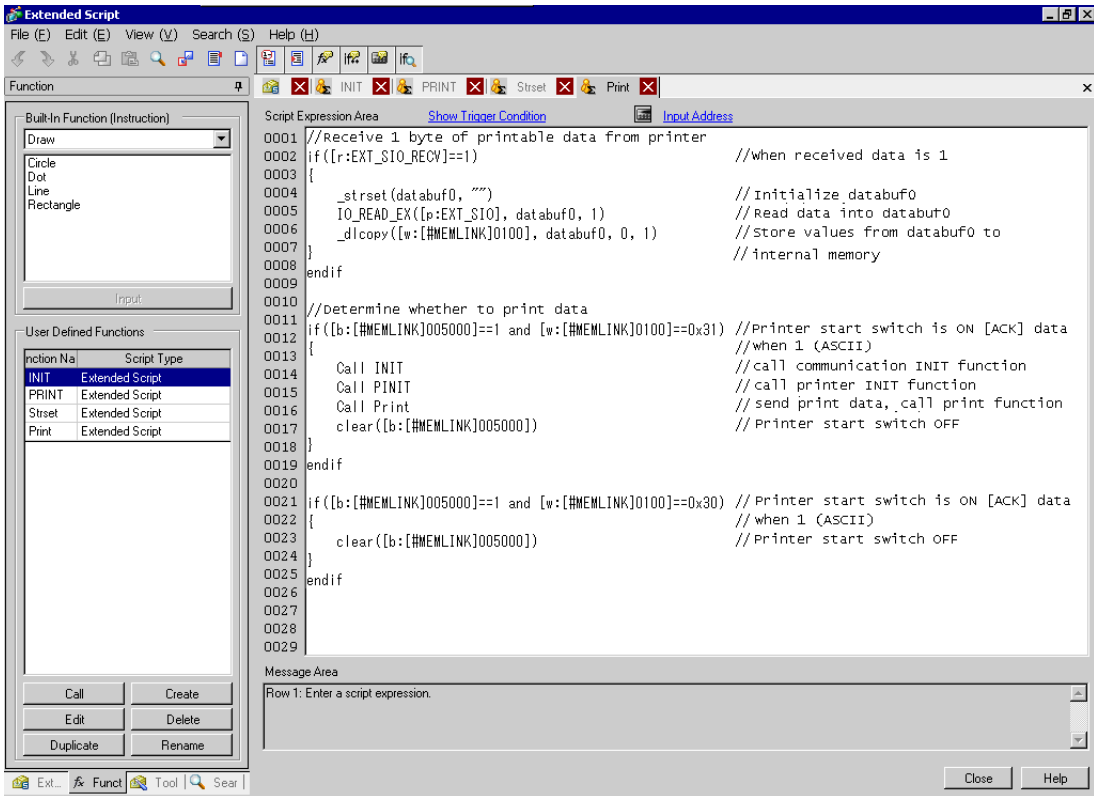
Function Summary

When the Printer Start Button (internal memory 005000) turns ON, decides whether or not to start printing from the 1st byte of Print Permit data.

The Print Permit data performs the following actions as an example of the printer specifications.

Print Preparation OK: Send 0x31 (ASCII code "1") to the device/PLC.
Print Preparation Invalid: Send 0x30 (ASCII code "0") to the device/PLC.

The GP receives the Print Permit data in databuf0 and this data is moved to accessible internal memory 100 with the following script handling.

When internal memory 100 = 0x31 (ASCII code for the value "1"), printing starts. When internal memory is 0x30 (ASCII code for "0") it returns to the beginning and repeats this process until the 0x31 data is received.
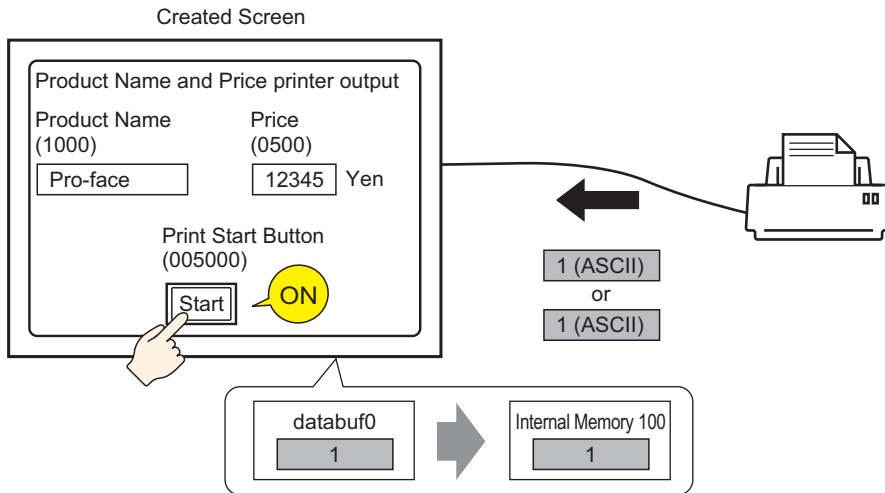
Created Screen

Product Name and Price printer output

Product Name (1000)

Price (0500)

Pro-face

12345 Yen

Print Start Button (005000)

Start ON

1 (ASCII)

or

1 (ASCII)

databuf0

1

Internal Memory 100

1

◆ INIT (User Defined Function)

Completed Script

```
Script Expression Area    Enlarge Script Expression Area    Input Address
0001 [c:EXT_SIO_CTRL00]=1    //Send buffer clear
0002 [c:EXT_SIO_CTRL01]=1    //Receive buffer clear
0003 [c:EXT_SIO_CTRL02]=1    //Error buffer clear
0004
```

Function Summary

Configure the Send Buffer, Receive Buffer, and Error initialization.

◆ PINIT (User Defined Function)

Completed Script

```
Script Expression Area    Enlarge Script Expression Area    [icon] Input Address
0001 //Printer initialization(ESC/P command"ESC+@")
0002
0003 _strset(databuf0,"")       //Clear databuf0
0004 _strset(databuf0, 0*1B)    //Set ASCII code"ESC"
0005 _strset(databuf1,"")       //Clear databuf1
0006 _strset(databuf1, 0*40)    //Set ASCII code"@"
0007 _strset(databuf0,databuf1)  //Add databuf1 to end of databuf0
0008 _strlen([t:0000],databuf0)  //Store data length to temporary address
0009                                                          --
0010 //Send data over serial port
0011
0012 IO_WRITE_EX([p:EXT_SIO],databuf0,[t:0000]) //Send databuf0,amount defined by temporary address value
0013
```

Function Summary

Initializes the printer. Send the ESC/P command "ESC+@" to the printer.

◆ Strset (User Defined Function)

Completed Script

```
Script Expression Area    Enlarge Script Expression Area    [icon] Input Address
0001 //String example, add "Price:" and "$"
0002 _strset(databuf0, "")                          //Initialize databuf0
0003 _strset(databuf0, "Price:")                    //Store text "Price:" to databuf0
0004 _bin2decasc(databuf0,[w:[#MEMLINK]0500])       //Convert value to string and store in databuf1
0005 _strcat(databuf0, databuf1)                    //Add databuf1 to end of databuf0
0006 _strset(databuf1, "")                          //Initialize databuf1
0007 _strset(databuf1, "$")                         //Store text "$" to databuf1
0008 _strcat(databuf0, databuf1)                    //Add databuf1 to end of databuf0
0009
0010 //Initialize temporary address
0011 [t:0001]=0
0012 [t:0002]=0
0013
0014 //Store to internal memory word units, consecutive characters into byte units(30 characters)
0015 loop()
0016 {
0017  [w:[#MEMLINK]2000]#[t:0002]=[w:[#MEMLINK]1000]#[t:0001]>>8      //Store top byte into bottom byte
0018  [w:[#MEMLINK]2001]#[t:0002]=[w:[#MEMLINK]1000]#[t:0001]& 0*FF  //Erase top byte and store in next address
0019  [t:0001]=[t:0001]+1                                            //Address offset + 1
0020  [t:0002]=[t:0002]+2                                            //Address offset + 2
0021  if([t:0001]==15)                                               //Store 2 words into 2 byte and repeat 15 times
0022  {
0023   break
0024  }
0025  endif
0026 }
0027 endloop
0028 _ldcopy(databuf2, [w:[#MEMLINK]2000],30)        //Store internal memory 2000~2030 to data buffer as characters
0029
0030 //Add string "Item:"
0031 _strset(databuf1, "")                          //Initialize databuf1
0032 _strset(databuf1, "Item:")                     //Store string "Item:" into databuf1
0033 _strcat(databuf1, databuf2)                    //Add databuf1 to end of databuf0
0034
0035 //Add Item and Price strings
0036 _strcat(databuf1, databuf0)                    //Add databuf0 to end of databuf1
```
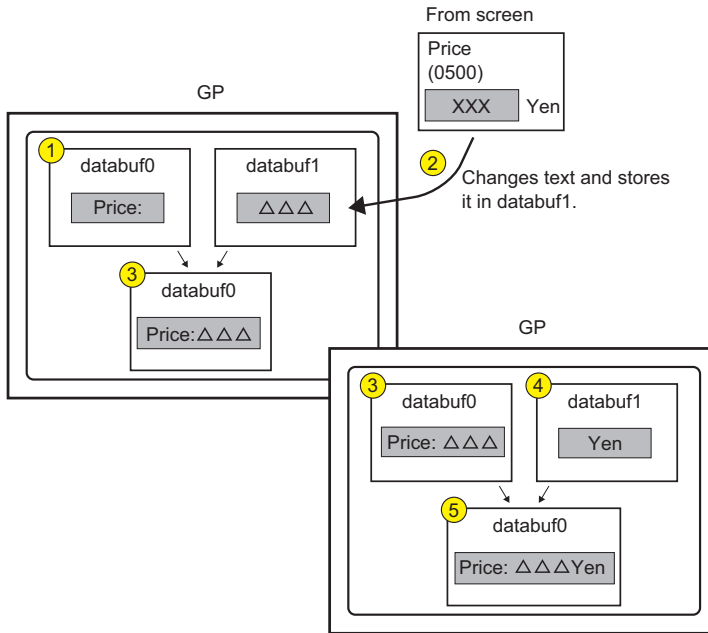
Function Summary

1 Append the text "Price:" and "Yen" to the price data stored internal memory 0500.

2 Change the data format in order to send print data to the printer. Divide the string data (Product Name) stored sequentially in internal memory 1000 into byte units, and store into internal memory 2000 to 2030 as low order byte string data. Use the function _ldcopy and store the data in databuf2 in order of the consecutive word address's lowest byte.

NOTE

• The _ldcopy function takes data stored as Words, and stores only the lower order bytes in the buffer, while higher order byte data is ignored.

From screen

Product Name (1000)
Pro-face
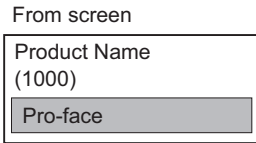
Stores data in byte units.

Internal Memory (16 bit)

2000	0	P
2001	0	r
2002	0	o
2003	0	-
2004	0	f
2005	0	a
2006	0	c
2007	0	e
2008	0	0
:	:	:
2030	0	0

Stores the low order byte to databuf 2.

databuf2 (8 bit)

databuf2[0]	P
databuf2[1]	r
databuf2[2]	o
databuf2[3]	-
databuf2[4]	f
databuf2[5]	a
databuf2[6]	c
databuf2[7]	e
databuf2[8]	0
	:
databuf2[30]	0

3 Append the text "Product Name:" and "Price" to databuf2.

GP

① databuf1

Product Name:

② databuf2

Pro-face

③ databuf1

Product Name: Pro-face

GP

③ databuf1

Product Name: Pro-face

④ databuf0

Price: XXX Yen

⑤ databuf1

Product Name: Pro-face Price: XXX Yen

◆ Print (User Defined Function)

Completed Script

```
Script Expression Area     Enlarge Script Expression Area        Input Address
0001 Call Street                        //Call string data function
0002 _strset(databuf0,"")              //Clear databuf1
0003
0004 //Text delimiter
0005
0006 _strset(detabuf0, 0*0d)           //Return to start of row
0007 _strset(databuf1, databuf0)       //Add databuf1 to end of databuf0
0008 _strset(databuf0, "")             //Clear databuf1
0009 _strset(databuf0, 0*0a)           //New line
0010 _strset(databuf1, databuf0)       //Add databuf1 to end of databuf0
0011
0012 _strset([t:0000], databuf1)        //Store data length to temporary address
0013
0014 //Send data over serial port
0015
0016 IO_WRITE_EX([p:EXT_SIO], databuf1, [t:0000] //Send databuf0, amount defined by temporary address value
0017
```

Function Summary

1 Append a "line feed" to allow for continuous printer output.

GP-Pro EX Reference Manual   20-35
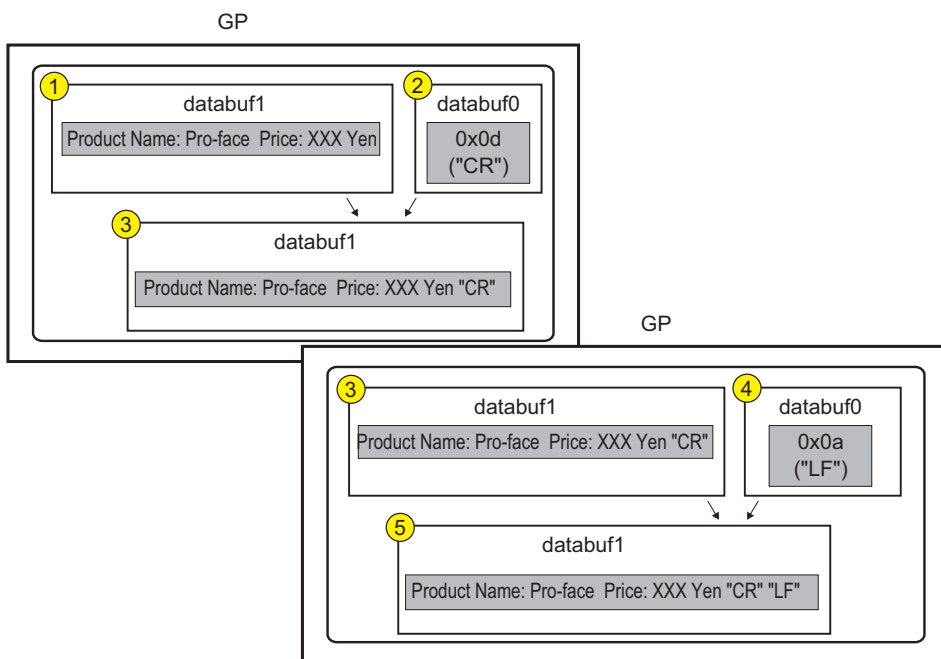
2 Set the print data to the printer.
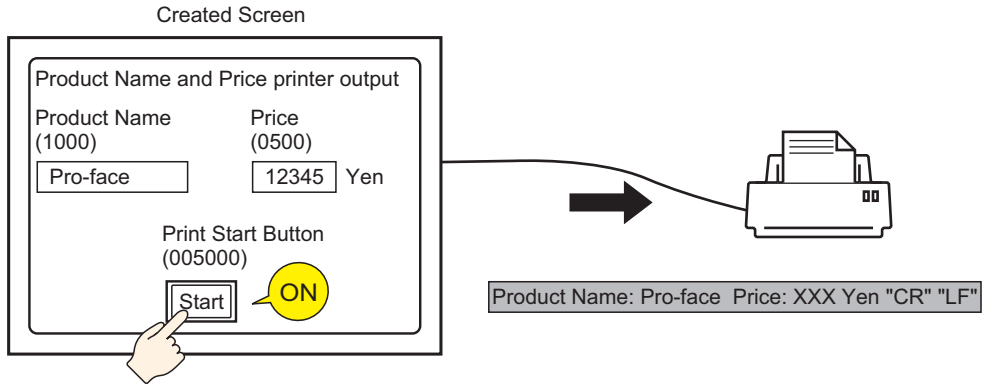
Created Screen

Product Name and Price printer output

Product Name
(1000)

Price
(0500)

Pro-face

12345   Yen

Print Start Button
(005000)

Start   ON

Product Name: Pro-face Price: XXX Yen "CR" "LF"

■ Commands Used

Command	Function Summary
if ( )	When a condition enclosed becomes true, the process following the "if ( )" statement is executed.
Label Settings [r:EXT_SIO_RECV]	Shows the quantity of data (number of bytes) that has been received at that time. The received data size is a read-only feature.
Equivalent (==)	True if N1 is equal to N2 (N1 = N2).
Text Settings (_strset)	Stores a fixed string in the data buffer.
Extended Receive (IO_READ_EX)	Receives data of the size indicated in Received Data Size (bytes) from the Extended SIO and stores it in the data buffer.
From Data Buffer to Internal Device (_dlcopy)	Each byte of string data stored in the offset of the data buffer is copied to the LS area according to the number of strings.
Label Settings [c:EXT_SIO_CTRL **]	This control variable is used to clear the Send buffer, Receive buffer, and error status.
Connect Text (_strcat)	Concatenates a character string or character code with the text buffer.
Text Length (_strlen)	Obtains the length of the stored string.
Extended Send (IO_WRITE_EX)	Sends the data in the data buffer with Extended SIO according to the size of Number of Send Bytes.
Assignment (=)	Assigns the right side value to the left side.
Addition (+)	Adds a constant to a Word device's data.
Numeric Value Decimal String Conversion (_bin2decasc)	This function is used to convert an integer to a decimal string.
From Internal Device To Data Buffer (_ldcopy)	The data of the string stored in the LS area is copied to the data buffer according to the number of strings in a byte-by-byte transfer.

■ Creation Procedure

1 Set the Extended Script used to communicate. From the [Project (F)] menu, point to [System Settings (C)] and select [Script Settings]. Set the [Type] to [Extended Script].
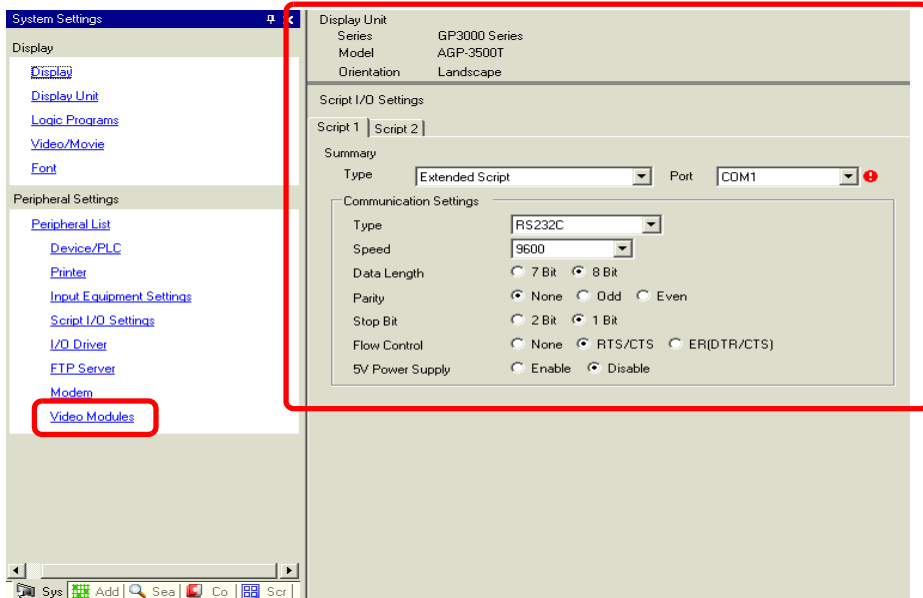
There are two tabs for the script settings.

Set the [Port] to COM1 or COM2. Set the [Communication Settings] to match the Extended SIO.

2 From the [Common Settings (R)] menu, select [Extended Script Settings (E)].
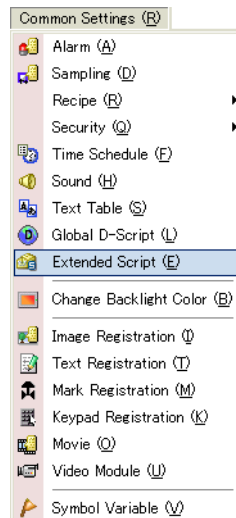
3 Register "INIT" as a User-Defined Function. Click the [Function] tab and click the user-defined function frame's [Create] button.

4 Input [INIT] as the function name, click [OK]. The following screen appears.

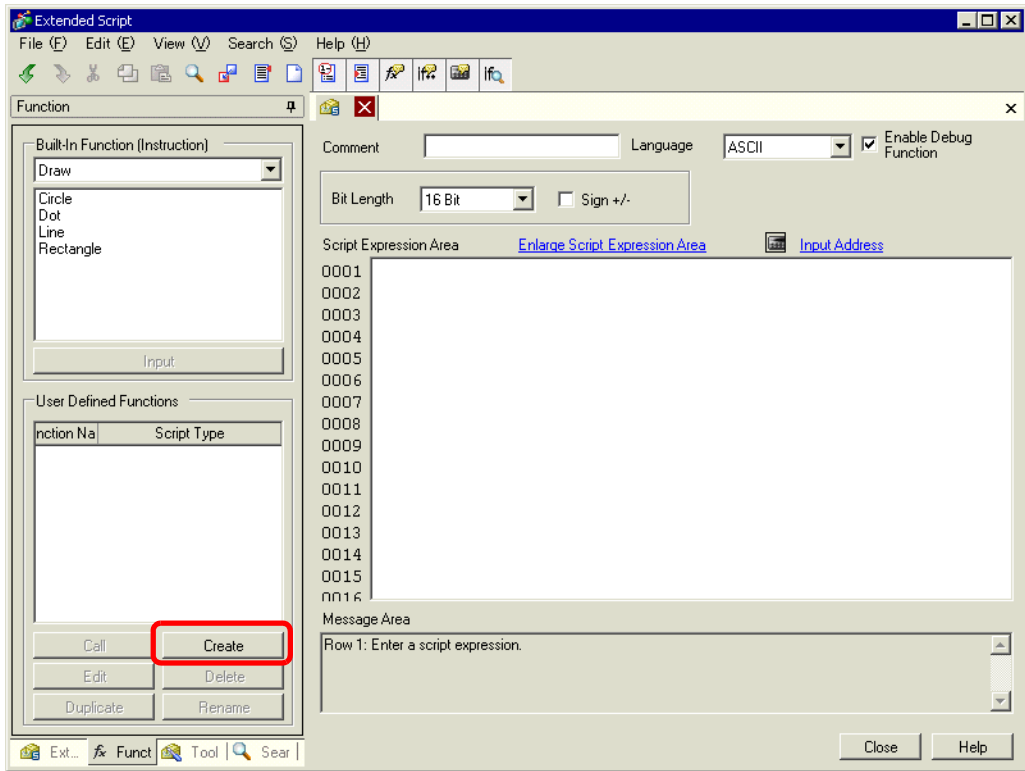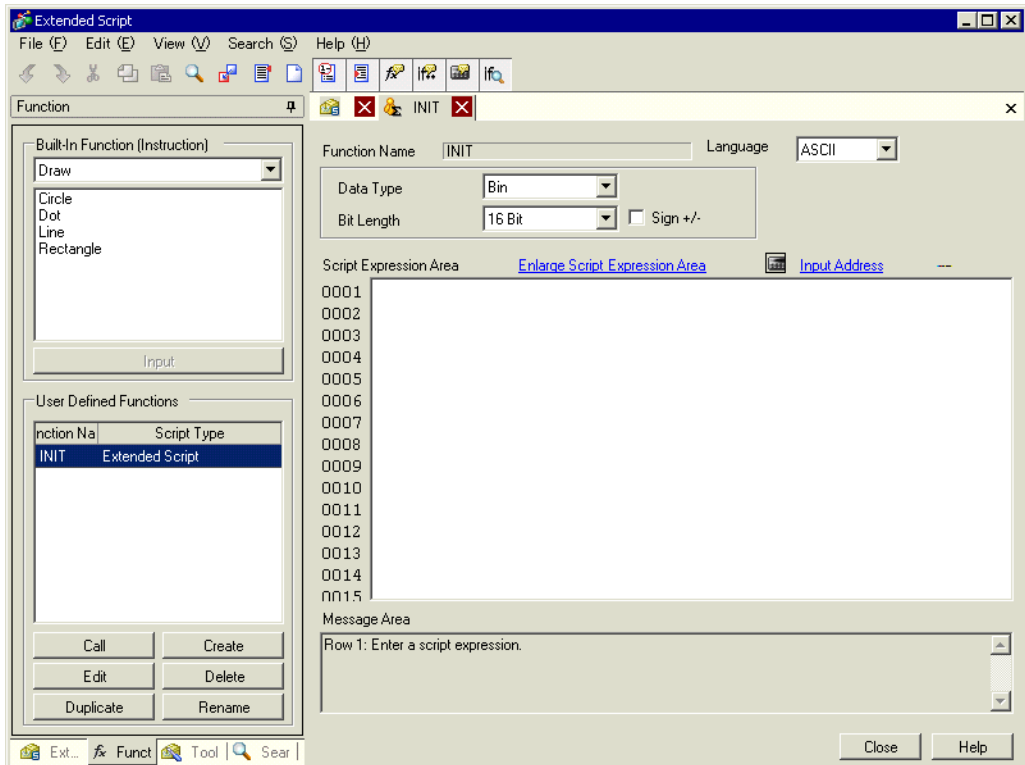5 Create a script in the Execution Expression with Commands, Statements, and Constant input.

```
Script Expression Area     Enlarge Script Expression Area      Input Address
0001 [c:EXT_SIO_CTRL00]=1     //Send buffer clear
0002 [c:EXT_SIO_CTRL01]=1     //Receive buffer clear
0003 [c:EXT_SIO_CTRL02]=1     //Error buffer clear
0004
```

6 In the same manner, register "PINIT" as a User-Defined Function. Enter [PINIT] as the function name and create the following script in Execution Expression.

```
Script Expression Area     Enlarge Script Expression Area      Input Address
0001 //Printer initialization(ESC/P command"ESC+@")
0002
0003 _strset(databuf0,"")          //Clear databuf0
0004 _strset(databuf0, 0*1B)       //Set ASCII code"ESC"
0005 _strset(databuf1,"")          //Clear databuf1
0006 _strset(databuf1, 0*40)       //Set ASCII code"@"
0007 _strset(databuf0,databuf1)    //Add databuf1 to end of databuf0
0008 _strlen([t:0000],databuf0)    //Store data length to temporary address
0009                                                              --
0010 //Send data over serial port
0011
0012 IO_WRITE_EX([p:EXT_SIO],databuf0,[t:0000]) //Send databuf0,amount defined by temporary address value
0013
```

7 In the same manner, register "Strset" as a User-Defined Function. Enter [Strset] as the function name and create the following script in Execution Expression.

```
Script Expression Area     Enlarge Script Expression Area      Input Address
0001 //String example, add "Price:" and "$"
0002 _strset(databuf0, "")                    //Initialize databuf0
0003 _strset(databuf0, "Price:")              //Store text "Price:" to databuf0
0004 _bin2decasc(databuf0,[w:[#MEMLINK]0500]) //Convert value to string and store in databuf1
0005 _strcat(databuf0, databuf1)              //Add databuf1 to end of databuf0
0006 _strset(databuf1, "")                    //Initialize databuf1
0007 _strset(databuf1, "$")                   //Store text "$" to databuf1
0008 _strcat(databuf0, databuf1)              //Add databuf1 to end of databuf0        (1)
0009
0010 //Initialize temporary address
0011 [t:0001]=0
0012 [t:0002]=0
0013
0014 //Store to internal memory word units, consecutive characters into byte units(30 characters)
0015 loop()
0016 {
0017  [w:[#MEMLINK]2000]#[t:0002]=[w:[#MEMLINK]1000]#[t:0001]>>8    //Store top byte into bottom byte
0018  [w:[#MEMLINK]2001]#[t:0002]=[w:[#MEMLINK]1000]#[t:0001]& 0*FF  //Erase top byte and store in next address
0019  [t:0001]=[t:0001]+1                                          //Address offset + 1
0020  [t:0002]=[t:0002]+2                                          //Address offset + 2
0021  if([t:0001]==15)                                             //Store 2 words into 2 byte and repeat 15 times  (2)
0022  {
0023   break
0024  }
0025  endif
0026 }
0027 endloop
0028 _ldcopy(databuf2, [w:[#MEMLINK]2000],30)        //Store internal memory 2000~2030 to data buffer as characters
0029
0030 //Add string "Item:"
0031 _strset(databuf1, "")                    //Initialize databuf1
0032 _strset(databuf1, "Item:")               //Store string "Item:" into databuf1
0033 _strcat(databuf1, databuf2)              //Add databuf1 to end of databuf0        (3)
0034
0035 //Add Item and Price strings
0036 _strcat(databuf1, databuf0)              //Add databuf0 to end of databuf1
```

8 In the same manner, register "Print" as a User-Defined Function. Enter [Print] as the function name and create the following script in Execution Expression.

```
Script Expression Area    Enlarge Script Expression Area    Input Address

0001 Call Street                          //Call string data function
0002 _strset(databuf0,"")                 //Clear databuf1
0003
0004 //Text delimiter
0005
0006 _strset(databuf0, 0*0d)              //Return to start of row
0007 _strset(databuf1, databuf0)          //Add databuf1 to end of databuf0
0008 _strset(databuf0, "")                //Clear databuf1
0009 _strset(databuf0, 0*0a)              //New line
0010 _strset(databuf1, databuf0)          //Add databuf1 to end of databuf0
0011
0012 _strset([t:0000], databuf1)          //Store data length to temporary address
0013
0014 //Send data over serial port
0015
0016 IO_WRITE_EX([p:EXT_SIO], databuf1, [t:0000] //Send databuf0, amount defined by temporary address valu
0017
```

9 Create the main script. Create the following script in Execution Expression and the settings are complete.

```
Script Expression Area    Enlarge Script Expression Area    Input Address

0001 //Receive 1 byte of printable data from printer
0002 if([r:EXT_SIO_RECV]==1)                                    //When received data is 1
0003 {
0004     _strset(databuf0,"")                                   //Initialize databuf0
0005     IO_READ_EX([p:EXT_SIO], databuf0, 1)                   //Read data into databuf0
0006 _dlcopy([w:[#MEMLINK]0100], databuf0, 0, 1)                //Store values from databuf0 to internal
0007 }
0008 endif
0009
0010 //Determine whether to print data
0011 if([b:[#MEMLINK]005000]==1 and [w:[#MEMLINK]0100]==0*31)   //Printer start switch is ON [ACK] data
0012 {                                                          //when 1 (ASCII)
0013     Call INIT                                              //call communication INIT function
0014     Call PINIT                                             //call printer INIT function
0015     Call Print                                             //send print data, call print function
0016     clear([b:[#MEMLINK]005000])                            //Printer start switch OFF
0017 }
0018 endif
0019
0020 if([b:[#MEMLINK]005000]==1 and [w:[#MEMLINK]0100]==0*30)   //Printer start switch is ON [ACK] data
0021 {                                                          //when 0 (ASCII)
0022     clear([b:[#MEMLINK]005000])                            //Printer start switch OFF
0023 }
0024 endif
```

NOTE
• When placing the user-defined functions created in steps 3 to 9 into the main script, select the function to be placed and click [Call] on the [Function] tab. The function will be placed using "Call Function Name".
• When selecting text, press the [Ctrl] Key + [Shift] Key + [->] Key or [<-] Key to select the entire text block.
• Press the [Ctrl] Key + [F4] Key to close the currently selected display.
• Press the [Esc] Key to overwrite the script or to delete and exit.

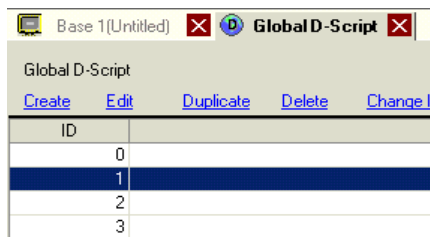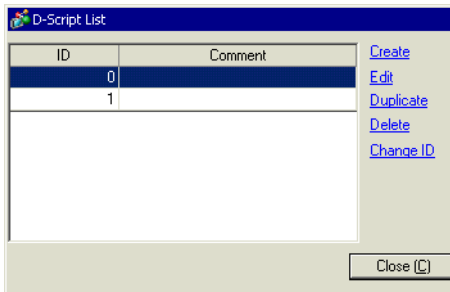## 20.6 Procedure for Creating Scripts

20.6.1 Procedure for Creating D-Scripts/Global D-Scripts
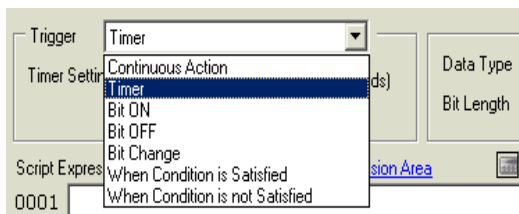
From the [Parts (P)] menu, select [D-Script (R)].	From the [Common Settings (R)] menu, select [Global D-Script Settings (L)].

Click [Create]. To view an existing script, select the ID number and click [Edit], or double-click the ID Number row.
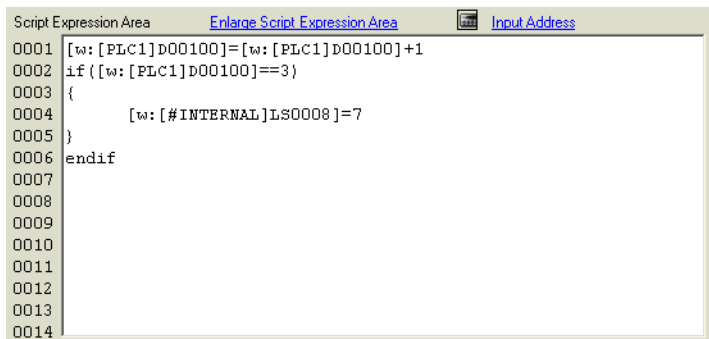

Click [Create]. To view an existing script, select the ID number and click [Edit], or double-click the ID Number row.

Set the trigger condition that causes the script to run. For more information about this function, please refer to "20.7 Triggered Condition Setup" (page 20-46) .

Create the script (Execution Expression). For more information about commands and functions, please refer to "21.13 Command List" (page 21-100) .
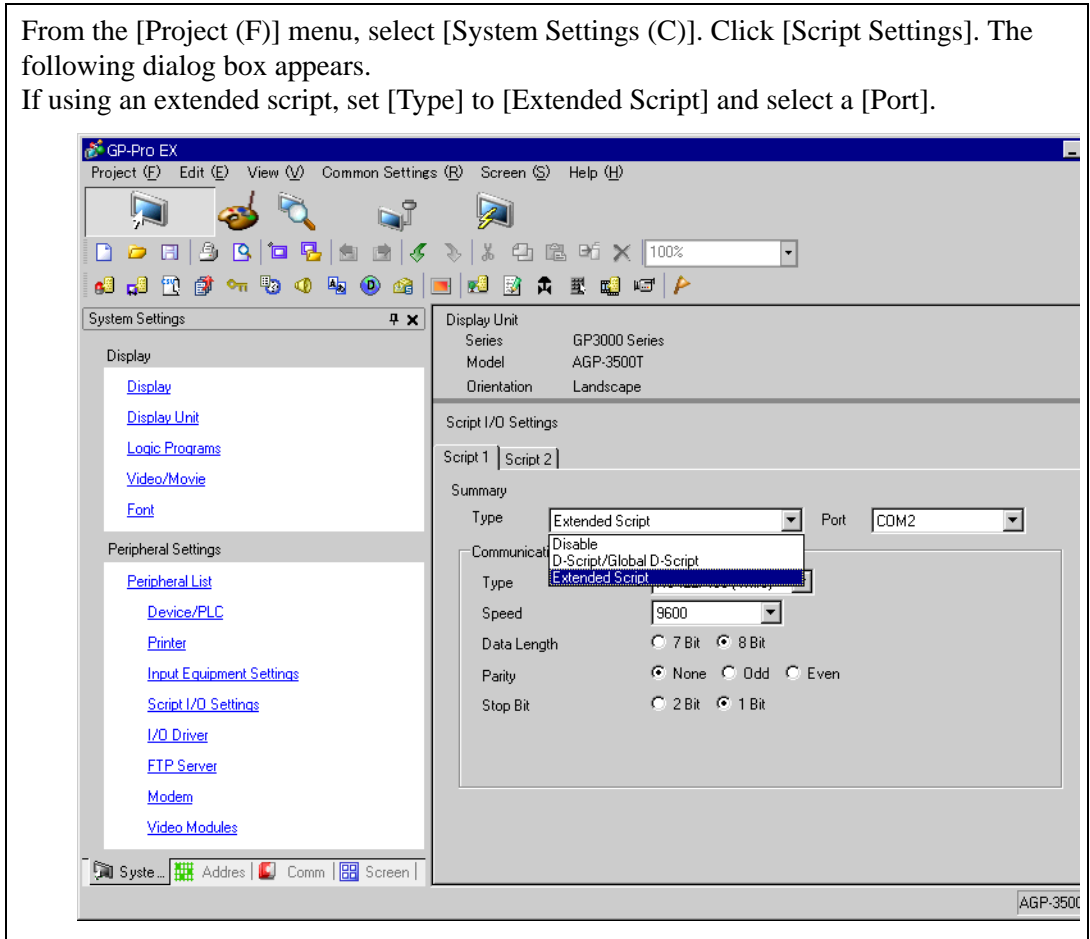
20.6.2 Procedure for Creating Extended Scripts

From the [Project (F)] menu, select [System Settings (C)]. Click [Script Settings]. The following dialog box appears.
If using an extended script, set [Type] to [Extended Script] and select a [Port].

From the [Common Settings (R)] menu, select [Extended Script Settings (E)].

Create the script (Execution Expression). For more information about commands and functions, please refer to "21 Commands and Descriptions" (page 21-1) .

20.6.3 Setting Up User-Defined Functions

Register an existing script as a user-defined function so you can use it within other scripts. The registered function can be used by a D-Script, Global D-Script, or Extended Script.

■ **Setting Procedure**

When creating a new User-Defined Function
Click on [Create]. The User-Defined Function dialog box appears.

When edited a previously registered User-Defined Function
Select the User-Defined Function you want to modify and click [Edit]. The User-Defined Function dialog box appears.
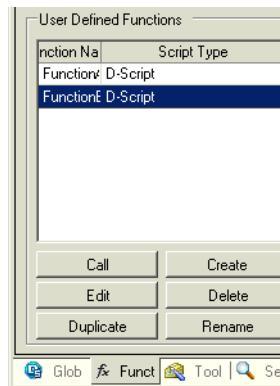
Enter the function name and create the script in the Execution field. Click [OK] to save the user-defined function.
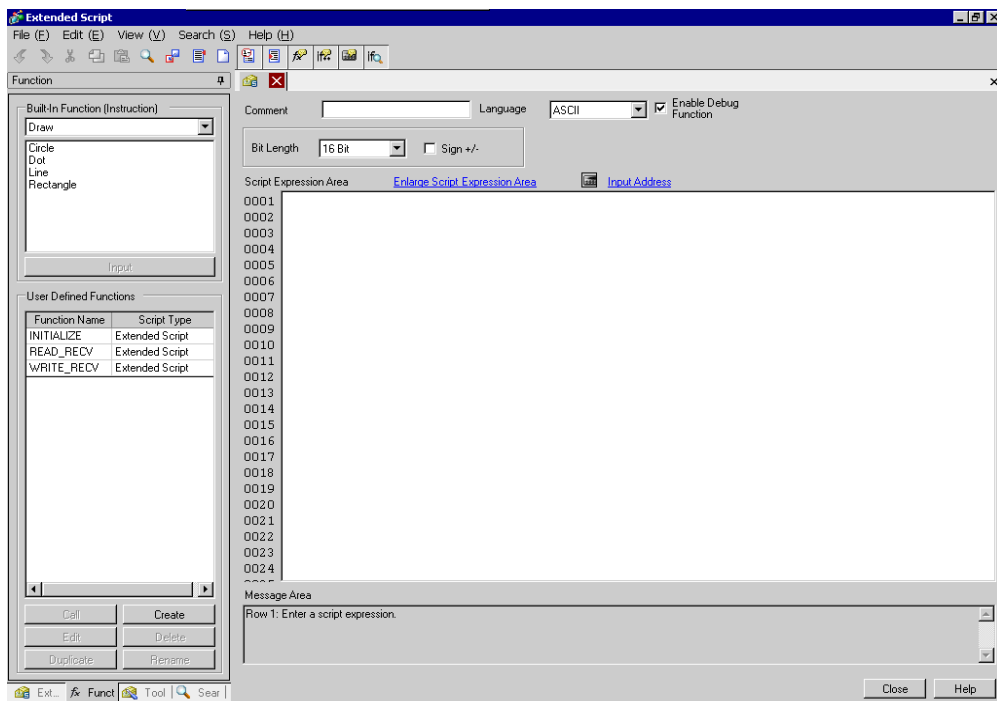
NOTE

• Restrictions apply to Function Names. For more information, see "20.9.3 Restrictions on User-Defined Functions" (page 20-65) .

Select the user-defined function to call, click [Call] and "Call Function Name" will be placed in the Execution field.
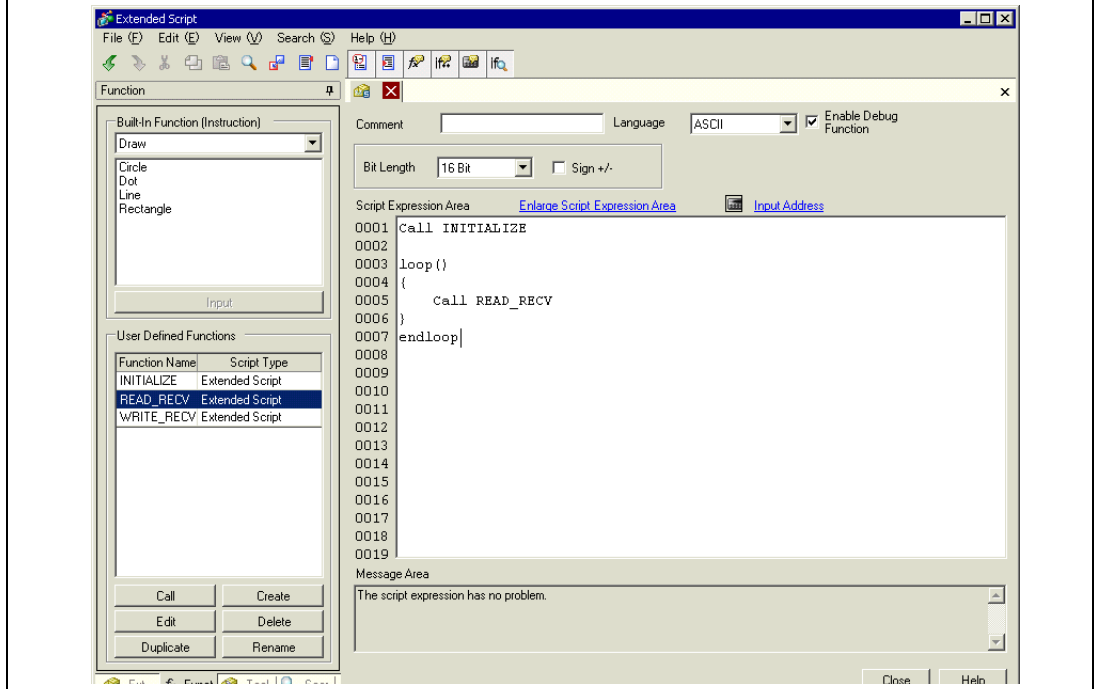
IMPORTANT	• When a user-defined function calls another script, it cannot use functions created in an Extended Script in D-Scripts or Global D-Scripts.

20.7 Triggered Condition Setup

A created script can use any of the following 7 types of trigger conditions.

Setting		Description
Continuous Action		The script is triggered regularly.
Timer		The script is triggered after a designated time elapses.
Bit	Bit ON	When the GP detects the designated bit rise from 0 to 1, the script is triggered.
	Bit OFF	When the GP detects the falling edge of the designated bit, the script is triggered.
	Bit Change	When the GP detects the designated bit rise from 0 to 1 or fall from 1 to 0, the script is triggered.
Condition Expression	When Condition is True	When the GP detects true for a designated expression, the script is triggered.
	When Condition is False	When the GP detects false for a designated expression, the script is triggered.

20.7.1 Continuous Action

Executes each display scan time.

20.7.2 Timer

■ Timer

Each time the designated time elapses, the script is executed one time. The timer duration can be set from 1 to 32,767 seconds.
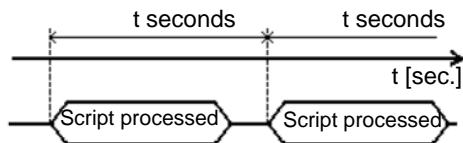
NOTE

- When setting the timer function's time, the time value includes the set time + display scan time error. Also, depending on the time taken to draw a screen item or to printout data, the timer function may be slowed. For more information about the Display Scan Time, please refer to " ■ Restrictions on the Triggered Bit" (page 20-50) .
- When using D-Script, switching the screen causes the timer function to restart counting from 0.

20.7.3 Bit

■ Bit ON

When the GP detects the designated bit address (trigger bit) rise from 0 to 1, the script is triggered.
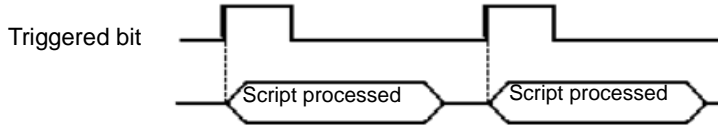
| NOTE | • For the trigger bit's ON/OFF, make sure to leave an interval longer than the communication cycle time or display scan time, whichever is longer. For more information about this function, please refer to | " ■ Restrictions on the Triggered Bit" (page 20-50) . |

■ Bit OFF

When the GP detects the designated bit address (trigger bit) fall from 1 to 0, the script is triggered.

| NOTE | • For the trigger bit's ON/OFF, make sure to leave an interval longer than the communication cycle time or display scan time, whichever is longer. For more information about this function, please refer to " ■ Restrictions on the Triggered Bit" (page 20-50) . |

■ Bit Change

When the GP detects the designated bit address (trigger bit) rise from 0 to 1 or fall from 1 to 0, the script is triggered.
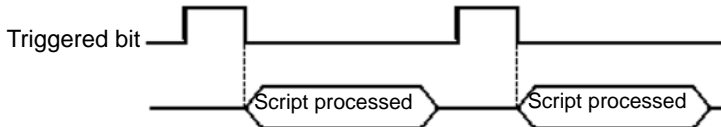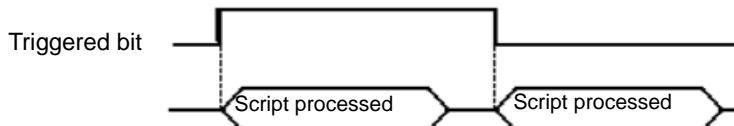

| NOTE | • For the trigger bit's ON/OFF, make sure to leave an interval longer than the communication cycle time or display scan time, whichever is longer. For more information about this function, please refer to " ■ Restrictions on the Triggered Bit" (page 20-50) . |

20.7.4   Condition Expression

■ When Condition is True

When the GP evaluates the trigger condition as true, the script runs one time.

For example, When the Triggered Condition is 100>[D100]>50, the script will run with the following timing.

[False] → [True] is detected, the script executes, and 70 is assigned to D100.

The script does not execute when [True] → [True].

NOTE	• For the Triggered Condition, leave an interval longer than the communication cycle time or display scan time, whichever is longer. For more information about this function, please refer to " ■ Restrictions on the Triggered Bit" (page 20-50) .

■ When Condition is not Satisfied

When the GP detects false for a designated expression in a triggering program, the script is executed once.

When the Triggered Condition is set to 100>[D100]>50, the script will execute with the following timing. [True]->[False] is detected, the script executes, and 20 is assigned to D100 The script does not execute when [False]->[False].

NOTE • For the Triggered Condition, make sure to leave an interval longer than the communication cycle time or display scan time, whichever is longer. For more information about this function, please refer to " ■ Restrictions on the Triggered Bit" (page 20-50) .
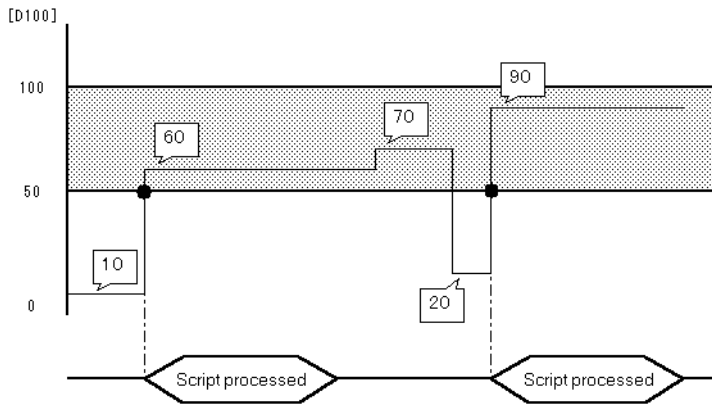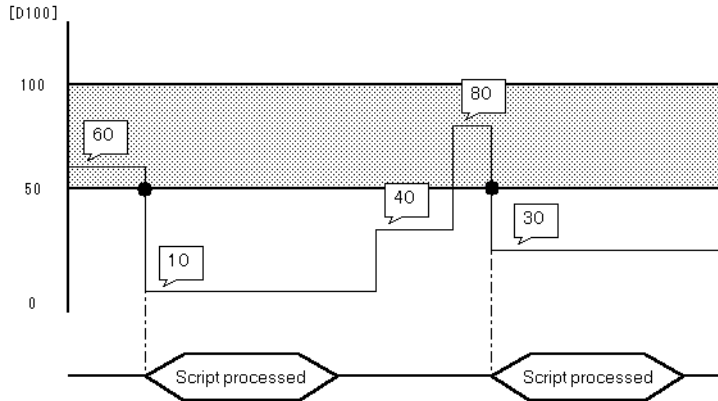
■ Restrictions on the Triggered Bit

- Make sure to leave an interval longer than the communication cycle time for executing write operations onto the connected device. When write operations onto the connected device are executed frequently by using the scan counter of GP internal special relay, communication errors or system errors may result.
- When the bit used for the D-Script Triggered Condition is set for "touch" and that bit turns OFF during D-Script processing, the timing used when pressing the touch area repeatedly can prevent the detection of the bit's rise. The D-Script trigger compares the previously read out value to the currently read out value to determine if the trigger is now "True". However, during a single scan, the value that is stored in the bit address used during the Triggered operation is kept the same, even if the value is changed during execution. The new value is read out only after the next scan begins.

Communication Cycle Time:	The communication cycle time is the time it takes to request and take in data from the GP unit to the PLC. It is stored in the internal device LS2037 as binary data. The unit is milliseconds (ms). There is an error of ±10ms.
Display Scan Time:	The display scan time is the time it takes to display/calculate 1 screen. It is stored in the internal device LS2036 as binary data. The unit is milliseconds (ms). There is an error of ±10ms.

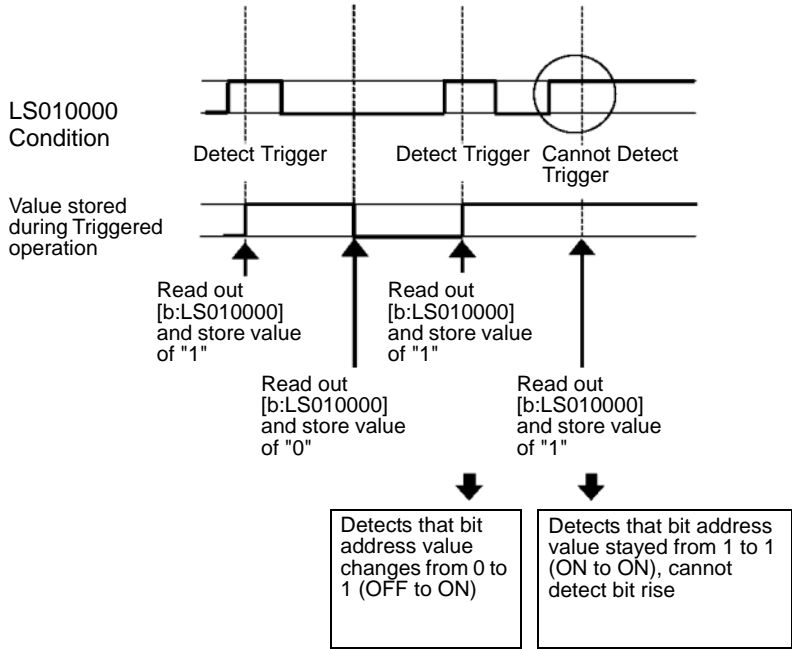For example, When Touch is used to turn ON the trigger bit (LS010000), and D-Script turns the value OFF:

     Triggered Condition: Bit ON [#INTERNAL] LS010000

     Execution Expression: clear ([b:[#INTERNAL] LS010000])

◆ **D-Script Processing Timing Chart**

LS010000
Condition

Detect Trigger   Detect Trigger   Cannot Detect
Trigger

Value stored
during Triggered
operation

Read out
[b:LS010000]
and store value
of "1"

Read out
[b:LS010000]
and store value
of "1"

Read out
[b:LS010000]
and store value
of "0"

Read out
[b:LS010000]
and store value
of "1"

| Detects that bit address value changes from 0 to 1 (OFF to ON) | Detects that bit address value stayed from 1 to 1 (ON to ON), cannot detect bit rise |

For example, if the D-Script touch timing is not used, and only detection is performed, the processing is as follows.

Using an if ( ) statement to detect a trigger:
Use an if statement to determine if a touch operation sets the bit. Each time the if () statement runs, it reads the value and runs a comparison operation.

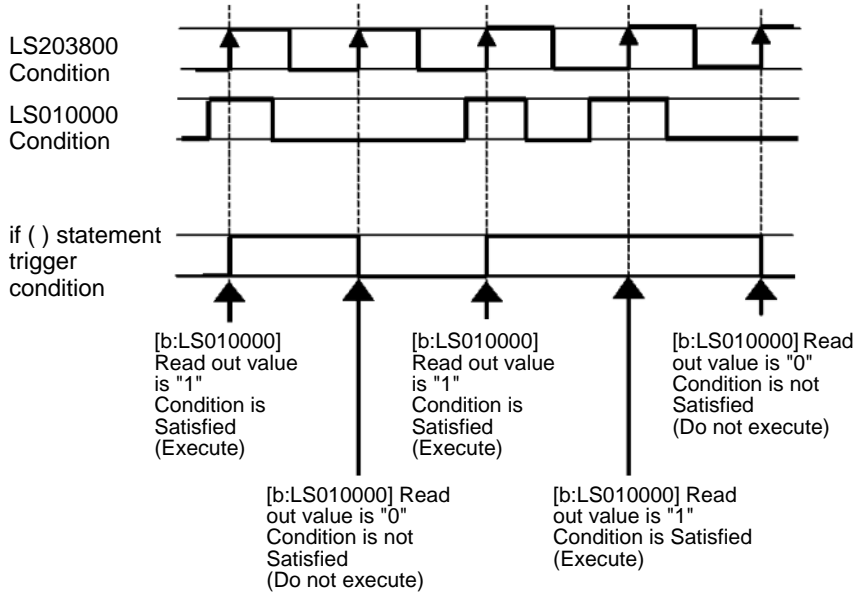Triggered Condition: Bit ON ([#INTERNAL]LS203800 *1)
Execution Expression: if ([b:[#INTERNAL]LS010000]==1)
{
clear ([b:[#INTERNAL]LS010000])
:
:

GP internal counter. The counter increments each time the Part set on the display screen
processes.

When this type of D-Script is created, even if touch input is done repeatedly, the tag scan is performed as shown in the following timing
chart. Here, each tag scan value is read out, the condition is compared and, regardless of the previous value,
if it agrees with the condition, the processing is performed.

◆ D-Script Processing Timing Chart

LS203800
Condition

LS010000
Condition

if ( ) statement
trigger
condition

[b:LS010000]
Read out value
is "1"
Condition is
Satisfied
(Execute)

[b:LS010000] Read
out value is "0"
Condition is not
Satisfied
(Do not execute)

[b:LS010000]
Read out value
is "1"
Condition is
Satisfied
(Execute)

[b:LS010000] Read
out value is "1"
Condition is Satisfied
(Execute)

[b:LS010000] Read
out value is "0"
Condition is not
Satisfied
(Do not execute)

20.8 Settings Guide

20.8.1 D-Script/Common [Global D-Script] Settings Guide

The following is the Common [Global D-Script] dialog box. The settings you can specify for the D-Script are the same as those in the dialog box. ID and trigger settings are not specified for Common [Extended Script], however, the other settings are the same.

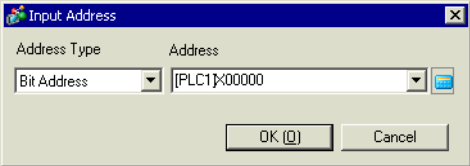| Setting	Description
Export	This can be selected from the File menu. Export writes a created script to a text file (.txt) which can then be imported into other scripts.
Import	This can be selected from the File menu. Import reads in an exported script (text file).
Row Number	Shows the row number to the right of the program.
Auto Indent Control	Automatically indents statements as below.
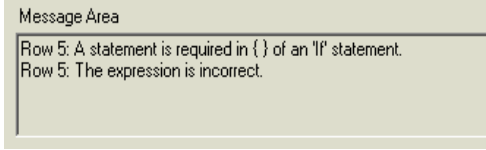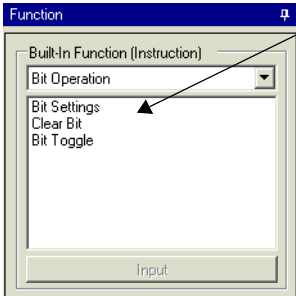
Continued

Setting	Description
Function Input Assistance 🖉	When the function and the initial bracket "(" are inputted as below, the function's format gets displayed. Execution Expression Enlarge Execution Expression 🖼 Address Input 0001 memcpy(0002 memcpy(Copy To Address, Copy From Address, No. of Words) 0003 0004 0005 0006 0007 0008 0009
Auto Syntax Completion 🖭	When "if" or "loop" is entered from the keypad, the remaining syntax is automatically placed.

Continued

Setting	Description
Address Input 🖼️	When creating a script, enter a left square bracket ([ ) to display the [Input Address] dialog box.

Select the address type from [Bit Address], [Word Address], [Temporary Address].
- Bit Address
  You can specify the Device/PLC address, GP Internal Device and Bit Variable.
- Word Address
  You can specify the Device/PLC address, GP Internal Device and Integer Variable.
- Temporary Address
  This address can only be used for scripts.

Refer to the following for details on the internal device.

) "A.1.2 Communicating with a Device/PLC Using the Direct Access Method" (page A-4)

) "A.1.3 Using the Memory Link Method with Unsupported Devices/PLCs" (page A-6)

IMPORTANT

- In the scripts, please do NOT set any passwords, etc., that begin with "0". All numeric values beginning with "0" will be processed as Oct (base-8) data.
- How to describe different input data formats
  For example,

DEC (Base-10)	: Non-zero starting value
	For example, 100

Hex (Base-16)	: Value starting with 0x
	For example, 0x100

Oct (Base-8)	: Value starting with 0x
	For example, 0100

- Example of operation with different data formats using the operator AND (Hex and BCD)

Hex only	
0x270F & 0xFF00	Result: 0x2700

BCD and Hex	
9999 & 0xFF00	Result: 0x9900

Continued

Setting	Description
Auto Syntax Analysis	Checks the syntax during script creation. The check results will be displayed in the bottom portion of the window.
ID	Scripts are managed by an ID number. When creating multiple scripts with different trigger conditions, set a value from 0 to 65,535.
Comment	Input a comment for the script.
Language	Choose a language from the drop-down list: [ASCII], [Japanese], [Chinese (Traditional)], [Chinese (Simplified)], or [Korean].
Enable Debug Function	Set whether or not to enable the debug function. If the _debug function exists in the body of the script, the _debug function will execute. For more information about this function, please refer to "21.7.1 Debug Function" (page 21-65) .
Triggered Condition	Set the trigger condition that causes the script to execute. For more information about this function, please refer to "20.7 Triggered Condition Setup" (page 20-46) . Extended scripts do not have the trigger condition setting.
Data Type	Set the data format for the script to Bin or BCD. For Extended Scripts, Bin is fixed.
Bit Length	Set the data length for the script to 16 bit or 32 bit.
Sign +/-	Select this when you want to insert negative numbers. This can only be set when the data type is Bin.
Execution Expression	The contents of the script.
Built-in Function (Instruction)	From the toolbar, select commands and functions to more easily add them to the script. For more information about available commands and functions that can be used, see) "21.13 Command List" (page 21-100) Built-in Functions Select a category from [Built-In Function (Instruction)]. The related functions appear in the bottom area. Select the function and click [Input]. The corresponding settings dialog box appears.

Setting	Description
User Defined Function	Register a script as a user-defined function and it can be used by other scripts. **NOTE** • For more details about user-defined functions, see "20.8.2 User-Defined Functions Settings Guide" (page 20-58).
Tool Box	As a shortcut, select commands from the Toolbox to use in the script. Also, you can select commands such as search and position text used in scripts. For more information about available commands, see "Chapter 21 Commands and Descriptions" (page 21-1).
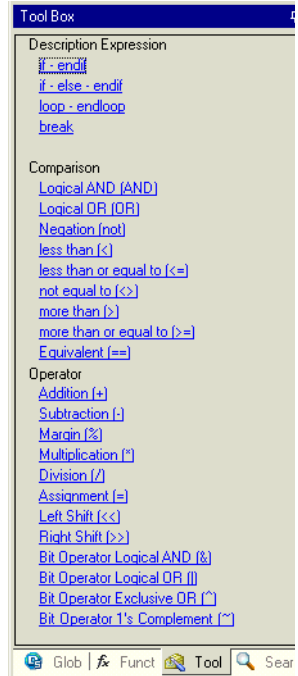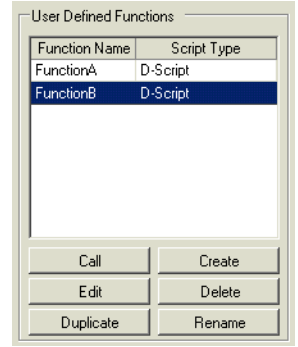
20.8.2 User-Defined Functions Settings Guide
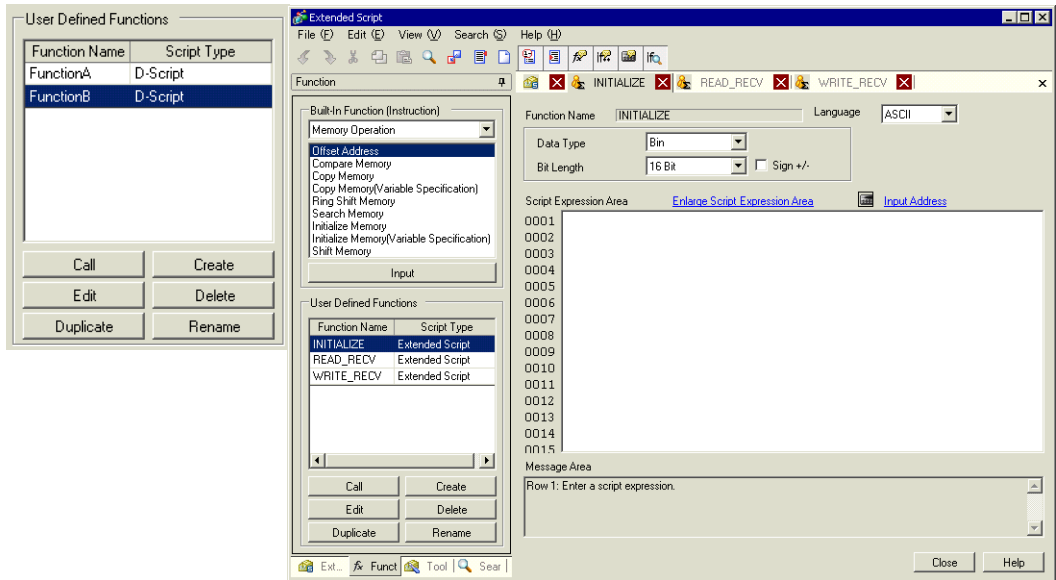
Setting	Description
Call	Call a created function. Select the function to call, click [Call] and "Call Function Name" is placed in the Execution field.
Create	Create a new function. Click on [Create]. The [Function Name] dialog box appears.
Edit	Edit an existing function. Select the function to edit, click on [Edit]. The [D-Script Function] dialog box appears.
Delete	Delete an existing function. Select the function to delete and click [Delete].
Duplicate	Copies the created function. Select the function to copy and click [Duplicate] and when the [Input Function Name] dialog box appears, enter the new function name.
Rename	Change the name of an existing function. Click on [Rename]. The Rename Function dialog box appears.

20.9 Restrictions

20.9.1 D-Script/Global D-Script Restrictions

- In D-Script programming, three addresses occupy the same amount of memory as one Part. The maximum number of addresses available for a D-Script is 255. Use the fewest possible addresses, since the more devices that are used, the slower the response.
- The Convert Address feature cannot convert addresses used in D-Script. Open the D-Script Editor to change these addresses.
- When the Connected Device Type settings are changed via the Project Manager menu >Project > Save As, the address used by the D-Script cannot be changed. Use the D-Script Settings dialog box to change the addresses.
- D-Script cannot run calculations on floating point values (Float Variables or Real Variables) or structure variables. However, you can run calculations on individual elements from structure variables.
- The size of a D-Script affects the Display Scan Time. Note that using a large number of addresses may significantly degrade the program performance.
- Do not specify [Continuous Action] in the Trigger Conditions for the script to write to device/PLC addresses. An error will be displayed because the communication processing cannot keep up with the large amount of write instructions. To enable [Continuous Action], use the GP internal device or temporary address.
- When calling a function within a function, the maximum number of nested levels is 9. Do NOT create more than 9 levels.
- Up to 9 levels of nested calls can be created.
- Up to 254 Functions can be created.

◆ **Depending on the devices specified for trigger conditions, the D-script operations activated by a trigger after the screen changes are as follows:**

Triggered Condition		Connection Device or GP Internal Device (LS/USR)				Memory Link			
	Current Value or Condition	Bit "0"	Bit "1"	Condition is not Satisfied	Condition is Satisfied	Bit "0"	Bit "1"	Condition is not Satisfied	Condition is Satisfied
Leading edge of bit		Disable	Enable	–	–	Disable	Disable	–	–
Falling edge of bit		Enable	Disable	–	–	Disable	Disable	–	–
Bit Change		Enable	Enable	–	–	Disable	Disable	–	–
Timer setting		Disable	Disable	Disable	Disable	Disable	Disable	Disable	Disable
Detecting true		–	–	Disable	Enable	–	–	Disable	Enable
Detecting false		–	–	Enable	Disable	–	–	Enable	Disable

 Operation is performed immediately after the screen is changed, or the power is turned ON.

 Operation is not performed immediately after the screen is changed, or the power is turned ON.

- When the timer is operating, the timer starts counting immediately after the screen changes.
- When using Global D-Script, the operations mentioned above are performed only when the GP power is turned ON. When the GP screen changes, however, the operation mentioned above is not performed and the monitor operates using the trigger conditions that have been set.
- When a Global D-Script includes a timer, the timer starts counting immediately after the GP power is turned ON.

NOTE
- Do not use the touch panel key to set the trigger bit or to operate the start bit in a program. The timing of the touch input may not be correct, resulting in the bit being improperly entered.

◆ **When a value is assigned to an address for switching screens while a D-Script command is being executed, the screen switching operation is processed after all D-Scripts have been processed.**

For example,

ID	00000			
Data Type	Bin	Data Length 16 Bit	Sign +/-	None
Trigger	Leading Bit([b:M0000])			

[w:[PLC1]D0100]=0   // (1)

[w:[#INTERNAL]LS0008]=30// (2) Switches to Base screen Number 30

[w:[PLC1]D0101]=1   // (3)

[w:[PLC1]D0102]=2   // (4)

When the above D-Script is executed, processing of the screen switch is performed after (3) and (4) have been processed.

◆ **When data used for D-Scripts is set by touch from the GP, detect that all the data has been written and then execute D-Scripts.**

◆ **Restrictions Specific to Global D-Script**

- When the GP power is turned ON, the actions shown in the table on the previous page are performed. At the screen change, the above table is not applied, and the trigger conditions are continuously monitored.

- Global D-Script operation is suspended during screen changes or other GP operations.

- After the GP power is turned ON, Global D-Script actions are not performed until all data reads are completed for the initial screen. After the initial screen changes, Global D-Script actions may be performed before the data reads are completed.

- The maximum number of devices in Global D-Scripts is 255. When this number is exceeded, the D-Script does not function. Since these devices always read data regardless of the screens, be sure to use only the minimum number of device settings in your D-Script. Otherwise, operation performance can be degraded.

- The maximum number of Global D-Scripts available is 32. The currently used function also counts as one Global D-Script. When the number of the Global D-Scripts reaches 32, any subsequent Global D-Scripts are ignored.

◆ **Restrictions for SIO Port Operations**

- Addresses designated in the Send/Receive functions are not added to the D-Script address count.
- The Control is a write-only variable, while Status and Received Data are read-only variables. Reading the Control variable or writing data to the Status variable causes the operation to fail.
- Create independent D-Scripts (or functions) for Send and Receive operations. For more information about the flow charts of data transfers, see
  ☞ " ■ Flow Chart" (page 20-24)
- The User area in the LS device (LS20 to LS2031 and LS2096 to LS8191) can store data for Send/Receive functions.
- In [System Settings] > [Script Settings], when no [D-Script/Global D-Script] is set, the 13th bit of LS2032 turns ON when the readout of the Send function, Receive function, Control, Status variable, and Received Data Size is executed. For information on the structure of the internal device, see
  ☞ "A.1.4.3 Special Relay" (page A-19)
- When using the Send/Receive functions, set the bit length of the D-Script to 16 bits. Note that the operation fails if the bit length is set to 32 bits.
- The size of the Send buffer is 2048 bytes, while the Receive buffer is 8192 bytes. The ER signal (output) RS signal (output) is turned OFF after at least 80% of the Receive buffer is full of received data.

◆ **Limitations on BCD Format Operations**

If a value which cannot be converted into BCD format is found during operation, the program stops running.

These values include A to F in hexadecimal format. Do not use such values.

If the program stops due to non- BCD values, bit 7 in common relay information (LS2032) in the GP turns ON. This bit does not turn OFF until the GP is turned OFF or goes offline.

For example,

[w:[PLC1]D0200]=([w:[PLC1]D0300]<<2)+80

If D300 is 3, shifting two bits to the left results in 0x000C, which cannot be converted into BCD format, and interrupts program execution.

[w:[PLC1]D0200]=[w:[PLC1]D0300]<<2

If D300 is 3, shifting two bits to the left results in 0x000C. Unlike the above example, 0x000C is the result of the operation to be stored in the memory, and does not cause the program to stop.

◆ **Limitations of Zero Operations**

If you divide by zero in division (/) and remainder (%) operations, execution will stop. Do not divide by zero.

If the program stops due to non- BCD values, bit 8 in common relay information (LS2032) in the GP turns ON. This bit does not turn OFF until the GP is turned OFF or goes offline.

◆ Notes on Delay During Assign Operation

Using a device address in an assign operation may cause write delay because the GP has to read the address data from the connected device. Consider the following:

For example,

    [w:[PLC1]D0200]=([w:[PLC1]D0300]+1 ...
    [w:[PLC1]D0201]=([w:[PLC1]D0200]+1 ...

Statement (1) assigns (D0300+1) into D0200. However, in statement (2), the result of statement (1) has not been assigned in D0200 because of time-consuming communication with the device/PLC. In such cases, program so that the result of statement (1) is stored in the LS area before it is executed, as shown below.

    [w:[#INTERNAL]LS0100]=[w:[PLC1]D0300]+1
    [w:[PLC1]D0200]=[w:[#INTERNAL]LS0100]
    [w:[PLC1]D0201]=[w:[#INTERNAL]LS0100]+1

20.9.2 Extended Script Restrictions

- For Device Addresses, only the LS Area and USR Area (Extended User Area) can be used.

- The temporary addresses of D-Scripts and Global D-Scripts are managed independently from the temporary address of Extended Scripts. Therefore, changes made to the temporary addresses of D-Scripts and Global D-Scripts are not reflected in the temporary address of Extended Scripts.

- You can call user-defined functions created with D-Script/Global D-Script, but if you access a device address outside of the internal device inside the function, it may not operate normally. Also, when transferred (during the creation of data for the GP), user-defined functions are created independently for D-Scripts, Global D-Scripts, and Extended Scripts.

- When calling a function from a function, the maximum number of nested levels is 9.

- Up to 254 functions can be called. (The number of functions available with "Call" is 254.)

- Extended Script does not affect the tag count.

- Functions supported only by Extended Script, for example string operations, do not function if called with D-Script or Global D-Script.

- The available data format is Bin. BCD data format is disabled.

- The size of the Send buffer is 2048 bytes, while the Receive buffer is 8192 bytes. The CTS line is turned OFF after at least 80% of the Receive buffer is full of received data.

- General protocol and Extended Script cannot be selected simultaneously. The following table shows more information about combinations.

Extended SIO Setting	D-Script/ Global D-Script Extended SIO function for Extended Script	Extended SIO function for Extended Script
General protocol	OK: Operation enabled	Invalid: Operation disabled
Extended Script	Invalid: Operation disabled	OK: Operation enabled

- Notational conventions for the character string setting

When using character strings with "_strset ( )" and other functions, enclose the character string in double quotation marks ("). To display double quotation marks in the character strings, append the "\" symbol and express as [\"]. There is no way to represent a single "\" symbol. When necessary, use the character code format setting (_strset (databuf0, 92).
For example,

"ABC\"DEF"→ ABC"DEF
"ABC\DEF"→   ABC\DEF
"ABC\\"DEF"→ ABC\"DEF
"ABC\\DEF" → ABC\\DEF

◆ **The following table shows the sizes of the dedicated buffers for Extended SIO, databuf0, databuf1, databuf2, and databuf3.**

Buffer	Buffer Name	Character Size
Data Buffer 0	databuf0	1KB
Data Buffer 1	databuf1	1KB
Data Buffer 2	databuf2	1KB
Data buffer 3	databuf3	1KB

20.9.3　Restrictions on User-Defined Functions

- Portions of the commands that can be used differ with each script. When using commands, please refer to "21.13 Command List" (page 21-100) .

- For the function name, you may use any English letters or the underscore character "_." (However, the function name must begin with an alphanumeric character.)

- Do not use the following as Function Names.

and	b_call	Bcall	_bin2hexasc	break	Call
_CF_delete	_CF_dir	_CF_read	_CF_read_csv	_CF_rename	_CF_write
clear	databuf0	databuf1	databuf2	databuf3	_decasc2bin
_dlcopy	dsp_arc	dsp_circle	dsp_dot	dsp_line	dsp_rectangle
else	endif	fall	_hexasc2bin	if	IO_READ
IO_READ_EX	IO_READ_WAIT	IO_WRITE	IO_WRITE_EX	loop	_memcmp
memcpy	_memcpy_EX	memring	_memsearch	memset	_memset_EX
_memshift	not	or	return	rise	rise_expr
set	_strcat	_strlen	_strmid	_strset	timer
toggle	_wait				

20.9.4　　Notes on Operation Results

■ Overflowing Digits

Overflowing digits resulting from operations are rounded.
When performing an operation on unsigned 16-bit data:

- 65535 + 1 = 0 (Produces overflowing digits)

- (65534 ∗ 2) / 2 = 32766 (Produces overflowing digits)

- (65534 / 2) ∗ 2 = 65534 (Does not produce overflowing digits)

■ Difference of Residual Processing

The result of a residual processing depends on whether the left and right sides are signed or unsigned.

- –9 % 5 = –4

- 9 % –5 = 4

■ Rounded Decimal Places

Decimal places resulting from a division are rounded.

- 10 / 3 ∗ 3 = 9

- 10 ∗ 3 / 3 = 10

■ Notes on Operating BCD Data

A BCD-data operation which produces overflowing digits does not give the correct result.

20.9.5 Errors

The following error message is displayed when a Script is configured incorrectly. The error will be displayed on the bottom of the GP screen.

Error codes are written to the LS91XX addresses. The number written in the error code area will be the number portion following RAAA in the table below. (For example, when error RAAA130 occurs, '130' will be written.)

List of Script Error Codes

D-Script (Error Address=LS9120)	Global D-Script (Error Address=LS9110)	Extended Script (Error Address=LS9100)
-	RAAA130	RAAA140
Unused	Global D-Script Error. (The Total Number of Global D-Scripts exceeds the maximum of 32.)	Extended D-Script Error (The total no. of functions exceeds the max of 255.)
-	RAAA131	-
Unused	Global D-Script Error. (The total no. of devices exceeds the maximum of 255.)	Unused
RAAA120	RAAA132	RAAA141
D-Script Error (The specified function does not exist or the function has an error.)	Global D-Script Error (The specified function does not exist or the function has an error.)	Extended D-Script Error (The specified function does not exist or the function has an error.)
RAAA121	RAAA133	RAAA142
D-Script Error (These functions are nested to 10 levels or more.)	Global D-Script Error (These functions are nested to 10 levels or more.)	Extended D-Script Error (These functions are nested to 10 levels or more.)
RAAA122	RAAA134	RAAA143
D-Script Error (An expression exists, that is not supported by this version.)	Global D-Script Error (An expression exists, that is not supported by this version.)	Extended D-Script Error (An expression exists, that is not supported by this version.)
RAAA123	RAAA135	RAAA144
D-Script Error (The SIO operation function is used in a condition where no device/ PLC has been set.)	Global D-Script Error (The SIO operation function is used in a condition where no device/ PLC has been set.)	Extended D-Script Error (The SIO operation function is used in a condition where no device/PLC has been set.)
RAAA124	RAAA136	RAAA145
The D-Script has an error.	The Global D-script has an error.	The Extended D-Script has an error.