Pro-face

Digital Human Machine Interface

Device/PLC Connection Manuals

About the Device/PLC Connection Manuals

Prior to reading these manuals and setting up your device, be sure to read the "Important: Prior to reading the Device/PLC Connection manual" information. Also, be sure to download the "Preface for Trademark Rights, List of Units Supported, How to Read Manuals and Documentation Conventions" PDF file. Furthermore, be sure to keep all manual-related data in a safe, easy-to-find location.

Chapter 1: Memory Link Method

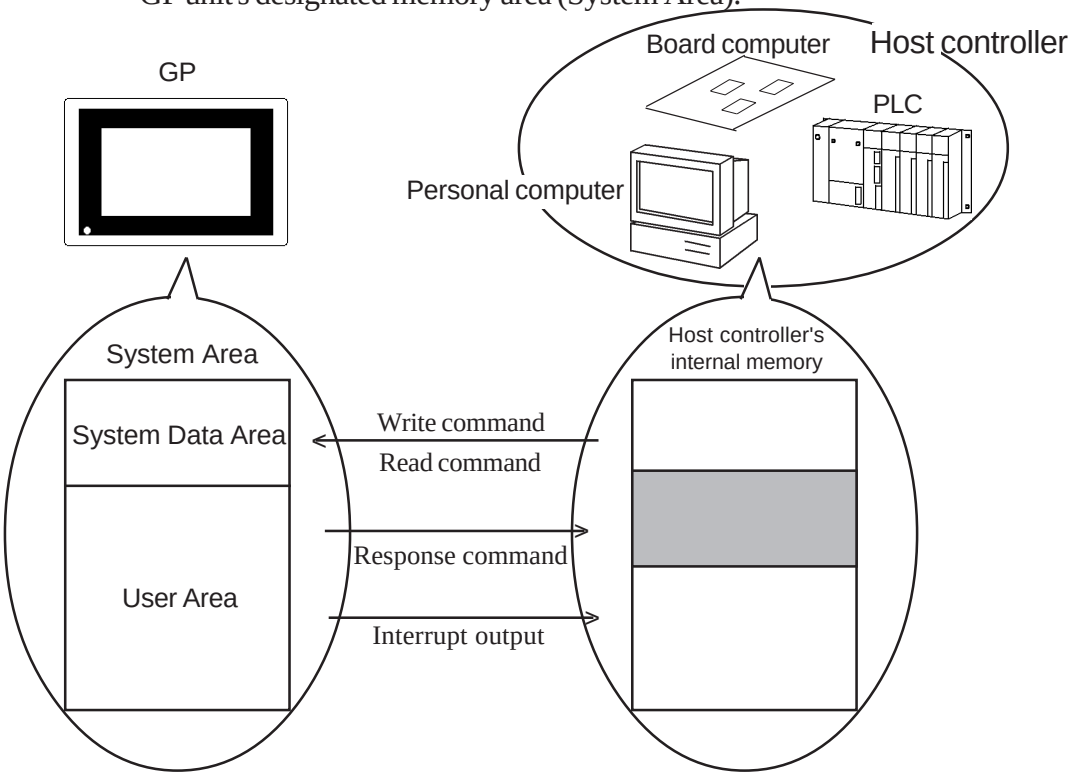
Please read this chapter prior to using the GP series' Memory Link system.
This chapter describes ethernet communication between a GP unit and a device with no specific protocol (e.g. personal computer, one-board microcomputer).

1-1 Memory Link Method

All data transmission between the GP unit and a host controller is performed according to the host controller's program.

The GP unit displays screens according to the display data transferred by the host controller's Write commands. Also, the GP unit sends the stored data to the host controller according to the host controller's Read commands. Thus, the host controller always controls all communications with the GP unit.

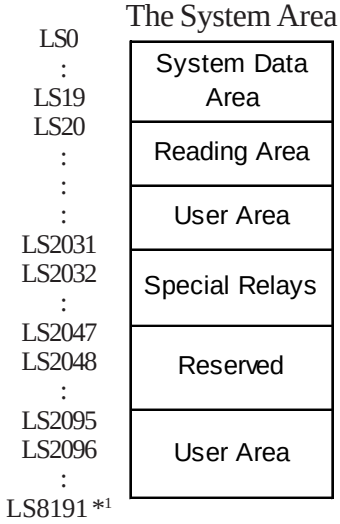
Data transmission between the GP unit and a host controller is performed via the GP unit's designated memory area (System Area).



1-1-1 System Area

Data transmission between the GP unit and a host controller is performed using a pre-defined System Area within the GP unit. The GP unit then displays screen images according to the data written in this System Area.

The System Area capacity is 8192 words [*1] and consists of the following areas:



◆System Data Area

Data required for GP unitoperations (e.g. GP panel screen control data, error information, etc.) is written into the System Area. The contents of the written data depends on the designation used for each address.

Reference *1-1-2 System Data Area Contents and Setting Range*

◆User Area

The User Area is used to receive data from the host controller, and to send GP data to the host controller.

To perform data transmission, the host controller first specifies the address of the GP unit to which the data will be written and then creates the writing program. The GP unit sets up the Parts and Tags to display the data that has been written to the specified address. To read the data specified by the K-Tag (numeric key input) and T-Tag (touch panel input), the host computer must have a program to read data from the GP unit.

■Special Relays

Special Relays are used to store various status data in the GP.

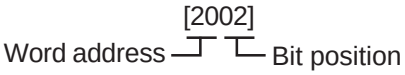
Note:

- When T-Tag data is written into address 13 of the System Data area, the interrupt output is activated. If the host controller receives a 1-byte interrupt output (e.g. by using the BASIC programming language's INPUT$ command) and this interrupt output is used to execute a jump command to each sub-routine, the program can be simplified.

**1 Except for GP2000 Series units, 4096 words (LS0 to LS4095) can be used.*

Note:

- To specify an address via a specific bit, assign a bit position (00 to 15) after the word address.
  <Example> To specify bit "02" for address 20 of the User Area:



◆Reserved

Please do not use this area. It is only for GP unit's internal use. If you use this area, the GP unit will not operate properly.

1-1-2 System Data Area Contents and Setting Range

When you wish to turn the GP unit's display OFF, use the Screen Display ON/OFF bit. Do not use the Control area's Backlight OFF bit. Be aware that this feature's system Data Area settings and range used during Memory Link Communication will differ from the settings used with Direct Access Communication.

Address	Detail	Function	Bit	Particulars
1	Status *1		0, 1	Reserved
			2	Now Printing *2
			3	Writes a set value *3
			4 ~ 7	Reserved
			8	K-tag entry error *4
			9	Display 0:ON, 1:OFF *5
			10	Backlight burnout detection *6
			11	Touch-Panel Input Error *7
			12 ~ 15	Reserved
3	Error Status Each bit changes according to the GP error function. When an error occurs, the corresponding bit will turn on. * A bit that has turned on remains on until the power is turned off and back on, or until RUN mode is re-entered from OFFLINE mode or details and the handling process about the Error Status contents, refer to the Section 1-1-4.		0, 1	Unused
			2	System ROM/RAM
			3	Screen Memory Checksum
			4	SIO Framing
			5	SIO Parity
			6	SIO Overrun
			7, 8	Unused
			9	Initialization of Internal Memory Checksum Necessary
			10	Timer Lock Error
			11 ~ 15	Unused
4	Clock Data (Year)	"Year / Month / Day / Hour / Minute" Data is stored in BCD's 2digits. (E.g.) 98/02/01 17:15	0 ~ 7	Stores the last 2 digits of the Calendar year as 2 BCD digits
			8 ~ 15	Unused
5	Clock Data (Month)		0 ~ 7	Stores 01 to 12 (Month) as 2 BCD digits
			8 ~ 15	Unused
6	Clock Data (Day)		0 ~ 7	Stores 00 to 31 (Day) as 2 BCD digits
			8 ~ 15	Unused
7	Clock Data (Hour)		0 ~ 7	Stores 00 to 23 (Hour) as 2 BCD digits
			8 ~ 15	Unused
8	Clock Data (Minute)		0 ~ 7	Stores 00 to 59 (Minute) as 2 BCD digits
			8 ~ 15	Unused
10	Interrupt Output (Touch OFF) *15	If you Write in word data, the bottom 8 bits will be output as an interrupt code after touching OFF.However FFh will not be output.		
11	Control *8		0	Backlight *9
			1	Buzzer ON *10
			2	Starts Printing
			3	Reserved
			4	Buzzer *10 - - - 0:enabled 1: disabled
			5	AUX Output *10 - - - 0:enabled 1: disabled
			6	Interrupt Output when touching panel to turn the display ON. (Interrupt Code:FFh) 0: Disabled 1: Enabled *11
			7	Reserved
			8	VGA display *12 - - - 0: Disabled 1: Enabled
			9, 10	Reserved
			11	Hard copy output *13 - - - 0: Enabled 1: Disabled
			12 ~ 15	Reserved

Address	Detail	Function	Bit	Particulars
12	Screen Display ON/OFF *14	FFFFh : Screen clears almost immediately 0h: Screen turns ON		
13	Interrupt Output *15	Using a Touch Tag or other method to write absolute value data from GP causes an output of the interrupt code using the contents of the bottom 8 bits (Will not out put FFh)		
15	Screen Display No.	Write the Screen No. in binary to change the screen display	0 ~ 14	Screen change number, 1 to 8999.(1 to 1999 when using BCD input)
			15	Forced Screen Change 0 : normal, 1: Forced Screen Change
16	Window Control *16		0	Display - - 0: OFF 1: ON
			1	Changing the order of window overlapping - - 0: Possible 1: Not Possible
			2 ~ 15	Reserved
17	Window Registration No. *16	Global Window registration number selected indirectly (BIN/BCD)		
18	Window Display Position (X-coordinate) *16	Global Window display position reached indirectly (BIN/BCD)		
19	Window Display Position (Y-coordinate) *16			

**1 • Monitor, in bit units, only the necessary bits.*

• Since reserved bits may be used for GP system maintenance, etc., their ON/OFF status is not defined.

**2 This bit turns ON during printing. Changing to OFFLINE mode in the middle of printing can cause the print output to become disordered.*

**3 Every time a value is written with the K-tag or Keypad Input Display, the bit is reversed.*

**4 If an (input value range) Error has been set for the K-tag data being entered, and a value outside the allowed range is entered, the bit turns ON. If, however, a value is entered that is within the Error range, or if the display screen is changed, this bit will turn OFF.*

**5 <Display ON/OFF status>*

The screen display ON/OFF can be detected from the PLC. This bit will change in the following cases:

(1) "FFFF" is written to the system data area's screen display ON/OFF bit (LS9 when using link type), to turn the screen display OFF. (Bit 9 = 1)

(2) After the stand-by time has elapsed, the screen display OFF bit is turned ON automatically. (Bit 9 = 1)

(3) The screen display OFF status has been changed to the screen display ON status via screen switching, etc. (Bit 9 = 0)

(4) The screen display ON/OFF status bit will not change via turning ON/OFF the system data area backlight OFF bit (Bit 0).

**6 <Backlight Burnout Detection>*

The bit turns ON when backlight burnout is detected. This feature is available only on units equipped with a backlight.

**7 <Touch-panel input error>*

The touch-panel input error bit is turned ON when input in the same position continues for longer than the specified time.

**8 Be sure to turn all reserved bits OFF since they may be used for GP system maintenance, etc.*

**9 With the GP series except GP-477R, GP-470, and GP-870 series units, the backlight turns OFF when this bit is ON(LCD display does not change) and turns ON when the bit is OFF.*

When the Control area's Backlight OFF bit turns ON, only the backlight will turn OFF, however, the LCD display will remain ON and all touch switches set up on the display can still be used. Use the Screen display ON/OFF bit to actually turn the screen display OFF.

**10 Control Bit 1 (Buzzer On) outputs as shown below.*

Buzzer Sound ..... While Control Bit1 is on, the GP internal buzzer is activated.

AUX Output ..... While Control Bit 1 is on, the AUX buzzer output is activated.

**11 Interrupt output when touching the panel to turn the display OFF to ON.*

- *Only when the display is turned ON by touching the panel, interrupt output will be operated.*
- *When using GP-H70, interrupt output will not be operated if the display is turned ON by the Operation Switch on the rear side.*

**12 When using GP-570VM and GP-870VM, the entire screen becomes a VGA display when this bit is on. Pressing the screen options position during a VGA display turns this function off.*

**13 Turning ON bit 11(Hard Copy Output) in the Control Area will cancel the current printing of the display's hard copy.*

- *After printing is cancelled, bit 11, however, will not turn OFF automatically. Therefore, after checking the Status Area's Now Printing bit, turn off the Control Area's bit 11.*
- *While bit 11 in the Control area is turned ON, hard copy cannot be created. If you cancel printing before it is completed, printing will stop after the last line data on the panel's current display has been output. Data already input in the printer buffer's memory will not be deleted.*

**14 After the System Data Area's "Screen Display ON/OFF" bit is set to turn the display OFF, simply touching the screen will turn the display ON again.*

**15 Do not write control codes 00 to 1F to word addresses 10 and 13. It may terminate data communication.*

16* ***Reference* *For more about windows, refer to GP-PRO/PBIII for Windows Tag Reference Manual."**2.26 U-tag (Window Display)**"*

- **Addresses 0, 2, 9, 14 are reserved areas. Do not write data to this area.**
- **Since addresses 3, 12, 13, 15 are utilized for System Control, displays that depend on tags do not function.**
- **Since addresses 12, 13, 15 are used to control word units, bit write cannot be performed.**
- **Writing FFFFh to address 12 causes the screen display to erase within moments. When you wish to erase the screen using the STANDBY MODE TIME entered in GP unit's INITIALIZE setup, write 0000h in address 12.**
- **Do not write the control code 00~1F in addresses 10 and 13. Data transmission may become impossible.**

1-1-3 Special Relays

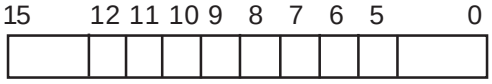
The structure of the Special Relay is as follows:

◆Reserved

The Reserved address value is undefined. Do not use this area.

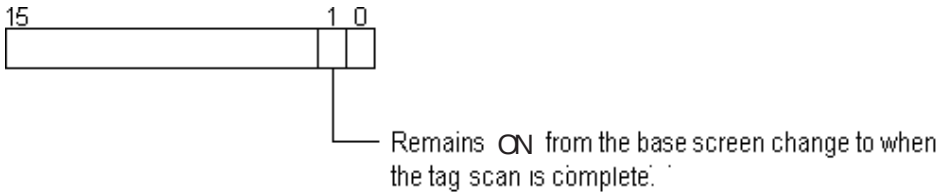
2032	Common Relay Information
2033	Base Screen Information
2034 2035	Reserved
2036	1 Second Binary Counter
2037	Tag Scan Time
2038	Reserved
2039	Tag Scan Counter
: : 2047	Reserved

◆Common Relay Information (2032)



Bit	Contents
0	Reserved
1	Remains ON from when a screen change (base or screen window) occurs until the tag scan is complete
2	Reserved
3	ON when displaying the initial screen after power up
4	Always ON
5	Always OFF
6	Turns ON when backup SRAM data has been deleted (Only for GP units equipped with backup SRAM)
7	Turns ON if a BCD error occurs while D-Script is being used. For more information about D-Script, ***Reference Tag Reference Manual, Section 3.1, "D-Script"***
8	Turns ON if a zero division error occurs while D-Script is being used
9	Turns ON if the filing data cannot be transferred to Backup SRAM by the Filing Data feature
10	When filing data transfer is triggered by the Control Word Address, this bit turns ON if the data cannot be transferred from PLC to SRAM. Also, when data transfer between PLC units is triggered by the Filing Data Display, only when the transfer complete bit address is used, this bit turns ON if the data is not transferred from the PLC to the SRAM.
11	Turns ON when Filing Data Display is used to transfer data to and from SRAM to LS area.
12	When using D-Script, turns ON if a communication error occurs when the function memcpy() is used, or when reading data using the designated Address Offset. Turns OFF when data read is normally completed.
13-15	Reserved

◆Base Screen Information (2033)



◆1 Second Binary Counter (2035)

Begins counting in one second intervals after the GP unit's power is turned ON. The data is binary.

◆Tag Scan Time (2036)

The time it takes to process all the tags setup on a screen. The data is stored as binary, millisecond units. This data is refreshed when preparation for all the screen tags is completed. The default value is 0. Data is accurate within + 10ms.

◆Tag Scan Counter (2038)

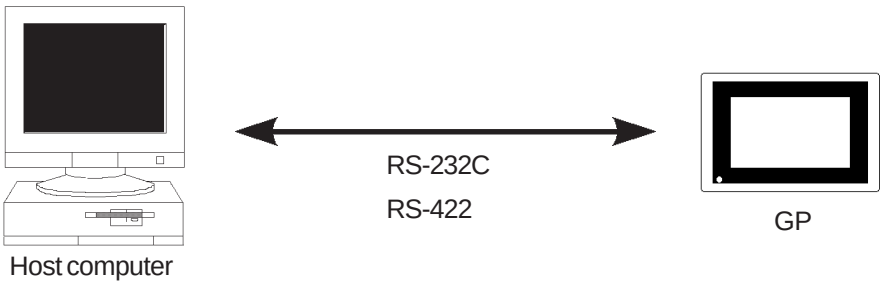
Counts until all the tags on a screen have been processed. The data is binary.

The Special Relay is not write protected. Do not turn this Relay's data ON or OFF using tags or other method.

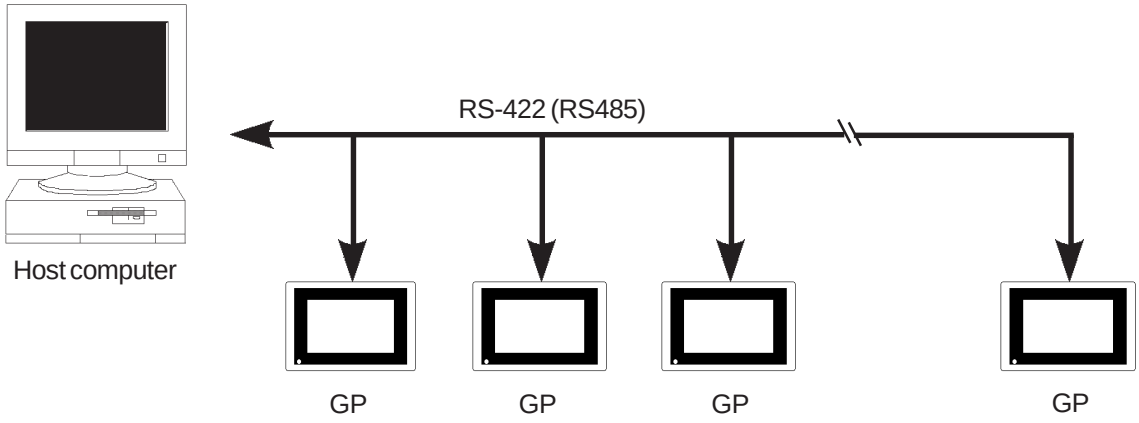
1-2 System Configuration

The memory link system allows you to connect up to 32 GP Series units to a single host computer.

■1:1 connection



■1:n connection (n < 32)



Up to 32 GP units; cable may extend up to 600 meters.

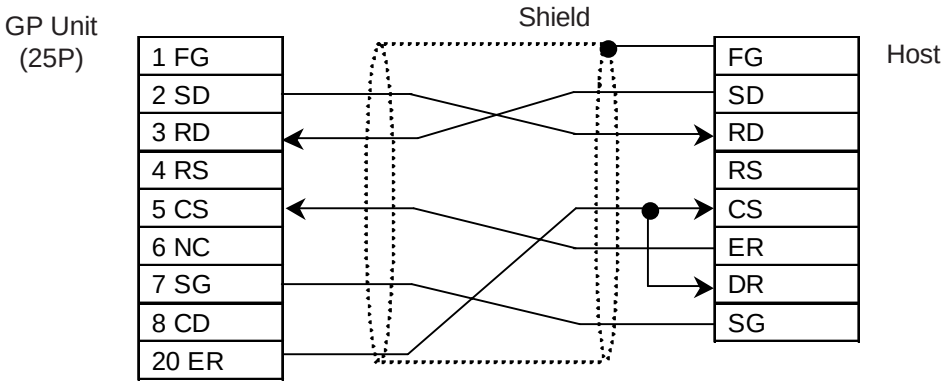
1-3 Connection Diagrams

This section describes how to connect the GP unit to the host computer.

1-3-1 RS-232C Interface Connection

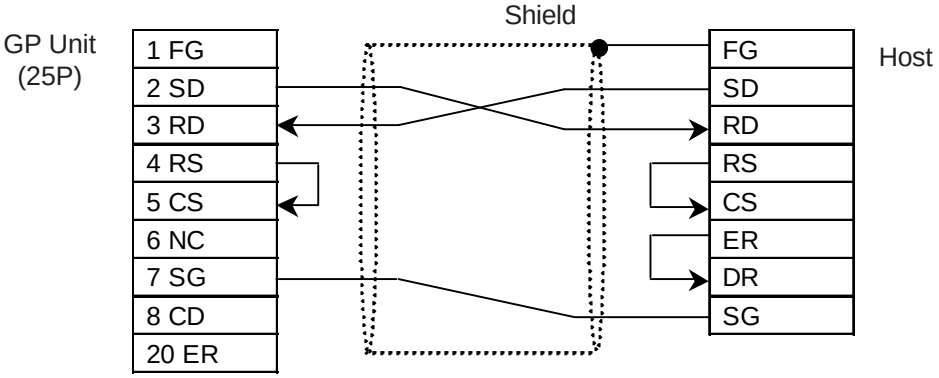
When using an RS-232C cable, there are two types of control formats: DTR (ER) Control and XON/XOFF Control. The GP-Host connection for each type is illustrated below.

■ DTR (ER) Control



STOP **Set the host's control software so that the host will not transmit data to the GP when the GP unit's ER is turned OFF.**

■ XON/XOFF Control



Important

- ***Depending on the host, the RS-232C connector's shape, pin numbers and corresponding signal names may differ. Be sure to follow the host interface specifications.***
- ***The maximum cable length is 15m.***

1-3-2 RS-422 Interface Connection

When using an RS-422 cable, only XON/XOFF control is available. The various types of GP-Host connections are illustrated below.

Careful!

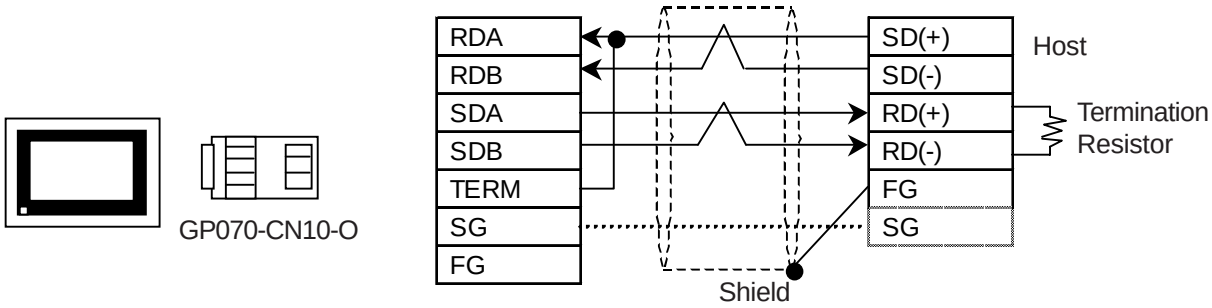
- *Be sure to insert a termination resistor between RDA-RDB.*
- *For a 24AWG line, use a twisted-pair cable with a static electrical capacity of 50pF/m, and a standard charactersistic impedance of 100Ω..*

Important

- ***The maximum length of the cable is 600m.***
- ***Depending on the Host connected, the connection method and termination resistance will differ.***
- ***When an SG wire is available, be sure to connect it.***

< 4-wire 1:1 Connection>

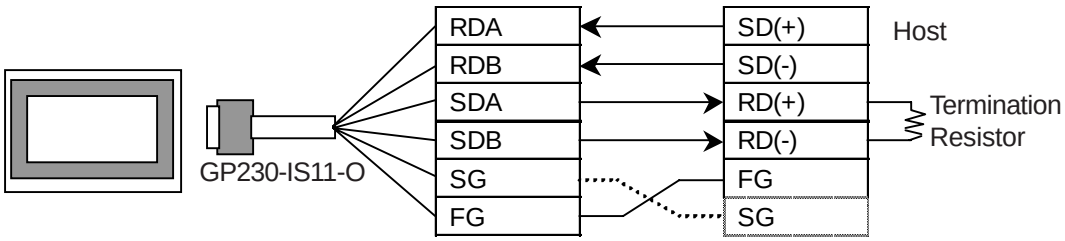
- When using Digital's RS-422 connector terminal adapter, GP070-CN10-O



Note:

Connecting the GP070-CN10-O's RDA and TERM wires inserts a 100Ω termination resistance between RDA-RDB on the GP.

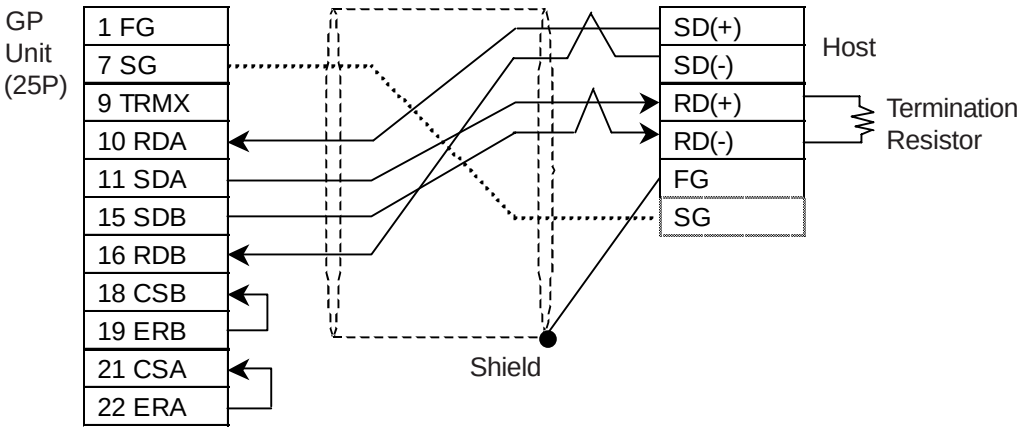
- When using Digital's RS-422 cable, GP230-IS11-O



Note:

Using the GP230-IS11-O cable inserts a termination resistance of 100Ω between RDA-RDB.

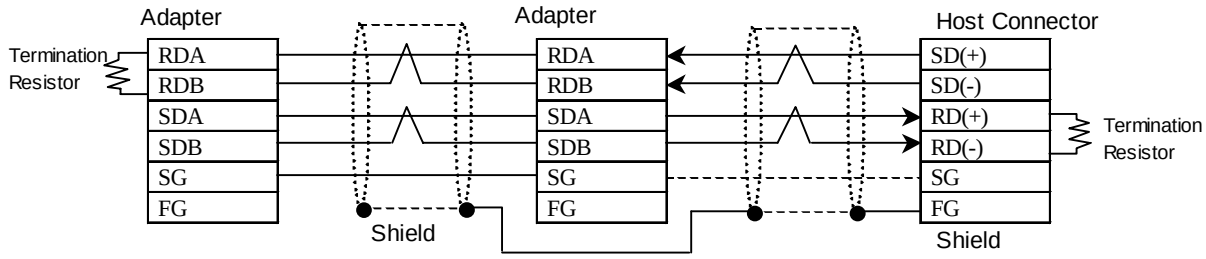
• When making your own cable connections



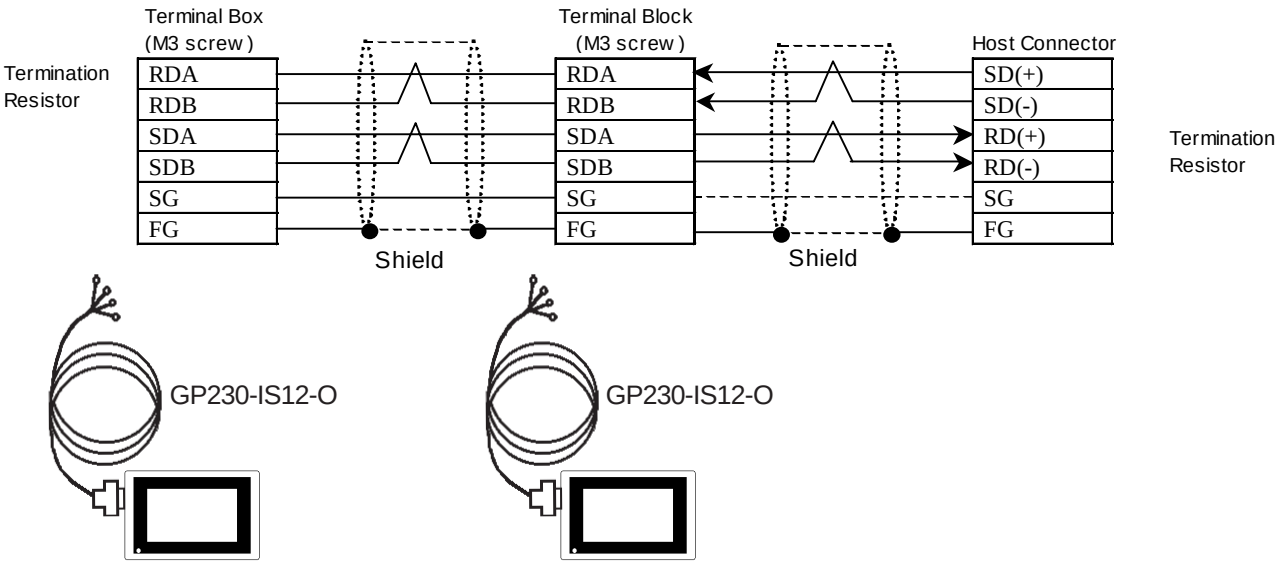
Note: **Connecting pins #9 and #10 in the GP Serial I/F inserts a termination resistance of 100Ω between RDA-RDB.**

< 4-wire 1:n Connection>

• When using Digital's RS-422 connector terminal adapter, GP070-CN10-O

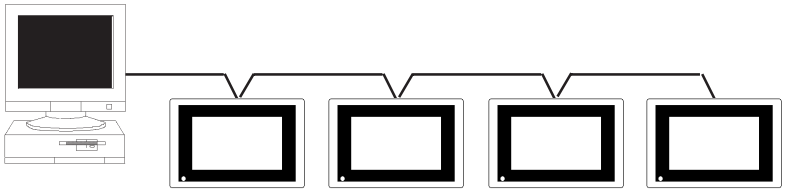


• When using Digital's RS-422 cable, GP230-IS12-O



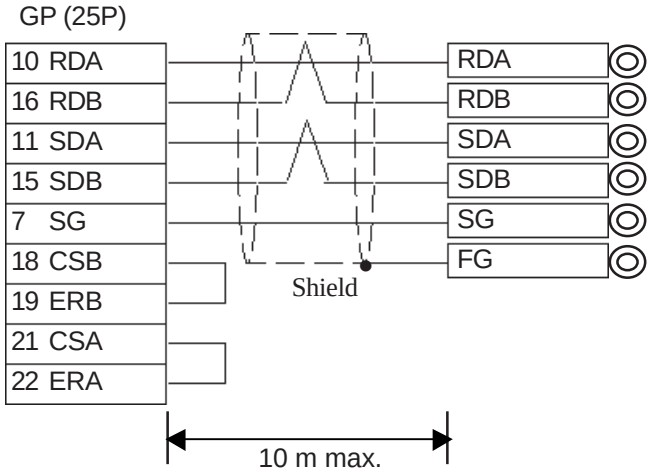
Important

- ***Be sure to connect all shields of the cable to the FG terminal on the host connector.***
- ***The FG wire of the GP230-IS12-0 cable is not connected to the FG terminal of the GP unit.***
- ***Make sure that the GP unit, connected at one end of the cable, and the host, connected at the other end of the cable, are terminated with resistors.***
- ***The host unit should be connected at the end of a communications network.***



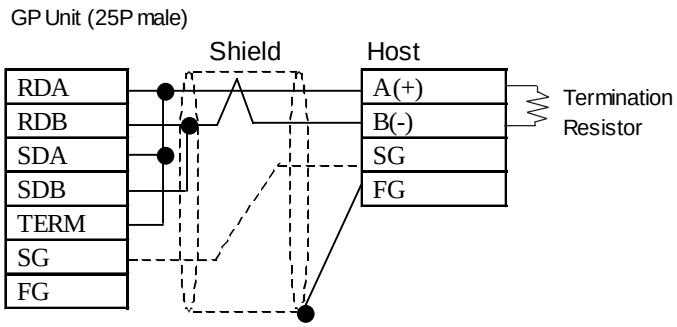
Note:

- When you wish to use an RS-422 cable of an other manufacturer, we recommend the Hirakawa Densen's H-9293A (CO-HC-ESV-EP*7/0.2).
- The following shows the connection required for this cable. At this time, be sure not to use a cable longer than 10 m for connecting the GP unit and the terminal box.



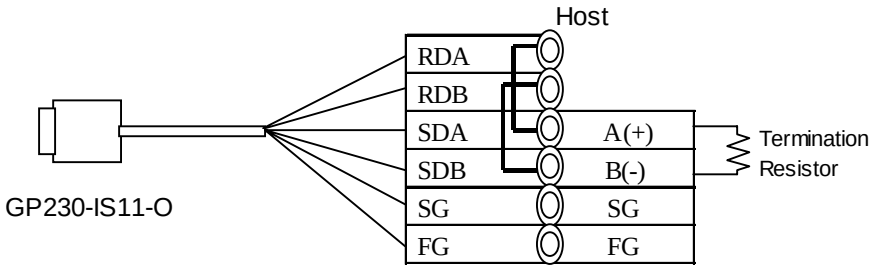
< 2-wire 1:1 Connection>

- When using Digital's RS-422 connector terminal adapter, GP070-CN10-O

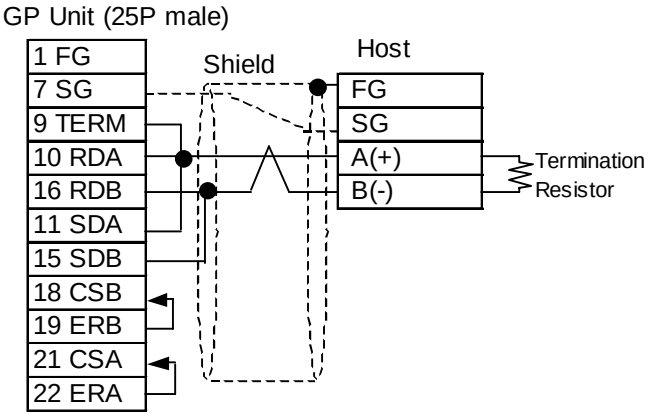


Connecting the GP070-CN10-O cable's RDA and TERM inserts a termination resistance of 100Ω on the GP Unit side.

• When using Digital's RS-422 cable, GP230-IS11-O



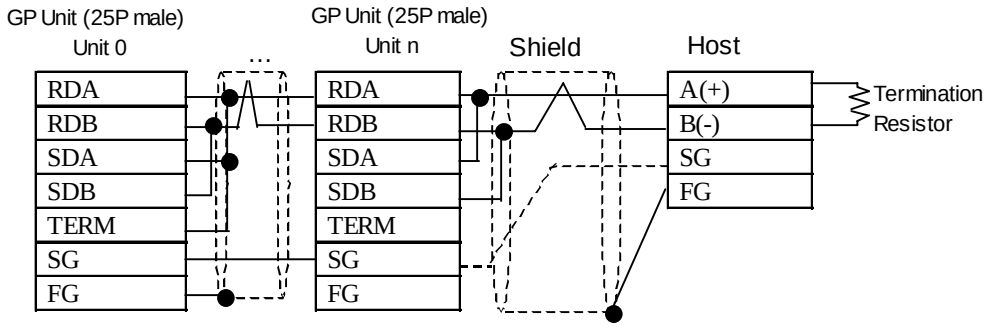
• When making your own cable connections



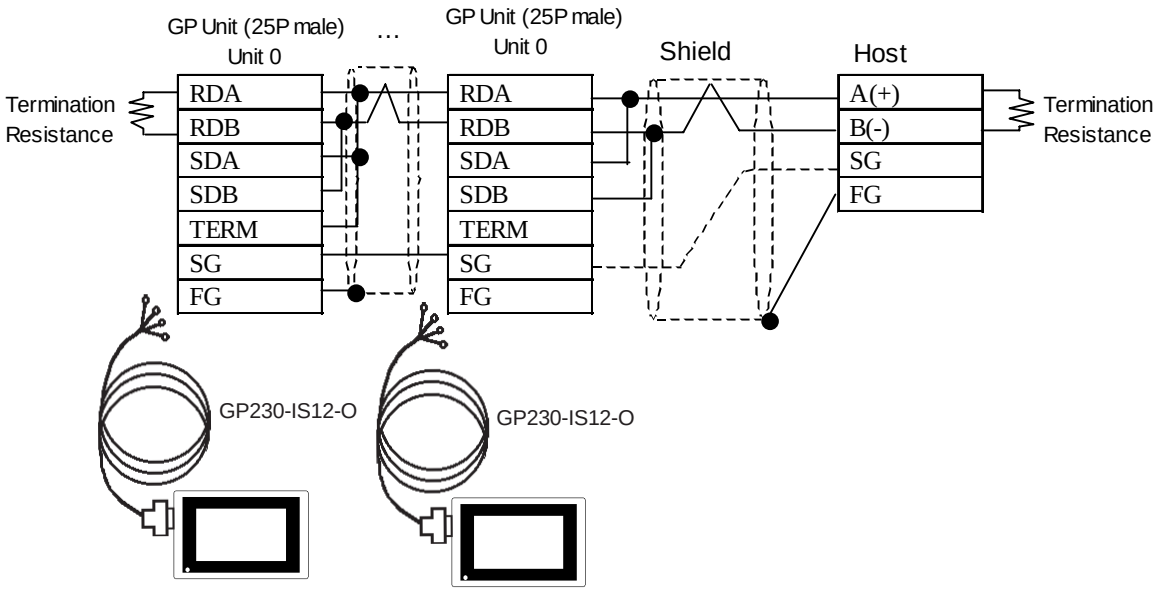
Connecting pins #9 and #10 in the GP Serial I/F inserts a termination resistance of 100Ω between RDA-RDB.

< 2-wire 1:n Connection>

• When using Digital's RS-422 connector terminal adapter, GP070-CN10-O



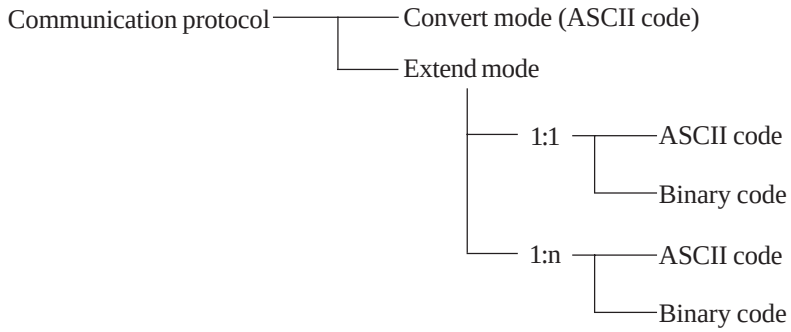
• When using Digital's RS-422 cable, GP230-IS12-O



MEMO

Chapter 2: Communication Protocol

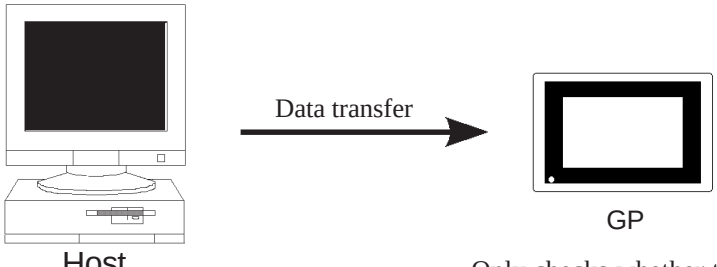
Communication protocol defines the format of data to be exchanged between the host and GP units and the procedures that allow such exchange of data. Broadly, two modes are available with the communication protocol of the GP unit as shown below. You need to switch between these modes in accordance, with the purpose for data exchange, and also the data processing performance of the host.
The host environment for program development and the system configuration are just a few examples of a variety of factors that must be taken into consideration when selecting the optimum mode. As the system administrator, you must review all possible factors to select the mode that best fits your needs.



■Convert Mode

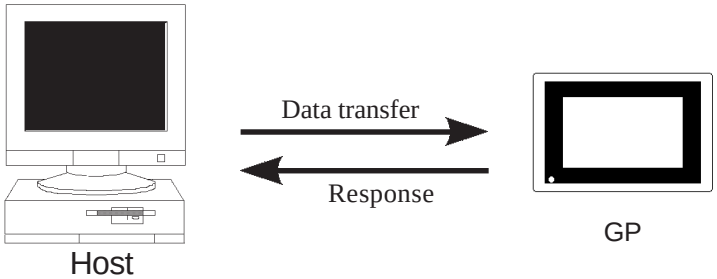
In this mode, only the write command (Esc W) for writing, and the read command (Esc R) for reading from the system area are used for communication.
In the convert mode, data is converted into ASCII codes only before transmission. Also, non-procedural data transmission is basically used in this mode. Therefore, the host is less burdened with communication control. On the other hand, data exchanged between the host and the GP units is less reliable.

◆ Non-procedural transmission



Only checks whether there is a communication error.

◆Procedural transmission



Checks the response to determine whether the data has been properly received.

Checks whether there is any communication error and determines whether the received data is valid.

■Extend Mode

In this mode, not only the read and the write commands but also drawing and other commands are available. This mode is designed for a multi-drop system in which a single host is connected to multiple GP units using drop wires.
In this mode, you can specify whether to send a sum check code together with the data block and also whether to send a response (ACK or NAK) for improved data reliability. This mode is further divided into ASCII and binary modes. Choose the mode that best fits your needs.

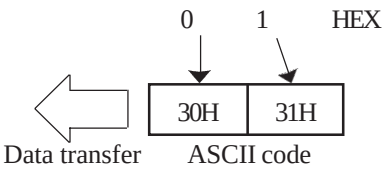
Note:

- For the extend mode, select 8 bits for data length when you specify the I/O Setup communication parameters for GP units.
- To improve data reliability, be sure to use the sum check code and response.

◆ASCII mode

In this mode, data (header and terminator excluded) are converted into ASCII codes before transmission.

<Example> When "01" (HEX) is sent

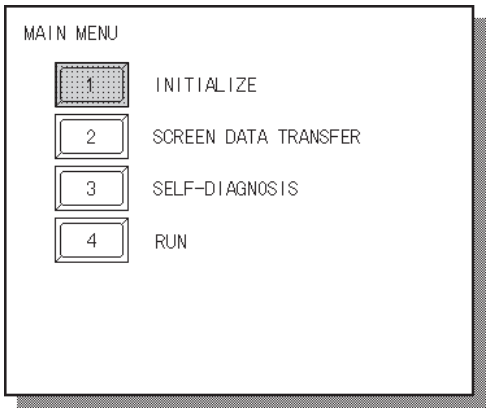


◆Binary mode

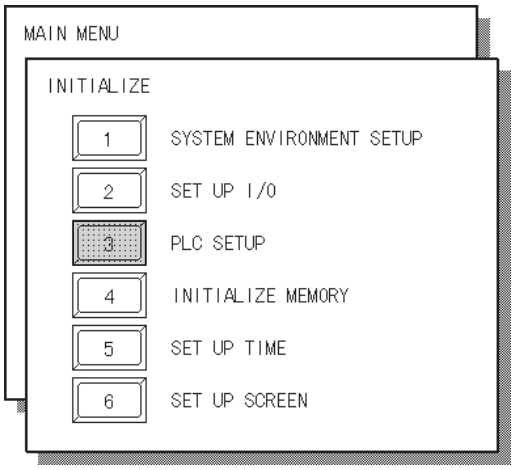
In this mode, data (header and terminator excluded) is converted into binary codes prior to transmission.

2-1 Setup

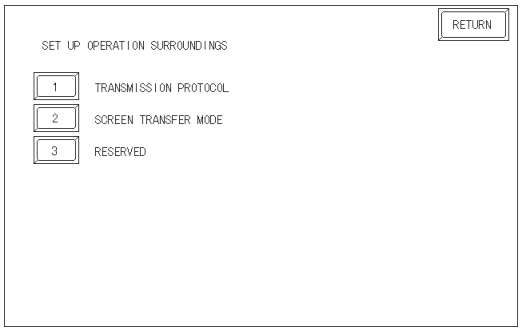
The following shows the steps you need to follow to specify the Data transmission protocol parameters:



1) Select [1] [INITIALIZE].



2) Select [3] [PLC SETUP].



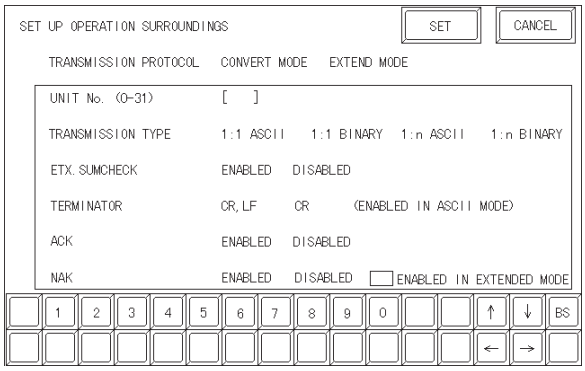
3) Select the desired options for each of [1] and [2].

1 TRANSMISSION PROTOCOL
Select the desired data transmission mode ([CONVERT MODE] or [EXTEND MODE]) and other transmission options.

2 SCREEN TRANSFER MODE
Select the screen data transfer mode.

3 RESERVED
This option cannot be selected.

Transmission Protocol



CONVERT MODE

In this mode, only the write command (W) for writing to and the read command (R) for reading from the system area are available for communication.

EXTEND MODE

In this mode, not only the read and the write commands but also the draw and some other commands are available.
When you select this mode, you need to select the communication options given in the rectangular box.

◆COMMUNICATION PROTOCOL

Select CONVERT MODE or EXTEND MODE.

- When you select CONVERT MODE, you do not need to select the communication options given in the rectangular box.

◆UNIT NO. (0-31)

Specify the number of the GP unit that will communicate with the host. Make sure that the number you specify matches that specified as one of the communication protocol options. Data is transferred between the host and the GP unit only when these numbers match.

- ***Enter "0" when a single GP unit communicates with the host.***
- ***You can connect up to 32 GP units to a single host. Be sure to assign a unique number to each GP unit. Improper assignment of numbers can result in improper communication.***

◆TRANSMISSION TYPE

Select any of the following:

1:1	ASCII	Data is exchanged between a single host and a single GP unit. Data is converted into ASCII codes before transmission.
1:1	BINARY	Data is exchanged between a single host and a single GP unit. Data is converted into binary codes before transmission.
1:n	ASCII	Data is exchanged between a single host and 'n' GP units. Data is converted into ASCII codes before transmission.
1:n	BINARY	Data is exchanged between a single host and 'n' GP units. Data is converted into binary codes before transmission.

◆ETX. SUM CHECK

You can select whether to add a sum check code to each data block or not, by toggling ENABLED or DISABLED.

◆TERMINATOR

You can select whether to use 'CR' or 'CR-LF' as the end code.

CR: CR (carriage return) is used as the end code.

CR-LF: CR (carriage return) and LF (line feed) are used as the end codes.

Note: This option is available when you select 1:1 ASCII or 1:n ASCII.

◆ACK

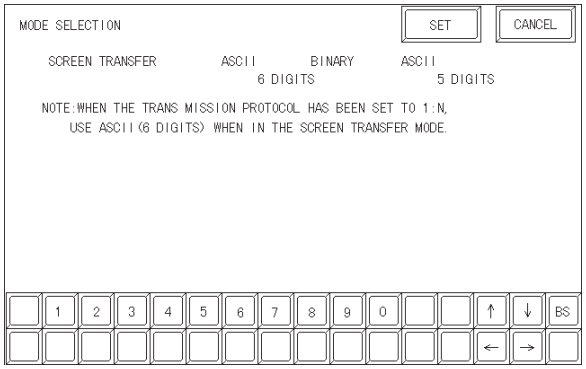
You can select whether to send ACK when data is received with no errors.

◆NAK

You can select whether to send NAK if an error occurs during receipt of data.

(For error codes displayed in case of NAK Reference *Chapter 6 Error Messages)*

MODE SELECT



6 DIGITS ASCII

Select this mode when you have selected 1:n for COMMUNICATION TYPE.

BINARY

Select this mode when you wish to transfer screen data using GP-PRO/PBIII.

5 DIGITS ASCII

Select this mode only when you need to use the GP-430 screens.

2-2 Data Transmission

Each data block exchanged between the host and a GP unit contains a command, control codes, and related data arranged in a preset sequence to ensure proper communication. This section describes the data blocks used for communication as well as the codes included in these blocks for data transmission control.

■Data Block Types and Control Codes

The data blocks and the control codes covered in this manual are as follows:

◆Data Blocks

There are three types of data blocks, each of which contains a command, control codes, and related data arranged in a preset sequence.

Command data block: Data block containing a command from the host
Response data block: Data block containing a response from the GP
Interrupt output data block: Data block containing GP touch panel input

◆Control Codes:

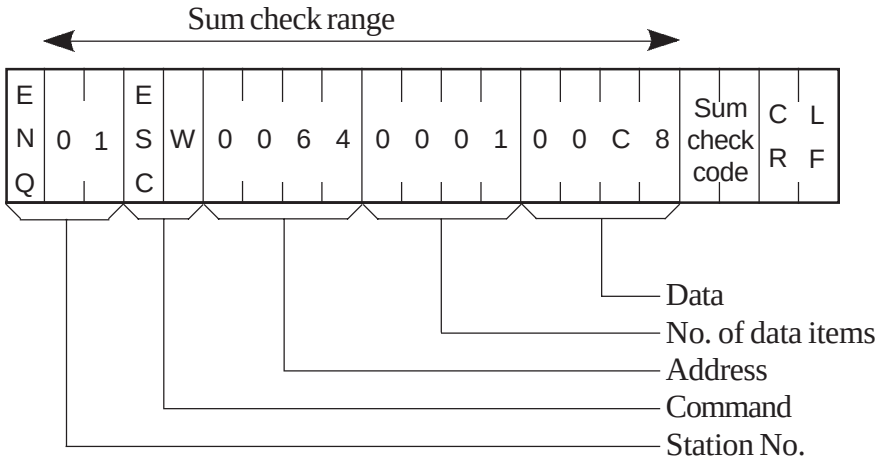
Code Name	Code (HEX)	Description
STX (Start of Text)	02	Start of response
ETX (End of Text)	03	End of response
ENQ (Enquiry)	05	Start of text
ACK (Acknowledge)	06	Data received with no errors
LF (Line Feed)	0A	Text terminator
CR (Carriage Return)	0D	
NAK (Negative Acknowledge)	15	Error during receipt of data
ESC (Escape)	1B	Start of command
DLE (Data Link Escape)	10	

◆Sum Check Code:

The sum check code is the lower byte (8 bits) of the sum of all data included in the sum check range.

In the ASCII mode, data is converted into ASCII code before summing. Then, the lower 2 digits of the hexadecimal sum of all data is used as the sum check code.
In the binary mode, the lower byte of the sum of all data is used as the sum check code.

<Example> Extended Mode , 1:n ASCII
The following data block writes "200" (decimal) to address 100 in the system area:



ASCII 30H + 31H + 1BH + 57H + 30H + 30H + 36H + 34H + 30H + 30H +
(0) (1) (ESC) (W) (0) (0) (6) (4) (0) (0)
ASCII 30H + 31H + 30H + 30H + 43H + 38H
(0) (1) (0) (0) (C) (8)

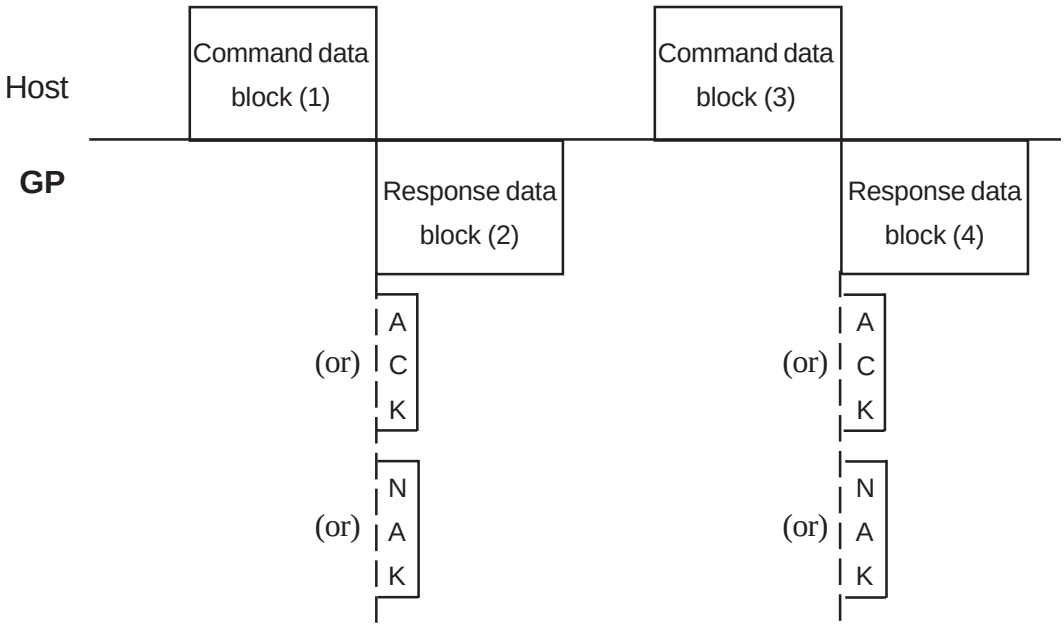
=339H
└── Lower 2 digits

Thus, the sum check code, represented by the 2 lower digits of the sum, is 39(33H, 39H).

■Basic Data Transmission Control Procedures

The following shows the basic procedures for data transmission control:

◆When the host sends data to a GP



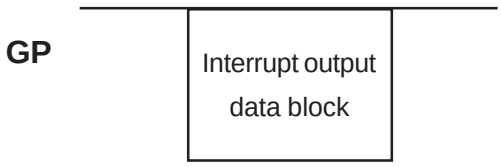
(1) and (3) (Command data block) : The host sends a command data block to the GP unit. The GP unit checks the command data block to determine whether to send the response data block ((2) and (4)) , ACK, NAK, or send no response at all.

Important

Be sure to send the command data (3) from the host to the GP unit only after receiving the response data (2) from the GP unit to the host. Sending the next command data from the host to the GP unit without first receiving the GP unit's response can cause a system error after a few hours of operation.

◆When the GP sends data to the Host

(1) Host



The GP unit sends an interrupt output data block when it receives touch panel input.

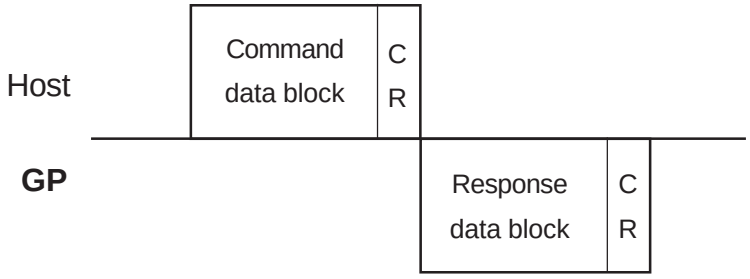
Note: Interrupt output is not possible when using 1:n or 2-wire connection.

Reference *3-12 Interrupt Output Requests [ESC I (large I)] ■Extend Mode, 1:1 ASCII,*

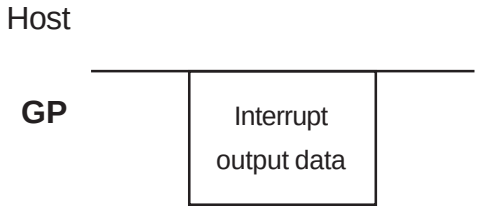
■Convert Mode Communication

Communication between the host and the GP unit is performed as shown below when you select "CONVERT MODE" as the communication protocol mode.

◆Host to GP



◆GP to Host (interrupt output)



Note:

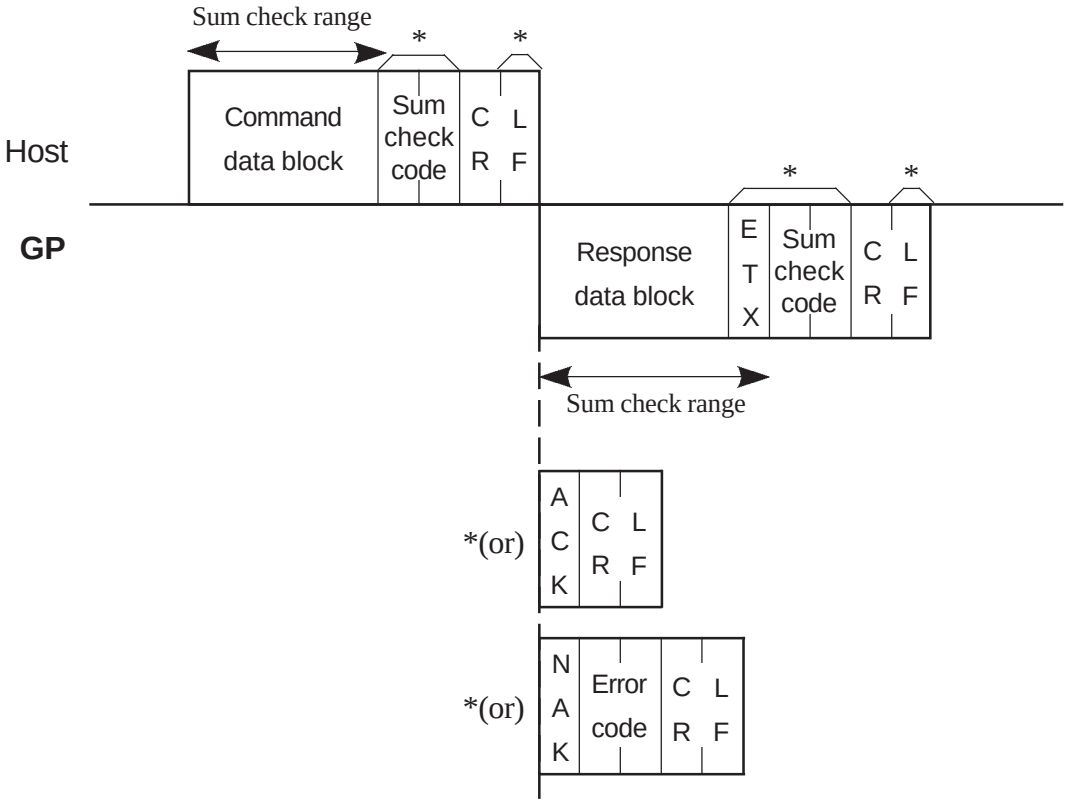
- Interrupt output data is the lower 8-bits of the 16-bit data written to address 10 of the system area.
- Convert mode command and response data must be of ASCII mode only. Interrupt output data is of binary format.
- Interrupt output data cannot be used when using a 2-wire type of connection.

■Extended Mode Communication

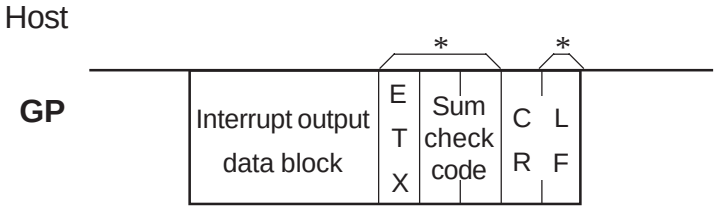
Communication between the host and the GP is achieved as shown below when you select "EXTEND MODE" as the communication protocol mode.

◆1:1 ASCII

<Host to GP>



<GP to Host>

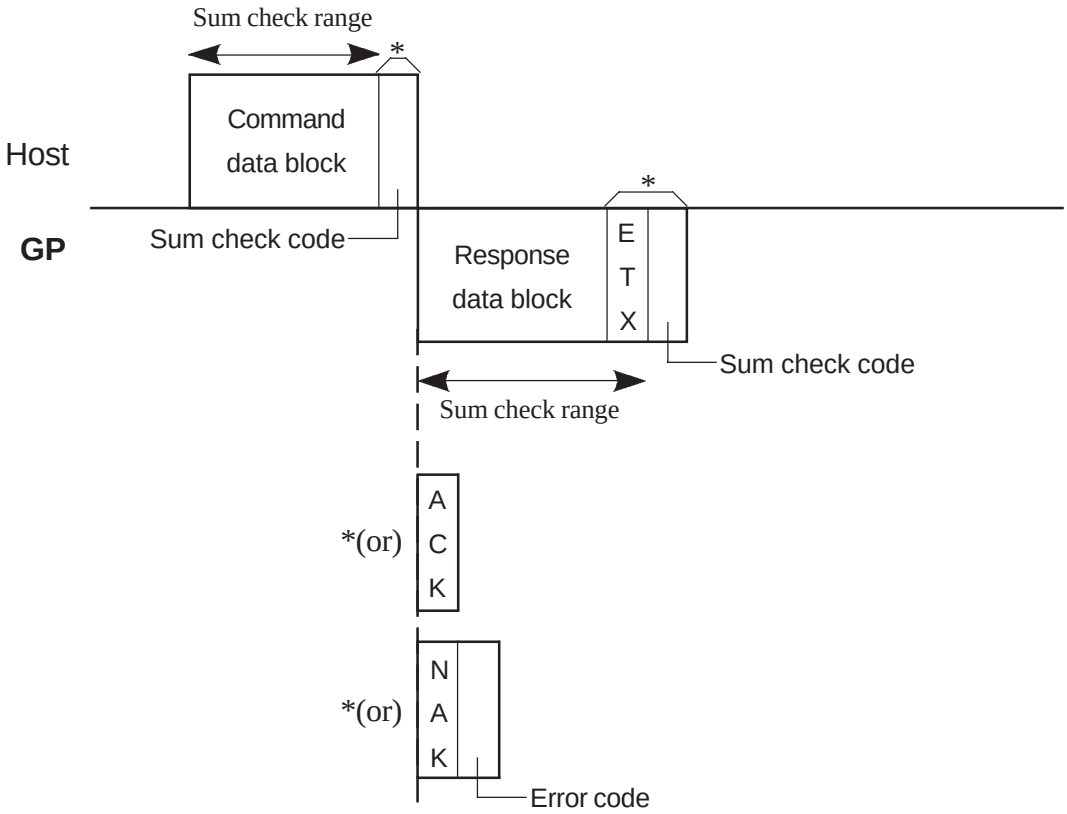


Note:

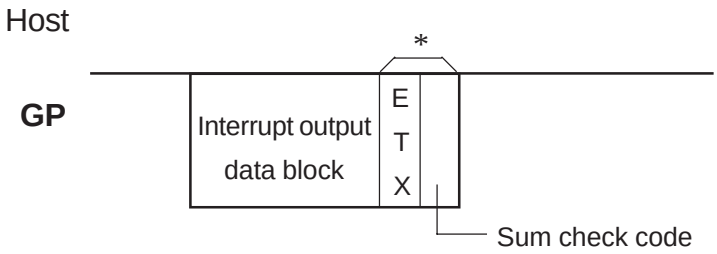
- *The "*" indicates that depending on GP settings, this block may or may not be added to the communications protocol.*
- *When using a 2-wire type of connection, be sure to use an "Interrupt Output Request" data block for the interrupt output.*

◆1:1 BINARY

<Host to GP>



<GP to Host>

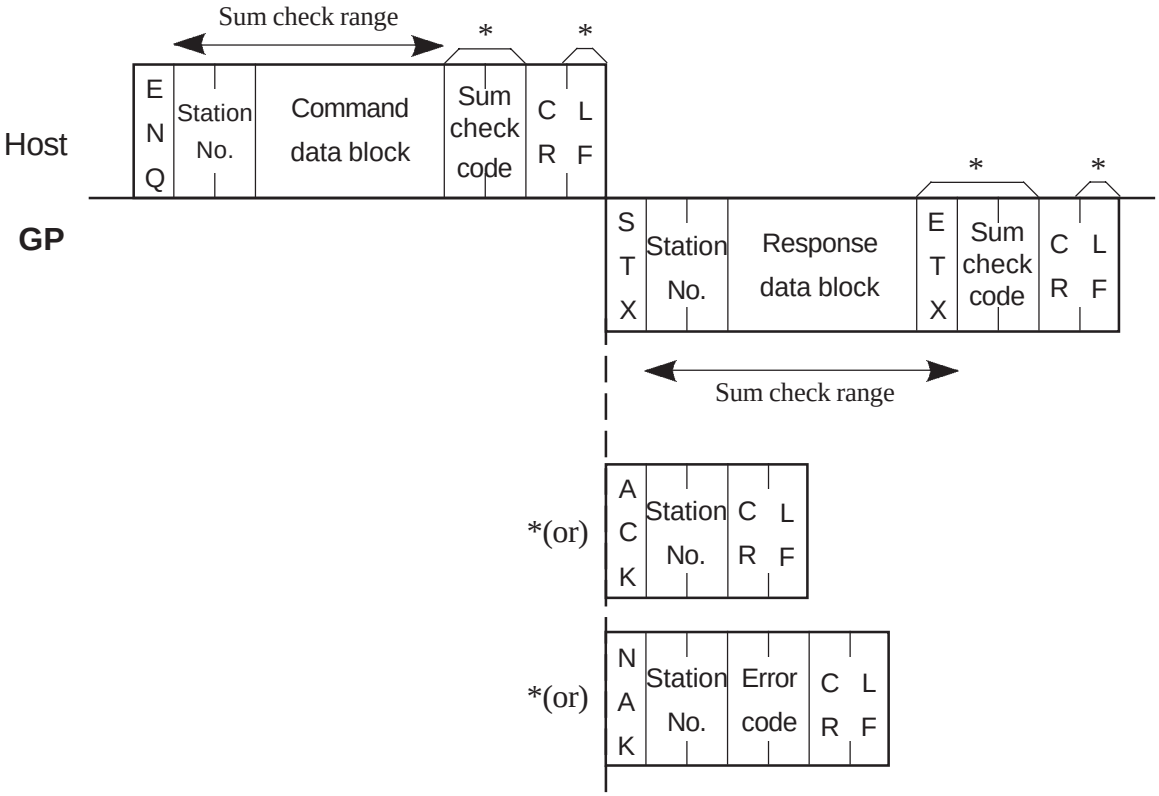


Note:

- *The "*" indicates that depending on GP settings, this block may or may not be added to the communications protocol.*
- *The XON/XOFF flow control method is not available in the binary mode. Use the ER flow control method and be sure to select "ENABLED" for either ACK or NAK (both can also be "ENABLED").*
- *When using a 2-wire type of connection, be sure to use an "Interrupt Output Request" data block for the interrupt output.*

◆1:n ASCII

<Host to GP>

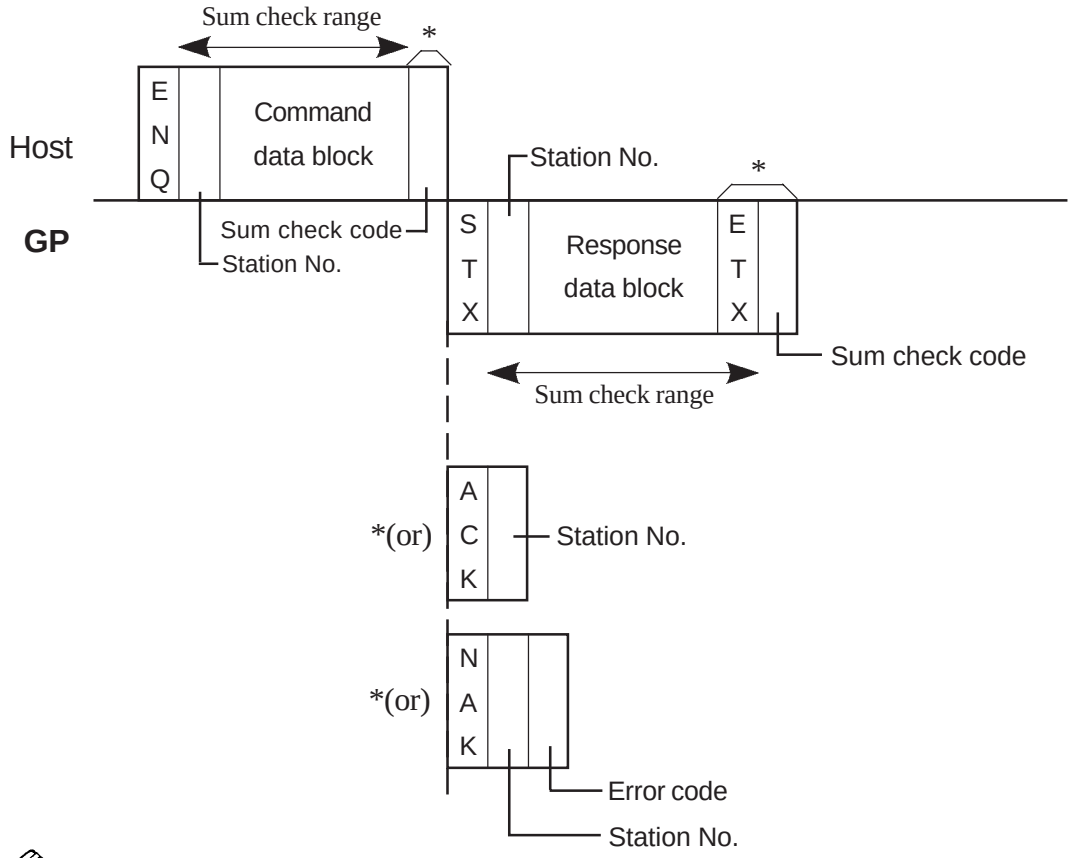


Note:

- *The "*" indicates that depending on GP settings, this block may or may not be added to the communications protocol.*
- *When you enter "FF" for the station No., the same data can be sent to all stations (GP units) at the same time. However, neither ACK nor NAK will be sent to the host in this case. After sending a data block, be sure there is a delay of at least 100ms before sending the next data block.Note also that you cannot use the system area read command (ESC R).*
- *In the 1:n ASCII or BINARY mode, the interrupt output enquiry command (ESC I) is used in the interrupt output data block.*

◆1:n BINARY

<Host to GP>

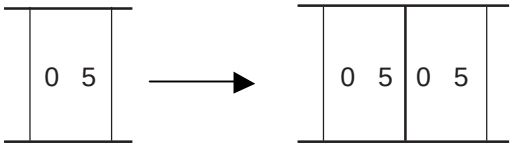


Note:

- *The "*" indicates that depending on GP settings, this block may or may not be added to the communications protocol.*
- *When you enter "FF" for station No., the same data can be sent to all stations (GP units) at the same time. However, neither ACK nor NAK will be sent to the host in this case.*
  After sending a data block, be sure there is a delay of at least 100ms before sending the next data block.
  Note also that you cannot use the system area read command (ESC R).
- *The XON/XOFF flow control method is not available in the binary mode. Use the ER flow control method and be sure to select "ENABLED" for either ACK or NAK (both can also be "ENABLED").*
- *In the 1:n ASCII or BINARY mode, the interrupt output enquiry command (ESC I) is used in the interrupt output data block.*

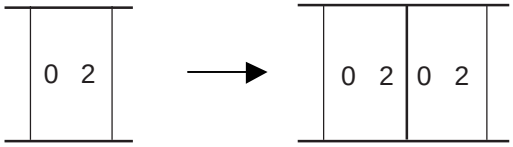
■When Using Binary Communication Method

- If "05h" (same as the ENQ control code) is included in the sum check range or sum check code, add another "05h" immediately before "05h" prior to sending the data.



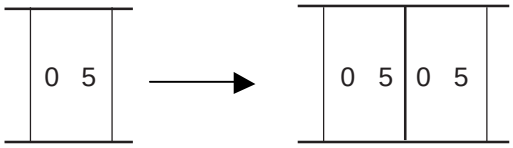
Be sure that the additional 05h is not included in the Number of data packets.

- If "02h" (same as the STX control code) is included in the sum check range or sum check code, the GP adds another "02h" immediately before "02h" prior to sending the data.



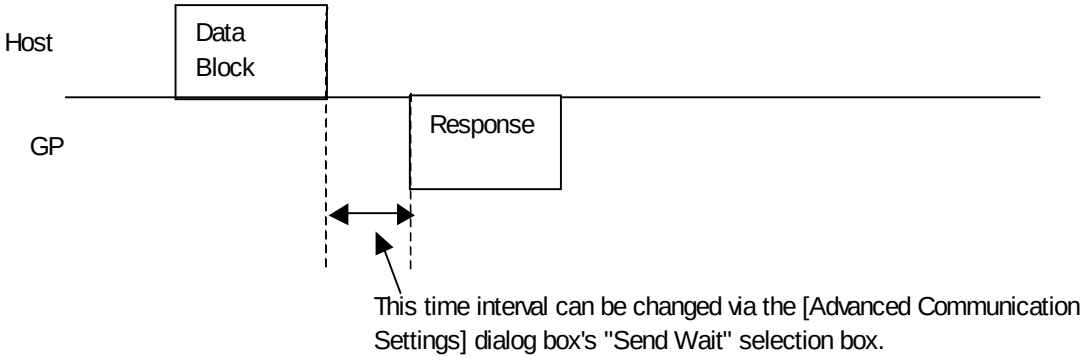
Be sure that the additional 02h is not included in the Number of data packets.

- When using a 2-wire 1:1 connection, if "05h" (same as the ENQ control code) is included in the sum check range or sum check code, add another "05h" immediately before "05h" prior to sending the data.

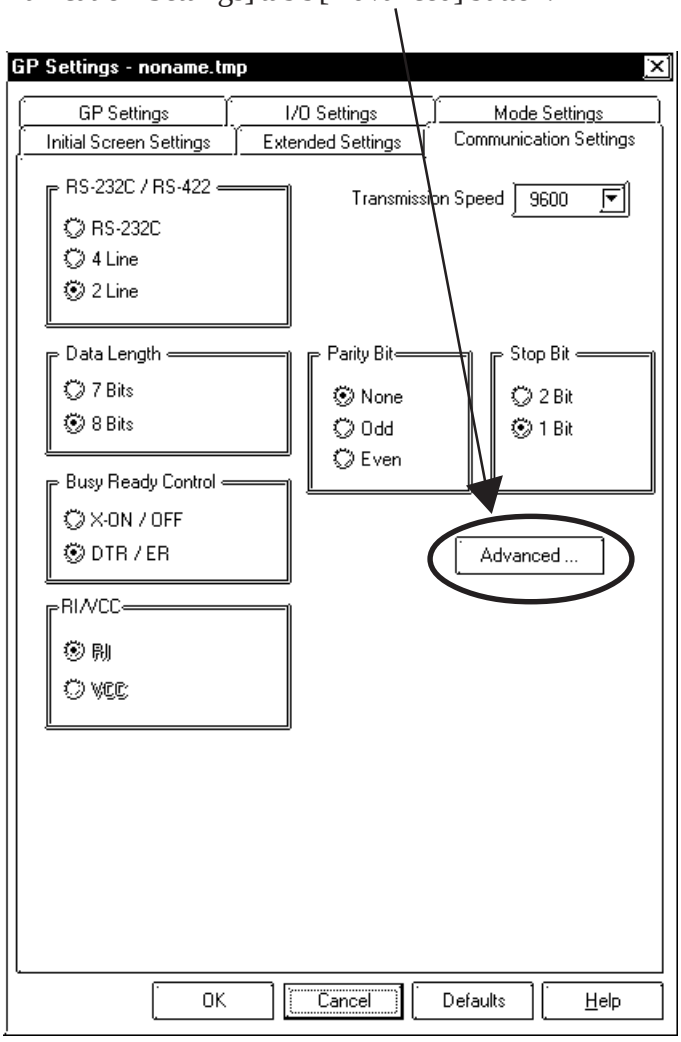


■Send Wait Settings

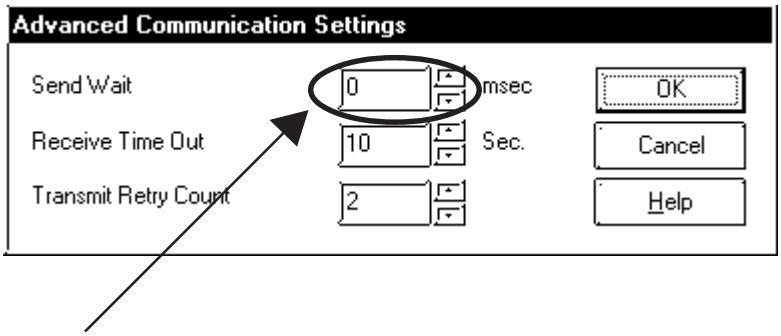
You can change the GP unit's response time.
Use this feature to set the GP unit's response time after the host has sent a block of data to the GP.



In GP-PRO/PBIII for Windows, click [Screen/Setup] -> [GP Setup], and click the [Communication Settings] tab's [Advanced] button.



In the following [Advanced Communication Settings] dialog box, enter or select the desired [Send Wait] time.



- **Send Wait**
  Indicates the delay before the GP sends a block of data to the host.
  Enter or select a value between 0 and 255 ms (Default setting: 0).
  Be sure the Send Wait time conforms to host specifications.

MEMO

Chapter 3: Command Data

This chapter describes the format of a command data block sent from the host to GP, and the format of a response data block sent from a GP, in response to the command data block. The formats of these blocks vary depending on the selected mode (ASCII or binary). The following shows the data block format used for each command in both the ASCII and binary modes:

3-1 Display Command Data

The list given below shows the display commands, including the read command (for reading from the system area), write command (for writing to the system area), and the graphics input command.

■Command list

Command	Description	Additional Features
ESC W	Writes data to the System Area.	
ESC R	Reads data from the System Area.	
ESC T*1	Displays a character string.	
ESC L*1	Displays a line.	
ESC B*1	Displays a rectangle.	
ESC S*1	Displays a filled rectangle.	
ESC C*1	Displays a circle.	
ESC A*1	Displays an arc.	
ESC G*1	Displays a pie shape.	
ESC P*1	Filling an object.	
ESC I (large I)	Interrupt Output Requests.	
ESC t*1	Displays a character string.	Rotation, Direction, Highlighting
ESC l (small L)*1	Displays a line.	Arrow
ESC b*1	Displays a rectangle.	Filling/Patterning, Beveling
ESC s*1	Displays a filled rectangle.	Filling/Patterning, Beveling
ESC c*1	Displays a circle.	Patterning
ESC g*1	Displays a pie shape.	Patterning
ESC #	Brightness/Contrast adjustment.	
ESC $	Brightness/Contrast Settings.	

Note:

- For control codes used for the command data block:
  Reference *2-2 Data Transmission*
- The maximum allowable X and Y coordinates for draw commands vary depending on the GP model used.

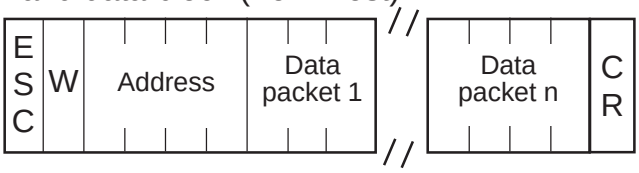
*1 64 color, 256 color and 3-speed blink features cannot be used.

3-2 Writing Data to the System Area [ESC W]

The host uses the ESC W command to write data to desired addresses in the system area. The format of the command data block containing the ESC W command (system area write command) is shown below.

■Convert Mode

Command data block (from Host)



<Setting range>

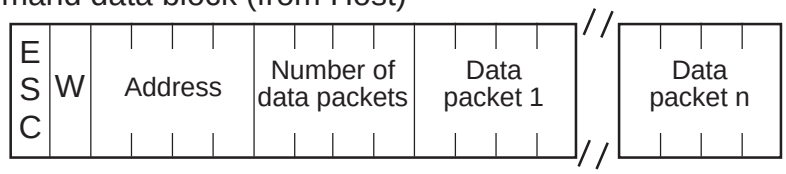
- Address : 0000H to 1FFFH (0 to 8191)
- Data : 0000H to FFFFH

Note:

- Be sure all data entered is in ASCII format.
- Data is written in order, starting from the specified write address.
- In convert mode there is no response data block sent from the GP.
- Issuing write commands in succession without an intermediate delay may cause GP screen display to fail to update.

■Extend Mode, ASCII

Command data block (from Host)



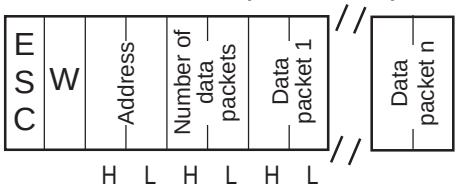
<Setting range>

- Address : 0000H to 1FFFH (0 to 8191)
- Number of data packets : 0001H to 0040H (1 to 64)
- Data : 0000H to FFFFH

GP Response data block (from GP)
ACK or NAK

■Extend Mode, Binary Mode

Command data block (from Host)



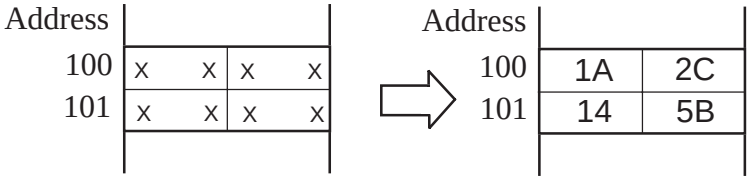
<Setting range>

- Address : 0000H to 1FFFH (0 to 8191)
- Number of data packets : 0001H to 0040H (1 to 64)
- Data : 0000H to FFFFH

GP Response data block from the GP
ACK or NAK

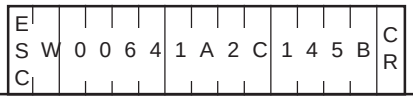
<Example>

The host writes "1A2CH" and "145BH" to addresses 100 and 101 in the system area, respectively.



■Convert Mode

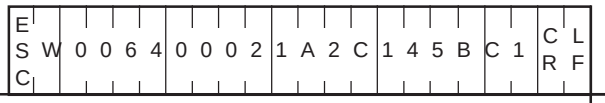
Host



■Extend Mode, 1:1 ASCII,

ETX. SUMCHECK: ENABLED, TERMINATOR: CR.LF, ACK: ENABLED, NAK: ENABLED

Host

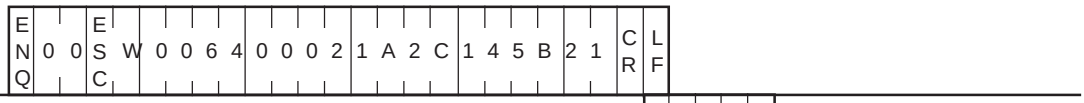


GP

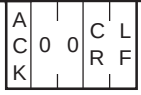
■Extend Mode, 1:n ASCII,

ETX. SUMCHECK: ENABLED, TERMINATOR: CR.LF, ACK: ENABLED, NAK: ENABLED

Host



GP

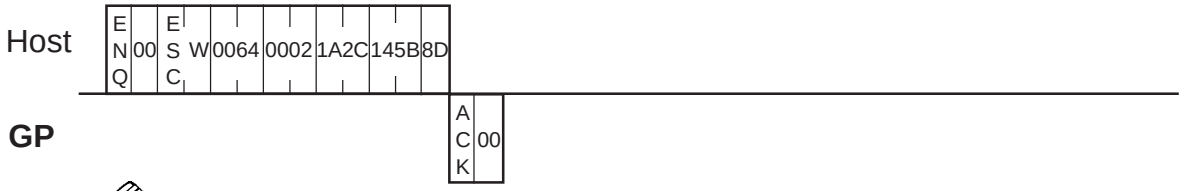


■Extend Mode, 1:1 BINARY,

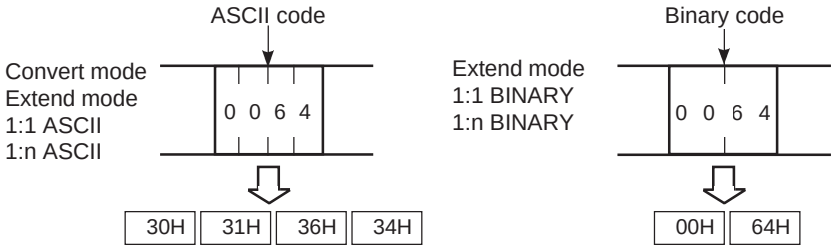
ETX. SUMCHECK: ENABLED, ACK: ENABLED, NAK: ENABLED

■Extend Mode, 1:n BINARY,

ETX. SUMCHECK: ENABLED, ACK: ENABLED, NAK: ENABLED



Note: ***The following shows the difference between ASCII and binary codes:***

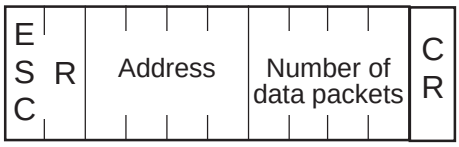


3-3 Reading Data from the System Area [ESC R]

The host uses the ESC R command to read data from desired addresses in the system area. The format of the command data block containing the ESC R command (system area read command) is shown below.

■Convert Mode

Command data block (from Host)

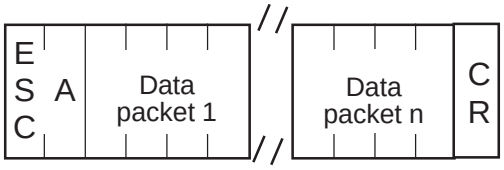


<Setting range>
- Address : 0000H to 1FFFH (0 to 8191)
- Number of data packets : 0001H to 0040H (1 to 64)

Note: Be sure to make all data entries in ASCII code format.

GP Response data block (from GP)

◆ When there is no error

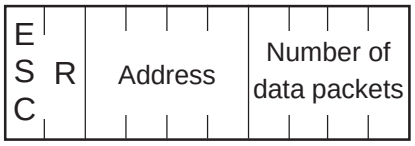


<Setting range>
- Data : 0000H to FFFFH

◆If an error occurs
NAK response.

■ Extend Mode, ASCII

Command data block (from Host)

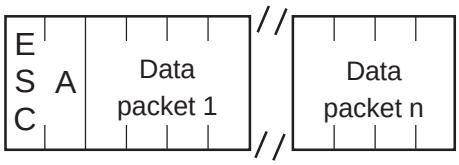


<Setting range>
- Address : 0000H to 1FFFH (0 to 8191)
- Number of data packets : 0001H to 0040H (1 to 64)

Note: Be sure to make all data entries in ASCII code format.

GP Response data block (from GP)

◆ When there is no error



<Setting range>

• Data : 0000H to FFFFH

◆If an error occurs

NAK response.

■ Extend Mode, Binary

Command data block (from Host)

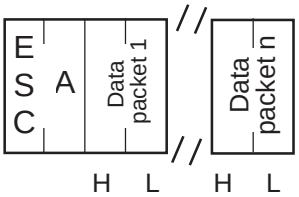
<Setting range>

• Address : 0000H to 1FFFH (0 to 8191)

• Number of data packets : 0001H to 0040H (1 to 64)

GP Response data block (from GP)

◆ When there is no error



<Setting range>

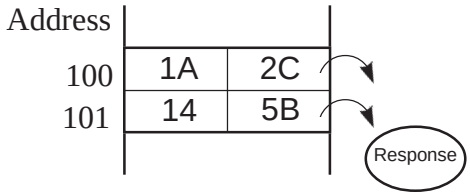
• Data : 0000H to FFFFH

◆ If an error occurs

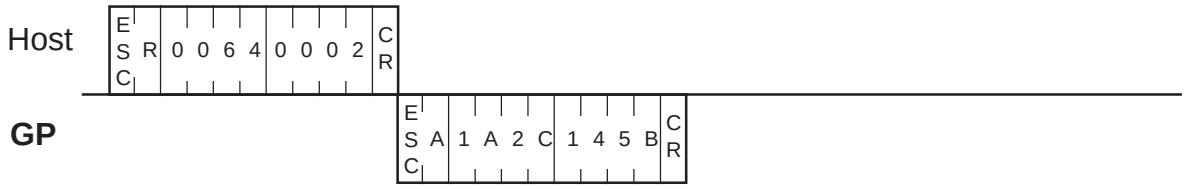
NAK response.

<Example>

The host reads hexadecimal data of 2 words in length from addresses 100 and 101 in the system area.

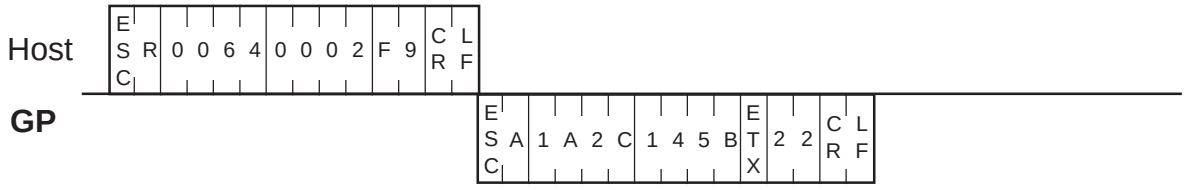


■Convert Mode



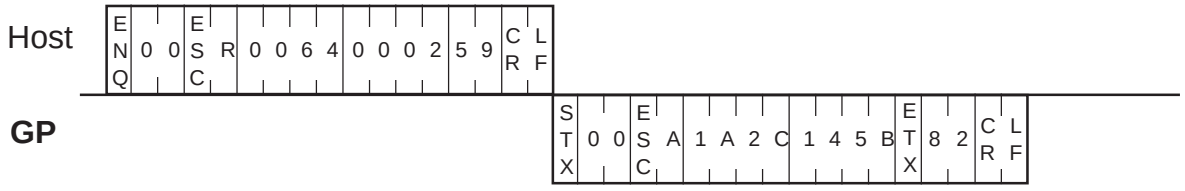
■Extend Mode, 1:1 ASCII,

ETX. SUMCHECK: ENABLED, TERMINATOR: CR.LF, ACK: ENABLED, NAK: ENABLED



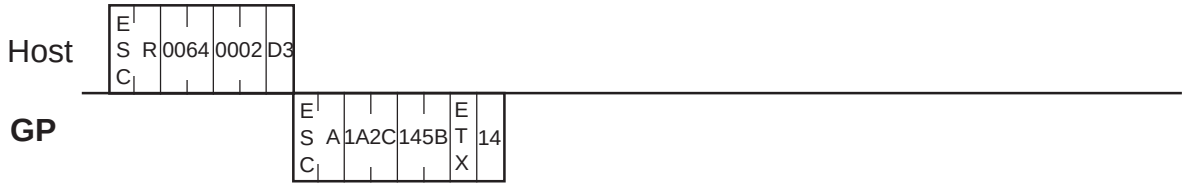
■Extend Mode, 1:n ASCII,

ETX. SUMCHECK: ENABLED, TERMINATOR: CR.LF, ACK: ENABLED, NAK: ENABLED



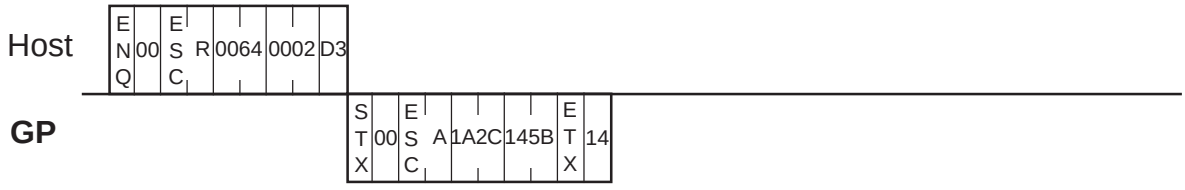
■Extend Mode, 1:1 BINARY,

ETX. SUMCHECK: ENABLED, ACK: ENABLED, NAK: ENABLED



■Extend Mode, 1:n BINARY,

ETX. SUMCHECK: ENABLED, ACK: ENABLED, NAK: ENABLED

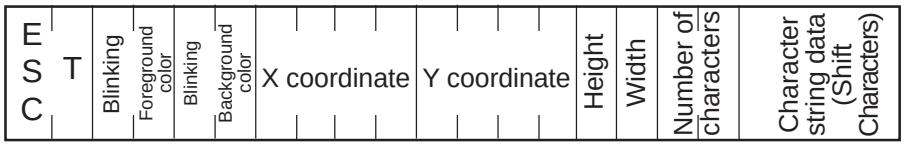


3-4 Displaying a Character String [ESC T]

The format of the command data block containing the ESC T command (character string display command) is shown below.

■ASCII Mode

Command data block (from Host)



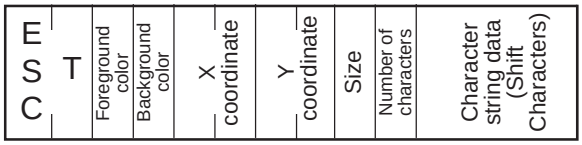
<Setting range>

- Blinking : 0 or 1 (0: Disabled, 1: Enabled)
- Foreground/Background color : 0 to 7 (0: Black, 1: Blue, 2: Green, 3: Light blue, 4: Red, 5: Purple, 6: Yellow, 7: White)
- Height, Width : 0 to 3 (0: 1 x, 1: 2 x, 2: 4 x, 3: 8 x)
- X coordinate : 0000 to 0799 (0 to 799)
- Y coordinate : 0000 to 0599 (0 to 599)
- Number of characters (bytes) : 01 to 80 (1 to 80)
- Character string data : ANK character is 1-byte long. All double-sized characters are 2-bytes long.

GP Response data block (from GP)
ACK or NAK

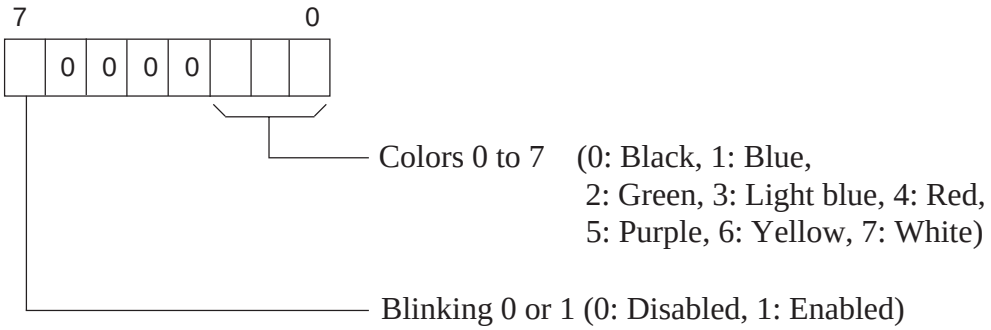
■Binary Mode

Command data block (from Host)

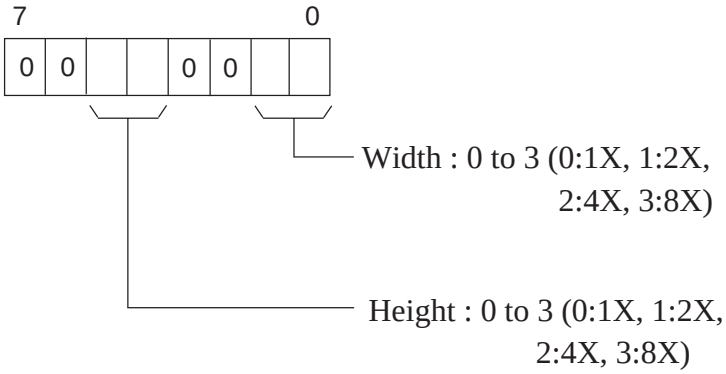


<Setting range>

- Foreground/Background color



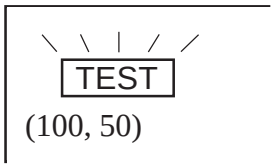
- X coordinate : 0000H to 031FH (0 to 799)
- Y coordinate : 0000H to 0257H (0 to 599)
- Size



- Number of characters (bytes) : 01H to 50H (1 to 80)
- Character string data : ANK character is 1 byte long. All double-sized characters are 2 bytes long.

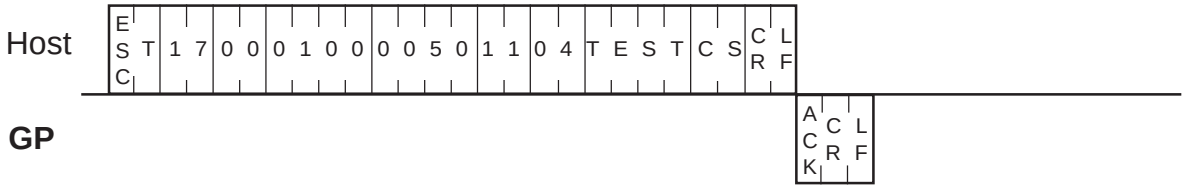
GP Response data block (from GP)
ACK or NAK

<Example>
"TEST" appears blinking to the right of point (100, 50).

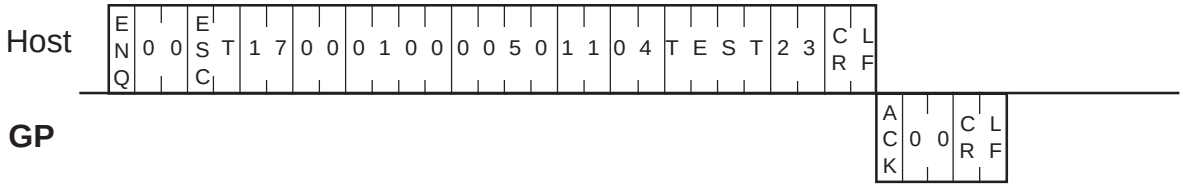


(Attribute)
Character size : 2 X 2

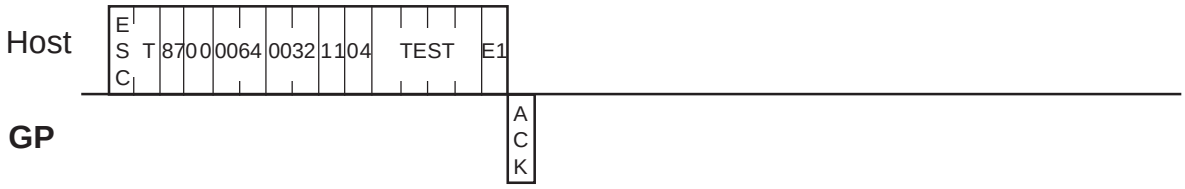
■Extend Mode, 1:1 ASCII,
ETX. SUMCHECK: ENABLED, TERMINATOR: CR.LF, ACK: ENABLED, NAK: ENABLED



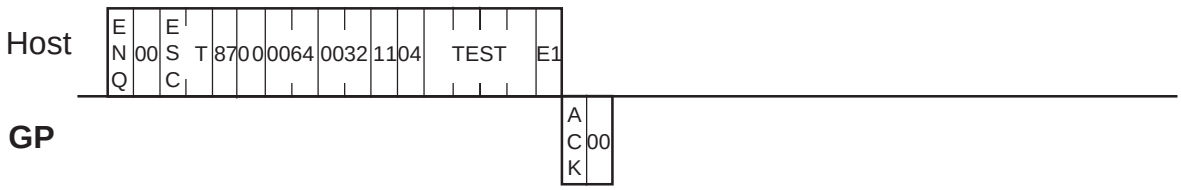
■Extend Mode, 1:n ASCII,
ETX. SUMCHECK: ENABLED, TERMINATOR: CR.LF, ACK: ENABLED, NAK: ENABLED



■Extend Mode, 1:1 BINARY,
ETX. SUMCHECK: ENABLED, ACK: ENABLED, NAK: ENABLED



■Extend Mode, 1:n BINARY,
ETX. SUMCHECK: ENABLED, ACK: ENABLED, NAK: ENABLED



3-5 Displaying a Line [ESC L]

The format of the command data block containing the ESC L command (straight line display command) is shown below.

■ASCII Mode

Command data block (from Host)

ESC	L	Blinking	Foreground color	Blinking	Background color	0	Line type	Start point X coordinate	Start point Y coordinate	End point X coordinate	End point Y coordinate

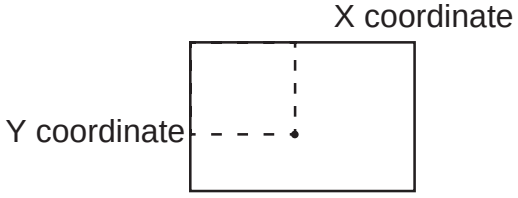
<Setting range>

- Blinking : 0 or 1 (0: Disabled, 1: Enabled)
- Foreground/ Background color : 0 to 7 (0: Black, 1: Blue, 2: Green, 3: Light blue, 4: Red, 5: Purple, 6: Yellow, 7: White)
- Line type : 0 to 7 (0: ———— , 1: _ _ _ _ ,2: — - — ,3: — — — 4: ———— ,5: _ _ _ _ ,6: — - — ,7: — — —)
- X coordinate : 0000 to 0799 (0 to 799)
- Y coordinate : 0000 to 0599 (0 to 599)

GP Response data block (from GP)
ACK or NAK

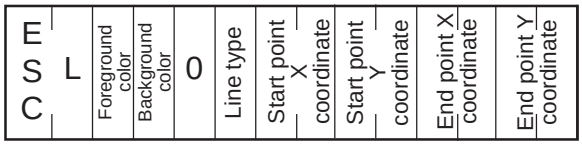
Note:

- Line types 0 through 3 are 1-dot line segments, while line types 4 through 7 are 2-dot line segments.
- When you wish to draw a point, be sure to specify the same value for the start and end points of the X coordinates, and the same value for the start and end points of the Y coordinates.



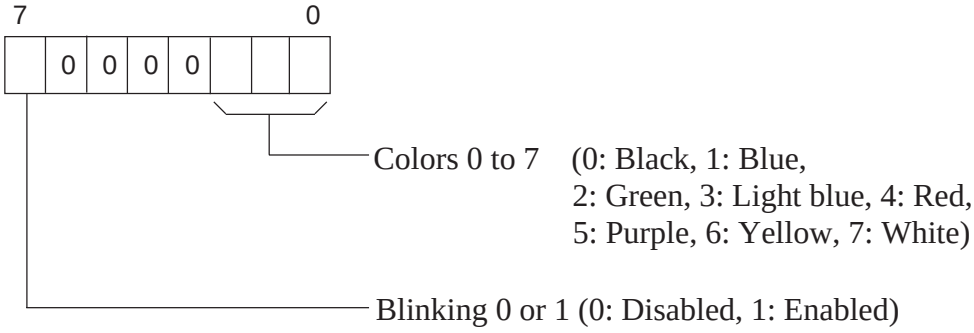
■Binary Mode

Command data block (from Host)



<Setting range>

- Foreground/Background color

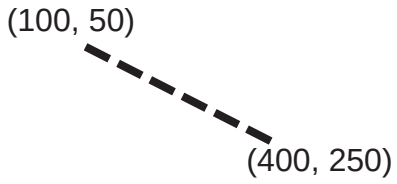


- Line type : 00H to 07H (00:———, 01:- - - - ,02: — - —, 03: — - - —,
  04:———, 05:- - - - ,06: — - —, 07: — - —,)
- X coordinate : 0000H to 031FH (0 to 799)
- Y coordinate : 0000H to 0257H (0 to 599)

GP Response data block (from GP)
ACK or NAK

<Example>

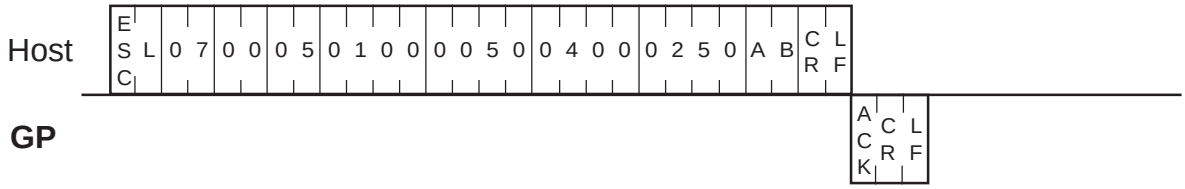
A dotted line (2-dot) is drawn between two points (100, 50) and (400, 250).



(Attributes)
Foreground color:
White
Background color:
Black

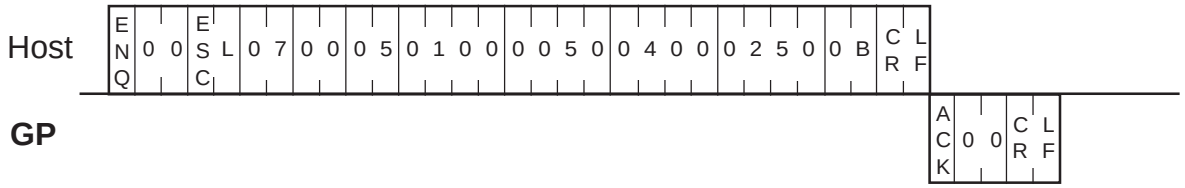
■Extend Mode, 1:1 ASCII,

ETX. SUMCHECK: ENABLED, TERMINATOR: CR.LF, ACK: ENABLED, NAK: ENABLED



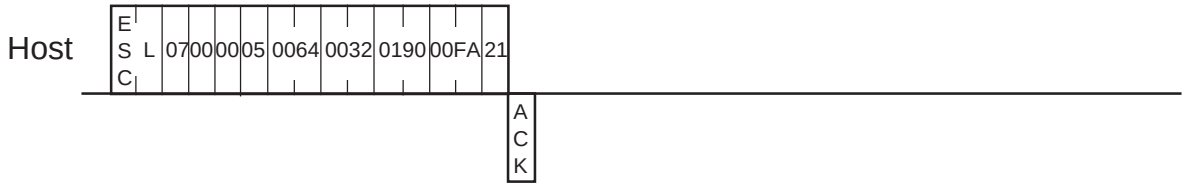
■Extend Mode, 1:n ASCII,

ETX. SUMCHECK: ENABLED, TERMINATOR: CR.LF, ACK: ENABLED, NAK: ENABLED



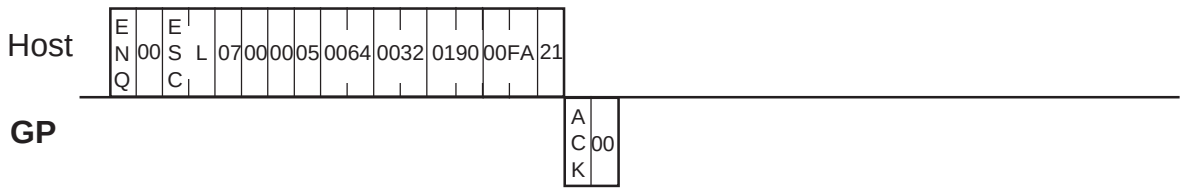
■Extend Mode, 1:1 BINARY,

ETX. SUMCHECK: ENABLED, ACK: ENABLED, NAK: ENABLED



■Extend Mode, 1:n BINARY,

ETX. SUMCHECK: ENABLED, ACK: ENABLED, NAK: ENABLED

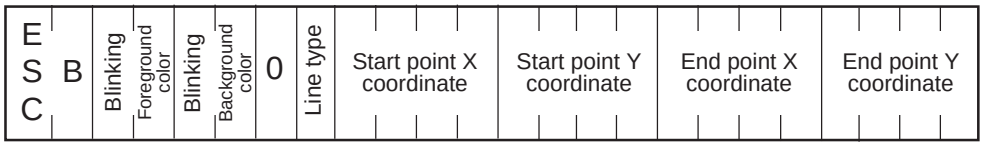


3-6 Displaying a Rectangle [ESC B]

The format of the command data block containing the ESC B command (rectangle display command) is shown below.

■ASCII Mode

Command data block (from Host)



<Setting range>

- Blinking : 0 or 1 (0: Disabled, 1: Enabled)
- Foreground/Background color : 0 to 7 (0: Black, 1: Blue, 2: Green, 3: Light blue, 4: Red, 5: Purple, 6: Yellow, 7: White)
- Line type : 0 to 3 (0:_____ ,1:_ _ _ _ ,2:__ - __,3:__ - - __)
- X coordinate : 0000 to 0799 (0 to 799)
- Y coordinate : 0000 to 0599 (0 to 599)

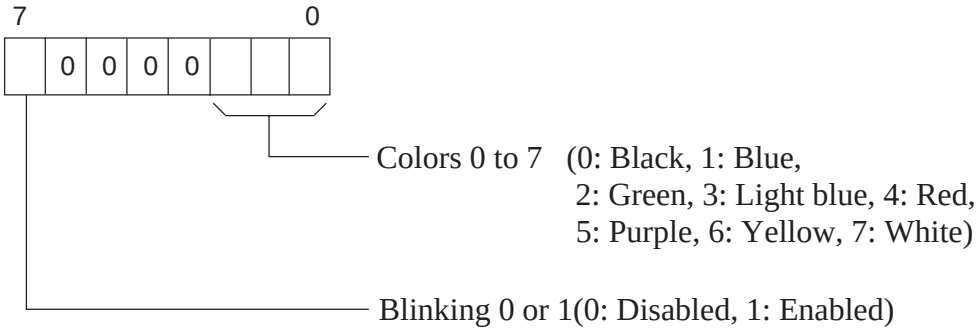
GP Response data block (from GP)
ACK or NAK

■Binary Mode

Command data block (from Host)

<Setting range>

- Foreground/Background color

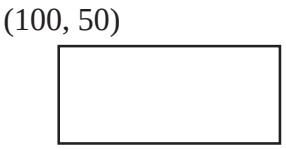


- Line type : 0 to 3 (0:——,1:– – – –,2: — - —,3:— - - —)
- X coordinate : 0000H to 031FH (0 to 799)
- Y coordinate : 0000H to 0257H (0 to 599)

GP Response data block (from GP)
ACK or NAK

<Example>

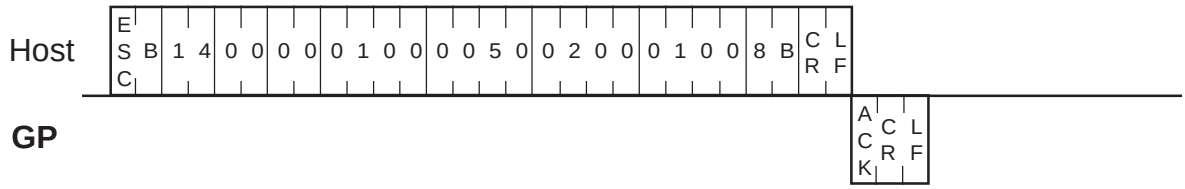
A rectangle is drawn with the upper left and lower right corners located respectively at points (100, 50) and (200, 100).



(Attributes)
Foreground color: Red, Blinking: Enabled
Background color: Black, Blinking: Disabled
Line type: Solid line (0)

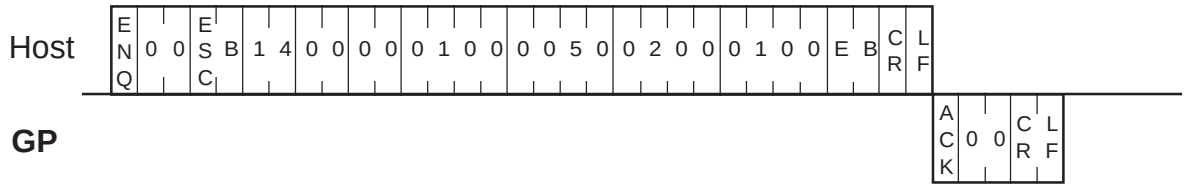
■Extend Mode, 1:1 ASCII,

ETX. SUMCHECK: ENABLED, TERMINATOR: CR.LF, ACK: ENABLED, NAK: ENABLED



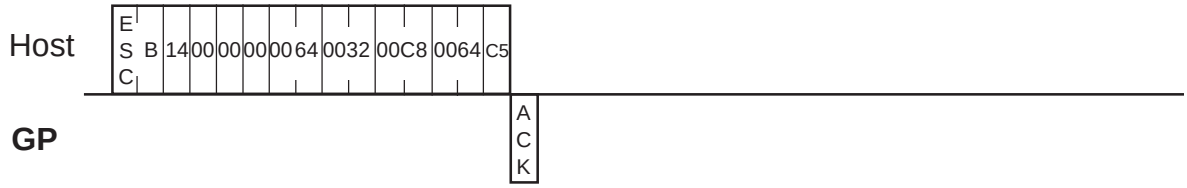
■Extend Mode, 1:n ASCII,

ETX. SUMCHECK: ENABLED, TERMINATOR: CR.LF, ACK: ENABLED, NAK: ENABLED



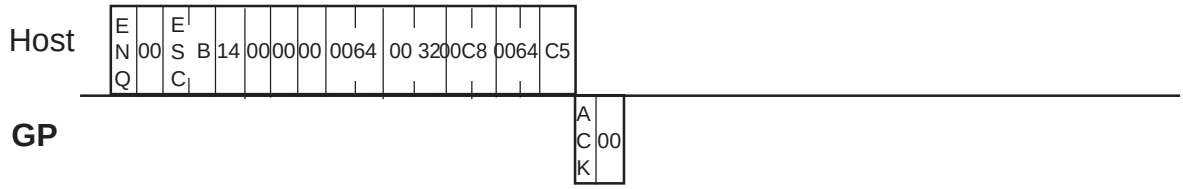
■Extend Mode, 1:1 BINARY,

ETX. SUMCHECK: ENABLED, ACK: ENABLED, NAK: ENABLED



■Extend Mode, 1:n BINARY,

ETX. SUMCHECK: ENABLED, ACK: ENABLED, NAK: ENABLED



3-7 Displaying a Filled Rectangle [ESC S]

The format of the command data block containing the ESC S command (filled rectangle display command) is shown below.

■ASCII Mode

Command data block (from Host)

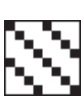
ESC	S	Blinking	Foreground color	Blinking	Background color	0	Tiling pattern	Start point X coordinate	Start point Y coordinate	End point X coordinate	End point Y coordinate

<Setting range>

- Blinking : 0 or 1 (0: Disabled, 1: Enabled)
- Foreground/Background color : 0 to 7 (0: Black, 1: Blue, 2: Green, 3: Light blue, 4: Red, 5: Purple, 6: Yellow, 7: White)
- Tiling pattern : 0 to 8 (*1)
- X coordinate : 0000 to 0799 (0 to 799)
- Y coordinate : 0000 to 0599 (0 to 599)

GP Response data block (from GP)
ACK or NAK

*1 Tiling Pattern Types

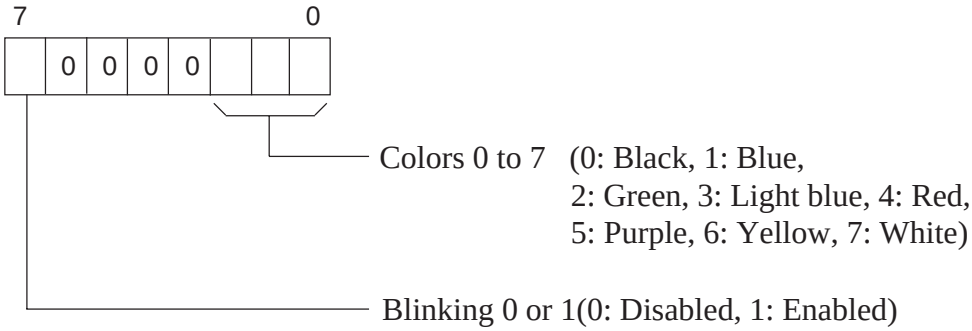
Tiling pattern No.	Tiling pattern	Tiling pattern No.	Tiling pattern	Tiling pattern No.	Tiling pattern
0		3		6	
1		4		7	
2	 8 dots 8 dots	5		8	

■Binary Mode

Command data block (from Host)

<Setting range>

- Foreground/Background color

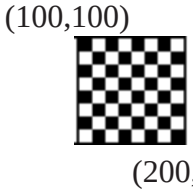


- Tiling pattern : 00H to 08H (See the tiling patterns on P.3-17)
- X coordinate : 0000H to 031FH (0 to 799)
- Y coordinate : 0000H to 0257H (0 to 599)

GP Response data block (from GP)
ACK or NAK

<Example>

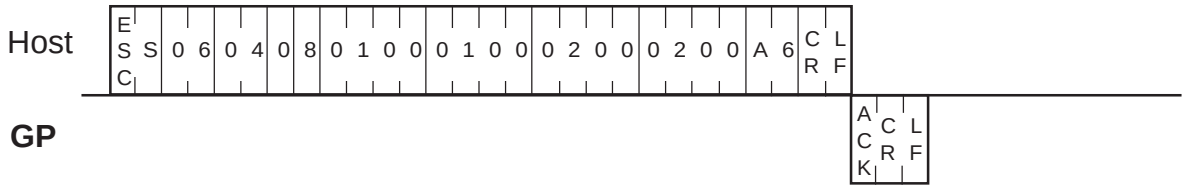
A filled rectangle is drawn with the upper left and lower right corners located respectively at points (100, 100) and (200, 200).



(Attributes)
Foreground color: Yellow, Blinking: Disabled
Background color: Red, Blinking: Disabled
Tiling pattern: 8

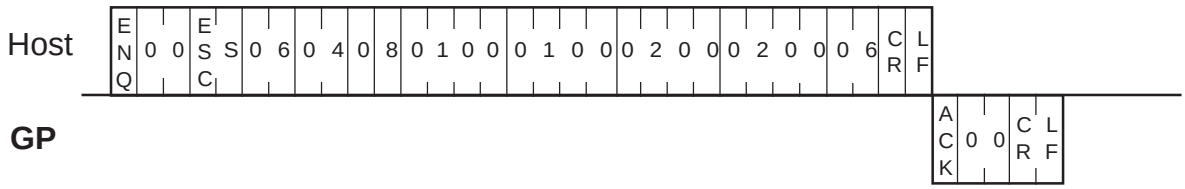
■Extend Mode, 1:1 ASCII,

ETX. SUMCHECK: ENABLED, TERMINATOR: CR.LF, ACK: ENABLED, NAK: ENABLED



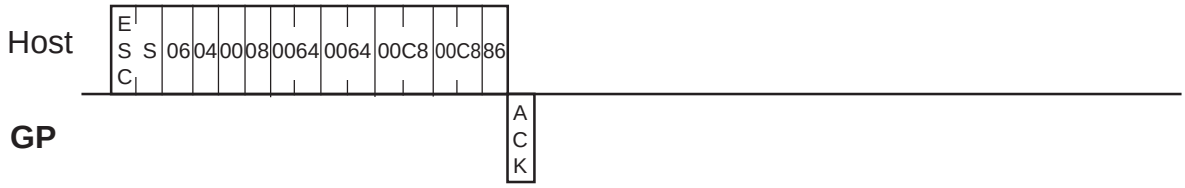
■Extend Mode, 1:n ASCII,

ETX. SUMCHECK: ENABLED, TERMINATOR: CR.LF, ACK: ENABLED, NAK: ENABLED



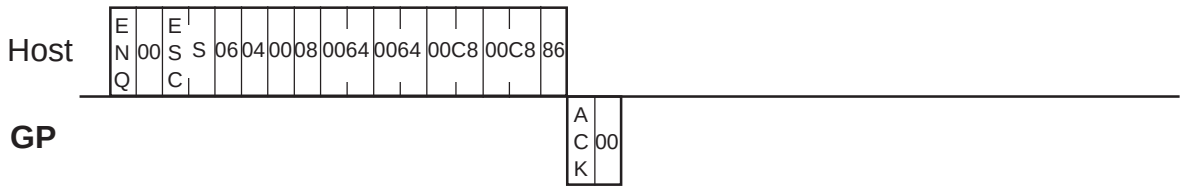
■Extend Mode, 1:1 BINARY,

ETX. SUMCHECK: ENABLED, ACK: ENABLED, NAK: ENABLED



■Extend Mode, 1:n BINARY,

ETX. SUMCHECK: ENABLED, ACK: ENABLED, NAK: ENABLED

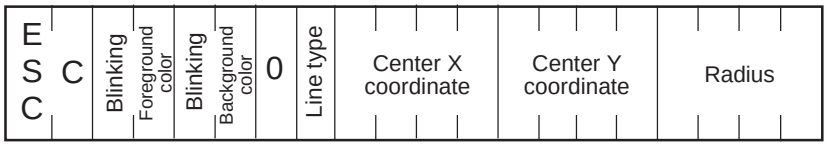


3-8 Displaying a Circle [ESC C]

The format of the command data block containing the ESC C command (circle display command) is shown below.

■ASCII Mode

Command data block (from Host)



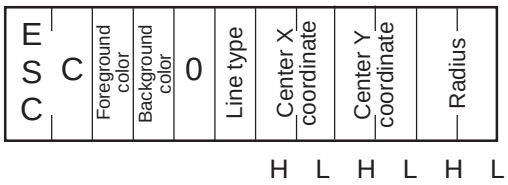
<Setting range>

- Blinking : 0 or 1 (0: Disabled, 1: Enabled)
- Foreground/ Background color : 0 to 7 (0: Black, 1: Blue, 2: Green, 3: Light blue, 4: Red, 5: Purple, 6: Yellow, 7: White)
- Line type : 0 to 3 (0: _____ ,1: _ _ _ _ ,2: __ - __ ,3: __ - - __)
- X coordinate : 0000 to 0799 (0 to 799)
- Y coordinate : 0000 to 0599 (0 to 599)
- Radius : 0001 to 0799 (1 to 799)

GP Response data block (from GP)
ACK or NAK

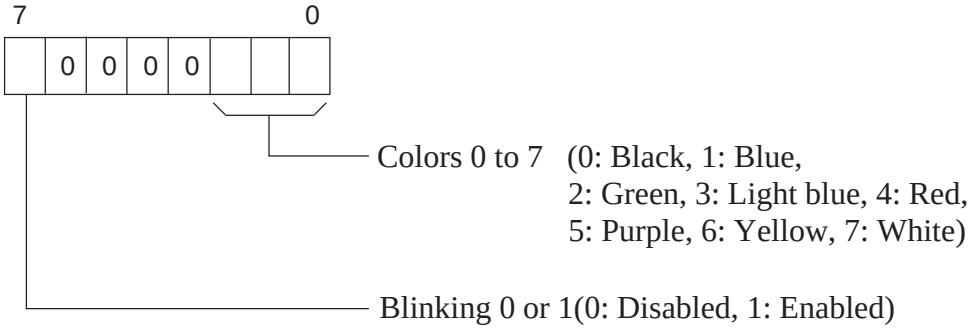
■Binary Mode

Command data block (from Host)



<Setting range>

• Foreground/Background color



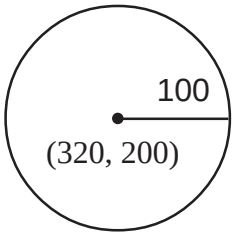
<Setting range

• Line type : 00H to 03H (00:_____ ,01:_ _ _ _ _ ,02:__ _ __ 03:__ _ __)
• X coordinate :0000H to 031FH (0 to 799)
• Y coordinate :0000H to 0257H (0 to 599)
• Radius : 0001H to 031FH (1 to 799)

GP Response data block (from GP)
ACK or NAK

<Example>

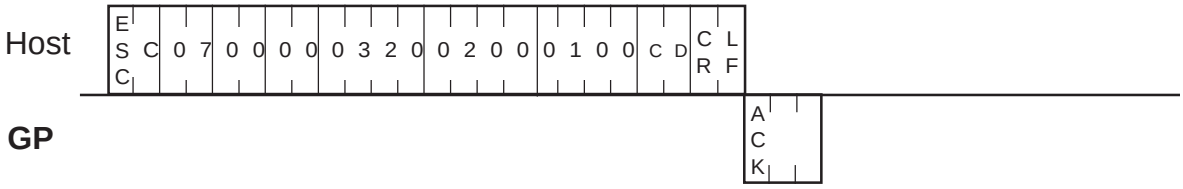
A circle is drawn with the center located at points (320, 200), and with a radius of 100.



(Attributes)
Foreground color: White
Background color: Black
Line type: Solid line (0)

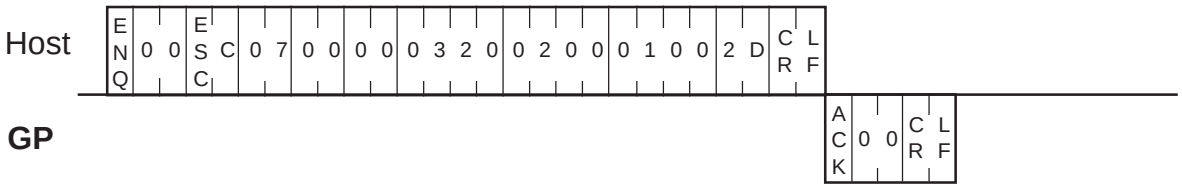
■Extend Mode, 1:1 ASCII,

ETX. SUMCHECK: ENABLED, TERMINATOR: CR.LF, ACK: ENABLED, NAK: ENABLED



■Extend Mode, 1:n ASCII,

ETX. SUMCHECK: ENABLED, TERMINATOR: CR.LF, ACK: ENABLED, NAK: ENABLED

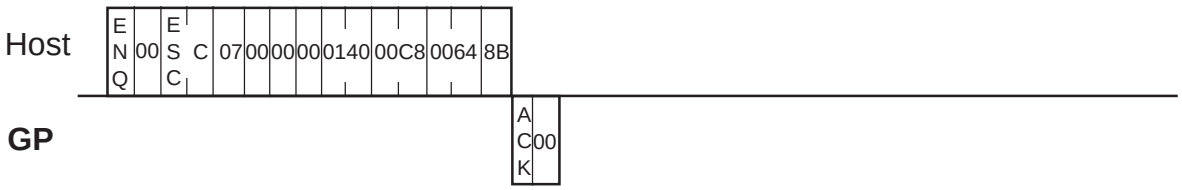


■Extend Mode, 1:1 BINARY,

ETX. SUMCHECK: ENABLED, ACK: ENABLED, NAK: ENABLED

■Extend Mode, 1:n BINARY,

ETX. SUMCHECK: ENABLED, ACK: ENABLED, NAK: ENABLED



3-9 Displaying an Arc [ESC A]

The format of the command data block containing the ESC A command (arc display command) is shown below.

■ASCII Mode

Command data block (from Host)

ESC A	Blinking Foreground color	Blinking Background color	0	Line type	Center X coordinate	Center Y coordinate	Radius	Start angle	End angle

<Setting range>

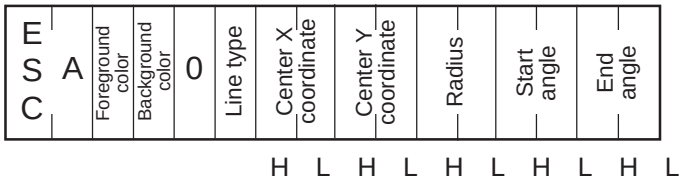
- Blinking : 0 or 1 (0: Disabled, 1: Enabled)
- Foreground/ Background color : 0 to 7 (0: Black, 1: Blue, 2: Green, 3: Light blue, 4: Red, 5: Purple, 6: Yellow, 7: White)
- Line type : 0 to 3 (0: ——— ,1: _ _ _ _ ,2: — - — ,3: — - - —)
- X coordinate : 0000 to 0799 (0 to 799)
- Y coordinate : 0000 to 0599 (0 to 599)
- Radius : 0001 to 0799 (1 to 799)
- Angle : 0000 to 0360 (0 to 360)

GP Response data block (from GP)
ACK or NAK

- Arcs are drawn counterclockwise.
- Be sure not to specify the same value for the start and end angles.

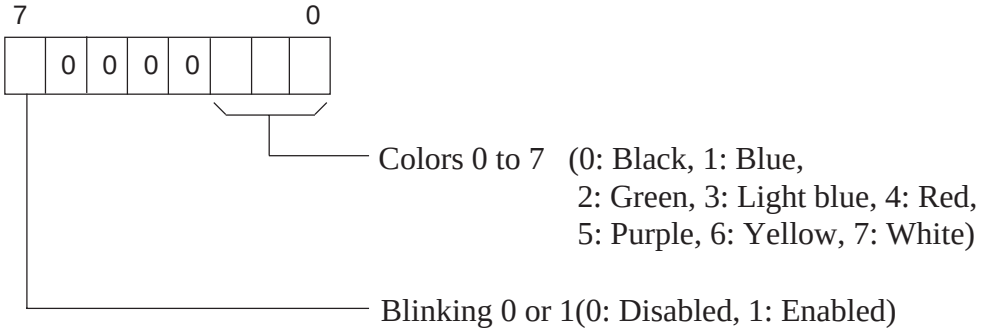
■Binary Mode

Command data block (from Host)



<Setting range>

• Foreground/Background color

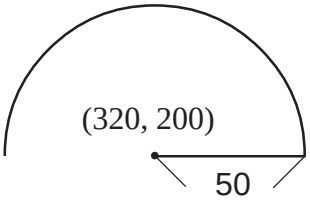


• Line type : 00H to 03H (00: ———,01: - - - - -,02: — - —
,03: — - - —)
• X coordinate : 0000H to 031FH (0 to 799)
• Y coordinate : 0000H to 0257H (0 to 599)
• Radius : 0001H to 031FH (1 to 799)
• Angle : 0000H to 0168H (0 to 360)

GP Response data block (from GP)
ACK or NAK

<Example>

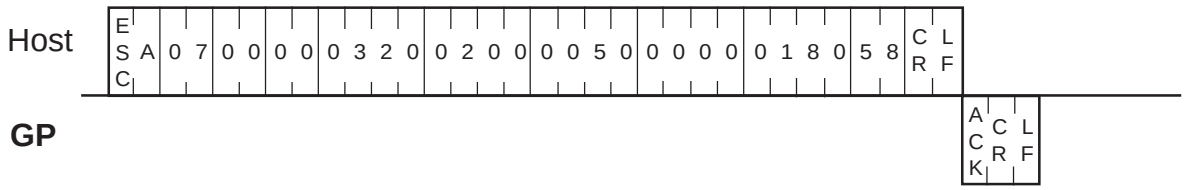
A semicircle (arc) is drawn with the center located at point (320, 200) with a radius of 50.



(Attributes)
Foreground color: White
Background color: Black
Line type: Solid line (0)

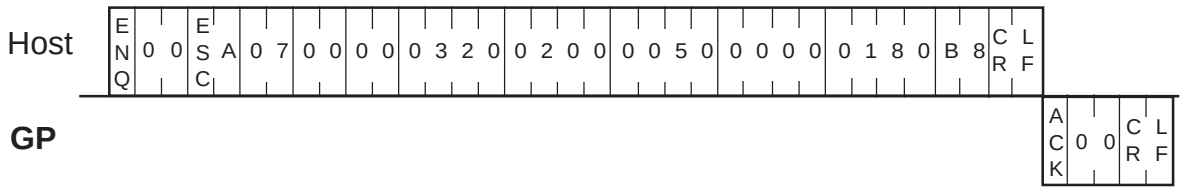
■Extend Mode, 1:1 ASCII,

ETX. SUMCHECK: ENABLED, TERMINATOR: CR.LF, ACK: ENABLED, NAK: ENABLED



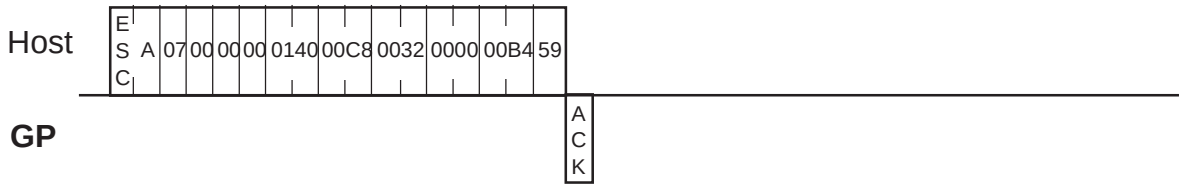
■Extend Mode, 1:n ASCII,

ETX. SUMCHECK: ENABLED, TERMINATOR: CR.LF, ACK: ENABLED, NAK: ENABLED



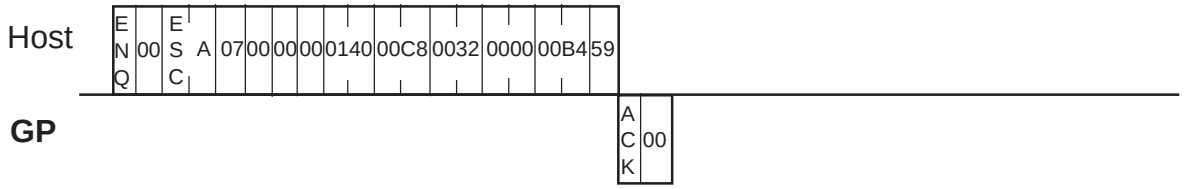
■Extend Mode, 1:1 BINARY,

ETX. SUMCHECK: ENABLED, ACK: ENABLED, NAK: ENABLED



■Extend Mode, 1:n BINARY,

ETX. SUMCHECK: ENABLED, ACK: ENABLED, NAK: ENABLED



3-10 Displaying a Pie Shape [ESC G]

The format of the command data block containing the ESC G command (pie display command) is shown below.

■ASCII Mode

Command data block (from Host)

ESC	G	Blinking	Foreground color	Blinking	Background color	0	Line type	Center X coordinate	Center Y coordinate	Radius	Start angle	End angle

<Setting range>

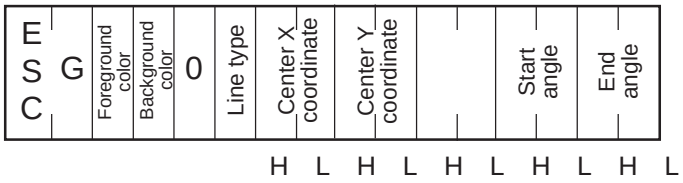
- Blinking : 0 or 1 (0: Disabled, 1: Enabled)
- Foreground/Background color : 0 to 7 (0: Black, 1: Blue, 2: Green, 3: Light blue, 4: Red, 5: Purple, 6: Yellow, 7: White)
- Line type : 0 to 3 (0:_____,1: _ _ _ _,2: __ - __ ,3: __ - - __)
- X coordinate : 0000 to 0799 (0 to 799)
- Y coordinate : 0000 to 0599 (0 to 599)
- Radius : 0001 to 0799 (1 to 799)
- Angle : 0000 to 0360 (0 to 360)

GP Response data block (from GP)
ACK or NAK

- Pie shapes are drawn counterclockwise.
- Be sure not to specify the same value for the start and end angles.

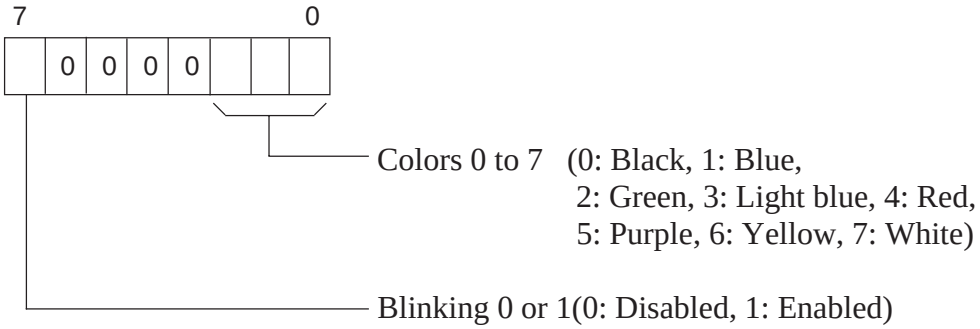
■Binary Mode

Command data block (from Host)



<Setting range>

- Foreground/Background color

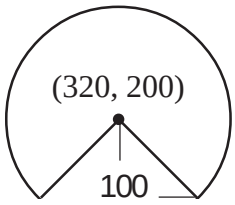


- Line type : 00H to 03H (00: ——— ,01: - - - - - ,02: — - — ,03: — - - —)
- X coordinate : 0000H to 031FH (0 to 799)
- Y coordinate : 0000H to 0257H (0 to 599)
- Radius : 0001H to 031FH (1 to 799)
- Angle : 0000H to 0168H (0 to 360)

GP Response data block (from GP)
ACK or NAK

<Example>

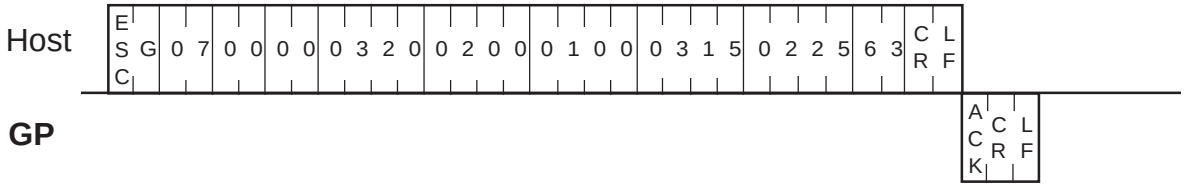
A pie is drawn with the center located at point (320, 200), and with a radius of 100.



(Attributes)
Foreground color: White, Start angle: 315°
Background color: Black, End angle: 225°
Line type: Solid line (0)

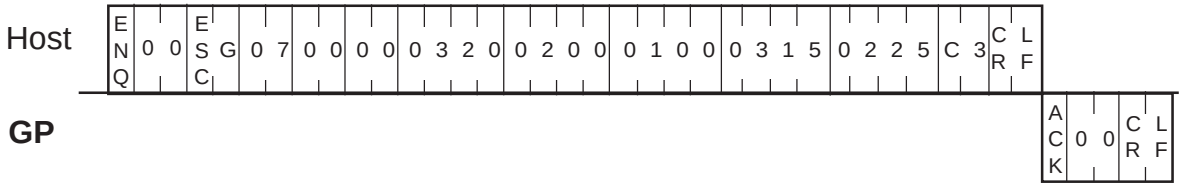
■Extend Mode, 1:1 ASCII,

ETX. SUMCHECK: ENABLED, TERMINATOR: CR.LF, ACK: ENABLED, NAK: ENABLED



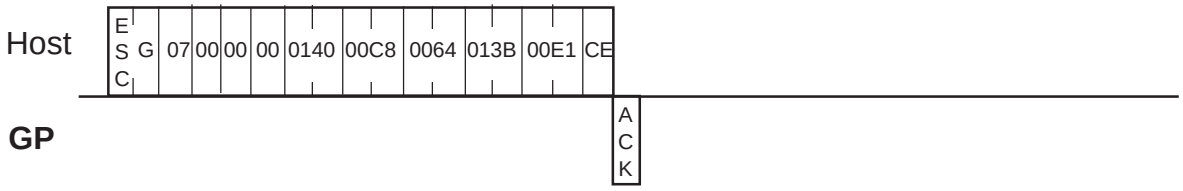
■Extend Mode, 1:n ASCII,

ETX. SUMCHECK: ENABLED, TERMINATOR: CR.LF, ACK: ENABLED, NAK: ENABLED



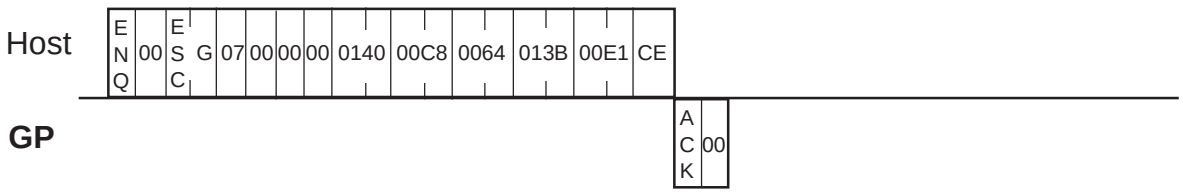
■Extend Mode, 1:1 BINARY,

ETX. SUMCHECK: ENABLED, ACK: ENABLED, NAK: ENABLED



■Extend Mode, 1:n BINARY,

ETX. SUMCHECK: ENABLED, ACK: ENABLED, NAK: ENABLED



3-11 Filling an Object [ESC P]

The format of the command data block containing the ESC P command (filled shape display command) is shown below.

■ASCII Mode

Command data block (from Host)

ESC	P	Blinking	Foreground color	Blinking	Background color	0	Tiling pattern	0	Border color	X coordinate	Y coordinate

<Setting range>

- Blinking: 0 or 1 (0: Disabled, 1: Enabled)
- Foreground/Background /Border color : 0 to 7 (0: Black, 1: Blue, 2: Green, 3: Light blue, 4: Red, 5: Purple, 6: Yellow, 7: White)
- Tiling pattern : 0 to 8 *1
- X coordinate : 0000 to 0799 (0 to 799)
- Y coordinate : 0000 to 0599 (0 to 599)

GP Response data block (from GP)
ACK or NAK

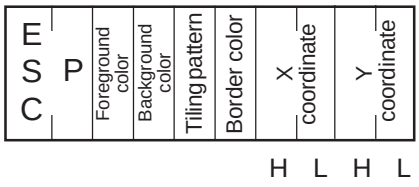
---

*1 Tiling Patterns

Tiling pattern No.	Tiling pattern	Tiling pattern No.	Tiling pattern	Tiling pattern No.	Tiling pattern
0		3		6	
1		4		7	
2	 8 dots 8 dots	5		8	

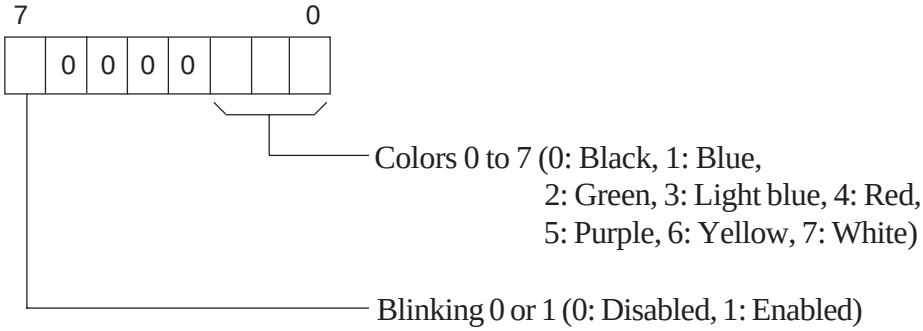
■Binary Mode

Command data block (from Host)



<Setting range>

• Foreground/Background/Border color



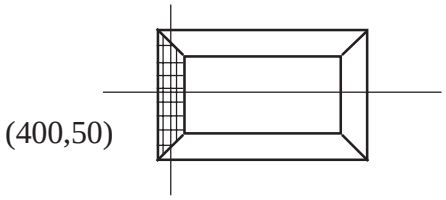
• Tiling pattern : 00H to 08H (See the tiling patterns on P.3-28.)
• X coordinate : 0000H to 031FH (0 to 799)
• Y coordinate : 0000H to 0257H (0 to 599)

Note: Be sure to disable border color blink.

GP Response data block (from GP)
ACK or NAK

<Example>

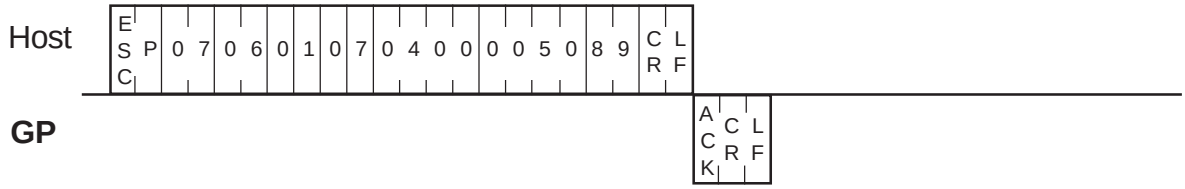
The section of a shape that contains point (400, 50) is filled with the pattern.



(Attributes)
Foreground color: White
Background color: Yellow
Border color: White
Tiling pattern: 1

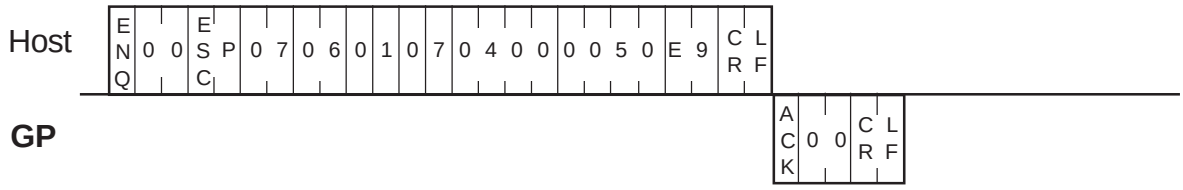
■Extend Mode, 1:1 ASCII,

ETX. SUMCHECK: ENABLED, TERMINATOR: CR.LF, ACK: ENABLED, NAK: ENABLED



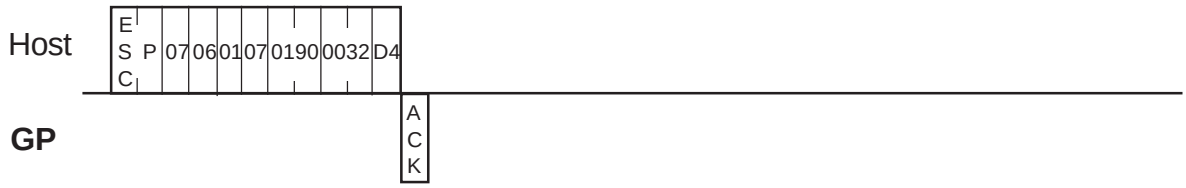
■Extend Mode, 1:n ASCII,

ETX. SUMCHECK: ENABLED, TERMINATOR: CR.LF, ACK: ENABLED, NAK: ENABLED



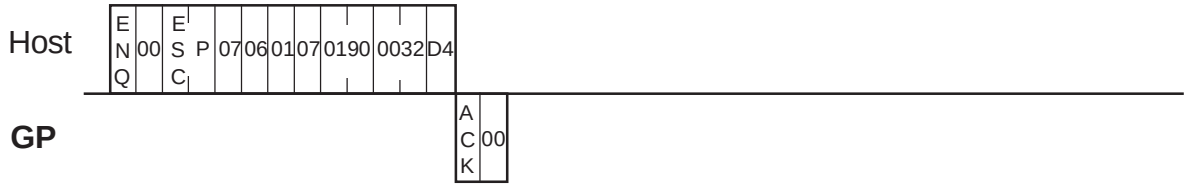
■Extend Mode, 1:1 BINARY,

ETX. SUMCHECK: ENABLED, ACK: ENABLED, NAK: ENABLED



■Extend Mode, 1:n BINARY,

ETX. SUMCHECK: ENABLED, ACK: ENABLED, NAK: ENABLED



3-12 Interrupt Output Requests [ESC I (large I)]

Here, commands are explanined that are used by the GP in Extend Mode, when using [1:n ASCII], [1:n Binary] or 2-wire type communication, to output an interrupt code via the GP unit's T-tag or System Area's Absolute Value Write, etc. from the GP unit to the Host.
When using 2-wire type connection, be sure to perform the following settings even for a 1:1 connection.

■ASCII Mode

Command data block (from Host)

GP Response data block (from GP)

◆When there is no error

◆If an error occurs
NAK response.

<Description>

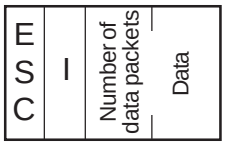
- Number of data packets : When an enquiry command is sent from the host, this value defines the previously issued interrupt output's number of data items.
  After all the previously issued interrupt output data is acquired, this data frequency (number) must be sent.
- Data : The data (00H to FEH) is converted into a 2-digit ASCII code (HEX) before being output. "00" will be entered in this field if there is no data to be output.

■Binary Mode

Command data block (from Host)

GP Response data block (from GP)

◆When there is no error



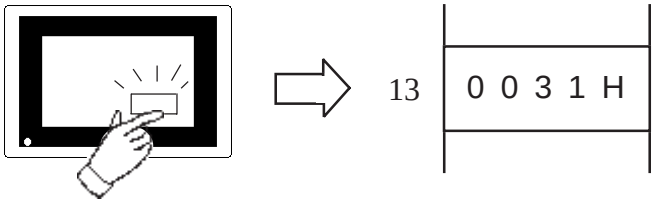
◆If an error occurs
NAK response.

<Description>

- Number of data packets : When an enquiry command is sent from the host, this value defines the previously issued interrupt output's number of data items.
  After all the previously issued interrupt output data is acquired, this data frequency number must be sent.
- Data : The data value (00H to FEH) is output. "00" will be entered in this field if there is no data to be output.

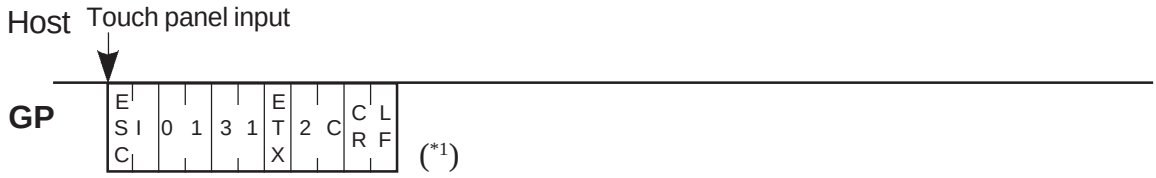
<Example>

"0031H" is written to the system data area 13 by the T-Tag (touch panel input).

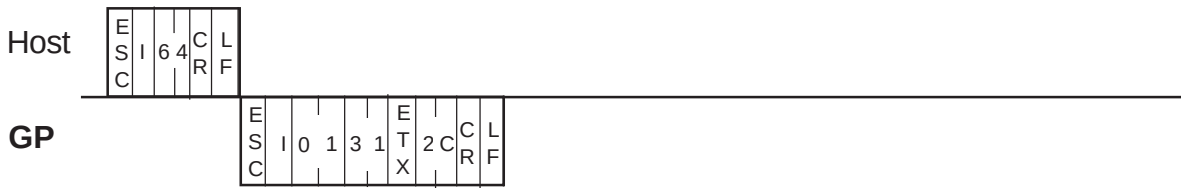


■Extend Mode, 1:1 ASCII,

ETX. SUMCHECK: ENABLED, TERMINATOR: CR.LF, ACK: ENABLED, NAK: ENABLED

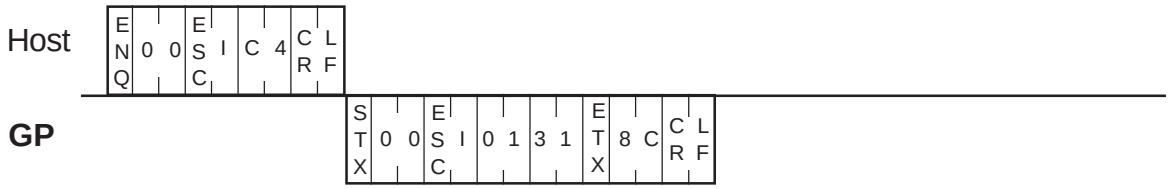


<2-Wire type>



■Extend Mode, 1:n ASCII,

ETX. SUMCHECK: ENABLED, TERMINATOR: CR.LF, ACK: ENABLED, NAK: ENABLED

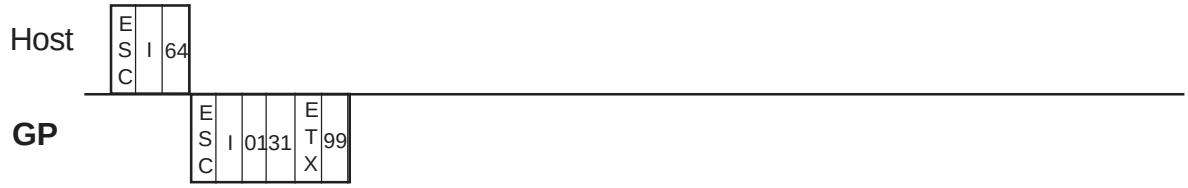


■Extend Mode, 1:1 BINARY,

ETX. SUMCHECK: ENABLED, ACK: ENABLED, NAK: ENABLED

*1 When the host sends an ESC I command, the GP sends the response data block containing 0 in the Number of data packets field and the data field. However, be sure to use the ESC I command when using the 2-wire type.

<2-wire type>



■Extend Mode, 1:n BINARY,
ETX. SUMCHECK: ENABLED, ACK: ENABLED, NAK: ENABLED

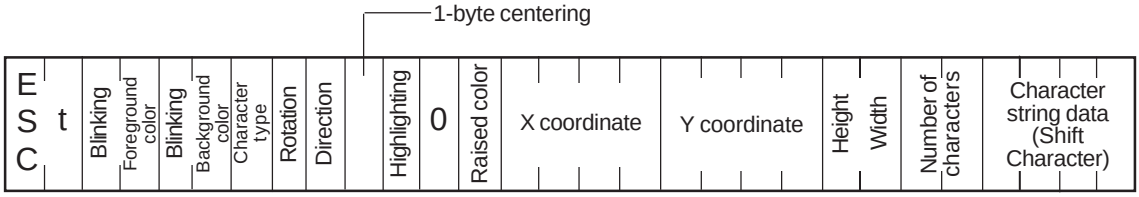
- ***When data is written to address 13 in the system data area by the T-Tag or by the ESC W command, the contents of the lower 8 bits at this address will be output as an interrupt code.***
- ***When [1:1 ASCII] or [1:1 Binary] are used in Extend Mode, and when a write is sent to Address 13, the Interrupt Code is output (Except for 2-wire type communication). Even if the Interrupt Output's enquiry command is sent, the answer is always "No. of data items = 0, Data =0".***
  Also, while in [1:1 ASCII], [1:1 Binary], [2-wire type 1:1 ASCII] and [2-wire type 1:1 Binary] modes, if an enquiry command sent from the Host is received (by the GP), an interrupt code is output.

3-13 Additional Character String Features [ESC t]

The format of the command data block containing the ESC t command (enhancement to the character string display command) is shown below. Rotation, direction, and highlighting are available as enhancements.

■ASCII Mode

Command data block (from Host)



<Setting range>

- Blinking: 0 or 1 (0: Disabled, 1: Enabled)

Foreground/Background color: 0 to 7 (0: Black, 1: Blue, 2: Green, 3: Light blue, 4: Red, 5: Purple, 6: Yellow, 7: White)

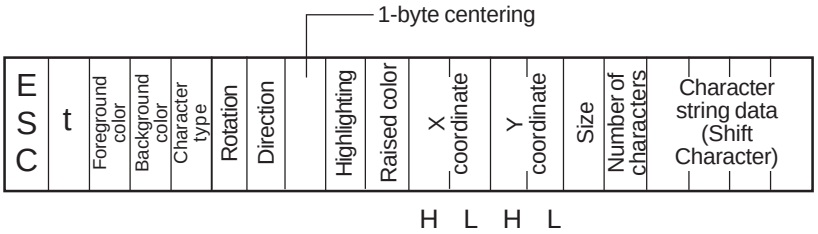
Character type: 00 to 09 (00: Half byte, 01: 1-byte, 02: 2-byte, 03: 5X7 font, 04: 7X9 font, 05: 11X16 font, 06: 24X32 font, 07: 7X9F font, 08: 11X16F font, 09: 1-byte double-sized character)

- Rotation: 0 to 3 (0: 0 deg., 1: 90 deg., 2: 180 deg., 3: 270 deg.)
- Direction: 0 or 1 (0: Horizontal, 1: Vertical)
- 1-byte centering: 0 or 1 (0: Disabled, 1: Enabled (available when "Vertical" is selected for Direction))
- Highlighting: 0 to 2 (0: Normal, 1: Bold, 2: Raised)
- Raised color: 0 to 7 (0: Black, 1: Blue, 2: Green, 3: Light blue, 4: Red, 5: Purple, 6: Yellow, 7: White)
- X coordinate: 0000 to 0799 (0 to 799)
- Y coordinate: 0000 to 0599 (0 to 599)
- Height, Width: 0 to 3 (0: 1X, 1: 2X, 2: 4X, 3: 8X)
- Number of characters: 01 to 80 (Number of bytes)
- Data: Character string data (Shift Character)

GP Response data block (from GP)
ACK or NAK

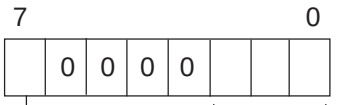
■Binary Mode

Command data block (from Host)



<Setting range>

• Foreground/Background color



Colors 0 to 7 (0: Black, 1: Blue,
2: Green, 3: Light blue, 4: Red,
5: Purple, 6: Yellow, 7: White)

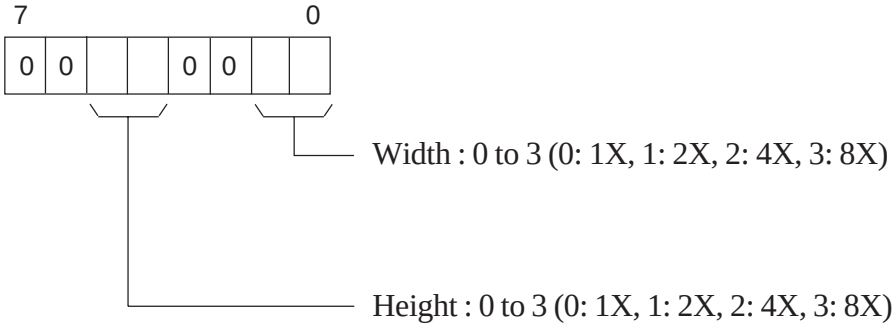
Blinking 0 or 1 (0: Disabled, 1: Enabled)

* Blinking is always enabled for a Raised character.

<Setting range>

• Character type :	00 to 09 (00: Half byte, 01: 1-byte, 02: 2-byte, 03: 5X7 font, 04: 7X9 font, 05: 11X16 font, 06: 24X32 font, 07: 7X9F font, 08: 11X16F font, 09: 1-byte double-sized character)
• Rotation :	0 to 3 (0: 0 deg., 1: 90 deg., 2: 180 deg., 3: 270 deg.)
• Direction :	0 or 1 (0: Horizontal, 1: Vertical)
• 1-byte centering :	0 or 1 (0: Disabled, 1: Enabled (available when "Vertical" is selected for Direction))
• Highlighting:	00 to 02 (0: Normal, 1: Bold, 2: Raised)
• Raised color :	00 to 07 (0: Black, 1: Blue, 2: Green, 3: Light blue, 4: Red, 5: Purple, 6: Yellow, 7: White)

- X coordinate : 0000H to 031FH (0 to 799)
- Y coordinate : 0000H to 0257H (0 to 599)
- Size

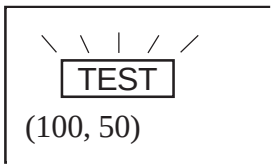


- Number of characters (bytes) : 01H to 50H (1 to 80)
- Character string data : An ANK character is 1-byte long. All double-sized characters are 2-bytes long.

GP Response data block (from GP)
ACK or NAK

<Example>

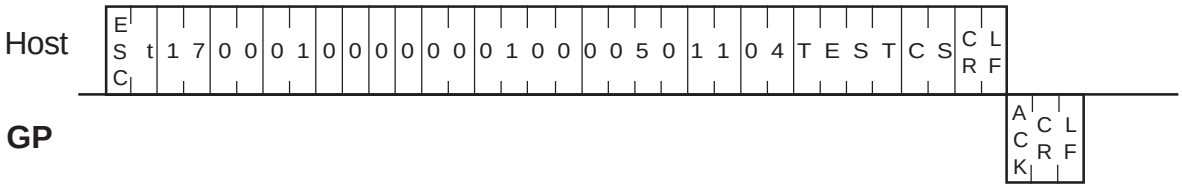
"TEST" appears blinking to the right of point (100, 50).



(Attribute)
Character size: 2 X 2

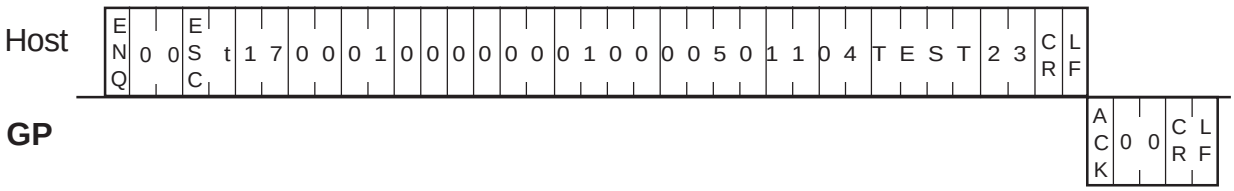
■Extend Mode, 1:1 ASCII,

ETX. SUMCHECK: ENABLED, TERMINATOR: CR.LF, ACK: ENABLED, NAK: ENABLED



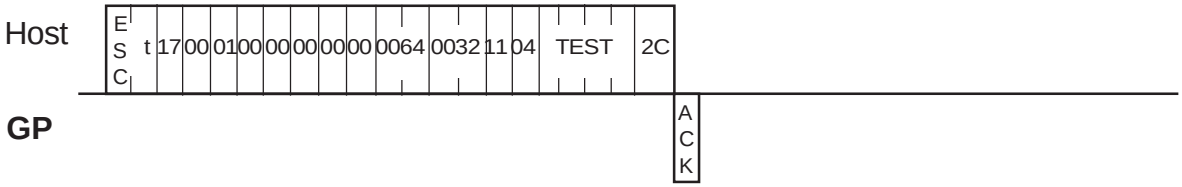
■Extend Mode, 1:n ASCII,

ETX. SUMCHECK: ENABLED, TERMINATOR: CR.LF, ACK: ENABLED, NAK: ENABLED



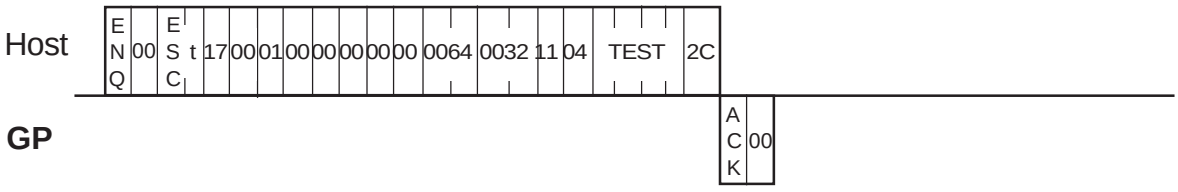
■Extend Mode, 1:1 BINARY,

ETX. SUMCHECK: ENABLED, ACK: ENABLED, NAK: ENABLED



■Extend Mode, 1:n BINARY,

ETX. SUMCHECK: ENABLED, ACK: ENABLED, NAK: ENABLED



3-14 Additional Line Features [ESC l (small L)]

The format of the command data block containing the ESC l command (enhancement to a straight line display command) is shown below. This additional feature is the use of an arrow.

■ASCII Mode

Command data block (from Host)

ESC	l	Blinking	Foreground color	Blinking	Background color	0	Line type	0	2	Arrow pattern	Arrow direction	Start point X coordinate	Start point Y coordinate	End point X coordinate	End point Y coordinate

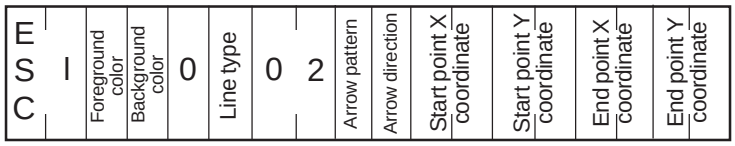
<Setting range>

- Blinking : 0 or 1 (0: Disabled, 1: Enabled)
- Foreground/Background color : 0 to 7 (0: Black, 1: Blue, 2: Green, 3: Light blue, 4: Red, 5: Purple, 6: Yellow, 7: White)
- Line type : 0 to 7 (0: ———,1: _ _ _ _,2: — - —,3: — - - —, 4: ———,5: _ _ _ _,6: — - —,7: — - - —)
- Arrow pattern : 0 or 1 (0: Disabled, 1: Enabled)
- Arrow direction : 0 or 1 (0: Both ends of line, 1: End point of line)
- X coordinate : 0000 to 0799 (0 to 799)
- Y coordinate : 0000 to 0599 (0 to 599)

GP Response data block (from GP)
ACK or NAK

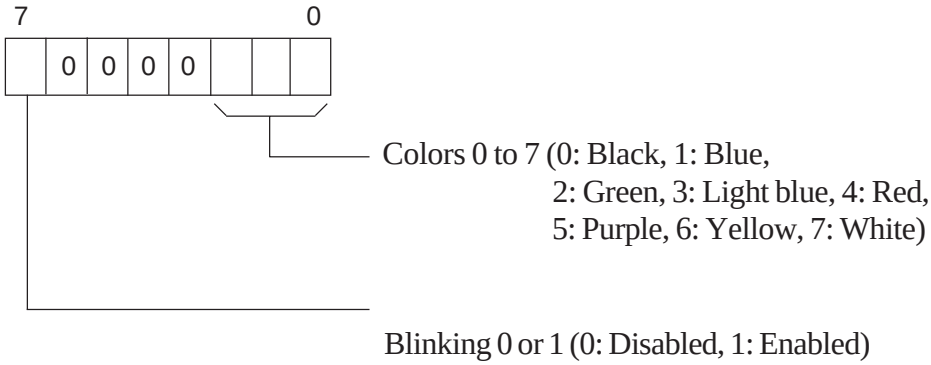
■Binary Mode

Command data block (from Host)



<Setting range>

• Foreground/Background color



<Setting range>

• Line type : 00H to 07H
(00: ——— ,01: - - - - ,02: — - — ,
03: — - - —,04: ——— ,05: - - - - ,
06: — - — ,07: — - - —)

• Arrow pattern : 0 or 1 (0: Disabled, 1: Enabled)

• Arrow direction : 0 or 1 (0: Both ends of line, 1: End point of line)

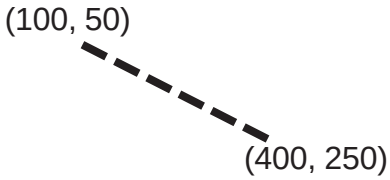
• X coordinate : 0000H to 031FH (0 to 799)

• Y coordinate : 0000H to 0257H (0 to 599)

GP Response data block (from GP)
ACK or NAK

<Example>

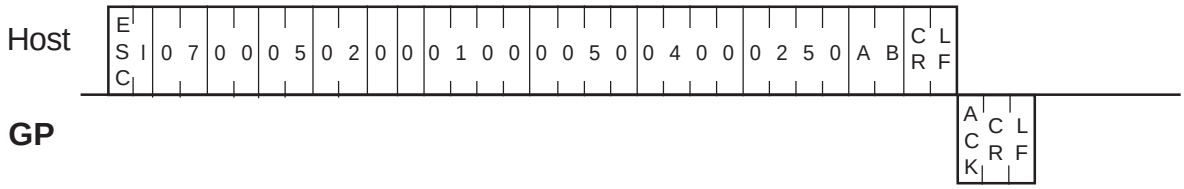
A bold dotted line is drawn between two points (100, 50) and (400, 250).



(Attributes)
Foreground color: White
Background color: Black

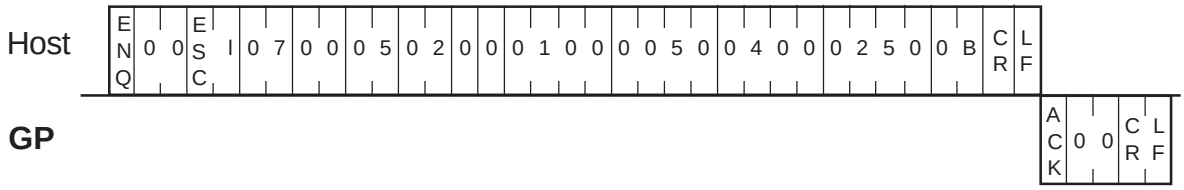
■Extend Mode, 1:1 ASCII,

ETX. SUMCHECK: ENABLED, TERMINATOR: CR.LF, ACK: ENABLED, NAK: ENABLED



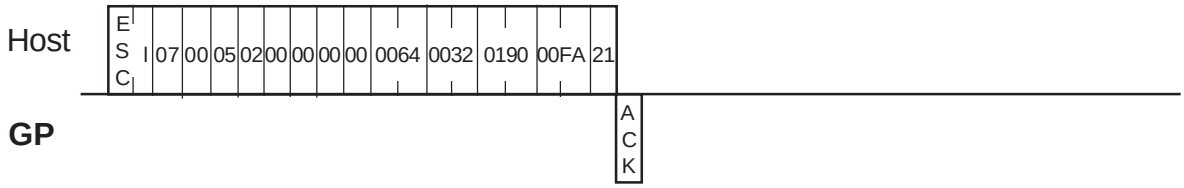
■Extend Mode, 1:n ASCII,

ETX. SUMCHECK: ENABLED, TERMINATOR: CR.LF, ACK: ENABLED, NAK: ENABLED



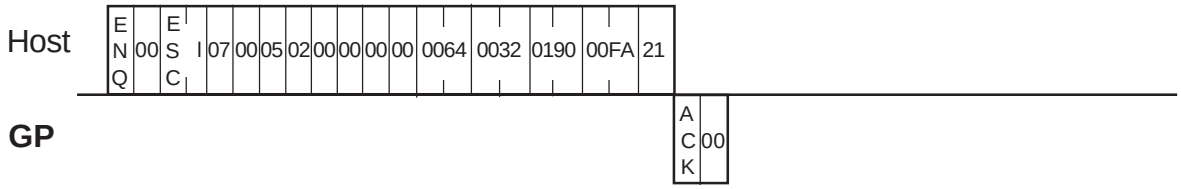
■Extend Mode, 1:1 BINARY,

ETX. SUMCHECK: ENABLED, ACK: ENABLED, NAK: ENABLED



■Extend Mode, 1:n BINARY,

ETX. SUMCHECK: ENABLED, ACK: ENABLED, NAK: ENABLED



3-15 Additional Rectangle Features [ESC b]

The format of the command data block containing the ESC b command (enhancement to the straight line display command) is shown below. This additional feature is the use of beveling.

■ASCII Mode

Command data block (from Host)

ESC	b	Blinking	Foreground color	Blinking	Background color	0	Line type	0	0	0	Beveling method	Beveling radius	Start point X coordinate	Start point Y coordinate	End point X coordinate	End point Y coordinate

<Setting range>

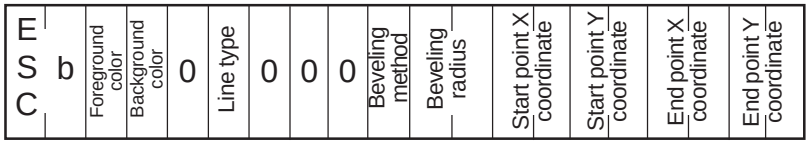
- Blinking: 0 or 1 (0: Disabled, 1: Enabled)
- Foreground/Background color: 0 to 7 (0: Black, 1: Blue, 2: Green, 3: Light blue, 4: Red, 5: Purple, 6: Yellow, 7: White)
- Line type: 0 to 3, 8, 9 (0: ——— ,1: - - - -,2: — - — ,3:— - - — , 8: ——— ,9: ▬▬▬)
- Beveling method: 0 to 2 (0: Disabled, 1: Curve, 2: Straight line)
- Beveling radius: 00 to 32
- X coordinate: 0000 to 0799 (0 to 799)
- Y coordinate: 0000 to 0599 (0 to 599)

GP Response data block (from GP)
ACK or NAK

Line types 0 to 3 are 1-dot lines, and line types 8 and 9 are 3-dot and 5-dot lines.

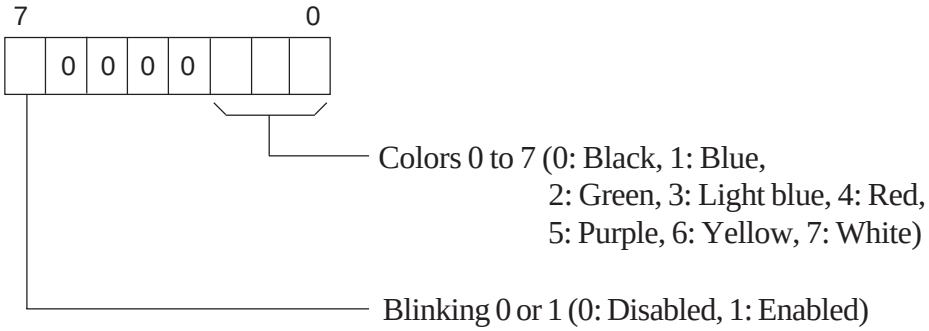
■Binary Mode

Command data block (from Host)



<Setting range>

• Foreground/Background color



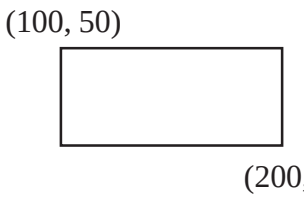
<Setting range>

- Beveling method : 00H: Disabled, 01H: Curve, 02H: Straight line
- Beveling radius : 00H to 32H
- Line type : 0 to 3, 8, 9
  0 to 3, 8, 9 (0: ———,1: - - - -,2: — - —,3: — - - —
  8: ———,9: ▬▬▬)
- X coordinate : 0000H to 031FH (0 to 799)
- Y coordinate : 0000H to 0257H (0 to 599)

GP Response data block (from GP)
ACK or NAK

<Example>

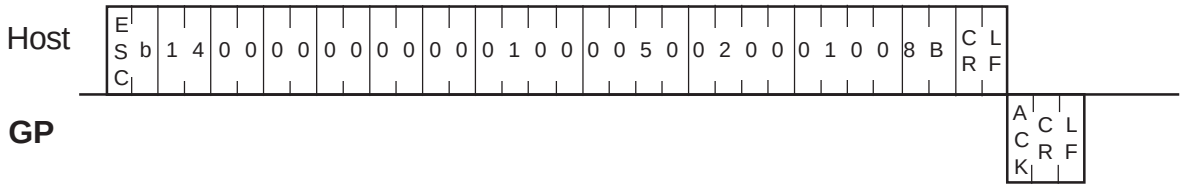
A rectangle is drawn whose upper left and lower right corners are located at points (100, 50) and (200, 100).



(Attributes)
Foreground color: Red, Blinking: Enabled
Background color: Black, Blinking: Disabled
Line type: Solid line (0)

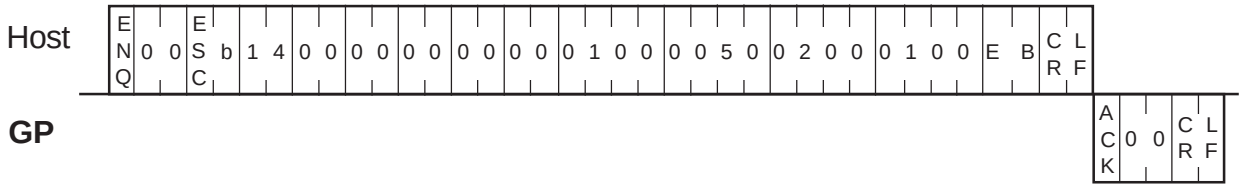
■Extend Mode, 1:1 ASCII,

ETX. SUMCHECK: ENABLED, TERMINATOR: CR.LF, ACK: ENABLED, NAK: ENABLED



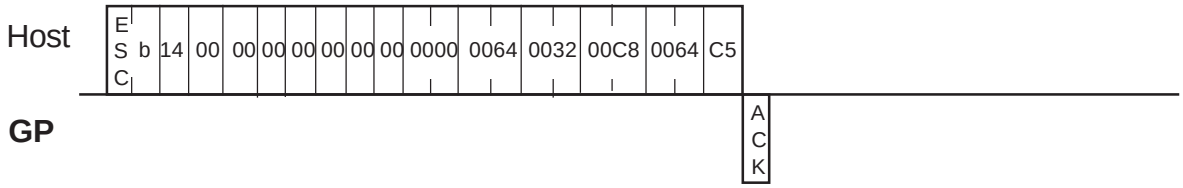
■Extend Mode, 1:n ASCII,

ETX. SUMCHECK: ENABLED, TERMINATOR: CR.LF, ACK: ENABLED, NAK: ENABLED



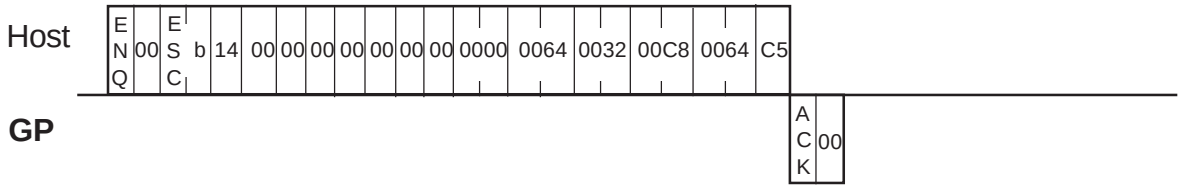
■Extend Mode, 1:1 BINARY,

ETX. SUMCHECK: ENABLED, ACK: ENABLED, NAK: ENABLED



■Extend Mode, 1:n BINARY,

ETX. SUMCHECK: ENABLED, ACK: ENABLED, NAK: ENABLED



3-16 Additional Filled Rectangle Features [ESC s]

The format of the command data block containing the ESC s command (enhancement to the filled rectangle display command) is shown below. This additional feature is the use of beveling.

■ASCII Mode

Command data block (from Host)

ESC	s	Blinking	Foreground color	Blinking	Background color	0	0	0	Tiling pattern	0	Beveling method	Beveling radius	Start point X coordinate	Start point Y coordinate	End point X coordinate	End point Y coordinate

<Setting range>

- Blinking: 0 or 1 (0: Disabled, 1: Enabled)
- Foreground/Background color: 0 to 7 (0: Black, 1: Blue, 2: Green, 3: Light blue, 4: Red, 5: Purple, 6: Yellow, 7: White)
- Tiling pattern: 0 to 8

Reference ***Tiling Patterns in Section 3-7 "Displaying a Filled Rectangle"***

- Beveling method: 0 to 2 (0: Disabled, 1: Curve, 2: Straight line)
- Beveling radius: 00 to 32
- X coordinate: 0000 to 0799 (0 to 799)
- Y coordinate: 0000 to 0599 (0 to 599)

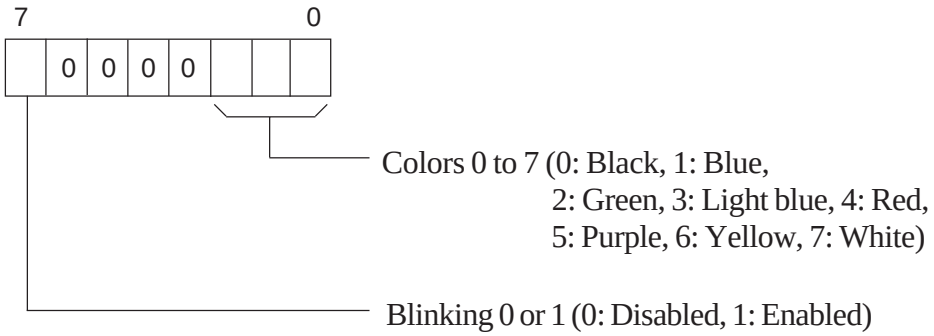
GP Response data block (from GP)
ACK or NAK

■Binary Mode

Command data block (from Host)

<Setting range>

• Foreground/Background color



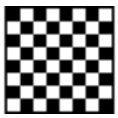
- Tiling pattern : 00H to 08H (See the tiling patterns on P.3-16.)
- Beveling method : 0: Disabled, 1: Curve, 2: Straight line
- Beveling radius : 00H to 32H
- X coordinate : 0000H to 031FH (0 to 799)
- Y coordinate : 0000H to 0257H (0 to 599)

GP Response data block (from GP)
ACK or NAK

<Example>

A patterned rectangle is drawn with the upper left and lower right corners located respectively at points (100, 100) and (200, 200).

(100, 100)

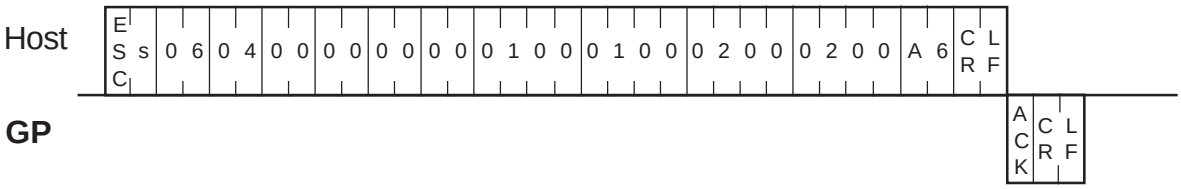


(200, 200)

(Attributes)
Foreground color: Yellow, Blinking: Disabled
Background color: Red, Blinking: Disabled
Tiling pattern: 8

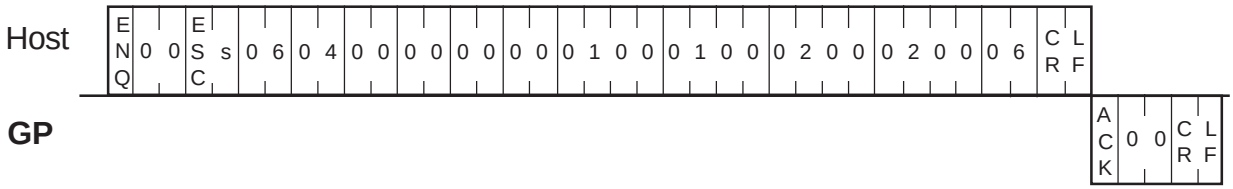
■Extend Mode, 1:1 ASCII,

ETX. SUMCHECK: ENABLED, TERMINATOR: CR.LF, ACK: ENABLED, NAK: ENABLED



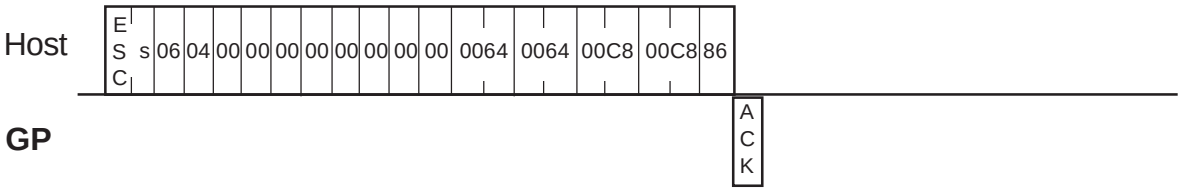
■Extend Mode, 1:n ASCII,

ETX. SUMCHECK: ENABLED, TERMINATOR: CR.LF, ACK: ENABLED, NAK: ENABLED



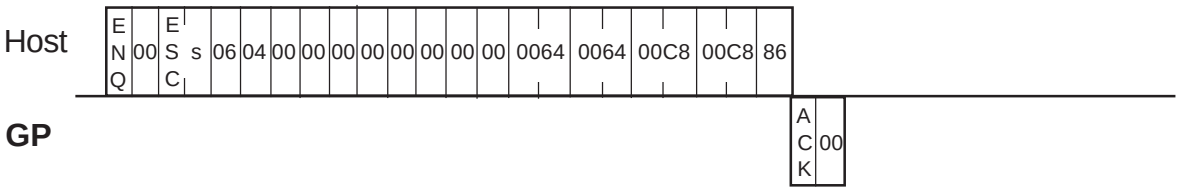
■Extend Mode, 1:1 BINARY,

ETX. SUMCHECK: ENABLED, ACK: ENABLED, NAK: ENABLED



■Extend Mode, 1:n BINARY,

ETX. SUMCHECK: ENABLED, ACK: ENABLED, NAK: ENABLED



3-17 Additional Filled Circle Features [ESC c]

The format of the command data block containing the ESC c command (enhancement to the filled circle display command) is shown below. This additional feature allows the use to Tiling.

■ASCII Mode

Command data block (from Host)

<Setting range>

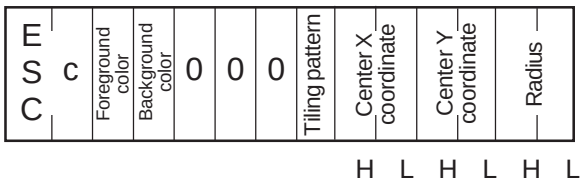
- Blinking: 0 or 1 (0: Disabled, 1: Enabled)
- Foreground/Background color: 0 to 7 (0: Black, 1: Blue, 2: Green, 3: Light blue, 4: Red, 5: Purple, 6: Yellow, 7: White)
- Tiling pattern: 0 to 8

  Reference ***Tiling Patterns in Section 3-7 "Displaying a Filled Rectangle"***
- X coordinate: 0000 to 0799 (0 to 799)
- Y coordinate: 0000 to 0599 (0 to 599)
- Radius: 0001 to 0799 (1 to 799)

GP Response data block (from GP)
ACK or NAK

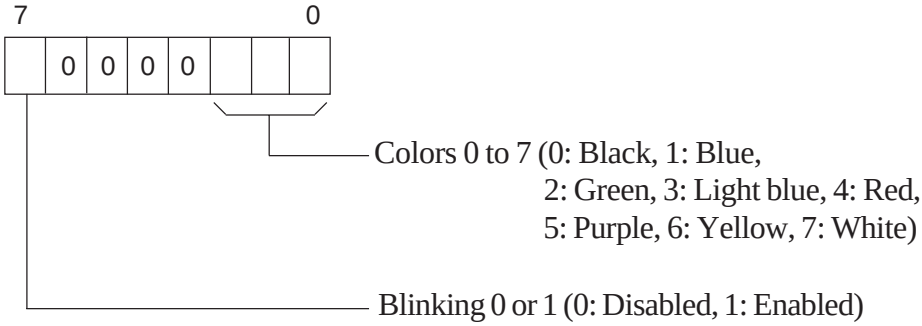
■Binary Mode

Command data block (from Host)



<Setting range>

- Foreground/Background color



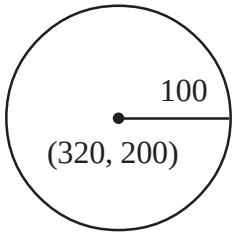
- Tiling pattern : 0 to 8

  Reference ***Tiling Patterns in Section 3-7 "Displaying a Filled Rectangle"***
- X coordinate : 0000H to 031FH (0 to 799)
- Y coordinate : 0000H to 0257H (0 to 599)
- Radius : 0001H to 031FH (1 to 799)

GP Response data block (from GP)
ACK or NAK

<Example>

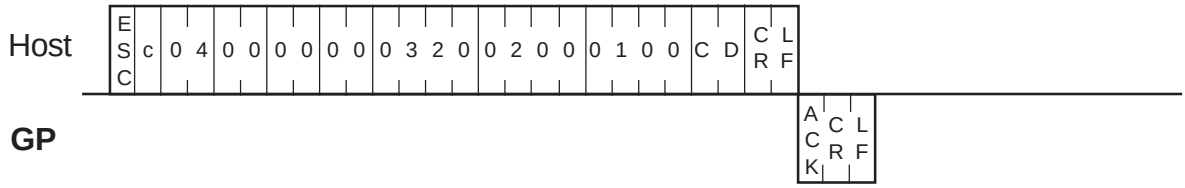
A circle is drawn with the center located at point (320, 200) and a radius of 100.



(Attributes)
Foreground color: White
Background color: Black
Line type: Solid line (0)

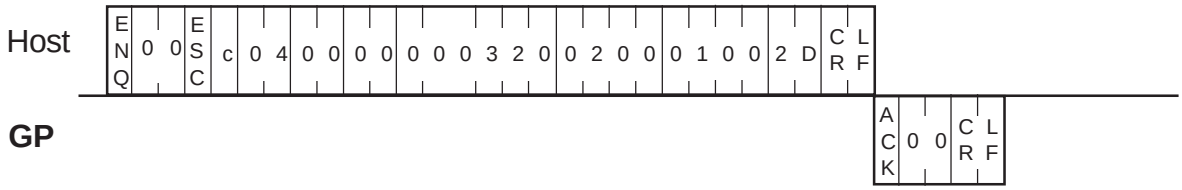
■Extend Mode, 1:1 ASCII,

ETX. SUMCHECK: ENABLED, TERMINATOR: CR.LF, ACK: ENABLED, NAK: ENABLED



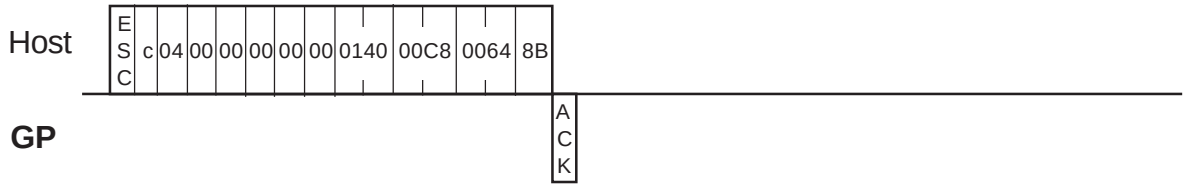
■Extend Mode, 1:n ASCII,

ETX. SUMCHECK: ENABLED, TERMINATOR: CR.LF, ACK: ENABLED, NAK: ENABLED



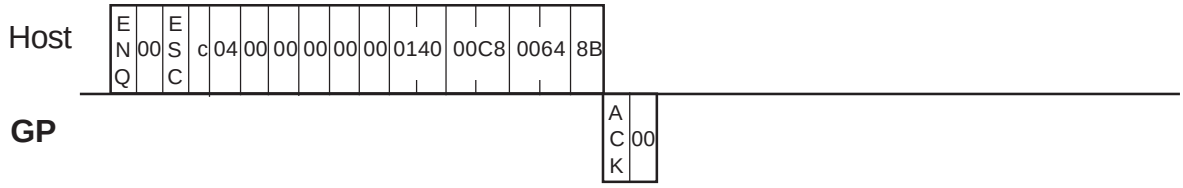
■Extend Mode, 1:1 BINARY,

ETX. SUMCHECK: ENABLED, ACK: ENABLED, NAK: ENABLED



■Extend Mode, 1:n BINARY,

ETX. SUMCHECK: ENABLED, ACK: ENABLED, NAK: ENABLED



3-18 Additional Pie Shape Features [ESC g]

The format of the command data block containing the ESC g command (enhancement to the pie display command) is shown below.
This addition feature allows the selection of the Line's type.

■ASCII Mode

Command data block (from Host)

ESC	g	Blinking	Foreground color	Blinking	Background color	0	Line type	0	0	Center X coordinate	Center Y coordinate	Radius	Start angle	End angle

<Setting range>

•Blinking:	0 or 1 (0: Disabled, 1: Enabled)
• Foreground/Background color:	0 to 7 (0: Black, 1: Blue, 2: Green, 3: Light blue, 4: Red, 5: Purple, 6: Yellow, 7: White)
• Line type :	0 to 3, 8, 9 (0: ——— ,1: _ _ _ _ ,2: ——— - ——— ,3: ——— - - ——— 8: ——— ,9: ———)
• X coordinate:	0000 to 0799 (0 to 799)
• Y coordinate:	0000 to 0599 (0 to 599)
• Radius:	0001 to 0799 (1 to 799)
• Angle:	0000 to 0360

GP Response data block (from GP)
ACK or NAK

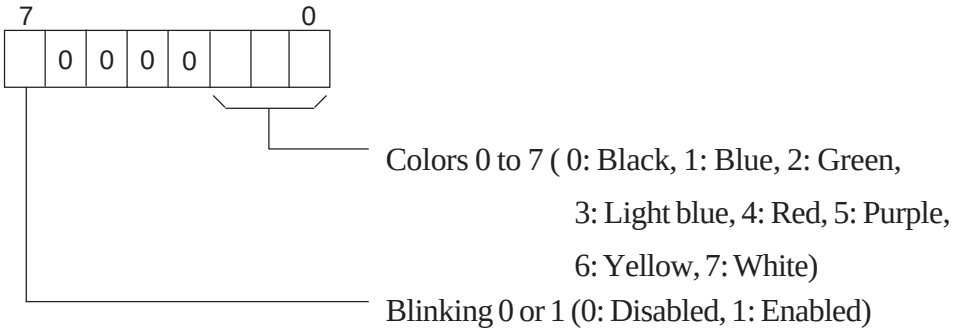
Note:
- Pies are drawn counterclockwise.
- Be sure not to specify the same value for start and end angles.
- Line types 0 to 3 are 1-dot lines while line types 8 and 9 are 3-dot and 5-dot lines, respectively.

■Binary Mode

Command data block (from Host)

<Setting range>

• Foreground/Background color

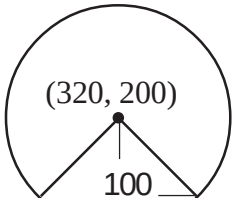


• Line type :	00 to 03H (00: ———,01: – – –,02: — - —,03: — - - —)
• X coordinate :	0000H to 031FH (0 to 799)
• Y coordinate :	0000H to 0257H (0 to 599)
• Radius :	0001H to 031FH (1 to 799)
• Angle :	0000H to 0168H (0 to 360)

GP Response data block (from GP)
ACK or NAK

<Example>

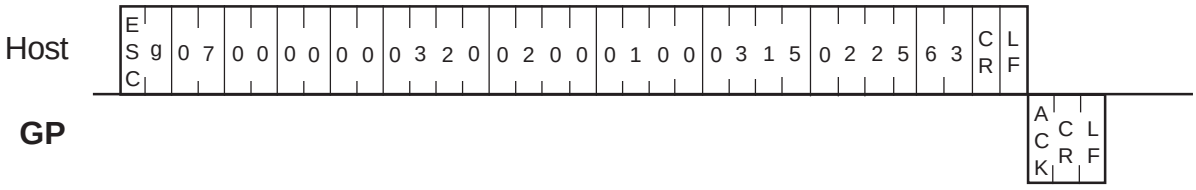
A pie is drawn with the center located at point (320, 200) and a radius of 100.



(Attributes)
Foreground color: White, Start angle: 315°
Background color: Black, End angle: 225°
Line type: Solid line (0)

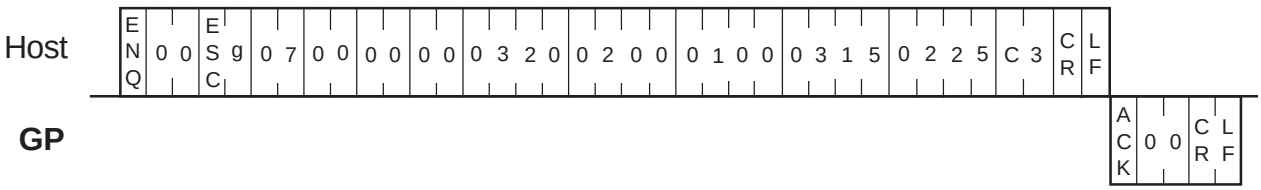
■Extend Mode, 1:1 ASCII,

ETX. SUMCHECK: ENABLED, TERMINATOR: CR.LF, ACK: ENABLED, NAK: ENABLED



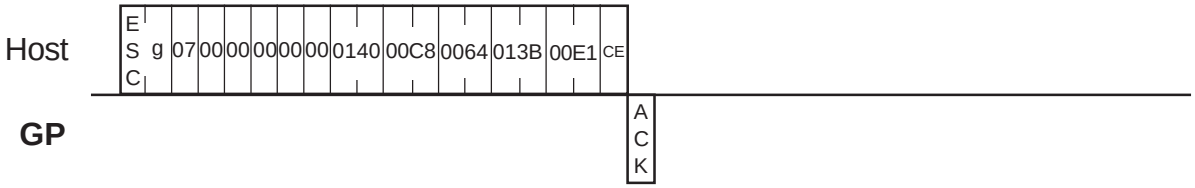
■Extend Mode, 1:n ASCII,

ETX. SUMCHECK: ENABLED, TERMINATOR: CR.LF, ACK: ENABLED, NAK: ENABLED



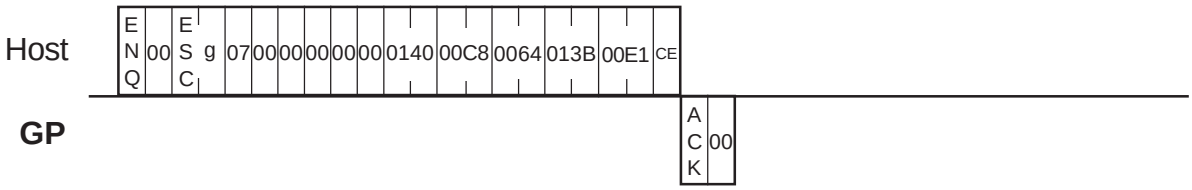
■Extend Mode, 1:1 BINARY,

ETX. SUMCHECK: ENABLED, ACK: ENABLED, NAK: ENABLED



■Extend Mode, 1:n BINARY,

ETX. SUMCHECK: ENABLED, ACK: ENABLED, NAK: ENABLED

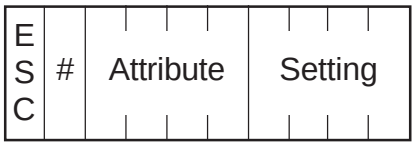


3-19 Brightness and Contrast Adjustments [ESC #]

The format of the command data block containing the ESC # command (brightness and contrast adjustment command) is shown below.
Note that brightness or contrast cannot be adjusted with some GP types.
Reference *Page 3-59, Brightness/Contrast Table*

■ ASCII Mode

Command data block (from Host)



<Setting range>
- Attribute: 0000 or 0001 (0:Contrast, 1:Brightness)
- Settings: Reference *Page 3-59, Brightness/Contrast Table*

GP Response data block (from GP)
ACK or NAK

■Binary Mode

Command data block (from Host)

<Setting range>

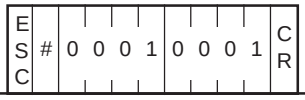
- Attribute: 0000 or 0001 (0:Contrast, 1:Brightness)
- Settings: Reference *Page 3-59, Brightness/Contrast Table*

GP Response data block (from GP)
ACK or NAK

■Brightness Adjustment

■Convert Mode

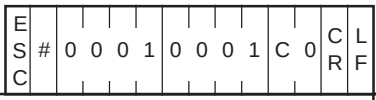
Host



■Extend Mode, 1:1 ASCII,

ETX. SUMCHECK: ENABLED, TERMINATOR: CR.LF, ACK: ENABLED, NAK: ENABLED

Host



GP

■Extend Mode, 1:1

BINARY, ETX. SUMCHECK: ENABLED, ACK: ENABLED, NAK: ENABLED

Host

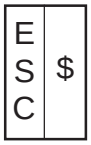
GP

3-20 Brightness and Contrast Settings [ESC $]

The format of the command data block containing the ESC $ command (current brightness and contrast levels command) is shown below.
Note that the brightness or contrast level is not available with some GP types. *1

■ASCII Mode

Command data block (from Host)



GP Response data block (from GP)

<Setting range>

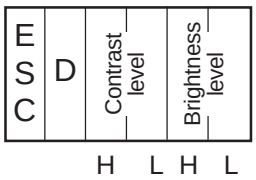
- Attribute: 0000 or 0001 (0:Contrast, 1:Brightness)
- Settings: Reference *Page 3-59, Brightness/Contrast Table*

Note: "FFFF" is output if your GP model does not offer the contrast or brightness adjustment function.

■Binary Mode

Command data block (from Host)

GP Response data block (from GP)



<Setting range>

- Attribute: 0000 or 0001 (0:Contrast, 1:Brightness)
- Settings: Reference *Page 3-59, Brightness/Contrast Table*

Note: "FFFF" is output if your GP model does not offer the contrast or brightness adjustment function.

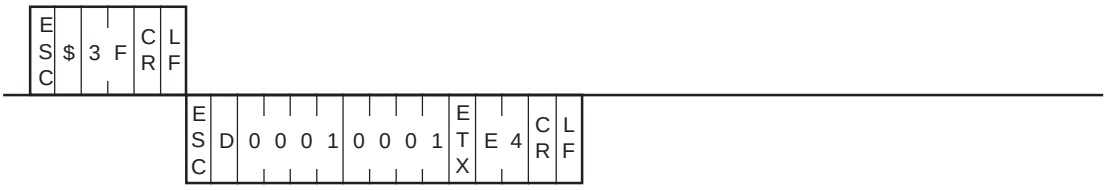
■Convert Mode

Host

■Extend Mode, 1:1 ASCII,

ETX. SUMCHECK: ENABLED, TERMINATOR: CR.LF, ACK: ENABLED, NAK: ENABLED

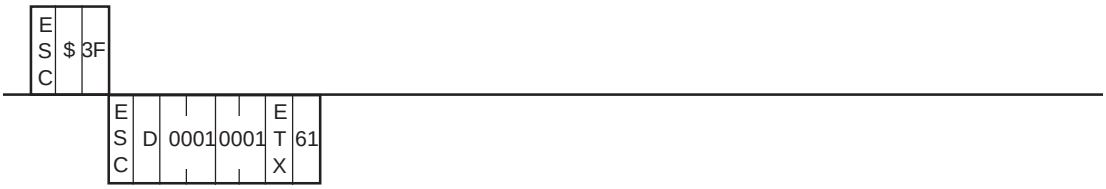
Host



■Extend Mode, 1:1 BINARY,

ETX. SUMCHECK: ENABLED, ACK: ENABLED, NAK: ENABLED

Host



■ Brightness/Contrast Table

	Brightness	Contrast
GP	Setting Range	Setting Range
GP270L		0(Dark) to 7(Bright)
GP270S		0(Dark) to 7(Bright)
GP370L	0(Bright) to 1(Dark)	0(Bright) to 7(Dark)
GP370S	0(Bright) to 1(Dark)	0(Bright) to 7(Dark)
GP470E	0(Bright) to 1(Dark)	
GP570S		0(Bright) to 7(Dark)
GP57JS		0(Bright) to 7(Dark)
GP570T		
GP570VM		
GP571T		
GP675S		0(Bright) to 7(Dark)
GP675T		
GPH70L		0(Bright) to 7(Dark)
GPH70S		0(Bright) to 7(Dark)
GP870VM		
GP57JS		0(Bright) to 7(Dark)
GP37W2		0(Bright) to 7(Dark)

Cannot be set

	Brightness	Contrast
GP/GLC	Setting Range	Setting Range
GP477RE	0(Dark) to 1(Bright)	
GP577RS	0(Bright) to 3(Dark)	0(Bright) to 7(Dark)
GP577RT	0(Bright) to 3(Dark)	
GP377RT	0(Bright) to 3(Dark)	
GP377L	0(Bright) to 3(Dark)	0(Bright) to 7(Dark)
GP377S	0(Bright) to 3(Dark)	0(Bright) to 7(Dark)
GP2600T	0(Bright) to 3(Dark)	
GP2601T	0(Bright) to 3(Dark)	
GP2500T	0(Bright) to 3(Dark)	
GP2501T	0(Bright) to 3(Dark)	
GP2500S	0(Bright) to 3(Dark)	0(Bright) to 7(Dark)
GP2501S	0(Bright) to 3(Dark)	0(Bright) to 7(Dark)
GP2500L	0(Bright) to 3(Dark)	0(Bright) to 7(Dark)
GP2501L	0(Bright) to 3(Dark)	0(Bright) to 7(Dark)
GP2400T	0(Bright) to 3(Dark)	
GP2401T	0(Bright) to 3(Dark)	
GP2300T	0(Bright) to 3(Dark)	
GP2300S	0(Bright) to 3(Dark)	0(Bright) to 7(Dark)
GP2300L	0(Bright) to 3(Dark)	0(Bright) to 7(Dark)
GP2301T	0(Bright) to 3(Dark)	
GP2301S	0(Bright) to 3(Dark)	0(Bright) to 7(Dark)
GP2301L	0(Bright) to 3(Dark)	0(Bright) to 7(Dark)
GP2301HS	0(Bright) to 3(Dark)	0(Bright) to 7(Dark)
GP2301HL	0(Bright) to 3(Dark)	0(Bright) to 7(Dark)
GP2401HT	0(Bright) to 3(Dark)	
ST	0(Bright) to 1(Dark)	0(Bright) to 7(Dark)
GLC2600T	0(Bright) to 3(Dark)	
GLC2400T	0(Bright) to 3(Dark)	
GLC2300T	0(Bright) to 3(Dark)	
GLC2300L	0(Bright) to 3(Dark)	0(Bright) to 7(Dark)

MEMO

Chapter 4: Transferring Screens

This chapter describes the commands used for transferring screen data.

4-1 Command List

Screen data is transferred through the serial interface. To do this, select SET UP OPERATION SURROUNDINGS / SCREEN TRANSFER MODE / MODE SELECTION, and then select "ASCII 6 DIGITS" for SCREEN TRANSFER.

Command	Description
DLE>C	Start data transfer(Communication check)
DLE>Q	End data transfer
DLE>L	Screen file information
DLE>M	Remaining screen memory
DLE>K	Deleting a screen file
DLE<D	Writing a screen file to internal memory
DLE<V	Issuing a system version request
DLE>B	Reading screen files into a specific memory bank
DLE<B	Writing screen files to all banks

- ***The mode (ASCII or BINARY) used for the command and response data blocks is selected in accordance with the currently selected option for SCREEN TRANSFER on the MODE SELECTION screen. Therefore, even if "1:1 BINARY" is selected for TRANSMISSION TYPE on the SET UP OPERATION SURROUNDINGS screen, the ASCII mode is selected for the command and response data blocks when "ASCII 6 DIGITS" is selected for SCREEN TRANSFER on the MODE SELECTION screen. Note, however, that other communication protocol options you have selected will remain valid.***
- ***Be sure not to select "ASCII 5 DIGITS" for SCREEN TRANSFER on the MODE SELECTION screen. This option is designed for use with the GP430.***
- ***If the screen editing software is password-protected on the GP, screen data transfer commands cannot be used.***
- ***When you select "BINARY" for SCREEN TRANSFER on the MODE SELECTION screen, you can transfer screen data from the screen editing software (GP-PRO/PB III for Windows 95).***
  To do this, you need to select "1:1 ASCII" or "1:1 BINARY" for TRANSMISSION TYPE on the SET UP OPERATION SURROUNDINGS screen. At this time, be sure to use the interface cable that allows the DT.R (ER) flow control via RS232C.

- ***A project file created using the screen editing software (GP-PRO/PB III for Windows 95) cannot be transferred as is. To transfer this file, you need to convert it into individual screen files using the file conversion program supplied with the screen editing software.***

For control codes used for command data block Data

Reference *Transmission Control Using Data Transmission in Section 2-2*

◆Error Codes

Code No.	Description	Remedy
0	Transfer complete with no errors	-----
1	Memory write error	The internal FEPROM may be defective. Perform self-diagnosis to check it If it is defective, contact your nearest Digital distributor.
2	Memory error	
3	Invalid command parameter	Check the contents of the command data block and send it again.
4	Insufficient storage capacity	Delete all unnecessary files and transfer again.
5	File checksum error	Send the correct screen file.
6	No file by the specified number	Specify a file number that exists.
8	No bank by the specified number	Specify a bank number that exists.

- Each code number is represented by decimal ASCII code.

4-2 Start Data Transfer (Communication Check) [DLE >C]

The GP assumes the screen data transfer mode when it receives the command data block containing the DLE>C command.

◆ Command data block (from Host)

DLE	>	C

◆ Response data block (from GP)

DLE	<	C

<Example>

The host sends the DLE>C command to check that the GP is in the screen data transfer mode.

■ **Extend Mode, 1:1 ASCII**

EXT. SUMCHECK: ENABLED, TERMINATOR: CR. LF, ACK: ENABLED, NAK: ENABLED

"RECEIVE DATA ERROR (02:FD)" may appear if the GP receives the command data block before it changes to the screen data transfer mode. However, this error message will disappear when the GP receives the command data block again after it resumes the screen data transfer mode.

4-3 End Data Transfer [DLE >Q]

The host changes to run mode when it sends the command data block containing the DLE>Q command.
(The GP is reset when it sends the response data block containing the DLE<Q command.)

◆ Command data block (from Host)

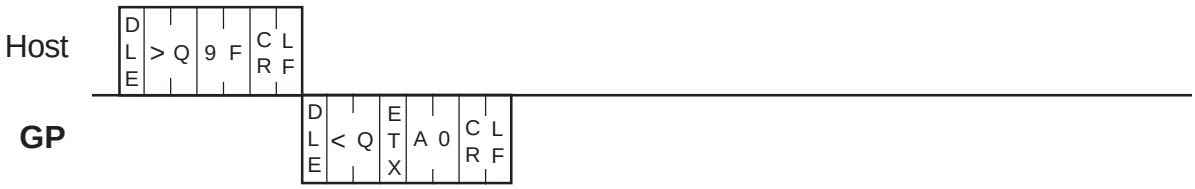
D L E	>	Q

◆ Response data block (from GP)

D L E	<	Q

<Example>
The host sends the DLE>Q command to change to run mode.

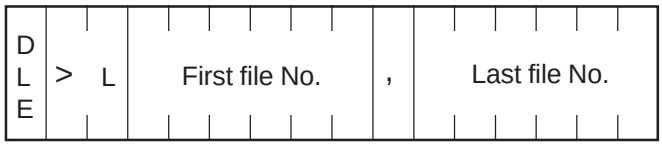
■ Extend Mode, 1:1
ASCII, ETX. SUMCHECK: ENABLED, TERMINATOR: CR.LF, ACK: ENABLED, NAK: ENABLED



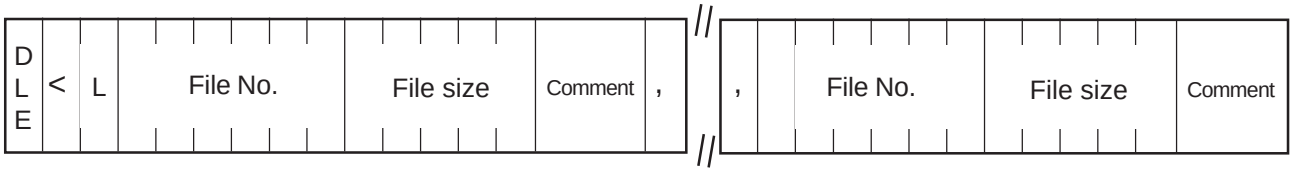
4-4 Screen File Information [DLE >L]

The GP outputs information on the specified screen files stored in the memory when it receives the command data block containing the DLE>L command from the host.

◆Command data block (from Host)



◆Response data block (from GP)



or

<Description of fields>

• File No. *1:B00001 to B13999 (B00001 to B13999 contain screens that are automatically generated when you finish positioning parts.)

M00001 to M09999
K00001 to K09999
A00001 to A09999
T00001 to T10019 (T10000 to T10019 contain screens that are automatically generated when you finish positioning parts.)
X00001 to X09999
I00001 to I09999
Q00001 to Q00004
S0
W0

• File size: 5-digit ASCII code in decimal notation
(Example) "01848" represents 1848 (decimal) bytes.

• Comment field contains ASCII code. Therefore, use this as character string data.

---

*1 Make sure that the device characters ("A", "B", etc.) used in the command data block match.
The following is not allowed: ~~B0001 to K0010~~

The following shows the error code numbers and the description of each error:

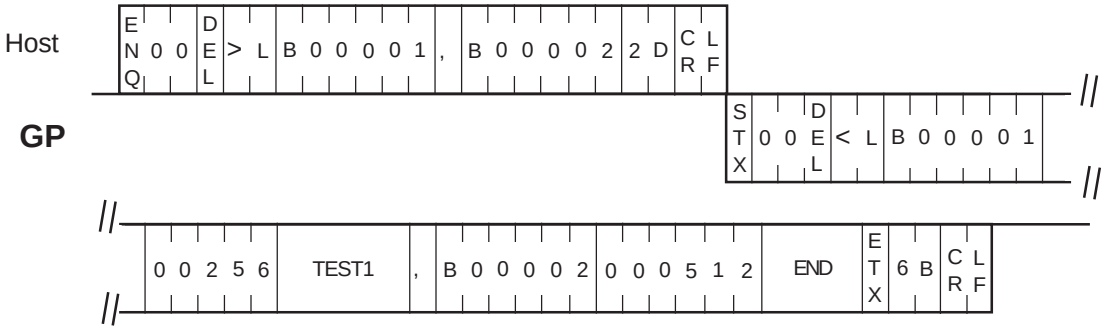
0: Transfer complete with no error
1: Memory write error
2: Memory error
3: Invalid command parameter
4: Insufficient storage capacity
5: File checksum error
6: No file by the specified number
8: No bank by the specified number

<Example>

The GP outputs the information on files B0001 and B0002.

■ Extend Mode, 1:n ASCII,

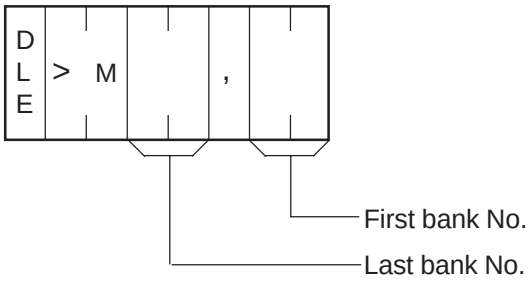
ETX. SUMCHECK: ENABLED, TERMINATOR: CR.LF, ACK: ENABLED, NAK: ENABLED



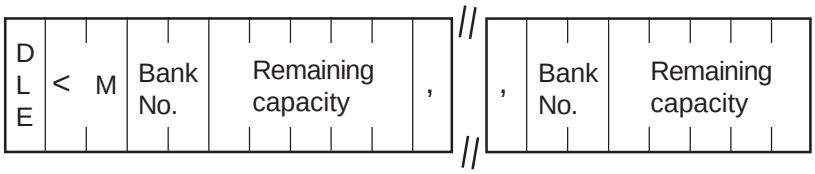
4-5 Remaining Screen Memory [DLE >M]

The GP outputs the remaining memory capacity per bank when it receives the command data block containing the DLE>M from the host.

◆ Command data block (from Host)



◆ Response data block (from GP)



or

<Description of fields>

- Bank No.: 00 to 64 (2-digit ASCII code in decimal notation)
- Remaining capacity: 5-digit ASCII code in decimal notation
  (Example) "59526" represents 59526 (decimal) bytes.
- The following shows the error code numbers and the description of each error:

0: Transfer complete with no errors
1: Memory write error
2: Memory error
3: Invalid command parameter
4: Insufficient storage capacity
5: File checksum error
6: No file by the specified number
8: No bank by the specified number

When you specify a bank number outside the permissible range, the GP returns only the DLE<M command with no error code.

Note:

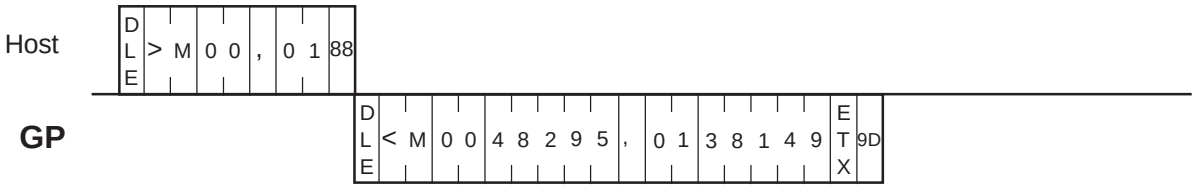
- The GP manages the internal memory space by dividing it into multiple sections of 64 Kbytes each. Each of these sections is called a bank.

<Example>

The GP outputs the remaining capacity in banks 00 and 01.

■Extend Mode, 1:1 BINARY,

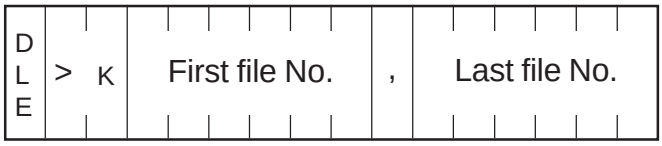
ETX. SUMCHECK: ENABLED, ACK: ENABLED, NAK: ENABLED



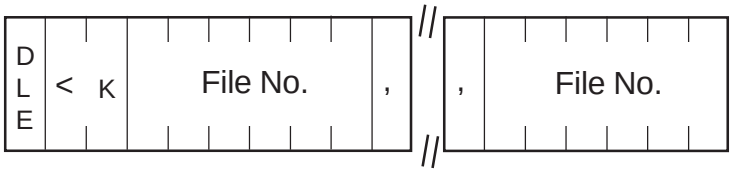
4-6 Deleting a Screen File [DLE >K]

The GP deletes the specified screen files stored in the internal memory when it receives the command data block containing the DLE>K command from the host.

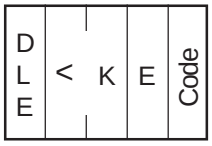
◆Command data block (from Host)



◆Response data block (from GP)



or



<Description of fields>

- File No. *1: B00001 to B13999 (B10000 to B13999 contain screens that are automatically generated when you finish positioning parts.)
  M00001 to M09999
  K00001 to K09999
  A00001 to A09999
  T00001 to T10019 (T10000 to T10019 contain screens that are automatically generated when you finish positioning parts.)
  X00001 to X09999
  I00001 to I09999
  Q00001 to Q00004
  S0
  W0
- The following shows the error code numbers and the description of each error:
  0: Transfer complete with no errors
  1: Memory write error
  2: Memory error
  3: Invalid command parameter
  4: Insufficient storage capacity
  5: File checksum error
  6: No file by the specified number
  8: No bank by the specified number

---

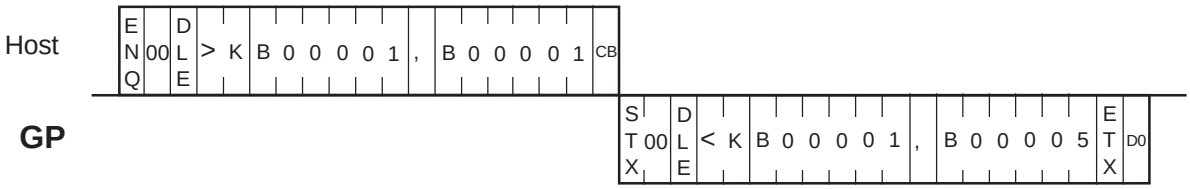
*1. Make sure that the device characters ("A", "B", etc.) used in the command data block match.
The following is not allowed: ~~B0001 to K0010~~

<Example>

The GP deletes screen files B00001 through B00010.

■Extend Mode, 1:1 BINARY,

ETX. SUMCHECK: ENABLED, ACK: ENABLED, NAK: ENABLED



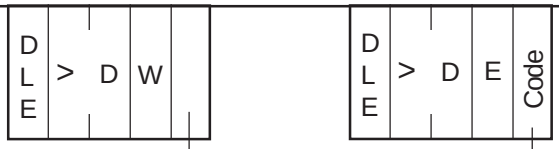
4-7 Writing a Screen File to Internal Memory [DLE <D]

The GP writes a specific screen file of a project file to internal memory when it receives the command data block containing the DLE<D command from the host.
To write screen files from a project file to internal memory, you need to convert this project file into individual screen files.

Reference *4-11 Project File Conversion Tool*

◆Command data block (from Host)

◆Response data block (from GP)



2: Writing data (requests the host to wait)

0: Data writing complete with no errors (Result output)

<Example>

The GP writes file B0001 to internal memory.

- When you wish to write a file of a specific screen, you need to check if any other files associated with that screen have been generated by the project file conversion tool. If so, you also need to write these files to internal memory together with the desired file. Remember that when screen elements are used on a screen, subordinate screens are automatically generated for those elements.
- File data must be converted from binary data to ASCII data. The divided files are binary data. Also, be sure to convert binary data (A to F) to ASCII data (41H to 46H).

  Eg.) Binary Data -> ASCII Data
  2AH -> 32H 41H

<Description of field>

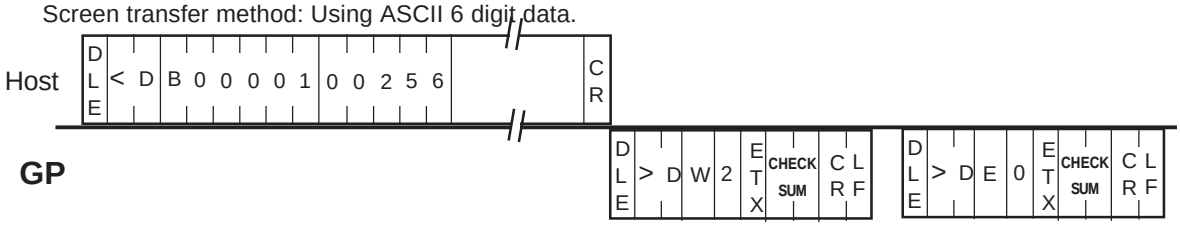
• The following shows the error code numbers and the description of each error:

0: Transfer complete with no errors
1: Memory write error
2: Memory error
3: Invalid command parameter
4: Insufficient storage capacity
5: File checksum error
6: No file by the specified number
8: No bank by the specified number

■Extend Mode, 1:1 ASCII,

EXT. SUMCHCK: ENBLED, TERMINATOR: CR,LF, ACK: ENABLED, NAK: ENABLED,

Screen transfer method: Using ASCII 6 digit data.



4-8 Issuing a System Version Request [DLE >V]

The host uses the DLE>V command to issue a request to the GP that information on the GP (GP model, version number, date, and target PLC be sent to the host.)

◆Command data block (from Host)

D L E	>	V

◆Response data block (from GP)

D L E	<	V	Model	,	Version No.	,	Date of manufacture	,	PLC No.

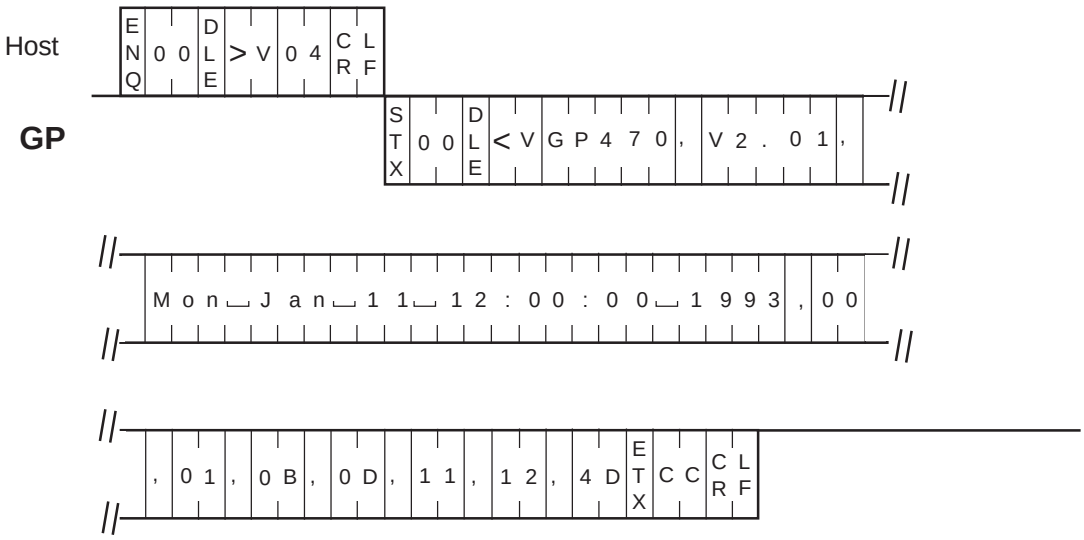
<Description of fields>

- Model: (Example) GP470
  Version No.: (Example) V2.01
  Date of manufacture: (Example) Mon_Jan_11_12:00:00_1993
  PLC No.: Reference *Your GP Series unit's User Manual - 6.4 Error Messages, Error Message List/OBJ.PLC HAS NOT BEEN SETUP*

<Example>

The host issues an enquiry about the version number of the GP system.

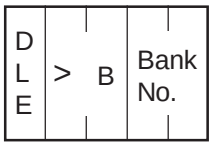
■ Extend Mode, 1:n ASCII,
ETX. SUMCHECK: ENABLED, TERMINATOR: CR.LF, E



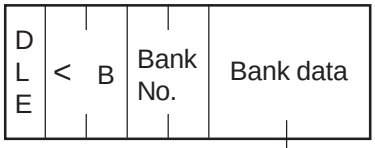
4-9 Reading Screen Files into a Specific Memory Bank [DLE >B]

The GP reads all screen files in the specified bank of internal memory when it receives the command data block containing the DLE>B command from the host.

◆Command data block (from Host)



◆Response data block (from GP)



or

<Description of field>

- Bank No.: 00 to 64
- Bank data : Internal memory
- The following shows the error code numbers and the description of each error:

0: Transfer complete with no errors
1: Memory write error
2: Memory error
3: Invalid command parameter
4: Insufficient storage capacity
5: File checksum error
6: No file by the specified number
8: No bank by the specified number

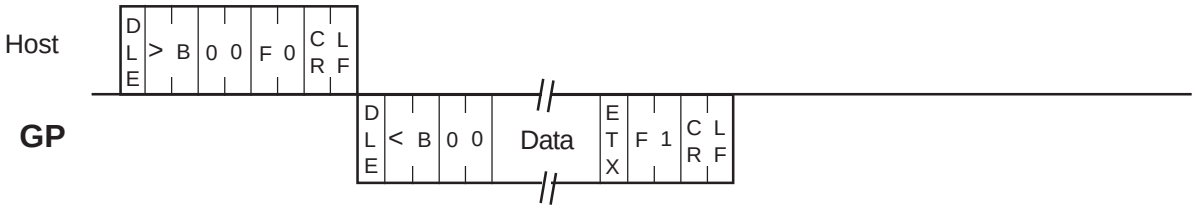
The DLE<B command contained in the response data block reads screen files in the specified bank of internal memory.

<Example>

The GP reads all screen files in bank 00.

■Extend Mode, 1:1 ASCII,

ETX. SUMCHECK: ENABLED, TERMINATOR: CR.LF, ACK: ENABLED, NAK: ENABLED



Note: Be sure to read screen files in all banks. You cannot read screen files in the specified bank.

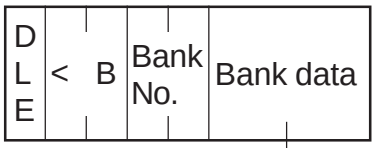
<Number of banks available>

Model	Banks
GP270 Series	4
GPH70 Series	16
GP370 Series	16
GP470 Series	16
GP570 Series (GP-571 excluded)	16
GP-571T	48
GP675 Series	32
GP870 Series	16
GP2301H Series	16
GP2401H Series	32
GP2300/2301 Series	32
GP2400/2401 Series	64
GP2500/2501 Series	64
GP2600/2601 Series	64
ST Series	64
GLC2300 Series	32
GLC2400 Series	64
GLC2600 Series	64

4-10 Writing Screen Files to All Memory Banks [DLE <B]

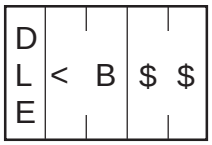
The GP writes screen files to all banks of internal memory when it receives the command data block containing the DLE<B command from the host.

◆Command data block (from Host)



128 Kbytes

or

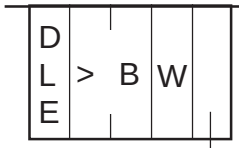


(Bank data transfer complete)

<Description of field>

• Bank No.: 00 to 64

◆Response data block (from GP)



1:Initializing
2: Writing data

(requests the host to wait)

0: Data writing complete with no errors

(Result output)

<Description of field>

• The following shows the error code numbers and the description of each error:

0: Transfer complete with no errors
1: Memory write error
2: Memory error
3: Invalid command parameter
4: Insufficient storage capacity
5: File checksum error
6: No file by the specified number
8: No bank by the specified number

Note:

- The GP manages the internal memory space by dividing it into multiple sections of 64 Kbytes each. Each of these sections is called a bank.
- Be sure to send multiple command data blocks to ensure that screen files are written to all banks of internal memory. Also, be sure to send a command data block containing the DLE<B$$ command at the end to indicate that there are no more command data blocks.

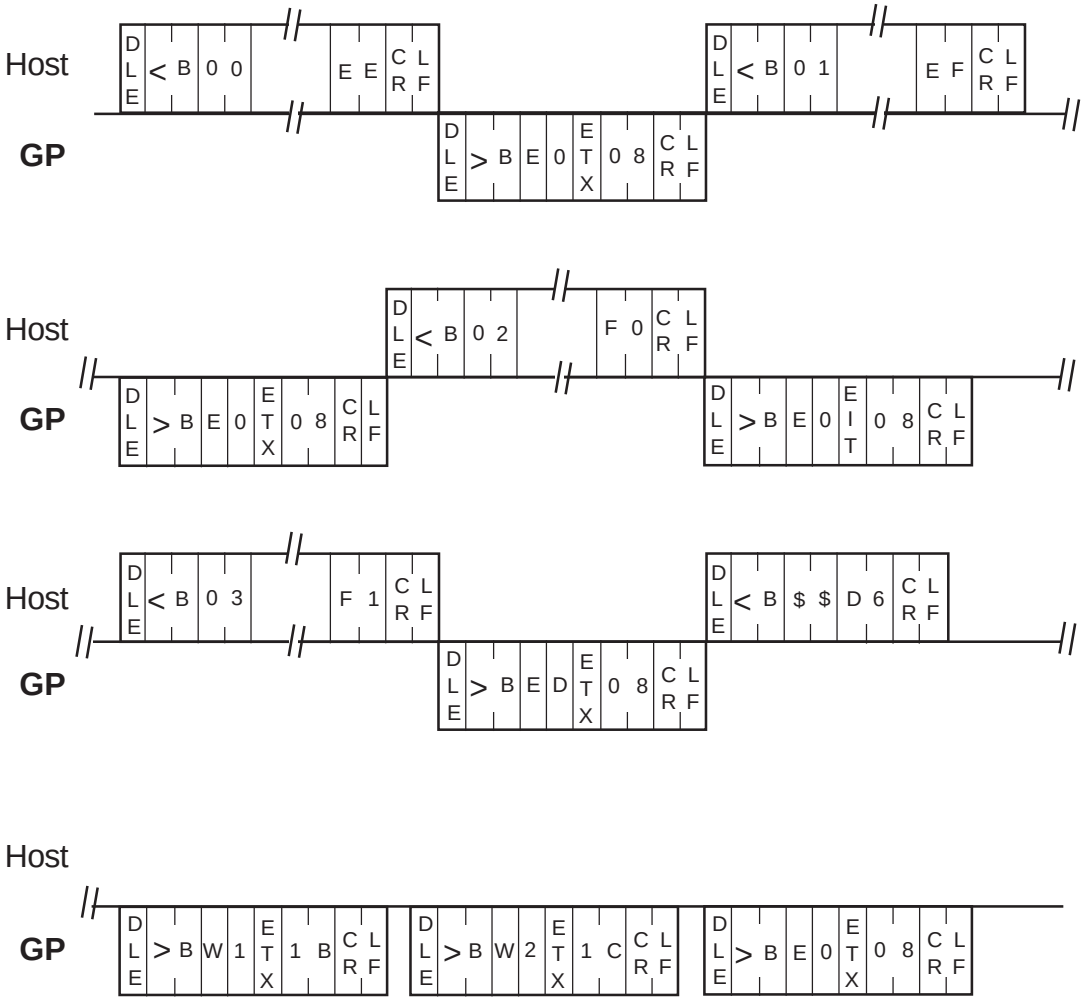
	(Bank No.)	(Acceptable: Yes/No)
GP-270:	No.0, No.1, - - - No.3, $$	Yes
GP-470:	No.0, No.1, - - - No.15, $$	Yes
	No.4, No.5, No.6, No.7, $$	No

<Example>

The GP writes screen files to all banks (banks 00 to 03).

■Extend Mode, 1:1 ASCII,

ETX. SUMCHECK: ENABLED, TERMINATOR: CR.LF, ACK: ENABLED, NAK: ENABLED



4-11 Project File Conversion Tool

This tool is designed to convert a GP-PRO/PBIII project file into screen files. A project file cannot be written to internal memory unless converted into individual screen files. This tool is included with the GP-PRO/PBIII for Windows 95 Ver.1.1 or later software.

◆File name

GPANALYZ.EXE

Included with the GP-PRO/PBIII for Windows 95 Ver.1.1 or later. This file will be automatically installed in the directory you specify when installing the GP-PRO/PBIII for Windows software.

◆How to use the project file conversion tool

Open the Start menu and select Programs|MS-DOS Prompt to display the DOS window. (This tool runs in the DOS environment.) Then, follow the steps below.

1 : Enter "GPANALYZ" and press the return key at the DOS prompt.

2 : Input Project File Name:
Enter the target project file name for "Input Project File Name:". Be sure to specify the full path including the drive name.

3 : Enter the GP screen file type for "Output GP File Attribute [*]".
The available types are shown on the next page. Enter "*" when you wish to select all types.

4 : Enter the first screen file number for "Output GP File Start Number [0]".
Normally, enter "0".

5 : Enter the last screen file number for "Output GP File End Number [9999]".
Normally, enter "9999".

◆Screen File Types

Screen file type	File No.	Description
B	1 to 13999	Base screen files (files 10000 to 13999 are generated automatically when screen elements are used)
M	1 to 9999	Mark screen files
T	1 to 10019	Trend graph screen files (files 10000 to 10019 are generated automatically when screen elements are used)
A	1 to 9999	Alarm Message/Summary files
K	1 to 9999	Keypad screen files
X	1 to 9999	Text screen files
I	1 to 9999	Image screen files
Q	1 to 4	Bit Alarm Log and Word Alarm Log files
S	0	GP system settings file
W	0	Window registration file

The extension for screen files is determined in accordance with the PLC type set for the project file.

◆Message

The following message will appear during conversion:
B1.DLM is Created
B10015.DLM is Created
Total create files:2

◆Error messages

Project File Not Found	The specified file cannot be found. Check the file name and path specification.
Cannot allocate memory	Not enough memory to start the program. Delete unnecessary files.
GP File XXX cannot be created	Output file cannot be created. Check the destination memory for available space and write protection.
File format error	The project file is not in the proper format. It may be damaged. Use the rebuild tool to restore it.
This project has not been prepared	The GP unit is not ready. Send a data transfer start command to cause the GP unit to assume the data transfer mode.

Chapter 5: Creating Programs

5-1 Sample System

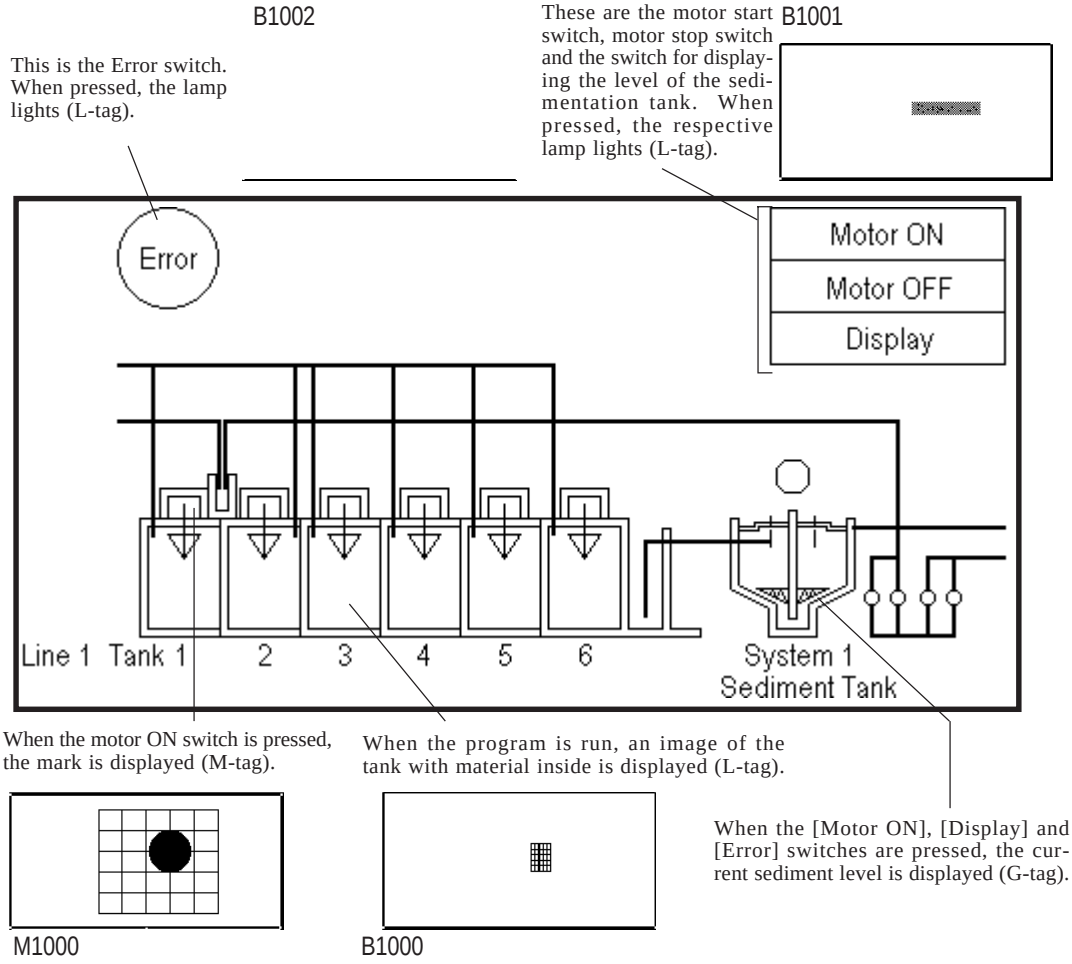
This section provides examples of the Host program and GP tag setup which are necessary for data transfer between the GP and Host. Plus, when the tag setup below is run with the sample program, it demonstrates a GP screen change.
Use the following steps to create the screens shown below.

When the [Motor ON], [Motor OFF], [Display], or [Error] switch is pressed, that switch's respective interrupt code is output to the host system, starting the following operations (T-tag).

Switch Explanation

[Motor ON]	Starts the motor to supply 50% of the sediment into the sedimentation tank.
[Motor OFF]	Stops the motor.
[Display]	"50% of the sediment is being supplied to the sedimentation tank."
[Error]	"Only 20% of the sediment has been supplied to the sedimentation tank."

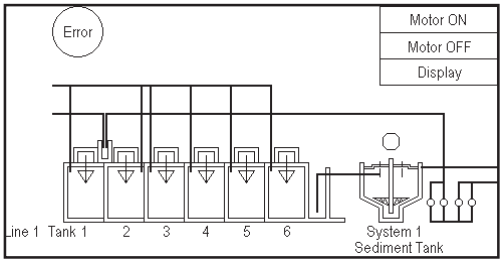
■ System Example



■ Screen Creation

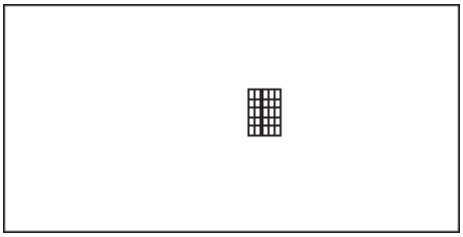
(1) Use the GP-PRO/PBIII for Windows software to create the following screens.

B1



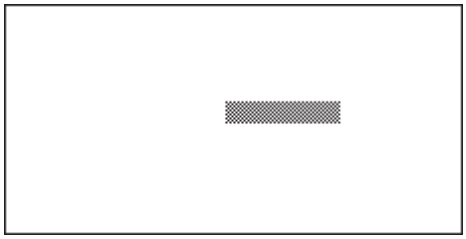
This screen is displayed when the GP is operating.

B1000



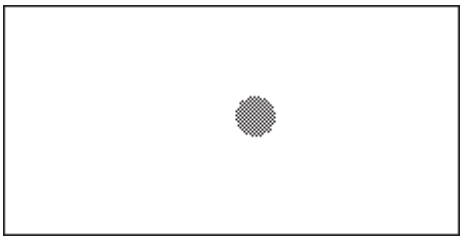
This screen represents the material to be put into the aeration tank.

B1001



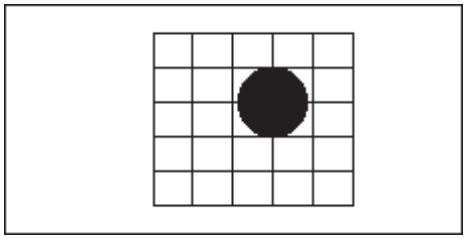
This screen shows the [Motor ON], [Motor OFF] and [Display] lamp's ON pattern.

B1002



This screen shows the [Error] switch's ON pattern.

M1000



This screen shows the mark displayed when the motor is started.

(2) Use the GP-PRO/PBIII for Windows software to setup Tags

◆ Tag Setup Example

T-tag List

File No.	Tag Name	Ope. Mode	Word Addr	Word Write	Fixed No.	Reverse Display	Starting point coordinate	Ending point coordinate	Details
B1	T1	Word	13	Word Set 16 bit	0031	On	Coordinates entered will vary according to screen being used.		Motor ON
B2	T2				0032				Motor OFF
B1	T3				0033				Display
B1	T4				0034				Error

L-tag List

File No.	Tag Name	Diplay Mode	Bit Addr	File Access	Direct Access File No.	Delete	Display coordincates	Detailas
B1	L1	0 --> 1 Erase Operation On	002000	Direct Access	B1000	On	Coordinates entered will vary according to screen being used.	Tank 1
B1	L2		002001					Tank 2
B1	L3		002002					Tank 3
B1	L4		002003					Tank 4
B1	L5		002004					Tank 5
B1	L6		002005					Tank 6
B1	L11		002100		B1001			"Motor ON" Reverse
B1	L12		002101					"Motor OFF" Reverse
B1	L13		002102					"Display" Reverse
B1	L14		002103		B1002			"Error" Reverse

When you wish to display L1~6 (L-tag) at the same time, turn all bits in address 20 ON.

M-tag List

File No.	Tag Name	Bit Addr	Diplay Mode	Color Attr. 0	Color Attr. 1	Magnify	File Access	Direct Access File No.	Display coordinates	Details
B1	M1	002200	Displays On/Off	Fg Clr:Black Bg Clr: Black Blink: Off	Fg Clr:White Bg Clr: Black Blink: Off	1 X 1	Direct Access	M1000	Coordinates entered will vary according to screen being used.	Motor 1
B1	M2	002201								Motor 2
B1	M3	002202								Motor 3
B1	M4	002203								Motor 4
B1	M5	002204								Motor 5
B1	M6	002205								Motor 6

When you wish to display L1~6 (M-tag) at the same time, turn all bits in address 20 ON.

G-tag List

File No.	Tag Name	Word addr	Data Type	Data Format	Code	Bit Length	Input Code	Color Attr.	Graph type	Dir.	Tiling Pttrn.	Alarm	Starting point coordinate	Ending point coordinate	Details
B1	G1	0023	Relative Value	BCD	+	16	None	Fg: White Bg: Black Blink: Off	Trend Graph	Up	2	Off	Coordinates entered will vary according to screen being used.		Sediment Tank

◆ **Address Map**

Tags used in the Sample program are allotted to their corresponding address as follows.

T-tag —> Address 13

Writing data to Address 13 (Interrupt) causes an output of the bottom 1 byte code from the RS-232C port. For this reason, the T-tag uses word write.

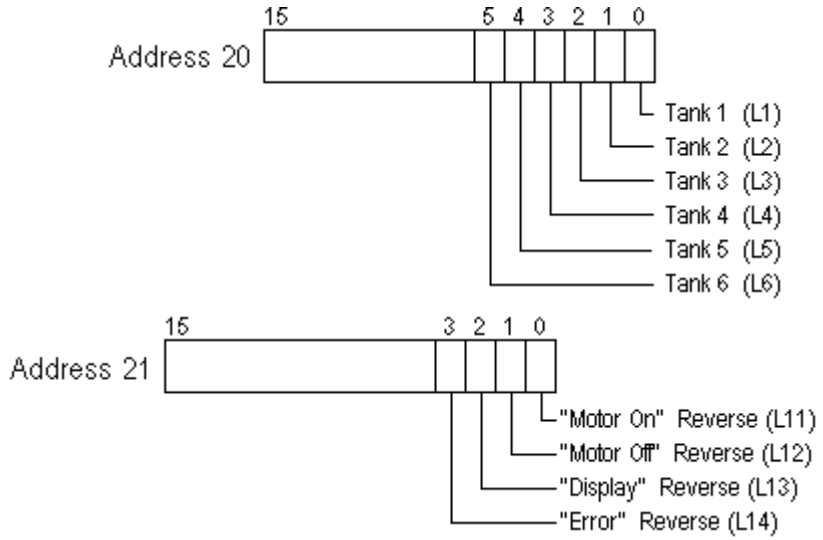
Motor ON (T1) ..... word write 0031 to address 13

Motor OFF (T2) ..... word write 0032 to address 13

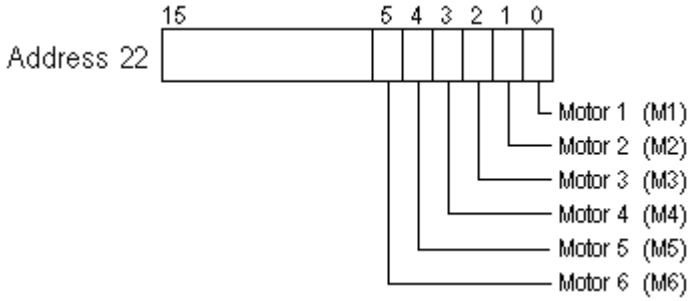
Display (T3) ..... word write 0033 to address 13

Error (T4) ..... word write 0034 to address 13

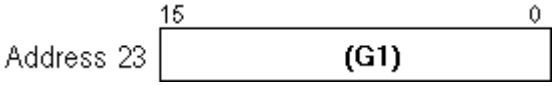
L-tag —> Address 20, Address 21



M-tag —>Address 22



G-tag —> Address 23



(3) The host unit's company creates the program for data transfer between the GP and the host.

◆ Sample Program

E.g. If Windows-compatible PC and the C language are used:

```
/*****************************************************************************************************************/
/*                                                                                                               */
/*  GP series     Sample program for memory link communications                                                  */
/*                                                                                                               */
/*****************************************************************************************************************/

#include<stdio.h>
#include<dos.h>
#include<string.h>
#include<stdlib.h>
#include<conio.h>

#define  data_size_str2            20        /*The data size of str2 is 20 bytes*/
#define  data_size_wr_data         24        /*The data size of wr_data is 24 bytes*/

#define  serial_port_BIOS          0x14      /*PC serial port BIOS*/
#define  serial_port_number        0x00      /*Serial port number used*/
#define  serial_port_INT           0xE7      /*The serial port is initialized.*/
#define  serial_port_parameter     0xE7      /*9600bps,8bit,stopbit;1,parity;none*/

#define  get_status                0x03      /*The status of the serial port is acquired.*/
#define  serial_port_write         0x01      /*The serial port is written.*/
#define  serial_port_read          0x02      /*The serial port is read out.*/

#define  status_bit_6000           0x60000   /*Port status bits 13 and 14*/
#define  status_bit_0020           0x0020    /*Port status bit 5*/

/*****************************************************************************************************************/
/*                Communications settings for the SI0                                                            */
/*****************************************************************************************************************/
void  open_SIO (void);                        /*Communications settings for RS232C*/

/*****************************************************************************************************************/
/*              Acquisition and identification of port status                                                    */
/*****************************************************************************************************************/
int   err_status (void);  /*The port status is acquired.*/
void  write_ready (void); /*The transmission buffer register and the transmission register statuses are acquired.*/
int.  read_ready (void);  /*Confirmation of data set status*/

/*****************************************************************************************************************/
/*                          Writing data                                                                         */
/*****************************************************************************************************************/
void  write_data (char wr_data);     /*The data is written to the registers.*/
void  write (char *wr_data);         /*The data is written to the GP.*/

/*****************************************************************************************************************/
/*                          Reading data                                                                         */
/*****************************************************************************************************************/
int   read_data (void);              /*The data is read from the GP.*/
void  change_screen (int interrupt_data);  /*The received data in an interruption from the GP is identified.*/
int   read (void);                   /*The received data in an interruption from the GP is read.*/

/*****************************************************************************************************************/
/*                    Confirmation of key entries                                                                */
/*****************************************************************************************************************/
int   kbhit (void);

/*****************************************************************************************************************/
/*                          Global variables                                                                     */
/*****************************************************************************************************************/
int   interrupt_data,port_status+;
char  *str2;
```

```
void main (void)
{
        int no_data;
         str2 =  (char*) malloc (sizeof (char) *data_sezi_str2); /*The memory for str2 is secured.*/
         char *wr_data =  (char*) malloc (sizeof (char) *data_size_wr_data);
                                        /*The memory for wr_data is secured.*/
         open_SIO ();          /*Communication settings for RS232C*/
         wr_data = "\x1bW000F0001\x0d\0" ; /*0x1 is written to address 15: screen number 1 setup*/
        write (wr_data);
        wr_data = "\x1bW0014003F\x0d\0";
              /*0x3F is written to address 20: Materials are put into aeration tanks Nos. 1 to 6.*/
        write (wr_data);
/*****************************************************************************/
/*      The data reception from the GP is identified.                        */
/*  If the Write key is pressed, the execution is completed.                 */
/*****************************************************************************/
        while (1)
        {
                no_data = read ();
                if (no_data == 1)      /*If there is any key entry, no_data=1.*/
                {
                     break;
                }
                else
                {
                      wr_data = str2;
                      write (wr_data);
                }
        }
        getch ();                     /*The codes for keys are removed from the key buffer.*/
        free (wr_data);                    /*The memory area for wr_data is freed up.*/
        free (str2);                  /*The memory area for str2 is freed up.*/
}

/*The transmission buffer register status and the transmission register status are acquired.*/
void write_ready (void)
{
        int err6000;

        err6000 = 0;
         while (status_bit_6000 != err6000)
        {
               err6000 = err_status () & status_bit_6000;
        }
        return;
}

/*Confirmation of data set status*/
int read_ready (void)
{
        int no_data,err0020;

        err0020 = 0;
         while (status_bit_0020 != err0020)
        }
               err0020 = 344_status () & status bit_0020;
              if (kbhit ())       /*Confirms whether there is a key entry or not.*/
              {
                     no_data = 1;/*If there is a key entry, no_data=1.*/
                      break;      /*The program is terminated.*/
              }
```

```
    }
     return (no_data);
}
/*Data is written to the GP.*/

void write (char *wr_data)
{
      while (*wr_data != '\0')        /*The data is written until it becomes NULL.*/
     {
            write_ready ();
             write_data (*wr_data);
              wr_data++;         /*The address pointed to by the pointer is incremented.*/
      }
     return;
}
 /*****************************************************************************/
/*          The interrupt data received from the GP is confirmed.         */
/*          The data is written to addresses 20, 21, 22, and 23.          */
 /*****************************************************************************/
void change_screen (int interrupt_data)
{
       switch (interrupt_data)
/*If interrupt_data is 1, 0x1 is written to address 21, 0x3F to address 22, and 0x50 to address 23.*/
                case   1: str2 = "\x1bW00150001003F0050\x0d\0";
                       break;

/*If interrupt_data is 2, 0x2 is written to address 21, 0x0 to address 22, and 0x0 to address 23.*/
                case   2: str2 = "\x1bW0015000200000000\x9d\0";
                       break;

/*If interrupt_data is 3, 0x4 is written to address 21, 0x0 to address 22, and 0x50 to address 23.*/
                case   3: str2 = "\x1bW00150000400000050\x0d\0";
                       break;

/*If interrupt_data is 4, 0x8 is written to address 21, 0x0 to address 22, and 0x20 to address 23.*/
                case   4; str2 = "\x1bW0015000800000020\x0d\0";
                       break;

/*If interrupt_data is other than 1 to 4, NULL is written.*/
               default  : str2 = "\0";
                      break;
       {
       return;
}

 /*****************************************************************************/
/*          The interrupt data received from the GP is read.              */
 /*  Reading is performed until the interrupt_data becomes other than NULL.  */
 /*****************************************************************************/
int read (void)
{
       int no_data;
       do
       {
              no_data = read_ready (); /*Confirmation of data set status*/
              if (no_data == 1)       /*If there is a key entry, no_data=1.*/
              {
                 break;
              }
              else
              {
```

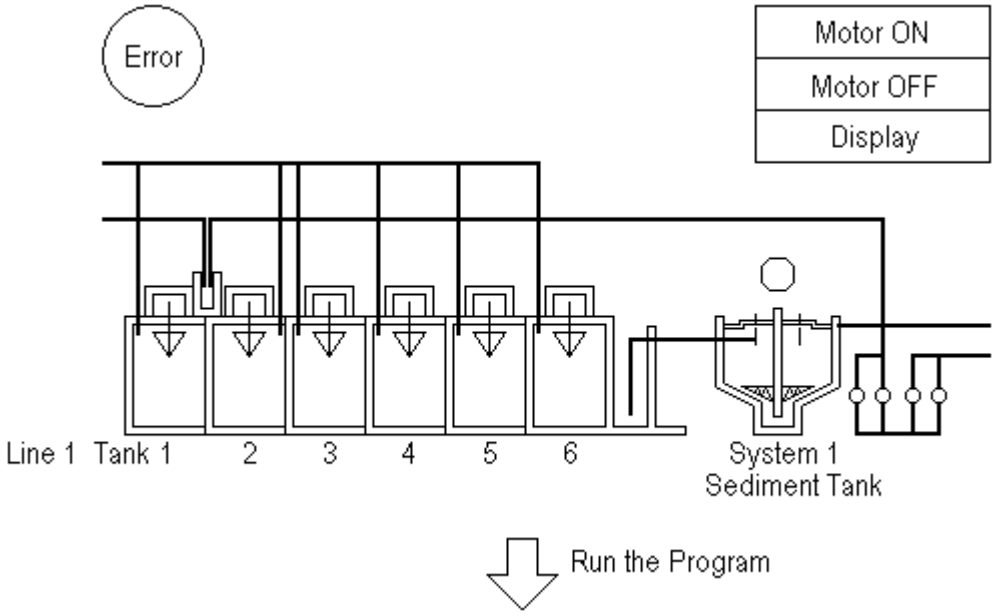
```
                    read_data ();      /*The data received from the GP is read out./*
                     change_screen (interrupt_data); /*The data received from the GP is identified.*/
                  }
            } while (*str2 == '\0');
            return (no_data);
    }
     /*Communications settings for RS232C*/

void open_SIO (void)
{
        union REGS regs ;
                regs.x.dx = serial_port_number;
                regs.h.ah = serial_port_INT;
                 regs.h.al = serial_port_parameter;
                 int86 (serial_port_BIOS,&regs,&regs);
       return;
}

/*The port status is acquired.*/
int err_status  (void)
{
        union REGS regs;
                regs.x.dx = serial_port_number;
                regs.h.ah = get_status;
                 int86 (serial_port_BIOS,&regs,&regs);
                port_status = regs.x.ax;
        return (port_status);
}

/*The data is written to the registers*/
void write_data (char wr_data)
{
        union REGS regs;
                regs.x.dx = serial_port number;
                 regs.h.ah = serial_port_write;
                regs.h.al = wr_data;
                 int86 (serial_port_BIOS,&regs,&regs);
       return;
}

/*The data is read from the GP*/
int read_data (void)
{
        union REGS regs;
                regs.x.dx = serial_port_number;
                 regs.h.ah = serial_port_read;
                 int86 (serial_port_BIOS,&regs,&regs);
                interrupt_data = regs.h.al;
        return (interrupt_data);
}
```

The availability of open_SI0 (void), err_status (void), write_data (char wr_data), and read_data (void) will depend on the PLC model used. If the program is written on a personal computer that is not Windows-compatible, it must be modified in order to be used.

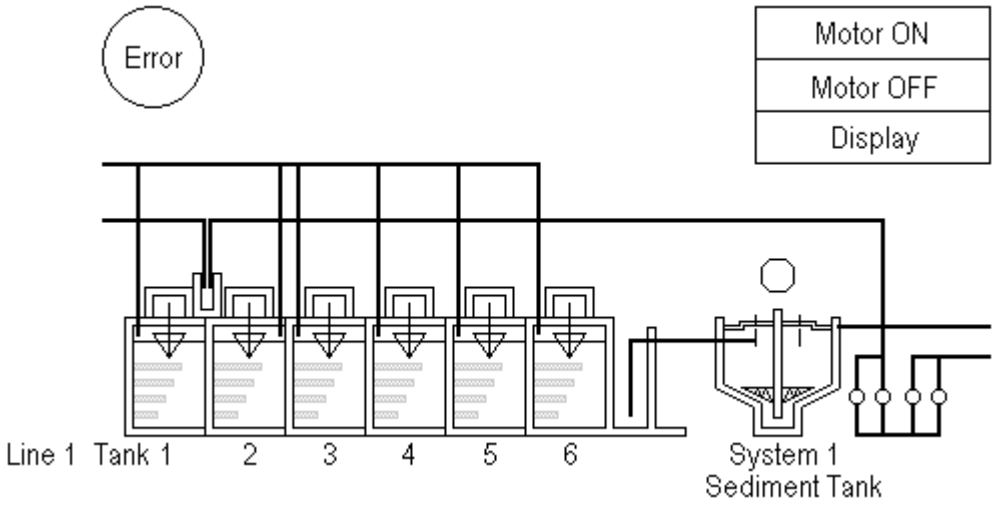
(4) After screen data is transferred to the GP, display (operation) can begin.

◆ **GP Run Screen**

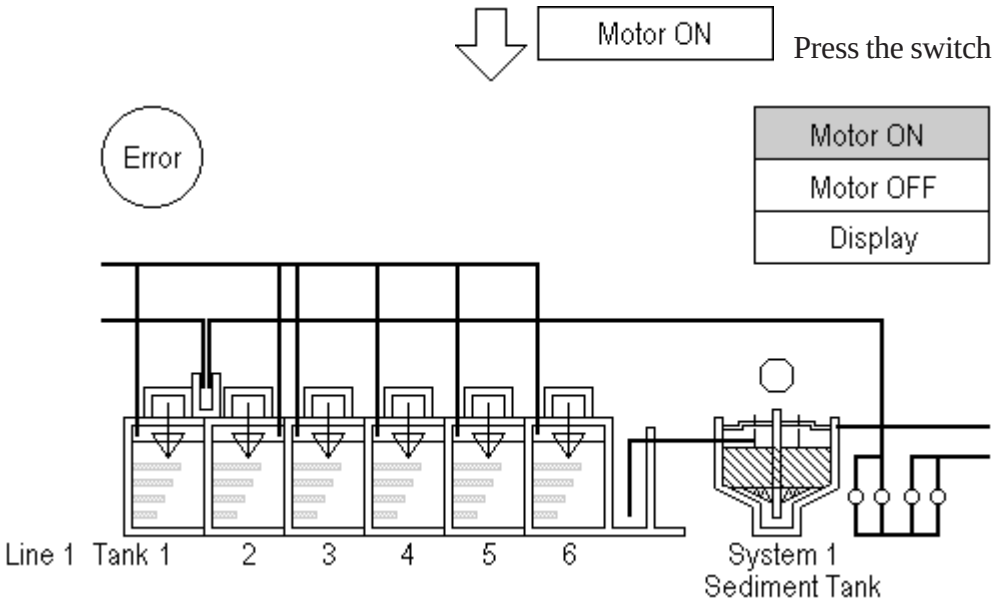
GP Screen (Before running program)



GP Screen (After running program)



Six Libraries  appear in the No. 90 screen display.



ASCII Code "31" = Data "1" is output to the Host, causing the screen to change.

5-2 Troubleshooting Multiple GP (Multi-drop) Communication

The host plays the following two roles when controlling multiple GP units:

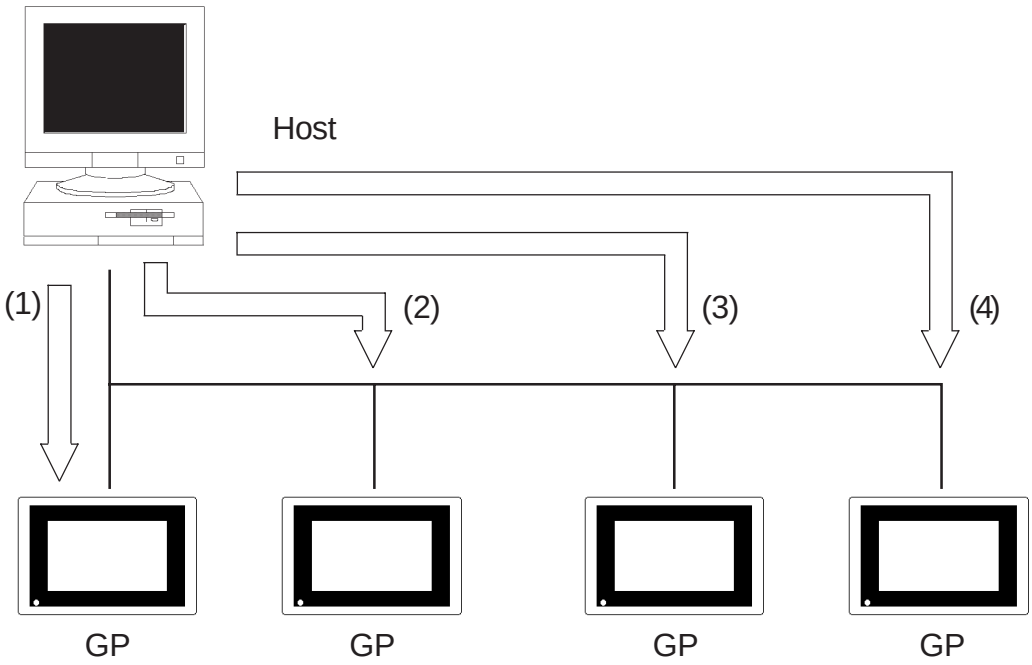
1. transferring data to be displayed
2. reading touch panel inputs from GP units through polling

Note that the more GP units to be controlled, and the more data to be transferred, the more burdened the host will be. In addition, an excessive number of GP units or an excessive amount of data can degrade the response speed of the GP units (slower display switching and slower response to touch panel inputs), substantially affecting the system operation.
Therefore, you should consider the number of GP units and amount of data when designing a multi-drop system.

■Sending Display Data to All GP units at the Same Time

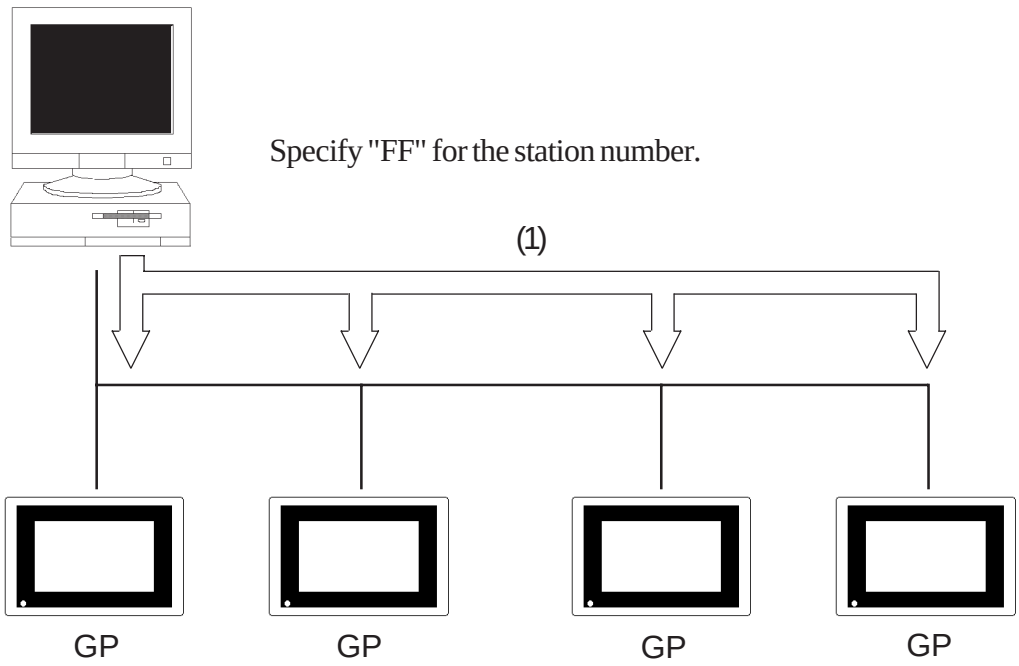
When you need to send the same data to all GP units, try sending it to all GP units at the same time for improved efficiency, instead of sending it to one GP unit at a time.
(This can be accomplished by specifying "FF" for the station number.)

◆When sending data to one GP at a time



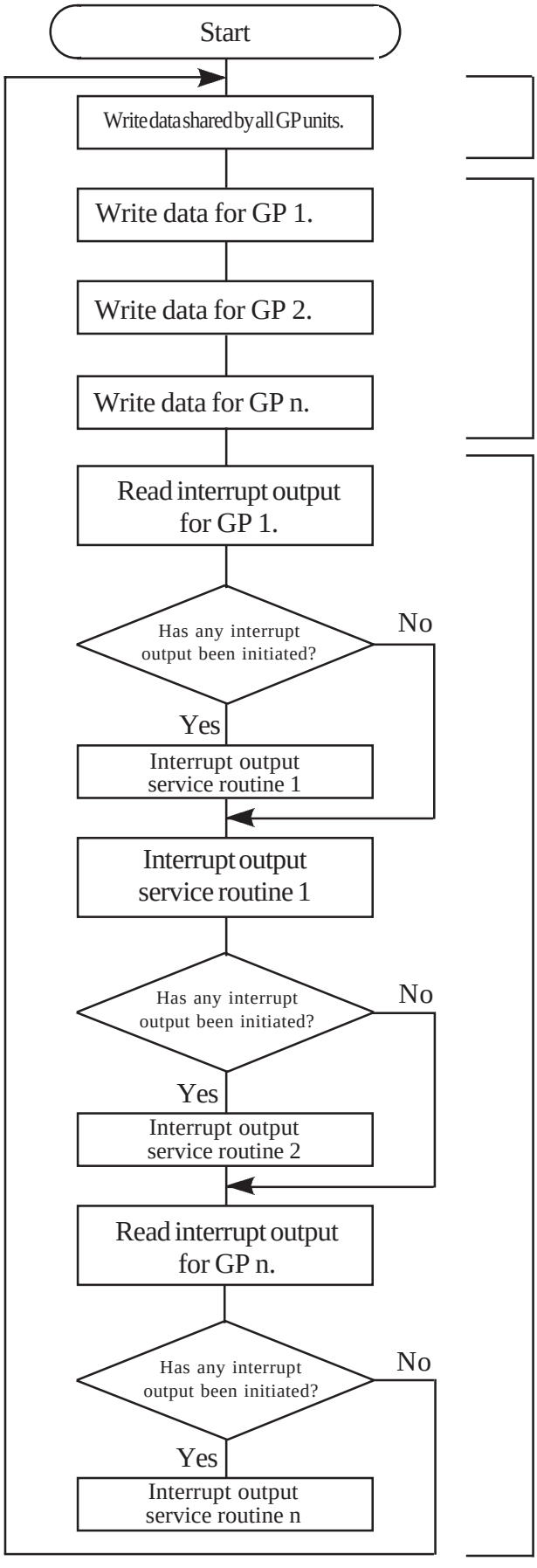
Excessive amount of time required (4 times longer than the case shown below)

◆When sending data to all GP units at the same time



Substantially reduced communication time (4 times shorter than the case shown above)

5-3 Program Flowchart for Multi-drop System



(1) Writing data shared by all GP units
Use the ESC W command to write display data shared by all GP units to the system area.
(At this time, specify "FF" for station number.)

(2) Writing data for a specific GP
Use the ESC W command to write data for a specific GP to the system area.

(3) Polling
Use the ESC I command to poll each GP unit to determine whether any touch panel input has been made. Touch panel inputs are serviced accordingly.

Note:

To improve GP response speed for touch panel input, insert a polling sequence after each write sequence (sequence in which data is written for a specific GP unit).
Make sure that the amount of data to be written to the system area is minimal.
In order to accomplish this, youcan, for example, choose to update only data items that have been changed.

MEMO

Chapter 6: Error Messages

This chapter describes error messages that may be displayed when the host and GPs communicate over the memory link. If an error message appears, take the proper action given in the table below. Then, reset and restart the GP.

6-1 Error Message List

An error message appears if communication parameters selected for the GP and the host do not match, or if the host sends invalid data to the GP.

In case of an error, "UPPER COMM. ERROR" appears first. Then, the following message appears:

UPPER COMM. ERROR (02:**)
**: ERROR CODE

Error Code	Description	Solution
06	Checksum does not match the data actually received.	- A SIO error may have occurred. Check for communication parameters and external problems, such as noise.
10	Undefined code has been recived.	-The host may be sending invalid data. -An SIO error may have occurred. Check for communication parameters and external problems, such as noise.
12	The specified number of data elements does not match the number of data elements received.	
15	The specified display attribute is out of the permissible range.	
16	The specified character size is out of the permissible range.	
17	The specified coordinate is out of the permissible range.	
18	The specified line type is out of the permissible range.	
19	The specified tiling pattern is out of the permissible range.	
1A	The specified radius is out of the permissible range.	
1B	The specified start/end angle is out of the permissible range.	
1C	The specified character type is out of the permissible range.	
1D	The specified rotation is out of the permissible range.	
1E	The specified direction is out of the permissible range.	
1F	The specified highlighting is out of the permissible range.	

Error Code	Description	Solution
20	The specified arrow pattern is out of the permissible range.	- The host may be sending invalid data. -An SIO error may have occurred. Check communication parameters and external problems, such as noise.
21	The specified arrow direction is out of the permissible range.	
22	The specified beveling method is out of the permissible range.	
23	The specified centering is out of the permissible range.	
FA	The specified address in the system area is out of the permissible range.	
FB	An attempt has been made to write to or read from outside the system area.	
FC	A data block of an improper format has been received by the GP.	
FF	The GP has been unable to send data for 10 seconds or more.	-The host may not be communicating properly with the GP. Check that the cable is properly connected.