

**CITIZEN**

# ***Command Reference***

---

**MOBILE LINE THERMAL PRINTER**  
**MODEL CMP-10**

Rev. 1.02 Newly issued on January 17, 2003

Japan CBM Corporation



# 1. CMP-10 COMMAND SET

	Code	Command Function	Page	
1	BEL	Sounds the Buzzer	4	*
2	HT	Horizontal Tab Command	5	
3	LF	Printing and Paper Feed Command	6	
4	CR	Print one line Command	7	
5	ESC RS	Sounds the Buzzer	7	
6	ESC SP	Setting the right space amount of the character	8	*
7	ESC \$	Specifying the Absolute Positions	9	
8	ESC %	Specifying/Canceling Download Character Set	10	
9	ESC &	Define user characters	11	
10	ESC !	Collective Specifying Printing Mode	12	
11	ESC *	Specifying the Bit Image Mode	14	*
12	ESC +	Switch OFF the printer	17	*
13	ESC -	Specifying/ Canceling Underline	17	
14	ESC .	Printer self test	18	*
15	ESC 2	Specifying 1/6-inch line feed rate	18	
16	ESC 3	Setting line feed rate of minimum pitch	19	
17	ESC =	Data Input Control	20	
18	ESC ?	Reading magnetic stripe reader	21	*
19	ESC @	Initializing the Printer	22	
20	ESC D	Setting Horizontal Tab Position	23	
21	ESC E	Specifying/canceling highlighting	24	
22	ESC G	Specifying/canceling Double Printing	25	
23	ESC J	Printing and feeding paper n/203 inch	26	
24	ESC R	Selecting Code table	26	
25	ESC S	Setting serial interface communication speed	27	*
26	ESC T	Printing Diagnostic information	27	*
27	ESC V	Specifying/Canceling 90°-right- turned Characters	28	
28	ESC Y	Specifying print density	29	*
29	ESC Z	Returning diagnostic information	30	*
30	ESC \	Specifying the relative positions	33	
31	ESC _	Setting the printer in default state	33	*
32	ESC '	Returning the battery voltage and Printer Head temperature	34	*
33	ESC a	Aligning the characters	35	
34	ESC c5	Enabling/Disabling Panel Switches	36	
35	ESC d	Printing and Feeding the paper by n lines	37	
36	ESC v	Transmitting the printer status	38	
37	ESC x	Selecting the time interval for automatically switching Off the printer	39	*
38	ESC {	Specifying/Canceling the Inverted Characters	40	
39	GS )	Setting of printer flags	41	*
40	GS *	Defining the Download Bit Image (LOGO)	42	*
41	GS /	Printing the Download, Bit Image	43	
42	GS :	Starting/Ending Macro Definition	44	
43	GS H	Selecting of Printing Position of HRI Code	45	

	<b>Code</b>	<b>Command Function</b>	<b>Page</b>
44	GS L	Setting the left margin	46
45	GS W	Selecting the horizontal size (Scale factor) of the Bar Code	47
46	GS ^	Executing the Macro	48
47	GS a	Enabling/Disabling ASB (Automatic Status Back)	49
48	GS f	Selecting the font of HRI characters	52
49	GS h	Selecting the height of the Bar Code	53
50	GS k	Printing the bar code	54
51	GS w	Setting the print area width	59

\* indicates a unique command or a non-compatible command.

## 2. COMMAND DETAILS

### 2.1 Description of Items

**XXXX**

**ALL**

---

<b>[Function]</b>	Command Function
<b>[Code]</b>	A sequence of code constituting a command is represented in hexadecimal number for < >H, binary number for < >B, and decimal number for < >, respectively; [ ]k represents a repeat count of k-times.
<b>[Range]</b>	Indicates an argument value (setting range) for the command in hexadecimal number (partly in decimal number). * When used for application other than defined in each control, an error may occur. Always use in the defined range.
<b>[Outline]</b>	Describes a command outline.
<b>[Caution]</b>	Describes a caution as required.
<b>[Default]</b>	Describes an initial value in hexadecimal number for the command when accompanied by an argument.
<b>[See Also]</b>	Describes the associated commands for use.
<b>[Sample Program]</b>	Describes a coding example in the Q-BASIC sample program. * This example is only for your reference and differs depending on the language used, version, and so on. For details, see the manual for the language used.

# BEL

---

<b>[Function]</b>	Sounds the Buzzer
<b>[Code]</b>	<07>H
<b>[Outline]</b>	By executing this command the buzzer will beep. This command functions in the same manner as ESC RS.

# HT

---

**[Function]** Horizontal Tab Command

**[Code]** <09>H

**[Outline]** Shifts the printing position to the next horizontal tab position.  
• Ignored when the next horizontal tab position has not been set.

**[Caution]**

- The horizontal tab position is set by ESC D.
- Initial setting of the horizontal tab position is each 8 characters in 9th, 17th, 25th, columns from FONT A.

**[See Also]** ESC D

**[Sample Program]**

```
PRINT #1, "0123456789012345678901";
PRINT #1, CHR$ (&HA);
PRINT #1, CHR$ (&H9) + "AAA";
PRINT #1, CHR$ (&H9) + "BBB";
PRINT #1, CHR$ (&HA);
PRINT #1, CHR$ (&H1B) + "D";
PRINT #1, CHR$ (3) + CHR$ (7) + CHR$ (14) + CHR$ (0);
PRINT #1, CHR$ (&H9) + "AAA";
PRINT #1, CHR$ (&H9) + "BBB";
PRINT #1, CHR$ (&H9) + "CCC" + CHR$ (&HA);
```

**[Print Results]**

```
123456789012345678901
    AAA  BBB      ←——  Initially set horizontal tab
AAA BBB  CCC      ←——  When set to the 4th, 8th, and 15th digits
```

# LF

---

**[Function]** Printing and Paper Feed Command

**[Code]** <0A>H

**[Outline]** Prints data inside the input buffer and feeds lines based on the line feed amount having been set.

- The head of the line becomes the next print starting position.

**[See Also]** ESC 2, ESC 3

**[Sample Program]**

```
PRINT #1, "AAA" + CHR$ (&HA);  
PRINT #1, "BBB" + CHR$ (&HA);  
PRINT #1, CHR$ (&HA);  
PRINT #1, "CCC" + CHR$ (&HA);
```

**[Print Results]**

```
AAA ←—— Print and line feed  
BBB ←—— Print and line feed  
      ←—— Line feed only  
CCC ←—— Print and line feed
```

## CR

---

<b>[Function]</b>	Print one line Command
<b>[Code ]</b>	<0D>H
<b>[Outline]</b>	This command is ignored or its action is the same as LF depending on the last execution of GS ) 2 command (FLAG SETTING COMMAND). If Flag 2 is 0, carriage return (CR) command is ignored. If Flag 2 is 1, CR comamnd will act as LF command.
<b>[Default]</b>	CR command is ignored. If you want tthat it acts like LF command set flag 2 to 1 by GS ) command.
<b>[See Also]</b>	GS )

## ESC RS

---

<b>[Function]</b>	Sounds the Buzzer
<b>[Code]</b>	<1B>H<1E>H
<b>[Outline]</b>	By executing this command the buzzer will beep. This command functions in the same manner as BEL.

## ESC SP n

---

<b>[Function]</b>	Setting the right space amount of the character
<b>[Code]</b>	<1B>H<20>H<n>
<b>[Range]</b>	{0 =< n =< 20(Hex)}
<b>[Outline]</b>	The rightward space amount is set in dot unit (1/203 inch unit).
<b>[Caution]</b>	The rightward space amount in doublewide mode is made double of the set volume.
<b>[Default]</b>	n = 0

### **[Sample Program]**

```
PRINT #1, CHR$ (&H1B) + “ “ + CHR$ (0);  
PRINT #1, “AAAAA” + CHR$ (&HA);  
PRINT #1, CHR$ (&H1B) + “ “ + CHR$ (1);  
PRINT #1, “AAAAA” + CHR$ (&HA);  
PRINT #1, CHR$ (&H1B) + “ “ + CHR$ (12);  
PRINT #1, “AAAAA” + CHR$ (&HA);
```

### **[Print Results]**

```
AAAAA           ←—— 0-dot space  
A A A A A       ←—— 1-dot space  
A A A A A A     ←—— 12-dot space
```

## ESC \$ n1 n2

---

**[Function]** Specifying the Absolute Positions

**[Code]** <1B>H<24>H<n1><n2>

**[Range]** {0 =< n1 =< FF(Hex)}  
{0 =< n2 =< 1(Hex)}

**[Outline]** The printing start position is specified in the number of dots (1/203 inch unit) from the beginning of line.

- The number of dots is divided by 256, whose quotient is taken as n2 and the residual as n1.
- Therefore, the printing start position is equal to  $n1 + n2 \times 256$  from the beginning of line.

**[Caution]** Specifying beyond the line end is ignored.

**[Default]** The initial value is not specified.

**[See Also]** ESC \

### **[Sample Program]**

```
PRINT #1, CHR$ (&H1B) + "$";  
PRINT #1, CHR$ (0) + CHR$ (0) + "A";  
PRINT #1, CHR$ (&H1B) + "$";  
PRINT #1, CHR$ (50) + CHR$ (0) + "B";  
PRINT #1, CHR$ (&H1B) + "$";  
PRINT #1, CHR$ (0) + CHR$ (1) + "C";  
PRINT #1, CHR$ (&HA);  
PRINT #1, CHR$ (&H1B) + "$";  
PRINT #1, CHR$ (100) + CHR$ (0) + "A";  
PRINT #1, CHR$ (&H1B) + "\";  
PRINT #1, CHR$ (&HC2) + CHR$ (&HFF) + "B";  
PRINT #1, CHR$ (&HA);
```

### **[Print Results]**

```
0    50    100    256  
↓    ↓    ↓    ↓  
A    B    ↓    C  
      B    A  
      ←→
```

## ESC % n

---

**[Function]** Specifying/Canceling Download Character Set

**[Code]** <1B>H<25>H<n>

**[Range]** {0 =< n =< FF(Hex)}

**[Outline]** Specifying/canceling download characters.  
Further, only the lowest bit (n0) is valid for n.

n (Hex)	Function
0	Selecting download character set
1	Canceling download character set

**[Caution]** This setting cannot be restored to initial value even with Initialize command or by turning power OFF. It can be restored to the initial value by ESC \_, a command for returning to initial state.

**[Default]** n = 0

**[See Also]** ESC &



## ESC ! n

---

**[Function]** Collective Specifying Printing Mode

**[Code]** <1B>H<21>H<n>

**[Range]** {0 =< n =< FF(Hex)}

**[Outline]** Printing mode is assigned.

Bit	Function	Value	
		0	1
0	Character Font	Font A	Font B
1	Undefined		
2	Undefined		
3	High-lighting	Canceled	Specified
4	Double height	Canceled	Specified
5	Double width	Canceled	Specified
6	Undefined		
7	Underline	Canceled	Specified

Each n bit indicates the following:

- [Caution]**
- With double height and double width being specified simultaneously, double wide and double high characters are consisted.
  - An underline is attached to the full character width, which, however, is not attached to the part having been skipped by the horizontal tab.  
Neither is it attached to 90°-right-turned characters.
  - The underline width is as having been specified by <ESC ->. (The default setting is 1 dot width.)
  - n case that double wide character and normal character exist in same one line, the layout of underline is consistent one.

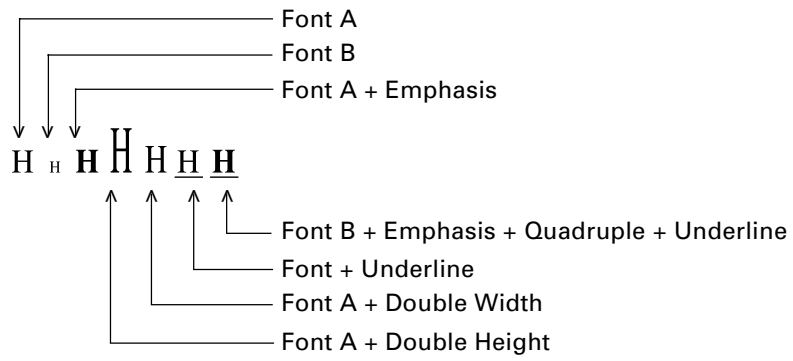
**[Default]** n = 0

**[See Also]** ESC E, ESC -

**[Sample Program]**

```
PRINT #1, CHR$(&H1B) + "!" + CHR$(&H00) + "H" ;  
PRINT #1, CHR$(&H1B) + "!" + CHR$(&H01) + "H" ;  
PRINT #1, CHR$(&H1B) + "!" + CHR$(&H08) + "H" ;  
PRINT #1, CHR$(&H1B) + "!" + CHR$(&H10) + "H" ;  
PRINT #1, CHR$(&H1B) + "!" + CHR$(&H20) + "H" ;  
PRINT #1, CHR$(&H1B) + "!" + CHR$(&HB9) + "H" ;  
PRINT #1, CHR$(&HA) ;  
END
```

**[Print Results]**



## ESC \* m n1 n2 [d] k

**[Function]** Specifying the Bit Image Mode

**[Code]** <1B>H<2A>H<m><n1><n2> [d] k

**[Range]** {m= 0, 1, 20, 21(Hex)}  
 {0 =< n1 =< FF(Hex)}  
 {0 =< n2 =< 1(Hex)}  
 {0 =< d =< FF(Hex)}  
 {k = n1 + FF(Hex) x n2} (m=0, 1(Hex)) Number of bytes containing image data  
 {k = (n1 + FF(Hex) x n2) x 3} (m=20, 21(Hex)) Number of bytes containing image data

**[Outline]** According to the number of dots specified in n1, n2, specify the bit image of mode m.

- The No. of dots printed is divided by 256, whose quotient is taken as n2 and residual as n1.
- Bit image data is sent starting from the top to bottom and from the left to right (vertical columns scanning). In modes m=0, and 1 only one byte per column is sent and in mode m=20h, 21h 3 bytes for each column are sent.
- d is bit image data, the bits subject to printing are taken as "1" and those not as "0".
- The bit image modes specified by m are shown as follows:

m (Hex)	Mode	Vertical Direction		Horizontal Direction	
		Dots	Dot Density	Dot Density	Max. Dots
0	8-dot single density	8	67 DPI	101 DPI	192
1	8-dot double density	8	67 DPI	203 DPI	384
20	24-dot single density	24	203 DPI	101 DPI	192
21	24-dot double density	24	203 DPI	203 DPI	384

**[Caution]**





- When the values set in m (bit image mode) are out of the above range, the data following after n1 is processed as normal printing data.
- After completion of bit image printing, printer returns to normal data processing mode.
- Line space is automatically set to 0.

### [Sample Program]

```
PRINT #1, CHR$(&H1B) + " * ";
PRINT #1, CHR$(0) + CHR$(20) + CHR$(0);
GOSUB IMG1
PRINT #1, CHR$(&HA);
PRINT #1, CHR$(&H1B + " * ";
PRINT #1, CHR$(1) + CHR$(20) + CHR$(0);
GOSUB IMG1
PRINT #1, CHR$(&HA);
PRINT #1, CHR$(&H1B) + " * ";
PRINT #1, CHR$(32) + CHR$(20) + CHR$(0);
GOSUB IMG2
PRINT #1, CHR$(&HA);
PRINT #1, CHR$(&H1B) + " * ";
PRINT #1, CHR$(33) + CHR$(20) + CHR$(0);
GOSUB IMG2
PRINT #1, CHR$(&HA);
END
```

```
IMG1:
PRINT #1, CHR$(&HFF);
FOR I=1 TO 18
  PRINT #1, CHR$(&H85);
NEXT I
PRINT #1, CHR$(&HFF);
RETURN
IMG2:
PRINT #1, CHR$(&HFF);
PRINT #1, CHR$(&HFF);
PRINT #1, CHR$(&HFF);
FOR I=1 TO 18
  PRINT #1, CHR$(&H80);
  PRINT #1, CHR$(&H00);
  PRINT #1, CHR$(&H05);
NEXT I
PRINT #1, CHR$(&HFF);
PRINT #1, CHR$(&HFF);
PRINT #1, CHR$(&HFF);
RETURN
```

### [Print Results]

	←	8-dot single density
	←	8-dot double density
	←	24-dot single density
	←	24-dot double density

**This command has one more version.**

**[Function]** Specifying the Bit Image Mode (PCX format)

**[Code]** <1B>H<2A>H<m><n>{<a><00>H} [d]k

**[Range]** {m= 10, 11 or 12(Hex)}  
{0 =< n =< 30(Hex)}  
{0 =< a =< 18(Hex)}  
{0 =< d =< FF(Hex)}  
{0 =< k =< n x 18(Hex)}

**[Outline]** Designates a bit image of n\*8 dots horizontal by 24 or a dots vertical.  
Dot density is fixed at 203 dpi both horizontally and vertically.  
Bit image mode selected by the value of m is shown in the following table.

<b>m (Hex)</b>	<b>Type</b>
10	Non-compressed data of 24 dots vertical (Argument "a" and 00(Hex) are not used.)
11	Compressed data of 24 dots vertical (Argument "a" and 00(Hex) are not used.)
12	Compressed data of a dots vertical

n denotes the number of bytes for horizontal size of a bit image.

d denotes a graphic data.

In 10(Hex) mode, data is in n x 24 bytes.

For data in 11(Hex) mode, the size in non-compressed format shall be n x 24 bytes.

For data in 12(Hex) mode, the size in non-compressed format shall be n x a bytes.

Compression system in 11(Hex) and 12(Hex) is the same as that for PCX file format (monochrome). The printer receives the compressed data, reproduces it in the following format and prints as bit-image data.

A If either of upper two bit of input byte is 0, the byte is treated as graphic data.

If both of upper two bit of graphic data to be printed are 0, it is necessary to send the data to printer in "B" format described below.

B If both of upper two bit of input byte are 0, the remaining 6 bit specifies the times the data is repeated. The number of times to be repeated is specified in 0-3F(Hex). The byte follows this byte is treated as graphic data to be repeated.

Graphics data should line up from left to right and from top to bottom.

## ESC +

---

<b>[Function]</b>	Switch OFF the printer
<b>[Code]</b>	<1B>H<2B>H
<b>[Range]</b>	None
<b>[Outline]</b>	This command is used for switching off the printer from the host. Sending this command causes the same operation as that by setting the power switch to off.

## ESC – n

---

<b>[Function]</b>	Specifying/ Canceling Underline
<b>[Code]</b>	<1B>H<2D>H<n>
<b>[Range]</b>	{0 =< n =< 02(Hex)}
<b>[Outline]</b>	Specifying/canceling an underline. • Types of underlines by n value are shown below:

n (Hex)	Type
0	Canceling an underline.
1	Specifying an underline for 1-dot width.
2	Specifying an underline for 2-dots width.

<b>[Caution]</b>	<ul style="list-style-type: none"><li>• An underline is attached to the full character width. It is, however, not attached to the part having been skipped by horizontal tab command.</li><li>• An underline is not attached to a 90°- right-turned characters.</li></ul>
<b>[Default]</b>	n = 0
<b>[See Also]</b>	ESC !

### [Sample Program]

```
PRINT #1, CHR$ (&H1B) + “-” + CHR$ (0);  
PRINT #1, “AAAAA” ;  
PRINT #1, CHR$ (&H1B) + “-” + CHR$ (1);  
PRINT #1, “AAAAA” + CHR$ (&HA);
```

### [Print Results]

```
Underline Canceled  
←────────────────→  
A A A A A A A A A A  
←────────────────→  
Underline Specified
```

## ESC .

---

<b>[Function]</b>	Printer self test
<b>[Code]</b>	<1B>H<2E>H
<b>[Range]</b>	None
<b>[Outline]</b>	Prints test page, downloaded bit image, and self-diagnostic information. The self-diagnostic information includes print density, print head temperature, battery voltage, Serial or IrDA interface, baud rate, and memory switch setting.
<b>[See Also]</b>	ESC T

## ESC 2

---

<b>[Function]</b>	Specifying 1/6-inch line feed rate
<b>[Code]</b>	<1B>H<32>H
<b>[Outline]</b>	The line feed rate per line is specified by 1/6 inch.

### [Sample Program]

```
PRINT #1, "AAAAA" + CHR$ (&HA);  
PRINT #1, CHR$ (&H1B) + "3" + CHR$ (0);  
PRINT #1, "AAAAA" + CHR$ (&HA);  
PRINT #1, CHR$ (&H1B) + "3" + CHR$ (50);  
PRINT #1, "AAAAA" + CHR$ (&HA);  
PRINT #1, CHR$ (&H1B) + "2";  
PRINT #1, "AAAAA" + CHR$ (&HA);  
PRINT #1, "AAAAA";  
PRINT #1, CHR$ (&H1B) + "J" + CHR$ (100);  
PRINT #1, "AAAAA" + CHR$ (&HA);  
PRINT #1, "AAAAA" + CHR$ (&HA);
```

### [Print Results]

A A A A A  
↑ 1/6-inch line feed  
A A A A A  
↓ 0/203-inch line feed  
A A A A A  
↑ 50/203-inch line feed  
A A A A A  
↑ 1/6-inch line feed  
A A A A A  
↑ 100/203-inch line feed  
A A A A A  
↑ 1/6-inch line feed  
A A A A A

## ESC 3 n

---

<b>[Function]</b>	Setting line feed rate of minimum pitch
<b>[Code]</b>	<1B>H<33>H<n>
<b>[Range]</b>	{0 =< n =< FF(Hex)}
<b>[Outline]</b>	The line feed rate per line is specified by n/203 inch.
<b>[Default]</b>	n = 22(Hex) 1/6-inch
<b>[Sample Program]</b>	See Sample Program and Print Results for ESC 2.

# ESC = n

**[Function]** Data Input Control

**[Code]** <1B>H<3D>H<n>

**[Range]** {0 =< n =< FF(Hex)}

**[Outline]** Selecting equipment in which data input from the host is effective.

- Each bit of n indicates as follows:

Bit	Equipment	Value	
		0	1
0	Printer	Invalid	Valid
1	Not defined		
2	Not defined		
3	Not defined		
4	Not defined		
5	Not defined		
6	Not defined		
7	Not defined		

- When the printer has not been selected, this printer abandons all the received data until it is selected by this command.

**[Caution]**

- Even when the printer has not been selected, it can become BUSY state through printer operation.
- When the printer is deselected, this printer discards all the data until it is selected with this command.

**[Default]** The initial value of n is "1".

**[Sample Program]**

```
PRINT #1, "AAAAA";  
PRINT #1, CHR$ (&H1B) + "=" + CHR$ (0);  
PRINT #1, "aaaaa" + CHR$ (&HA);  
PRINT #1, CHR$ (&H1B) + "=" + CHR$ (1);  
PRINT #1, "AAAAA" + CHR$ (&HA);
```

**[Print Results]**

```
A A A A A A A A A  
      ↑  
a a a a a
```

## ESC ? n

---

**[Function]** Reading magnetic stripe reader

**[Code]** <1B>H<3F>H<n>

**[Range]** {0 =< n =< 7(Hex)}

**[Outline]** When the command is received the bicolor LED is shining in RED and the printer is waiting for the magnetic card to be swiped through the reader. If even after 10 seconds the card is not swiped the command is aborted automatically. Printer returns the information read from the tracks followed by 00(Hex). Reads the data of the track in the following table by the value of n.

n (Hex)	Track
0	Invalid
1	Track 1
2	Track 2
3	Track 1 and 2
4	Track 3
5	Invalid
6	Track 2 and 3
7	Invalid

The format of the data read is as follows.

Track	Header	Card data	Footer
-------	--------	-----------	--------

Track	Track 1 is F1 (Hex) Track 2 is F2 (Hex) Track 3 is F3 (Hex)
Header	Track 1: "%" Track 2: ";" Track 3: "+" or ";"
Card data	ASCII data
Footer	"?"

If reading track is more than one (n=3 or 6), footer data of track 1 or 2 is followed by track data of track 2 or 3.

**[Caution]** The command for reading multiple tracks at a time may often result in read error. If reading is unsuccessful, repeat it.

## ESC @

---

<b>[Function]</b>	Initializing the Printer
<b>[Code]</b>	<1B>H<40>H
<b>[Outline]</b>	<p>Clears data stored in the print buffer and brings various settings to the initial state (Default state). Items not cleared are as shown below.</p> <ul style="list-style-type: none"><li>• Data inside the internal input buffer are not cleared.</li><li>• Does not clear any data in the internal input buffer.</li><li>• Does not clear downloaded characters.</li><li>• Does not clear defined downloaded characters.</li><li>• Does not clear defined downloaded bit image.</li><li>• Does not clear defined macro.</li><li>• Does not clear setting for print density.</li><li>• Does not clear baud rate setting of serial port.</li><li>• Does not clear valid/invalid setting for panel switch.</li><li>• Does not clear ASB mode setting.</li><li>• Does not clear setting for auto power-off time.</li><li>• Does not clear memory switch.</li></ul>

<b>[Caution]</b>	<p>At power off, the following settings are cleared in addition to the items cleared by this command.</p> <ul style="list-style-type: none"><li>• Clears input buffer.</li><li>• Clears macro.</li><li>• Return ASB mode according to the memory switch setting.</li><li>• Clears valid/invalid setting for panel switch.</li></ul>
------------------	---

### [Sample Program]

```
PRINT #1, CHR$ (&H1B) + “ ! “ + CHR$ (&H30) ;  
PRINT #1, CHR$ (&H1B) + “V” + CHR$ (1);  
PRINT #1, “AAA” + CHR$ (&HA);  
PRINT #1, CHR$ (&H1B) + “@”;  
PRINT #1, “AAA” + CHR$ (&HA);
```

### [Print Results]

```
! V @  
A A A
```

## ESC D [ n ] k NUL

---

<b>[Function]</b>	Setting Horizontal Tab Position
<b>[Code]</b>	<1B>H<44>H [ <n> ] k<00>H
<b>[Range]</b>	{0 =< n =< FF(Hex)} {0 =< k =< 20(Hex)}
<b>[Outline]</b>	<p>Specifying a horizontal tab position.</p> <ul style="list-style-type: none"><li>• “n” indicates the no. of columns from the beginning to the horizontal tab position. At this time, n= set position – 1 is to be specified. For example, to set the position at 9th column, n=8 is to be specified.</li><li>• k denotes the number of horizontal tab positions you want to set.</li><li>• The tab position is set at position where it is “character width × n” from the line beginning. The character width, at this time, includes the rightward space amount. In double wide characters, it is made double of the ordinary case.</li><li>• Tab positions can be specified are maximum 32. Specifying exceeding this is ignored. &lt;n&gt; k, which denotes a setting position, is input in the increasing order and ends at &lt;00&gt; H.</li><li>• ESC D NUL clears all the set tab positions. Following clearing, horizontal tab command is ignored.</li></ul>
<b>[Caution]</b>	When the data, <n> k, is equal to or smaller than its preceding data, <n> k-1, it is assumed that tab setting is finished. If this is the case, the next data onward will be processed as normal data. When the data, <n> k, exceeds a 1-line print area, set the horizontal tab position, assuming “Set digit position = Maximum print digits + 1.” The horizontal tab position does not change even if the character width is altered after setting the horizontal tab position.
<b>[Default]</b>	Initial value is specified every eight characters (9th, 17th, 25th, and on) of Font A.
<b>[See Also]</b>	HT
<b>[Sample Program]</b>	
<b>[Print Results]</b>	See Sample Program and Print Results for HT.

## ESC E n

---

**[Function]** Specifying/canceling highlighting

**[Code]** <1B>H<45>H<n>

**[Range]** {0 =< n =< FF(Hex)}

**[Outline]** Specifying/canceling the highlighting characters.

- “n” is valid only for the lowest bit (n0).

n (Hex)	Type
0	Specifying highlighting.
1	Canceling highlighting.

- This is effective only for font A (12 × 24 dots)
- Dot configuration of a highlighted character includes one extra dot added at its side.

**[Caution]** The double printing and highlighted character printing commands provide exactly the same operation.

**[Default]** n = 0

**[See Also]** ESC !

**[Sample Program]**

```
PRINT #1, CHR$ (&H1B) + “E” + CHR$ (0);  
PRINT #1, “AAABBB” + CHR$ (&HA);  
PRINT #1, CHR$ (&H1B) + “E” + CHR$ (1);  
PRINT #1, “AAABBB” + CHR$ (&HA);
```

**[Print Results]**

A A A B B B ← Highlighting canceled

**A A A B B B** ← Highlighting canceled

# ESC G n

---

**[Function]** Specifying/canceling Double Printing

**[Code]** <1B>H<47>H<n>

**[Range]** {0 =< n =< FF(Hex)}

**[Outline]** Specifying/canceling the double printing.

- “n” is valid only for the lowest bit (n0).
- Control by n is shown as follows.

n (Hex)	Type
0	Canceling double printing.
1	Specifying double printing.

- This is effective to font A (12 × 24 dots)

**[Caution]** • The print result of Double printing and highlight character printing is completely same.

**[Default]** n = 0

**[See Also]** ESC E

**[Sample Program]**

```
PRINT #1, CHR$ (&H1B) + “G” + CHR$ (0);  
PRINT #1, “AAABBB” + CHR$ (&HA);  
PRINT #1, CHR$ (&H1B) + “G” + CHR$ (1);  
PRINT #1, “AAABBB” + CHR$ (&HA);
```

**[Print Results]**

A A A B B B ← Highlighting canceled

**A A A B B B** ← Highlighting canceled

## ESC J n

---

<b>[Function]</b>	Printing and feeding paper n/203 inch
<b>[Code]</b>	<1B>H<4A>H<n>
<b>[Range]</b>	{0 =< n =< FF(Hex)}
<b>[Outline]</b>	Prints data in the print buffer and feeds paper by n/203 inch. <ul style="list-style-type: none"><li>• This function is temporary and does not affect the feed operation thereafter.</li><li>• The beginning of the line is to be considered as the next printing start position.</li></ul>
<b>[Default]</b>	Initial value is not defined.
<b>[Sample Program]</b>	See Sample Program and Print Results for ESC 2 on Page 18.

## ESC R n

---

<b>[Function]</b>	Selecting Code table
<b>[Code]</b>	<1B>H<52>H<n>
<b>[Range]</b>	{0 =< n =< D(Hex)}

n (Hex)	Character Set
0	U.S.A.
1	France
2	Germany
3	U.K.
4	Denmark I
5	Sweden
6	Italy
7	Spain I
8	Japan
9	Norway
A	Denmark II
B	Spain II
C	Latin America
D	Korea

<b>[Outline]</b>	Selects the necessary code table
<b>[Default]</b>	n = 0
<b>[See Also]</b>	Character Code Table (International Character Set)

## ESC S n

---

**[Function]** Setting serial interface communication speed

**[Code]** <1B>H<53>H<n>

**[Range]** {0 =< n =< 6(Hex)}  
{30 =< n =< 36(Hex)}

n (Hex)	Speed (bps)
0, 30	1200
1, 31	2400
2, 32	48000
3, 33	9600
4, 34	19200
5, 35	57600
6, 36	115200

**[Outline]** Sets the communication speed for the serial interface.  
The command is valid only when the printer is connected through a serial cable. It is not valid when using IrDA interface. The last setting is valid even after the printer is switched OFF.

**[Default]** The default value is 3 [9600 BPS]

## ESC T

---

**[Function]** Printing Diagnostic information

**[Code]** <1B>H<54>H

**[Outline]** Prints current printer attributes and diagnostics.

**[See Also]** ESC .

**[Print Results]** **CITIZEN Mobile Printer**  
////////////////////////////////////  
**Model CMP-10 Ver x.xxXX**  
Intensity: **100%**  
Timeout: **10 min**  
Temperature: **27°C**  
Battery: **7.5V [\*\*...]**  
Mode: **RS232**  
Speed: **9600 bps**  
Switches: **0 0 0 0 0 0 0**

# ESC V n

---

**[Function]** Specifying/Canceling 90°-right- turned Characters

**[Code]** <1B>H<56>H<n>

**[Range]** {0 =< n =< 1(Hex)}

**[Outline]** Specifying/canceling characters 90°-right- turned character.  
“n” means the followings.

n (Hex)	Condition
0	Canceling 90°-right- turned Characters
1	Specifying 90°-right- turned Characters

**[Caution]** No underlines are attached to 90°-right- turned characters.

**[Default]** The initial value of n is “0”.

**[Sample Program]**

```
PRINT #1, CHR$ (&H1B) + “V” + CHR$ (0);  
PRINT #1, “AAAAA”;  
PRINT #1, CHR$ (&H1B) + “V” + CHR$ (1);  
PRINT #1, “AAAAA” + CHR$ (&HA);
```

**[Print Results]**

```
90° Rotation Canceled  
←→  
A A A A A > > > > >  
←→  
90° Rotation Specified
```

## ESC Y n

---

**[Function]** Specifying print density

**[Code]** <1B>H<59>H<n>

**[Range]** {0 =< n =< 5(Hex)}

**[Outline]** Specifies the print density.  
“n” means the followings.

n (Hex)	Condition
0	70 % density
1	80 % density
2	90 % density
3	100 % density
4	120 % density
5	150 % density

**[Caution]** Higher density may lead to slower printing.  
Increasing the density allows Low battery to be detected easily.

**[Default]** n = 3 (100%)

**[Sample Program]**

```
PRINT #1, CHR$ (&H1B) + “Y” + CHR$ (0);  
PRINT #1, “AAAAA”;  
PRINT #1, CHR$ (&H1B) + “Y” + CHR$ (5);  
PRINT #1, “AAAAA” + CHR$ (&HA);
```

# ESC Z

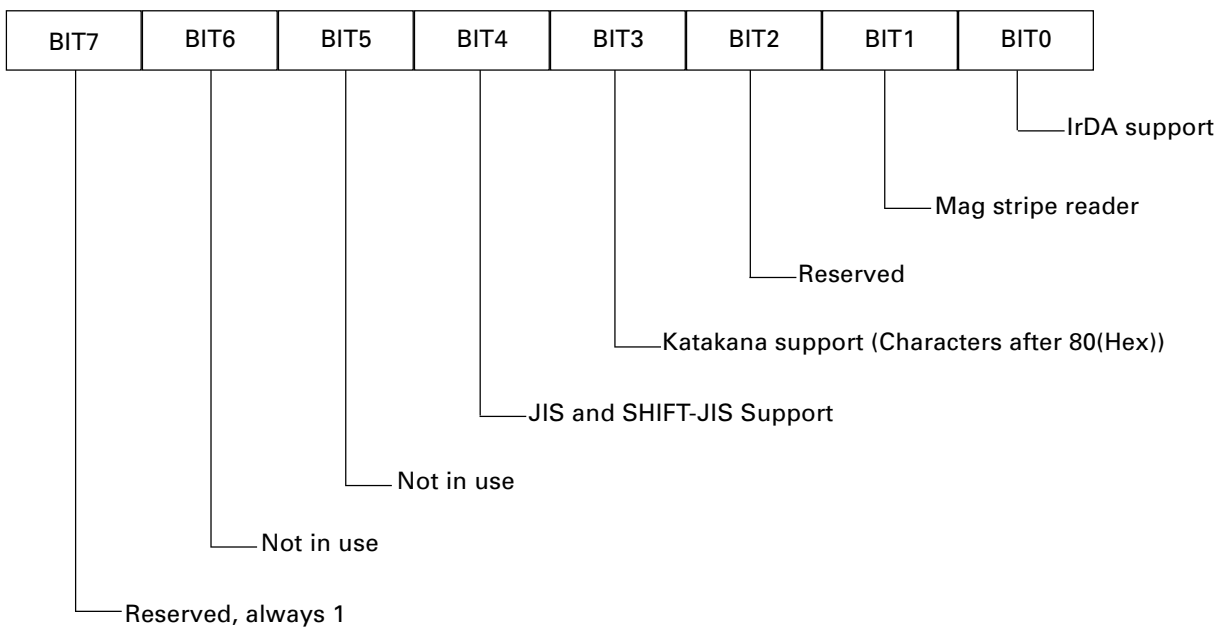
---

**[Function]** Returning diagnostic information

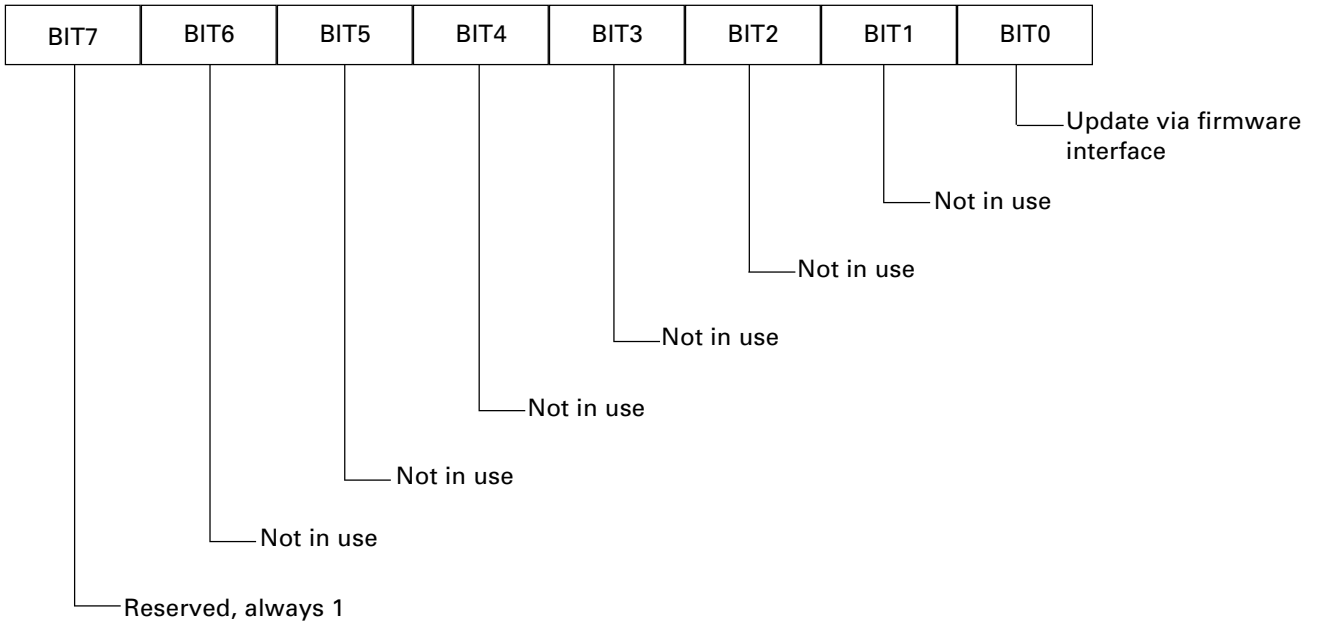
**[Code]** <1B>H<5A>H

**[Outline]** The printer will return 32 bytes of information with the following structure:  
1-22: Printer name up to 22 characters.  
23-25: Firmware version  
26-27: Language version, described by two characters.  
28-32: When the corresponding bit is 1, the function is supported and when 0, the function is not supported.

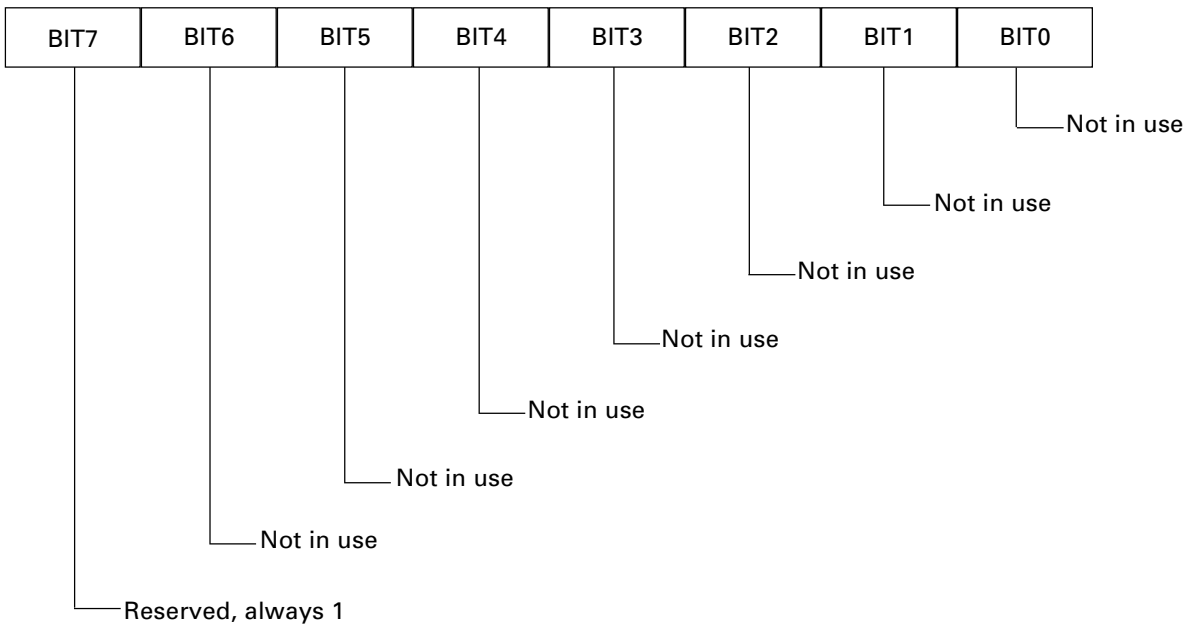
## BYTE 28



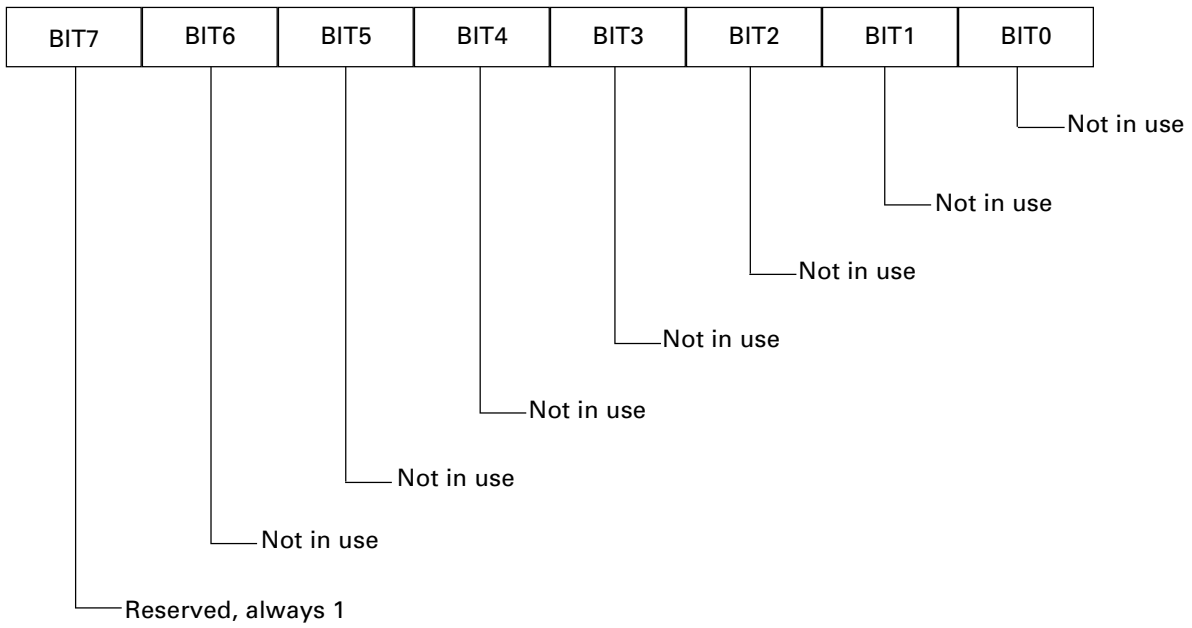
BYTE 29



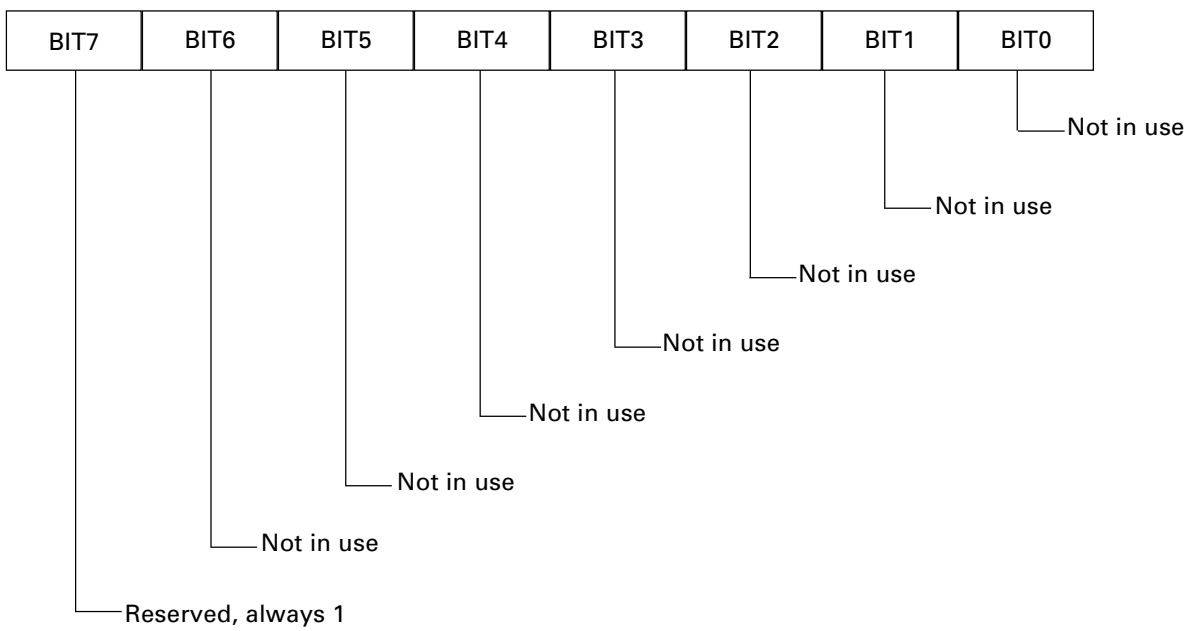
BYTE 30



BYTE 31



BYTE 32



## ESC \ n1 n2

---

<b>[Function]</b>	Specifying the relative positions
<b>[Code]</b>	<1B>H<5C>H<n1><n2>
<b>[Range]</b>	{0 =< n1 =< 255} {0 =< n2 =< 255}
<b>[Outline]</b>	<p>The printing start position is specified in the number of dots (1/203 inch unit) from the current position.</p> <ul style="list-style-type: none"><li>• Rightward direction is taken as plus and leftward direction as minus.</li><li>• To specify N dot in minus (left) direction, use a complement of N for assignment.</li><li>• N dots = 65536 – N</li><li>• The number of dots is divided by 256, whose quotient is taken as n2 and the residual as n1.</li></ul>
<b>[Caution]</b>	Specifying exceeding the top of line or the end of line is ignored.
<b>[See Also]</b>	ESC \$
<b>[Sample Program]</b>	See Sample Program and Print Results for ESC \$.

## ESC \_

---

<b>[Function]</b>	Setting the printer in default state
<b>[Code]</b>	<1B>H<5F>H
<b>[Outline]</b>	<p>Sets the printer in default state. The default state is as follows:</p> <ul style="list-style-type: none"><li>• All printing attributes like underline, rotating etc. are cleared.</li><li>• Internal font A (12 × 24) is selected.</li><li>• Pitch between lines is 1/6 inch.</li><li>• Pitch between characters is 0.</li><li>• Barcode height is 162 dots, and barcode width is 3.</li><li>• All downloaded fonts and bit images are cleared.</li><li>• Print density is 100%.</li><li>• Communication speed is set to 9600 bps.</li><li>• Auto power-off time is set at 10 min.</li><li>• Memory switch is set to “all 0”.</li></ul>
<b>[See Also]</b>	ESC @

## ESC ' ---

<b>[Function]</b>	Returning the battery voltage and Printer Head temperature
<b>[Code]</b>	<1B>H<60>H
<b>[Outline]</b>	<p>When this command is send to CMP-10, the printer will return the current value of battery voltage and the current temperature of the printer thermal head.</p> <ul style="list-style-type: none"><li>• Battery voltage is returned in the format: battery voltage <math>\times 10 + 20</math>H and head temperature is returned in the format: head temperature + 20H.</li></ul>
<b>[Sample Program]</b>	<pre>PRINT #1, CHR\$ (&amp;H1B) + " ' ";</pre>
<b>[Print Results]</b>	<p>Suppose the battery voltage is 7.8 volts and head temperature is 40°C. Then CMP-10 will return &lt;6C&gt;H, &lt;48&gt;H to the host. This values are calculated as follows: Volatge: <math>7.8 \times 10 = 78 + 32 = 110</math> which is 6CH. Temperaure: <math>40 + 32 = 72</math> which is 48H. 32 is equal to 20H.</p>

# ESC a n

---

**[Function]** Aligning the characters

**[Code]** <1B>H<61>H<n>

**[Range]** {0 =< n =< 2(Hex)}

**[Outline]** All the printed data within one line are aligned in the specified position. Depending on n value, positional alignment is carried out as in the table below:

n (Hex)	Position
0	Left end alignment
1	Centering
2	Right end alignment

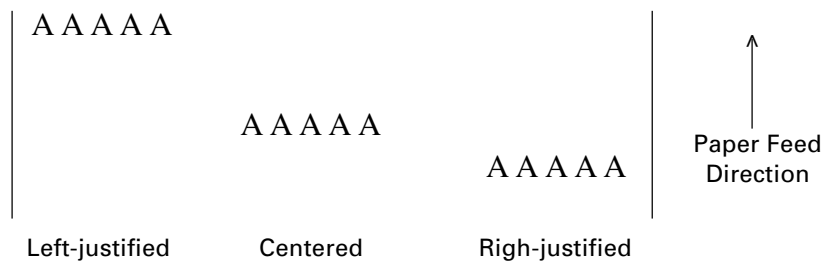
**[Caution]**

- This is valid only when n is inputted at the beginning of line.
- The initial value of n is "0".

**[Sample Program]**

```
PRINT #1, CHR$ (&H1B) + "a" + CHR$ (0);  
PRINT #1, "AAAAA" + CHR$ (&HA);  
PRINT #1, CHR$ (&H1B) + "a" + CHR$ (1);  
PRINT #1, "AAAAA" + CHR$ (&HA);  
PRINT #1, CHR$ (&H1B) + "a" + CHR$ (2);  
PRINT #1, "AAAAA" + CHR$ (&HA);
```

**[Print Results]**



## ESC c5 n

---

**[Function]** Enabling/Disabling Panel Switches

**[Code]** <1B>H<63>H<35>H<n>

**[Range]** {0 =< n =< FF(Hex)}

**[Outline]** Selecting the LF switch valid/invalid.

- “n” is valid only in the lowest bit (n0).

n (Hex)	Condition
0	LFSW valid.
1	LFSW invalid.

**[Caution]**

- When the panel switch is disabled with this command, the LF switch is disabled. Therefore, the paper cannot be fed by operating the LF switch.
- When LF switch is set to Invalid, it cannot be cleared by the Initialize command (ESC @) but is cleared by turning power off.

**[Default]** The initial value of n is “0”.

**[Sample Program]**

```
PRINT #1, CHR$ (&H1B) + “c5” + CHR$ (0); .....When enabling the LF switch  
PRINT #1, CHR$ (&H1B) + “c5” + CHR$ (1); .....When disabling the LF switch
```

# ESC d n

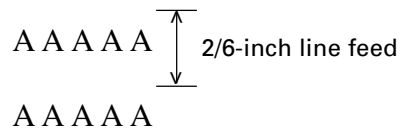
---

<b>[Function]</b>	Printing and Feeding the paper by n lines
<b>[Code]</b>	<1B>H<64>H<n>
<b>[Range]</b>	{0 =< n =< FF(Hex)}
<b>[Outline]</b>	Prints data inside the buffer and feeds paper by n lines. Specified line does not remain. The beginning of the line is to be considered as the next printing start position.
<b>[Default]</b>	The initial value is not defined.

**[Sample Program]**

```
PRINT #1, "AAAAA"  
PRINT #1, CHR$ (&H1B) + "d" + CHR$ (2);  
PRINT #1, "AAAAA" + CHR$ (&HA);
```

**[Print Results]**



## ESC v

---

**[Function]** Transmitting the printer status

**[Code]** <1B>H<76>H

**[Outline]** Current printer status is transmitted.  
Status sent out consists of 1 byte whose content is as in the table below.

Bit	Function	Value	
		0	1
0	Not defined		
1	Not defined		
2	Paper or paper cover	With paper and cover closed	No paper or cover open
3	Not defined		
4	Not used	Fixed to 0	–
5	Not defined		
6	Not defined		
7	Not defined		

**[Caution]** In paper end status, this command may be unreceptible state due to BUSY state.

**[Sample Program]**

```
OPEN "COM1:N81NN" AS #1 ;  
PRINT #1, CHR$ (&H1B) ; "v" ;  
A$ = INPUT$ (1, #1) ;  
CLOSE #1  
END
```

## ESC x n

---

<b>[Function]</b>	Selecting the time interval for automatically switching Off the printer.
<b>[Code]</b>	<1B>H<78>H<n>
<b>[Range]</b>	{1 =< n =< 3C(Hex)}
<b>[Outline]</b>	Sets the time interval after which the printer will be switched Off automatically if there is no incoming data through the Serial interface, there is no IrDA communication and LF button is not pressed. The maximum value for the interval is 60 minutes <3C>H .
<b>[Default]</b>	n = a(Hex) 10 minutes
<b>[Sample Program]</b>	<pre>PRINT #1, CHR\$ (&amp;H1B) + "x" + CHR\$ (H14);</pre> <p>After executing this command the printer will perform auto power off after 20 minutes, if there is no IrDA communication and LF button is not pressed.</p>

# ESC { n

**[Function]** Specifying/Canceling the Inverted Characters

**[Code]** <1B>H<7B>H<n>

**[Range]** {0 =< n =< FF(Hex)}

**[Outline]** Specifying/canceling inverted characters.

- "n" is valid only for the lowest bit (n0).

n (Hex)	Condition
0	Canceling inverted characters.
1	Specifying inverted characters.

**[Caution]**

- Inverted-printing means printing the line at 180° turned.
- This is valid only when this is specified at the beginning of a line.

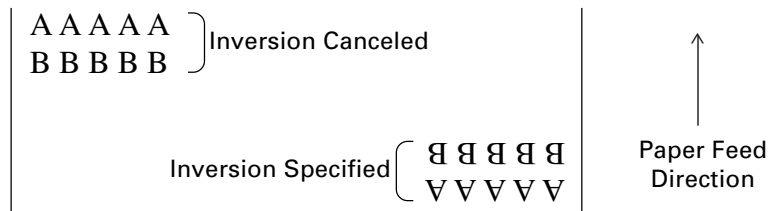
**[Default]**

- The initial value of n is "0".

**[Sample Program]**

```
PRINT #1, CHR$ (&H1B) + "{" + CHR$ (0);
PRINT #1, "AAAAA" + CHR$ (&HA);
PRINT #1, "BBBBB" + CHR$ (&HA);
PRINT #1, CHR$ (&H1B) + "{" + CHR$ (1);
PRINT #1, "AAAAA" + CHR$ (&HA);
PRINT #1, "BBBBB" + CHR$ (&HA);
```

**[Print Results]**



## GS ) n m

---

**[Function]** Setting of printer flags

**[Code]** <1D>H<29>H<n><m>

**[Range]** {0 =< n =< 8(Hex)}  
{m = 0 or 1(Hex)}

**[Outline]** This model has 8 memory switches and selecting, releasing, and changing a function is available with this command.

Memory switch setting is retained even after printer power off.

n specifies the number of memory switch to be operated.

m specifies a selection meaning as shown below.

0: when the value is 0, this function is invalid.

1: when the value is 1, this function is valid.

The contents of individual memory switches are as shown below.

Memory Switch	OFF (0)	ON (1)
1	Slach 0	Normal 0
2	CR is ignored	CR has the same action as LF
3	Default font A	Default font B
4	Prints "Low Battery".	Does not print "Low Battery".
5	ASB Invalid for default	ASB Valid for default
6	Not in use	
7	Not in use	
8	Shift JIS for default	JIS for default

**[Caution]** The setting of the changed memory switch is enabled by resetting the power switch. No. 8 is only for Japanese model.

# GS \* n1 n2 [ d ] n1 n2 D1 ..... Dn

**[Function]** Defining the Download Bit Image (LOGO)

**[Code]** <1D>H<2A>H<n1><n2> [ < d > ]

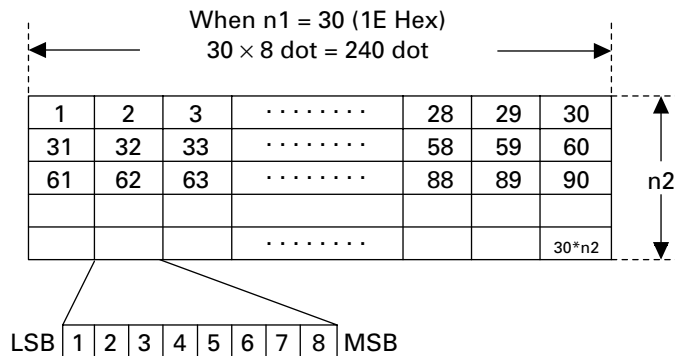
**[Range]** {1 =< n1 =< 7F} Defines horizontal size of downloaded image.  
 {1 =< n2 =< F8} Defines the vertical size of downloaded image.

**[Outline]** Defines downloading bit images of the number of dots specified by n1 and n2. The numbers of dots are n1 × 8 in horizontal direction and n2 × 8 in vertical direction. The number of horizontal bytes can be up to 7F, but only the first 48 (30H) bytes will be printed. The rest will be rejected.

- d indicates bit image data.
- The downloaded bit image thus defined is valid till it is redefined or ESC \_ is executed. After power off, the downloaded bit image is maintained even by the execution of Initialize command.

**[Caution]**

- This downloaded bit image is also printed at the self-test printing.
- The maximum size of the Bit image cannot exceed 16 KB.
- Relations between the bit image data and the dot defined are shown below.



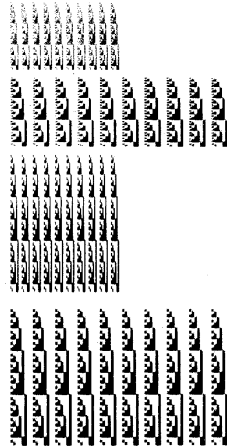
**[See Also]** GS /

**[Sample Program]**

```

GOSUB IMG
PRINT #1, CHR$ (&H1D) + “/” + CHR$ (0);
PRINT #1, CHR$ (&H1D) + “/” + CHR$ (1);
PRINT #1, CHR$ (&H1D) + “/” + CHR$ (2);
PRINT #1, CHR$ (&H1D) + “/” + CHR$ (3);
END
IMG:
n 1 = 10 : n 2= 50
PRINT #1, CHR$ (&H1D) + “*”;
PRINT #1, CHR$ (n1) + CHR$ (n2);
FOR I=1 TO n2
FOR J=1 TO n1
PRINT #1, CHR$ (J);
NEXT I
NEXT J
RETURN
    
```

**[Print Results]**



## GS / m

---

**[Function]** Printing the Download, Bit Image

**[Code]** <1D>H<2F>H<m>

**[Range]** {0 =< m =< 03(Hex)}

**[Outline]** Prints download bit image in a mode specified by m.  
• Modes can be selected by m are shown below.

m (Hex)	Name	Dot Density in Vertical Direction	Dot Density in Horizontal Direction
0	Normal mode	203 DPI	203 DPI
1	Double wide mode	203 DPI	101 DPI
2	Double high mode	101 DPI	203 DPI
3	Double wide/ double high mode	101 DPI	101 DPI

**[Caution]**

- When data exist inside the print buffer, this command is ignored.
- When a download bit image has not been defined, this command is ignored.
- A portion of a download bit image exceeding one line length is not printed.

**[See Also]** GS \*

**[Sample Program]** Refer to the programming example and print result of GS \*.

# GS :

---

**[Function]** Starting/Ending Macro Definition

**[Code]** <1D>H<3A>H

**[Outline]** Specifying starting/ending macro definition.  
Means termination when received while defining a macro.

**[Caution]**

- Maximum content available for macro definition is 3328 bytes.
- A portion exceeding 3328 bytes is not defined.
- Even with ESC @ (initialization of the printer) having been executed, defined content is not cleared. Therefore, it is possible to include ESC @ into the content of macro definition.
- Normal printing operation is carried out even while in macro definition.

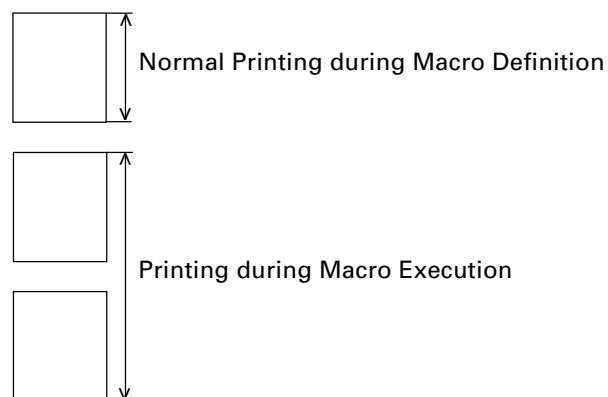
**[Default]** Initially, Macro is not specified.

**[See Also]** GS ^

**[Sample Program]**

```
PRINT #1, CHR$ (&H1D) + “: “;
PRINT #1, “+—+” + CHR$ (&HA);
PRINT #1, “||” + CHR$ (&HA);
PRINT #1, “+—+” + CHR$ (&HA);
PRINT #1, CHR$ (&H1D) + “: “;
PRINT #1, CHR$ (&H1D) + “^ “;
PRINT #1, CHR$ (2) + CHR$ (10);
PRINT #1, CHR$ (0);
```

**[Print Results]**





## GS L nL nH

---

<b>[Function]</b>	Setting the left margin
<b>[Code]</b>	<1D>H<4C>H<nL><nH>
<b>[Range]</b>	{0 =< nL =< FF(Hex)} {0 =< nH =< FF(Hex)}
<b>[Outline]</b>	This command sets the left margin specified by nL and nH. The value of the left margin is [(nL + nH × 256) × 1/203] inches.
<b>[Caution]</b>	<ul style="list-style-type: none"><li>• This command only works when it is entered at the beginning of a line.</li><li>• The maximum settable left margin is equal to the horizontal printable area. A setting greater than this maximum is trimmed to the maximum.</li><li>• When mapping character data, if the print area specified is not wide enough to accommodate one character of the current font, only the line for that character data is handled as follows:<ol style="list-style-type: none"><li>(1) The print area is extended toward the right to be equivalent to one character of the current font, but not wider than the printable area.</li><li>(2) If an area for one character cannot be provided as a result of step (1), the print area is extended toward the left. (So, the left margin is decreased.)</li></ol></li><li>• When mapping non-character data (Bit image, downloaded bit image, or bar code), if the print area specified is narrower than 9-bits, only the line for that data is handled as follows:<ol style="list-style-type: none"><li>(1) The print area is extended toward the left (So, the left margin is decreased) until it is 9-dot wide, but not wider than the printable area.</li></ol></li></ul>
<b>[Default]</b>	nL = 0, nH = 0

## GS W n

---

**[Function]** Selecting the horizontal size (Scale factor) of the Bar Code

**[Code]** <1D>H<77>H<n>

**[Range]** {2 =< n =< 4(Hex)}

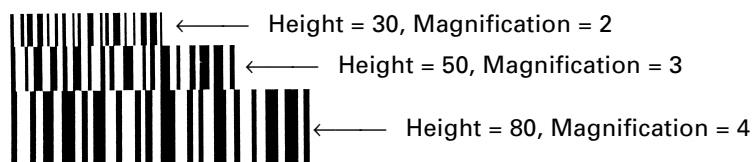
**[Outline]** Selecting bar code width.  
n denotes the number of dots in fine element width.

**[Default]** The initial value of this width is "3".

**[Sample Program]**

```
PRINT #1, CHR$ (&H1D) + "h" + CHR$ (30);  
PRINT #1, CHR$ (&H1D) + "w" + CHR$ (2);  
GOSUB BC  
PRINT #1, CHR$ (&H1D) + "h" + CHR$ (50);  
PRINT #1, CHR$ (&H1D) + "w" + CHR$ (3);  
GOSUB BC  
PRINT #1, CHR$ (&H1D) + "h" + CHR$ (80);  
PRINT #1, CHR$ (&H1D) + "w" + CHR$ (4);  
GOSUB BC  
END  
BC:  
PRINT #1, CHR$ (&H1D) + "k";  
PRINT #1, CHR$ (4);  
PRINT #1, "12" + CHR$ (0);  
RETURN
```

**[Print Results]**



## GS ^ n1 n2 n3

---

**[Function]** Executing the Macro

**[Code]** <1D>H<5E>H<n1><n2><n3>

**[Range]** {0 =< n1 =< FF(Hex)}  
{0 =< n2 =< FF(Hex)}  
{0 =< n3 =< 1(Hex)}

**[Outline]** Executing contents defined in macro.  
The following items are set for parameters n1 to n3.  
n1: The number of times of macro execution  
n2: Waiting time on macro execution. Waiting time of  $n2 \times 100$  msec is given for every execution.  
n3: Macro execution mode

n (Hex)	Mode
0	Continuous execution
1	Execution by LF switch

Continuous execution: The Macro is executed n1 times continuously at the time intervals specified by n2.

Execution by LF switch: After waiting for lapse of time specified by n2, the LF switch is waited to be pressed. When it is pressed, the macro is executed once.

This action is repeated n1 times.

**[Caution]**

- When this command is received while in macro definition, suspension of macro definition is indicated. At this time, the defined content is cleared.
- No execution takes place when macro is held undefined or  $n1=0$ .
- While in macro execution with  $n3=1$ , paper feed with the LF switch is not available.
- During execution of a macro, the power cannot be turned off.

**[See Also]** GS :

**[Sample Program]** See Sample Program and Print Results for GS :.

# GS a n

**[Function]** Enabling/Disabling ASB (Automatic Status Back)

**[Code]** <1D>H<61>H<n>

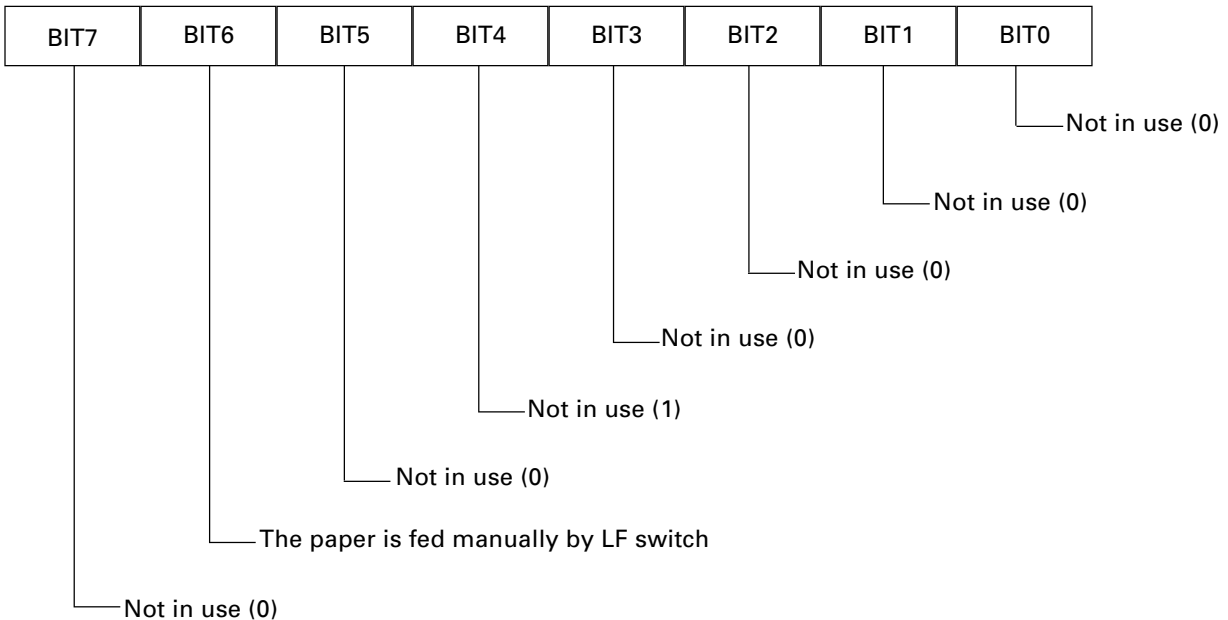
**[Range]** {0 =< n=< FF(Hex)}

**[Outline]** This command selects the status item to be addressed by ASB (Automatic Status Back.)

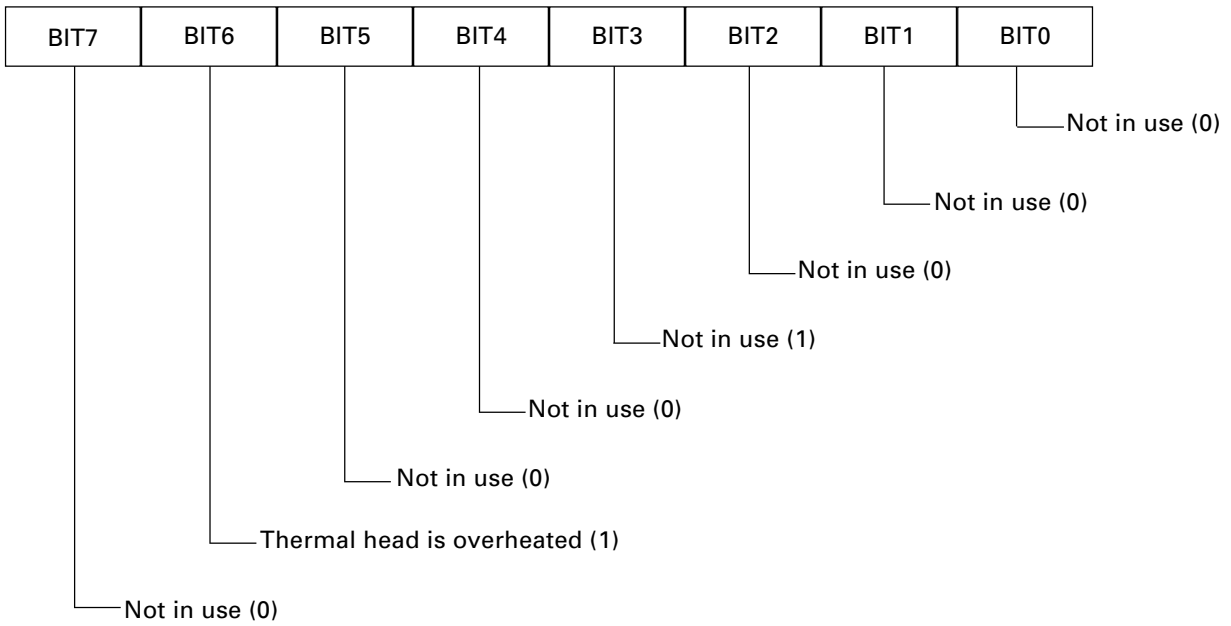
Bit	Status Item Addressed by ABS
0	Not defined
1	Defining LF button function
2	Thermal head temperature status
3	Paper end sensor and paper cover open status
4	Not defined
5	Not defined
6	Not defined
7	Not defined

When this command is executed, the printer will return 4 bytes with status information, with the following meaning:

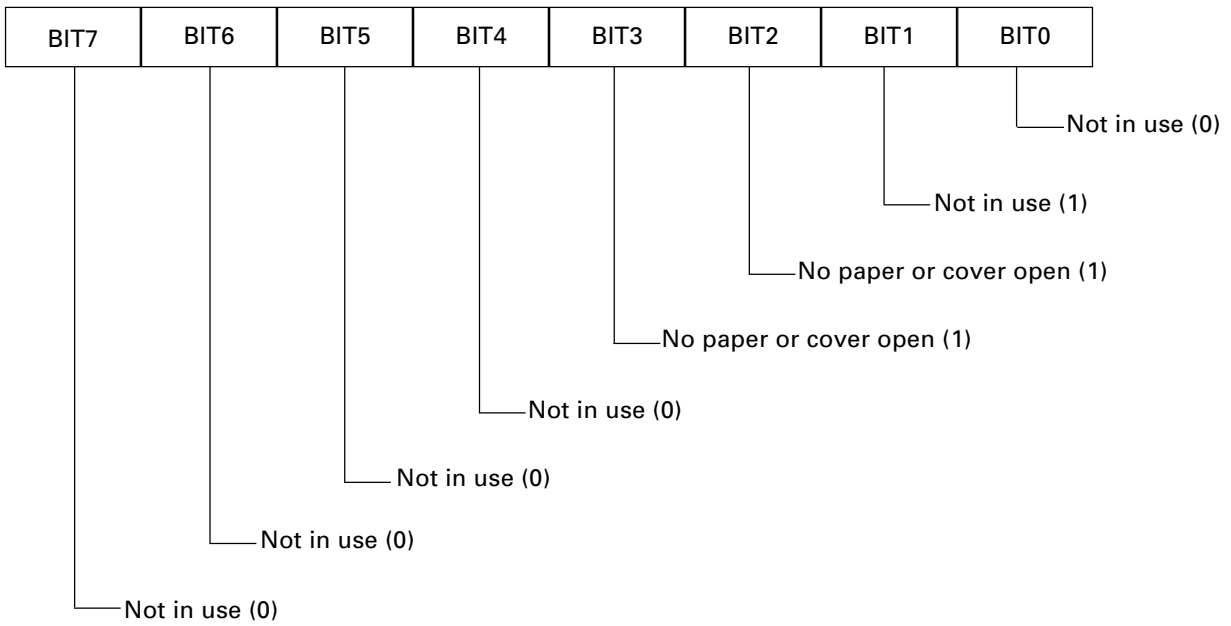
BYTE 1



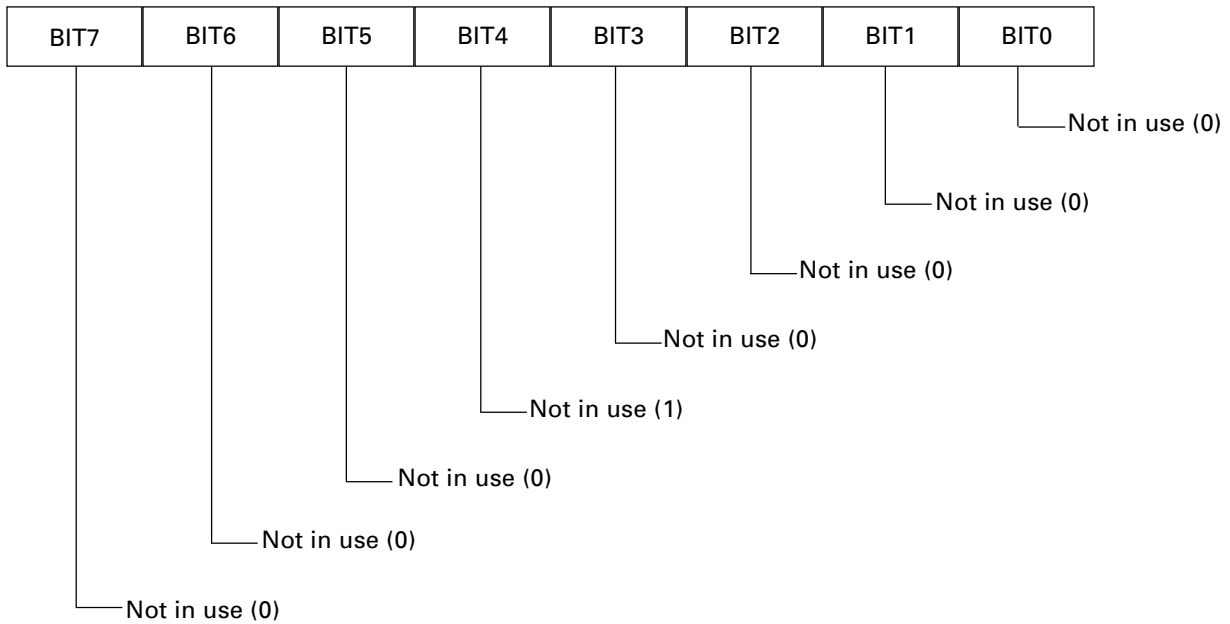
BYTE 2



BYTE 3



## BYTE 4



### [Caution]

- If any status item is enabled, the status is sent to the host when this command is executed. After that time on, the status is sent each time an enabled status item changes. Because each status item represents the current condition, status items disabled for ASB may also have changed.
- The ASB function is disabled if all status items are disabled.
- The printer sends 4 bytes of status shown in the tables below, without checking whether the host is ready to receive or busy. The 4 bytes of status is a continuous string except for XOFF code.
- Because this command is executed when data is mapped in the receive buffer, there may be a delay between command receiving and status sending depending on the condition of the receive buffer.
- Even if the printer is excluded from the selection of peripheral equipment (ESC =), the 4 bytes of status is sent to the host whenever status changes.

# GS f n

---

**[Function]** Selecting the font of HRI characters

**[Code]** <1D>H<66>H<n>

**[Range]** {n = 0, 1(Hex)}

**[Outline]** Selecting the font of HRI characters in printing bar code.  
The type of font can be selected with “n” as follows:

n (Hex)	Font
0	Font A (12 × 24)
1	Font B (9 × 16)

The HRI characters refer to the bar code-turned characters so that you can read them.

**[Caution]** The HRI characters are printed at the position specified with GS H.

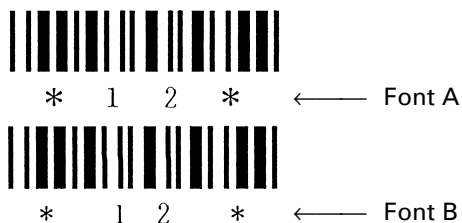
**[Default]** n = 0

**[See Also]** GS H

**[Sample Program]**

```
PRINT #1, CHR$(&H1D) + "h" + CHR$(50);  
PRINT #1, CHR$(&H1D) + "H" + CHR$(2);  
PRINT #1, CHR$(&H1D) + "F" + CHR$(0);  
GOSUB BC  
PRINT #1, CHR$(&H1D) + "f" + CHR$(1);  
GOSUB BC  
END  
BC:  
PRINT #1, CHR$(&H1D) + "k";  
PRINT #1, CHR$(4);  
PRINT #1, "12" + CHR$(0);  
PRINT #1, CHR$(&HA);  
RETURN
```

**[Print Results]**



## GS h n

---

<b>[Function]</b>	Selecting the height of the Bar Code
<b>[Code]</b>	<1D>H<68>H<n>
<b>[Range]</b>	{1 =< n =< FF(Hex)}
<b>[Outline]</b>	Selecting bar code height. n denotes the number of dots in the vertical direction.
<b>[Default]</b>	n = a2(Hex) 162 dots
<b>[Sample Program]</b>	See Sample Program and Print Results for GS w .

## GS k m [d1 ..... dk] NUL

## GS k m n [d1 ..... dn]

**[Function]** Printing the bar code

**[Code]** (1) <1D>H<6B>H<m> [d1.....dk] NUL  
 (2) <1D>H<6B>H<m><n> [d1....dn]

**[Range]** (1) 0 =< m =< 6 The definitions of "k" and "d" vary with the bar code system.  
 (2) 65 =< m =< 73 The definitions of "n" and "d" vary with the bar code system.

**[Outline]** Selects a bar code system and prints the bar code.

For (1):

m	Bar code system	Range of "k"	Range of "d"
0	UPC-A	k=11	48 =< d =< 57
1	UPC-E	k=11	48 =< d =< 57
2	JAN13 (EAN)	k=12	48 =< d =< 57
3	JAN8 (EAN)	k=7	48 =< d =< 57
4	CODE39	1 =< k	48 =< d =< 57, 65 =< d =< 90 32, 36, 37, 43, 45, 46, 47
5	ITF	1 =< k (Must be an even number)	48 =< d =< 57
6	CODABAR (NW-7)	1 =< k	48 =< d =< 57, 65 =< d =< 68 36, 43, 45, 46, 47, 58

For (2):

m(2)	Bar code system	Range of "n"	Range of "d"
65	UPC-A	n=11	48 =< d =< 57
66	UPC-E	n=11	48 =< d =< 57
67	EAN13 (JAN)	n=12	48 =< d =< 57
68	EAN8 (JAN)	n=7	48 =< d =< 57
69	CODE39	1 =< n	48 =< d =< 57, 65 =< d =< 90 32, 36, 37, 43, 45, 46, 47
70	ITF	1 =< n (Must be an even number)	48 =< d =< 57
71	CODABAR	1 =< n	48 =< d =< 57, 65 =< d =< 68 36, 43, 45, 46, 47,58
72	CODE93	1 =< n	0 =< d =< 127
73	CODE128	1 =< n	0 =< d =< 127

## [Caution]

### For (1):

- This command ends with a NUL code.
- For UPC-A or UPC-E, the bar code is printed when 12 bytes of bar code data have been entered, and the subsequent data is handled as normal data.
- For JAN13, the bar code is printed when 13 bytes of bar code data have been entered, and the subsequent data is handled as normal data.
- For JAN8, the bar code is printed when 8 bytes of bar code data have been entered, and the subsequent data is handled as normal data.
- The data of ITF bar code must have an even number of columns. Should the data have an odd number of columns, the last column is ignored.

### For (2):

- Numeral "n" indicates the number of data items, and the subsequent "n" bytes of data are handled as bar code data.
- If "n" is out of the range, the processing of the command is aborted, and the subsequent data is handled as normal data.

### For STANDARD MODE:

- If "d" is out of the range, only a paper feed is executed, and the subsequent data is handled as normal data.
- If the bar code is wider than the print area for one line, the bar code is not printed, but only a paper feed is executed.
- The amount of paper feed corresponds to the height of the bar code (Including the HRI characters if HRI character printing is specified), irrespective of the line feed width set by a command such as ESC 2 or ESC 3.
- This command only works if no data exists in the print buffer. If any data exists in the print buffer, the data subsequent to "m" is handled as normal data.
- After the bar code is printed, the beginning of the line is taken as the start position for the next print.
- This command is not affected by any print modes (Emphasis, double strike, underline, and character size), except for the inverted character mode.

## [Description of Bar Codes]

<For print examples, refer to page 56>

### UPC-A

This bar code, consisting of numerals only, has a fixed length of 12 columns; a 11-column number entered from the host or application software plus a check digit (12th column) automatically calculated inside the printer. If the 12th-column numeral is sent from the host, the entire bar code will be printed as it is.

### UPC-E

This bar code, consisting of numerals only, has a fixed length of 8 columns; the first number system character is "0" stationary. A 12-column numeral entered from the host or application software is compressed to 8 columns with a check digit attached and then is printed. The 12th-column check digit is automatically calculated inside the printer. If it is and sent from the host, the entire bar code will be printed, compressed to 8 columns.

### JAN-13(EAN)

This bar code, consisting of numerals only, has a fixed length of 13 columns; a 12-column number entered from the host or application software plus a check digit (13th column) automatically calculated inside the printer. If the 13th-column numeral is sent from the host, the entire bar code will be printed as it is.

## **JAN-8(EAN)**

This bar code, consisting of numerals only, has a fixed length of 8 columns; a 7-column number entered from the host or application software plus a check digit (8th column) automatically calculated inside the printer. If the 8th-column numeral is sent from the host, the entire bar code will be printed as it is.

## **CODE39**

This bar code, consisting of upper-case alphabetic characters and numerals, has a variable length of columns. The start/stop code "\*" is automatically added by the printer. The available characters include space and "\$ % + - • / 0 1 2 3 4 5 6 7 8 9" and upper-case alphabetic characters.

A check digit is added and printed automatically.

## **ITF**

This bar code, consisting of only numerals, has a variable length of even-number columns. If a code of odd-number columns is sent, the bar code will not be printed.

## **CODABAR (NW-7)**

This bar code, consisting of alphanumerics, has a variable length of columns. Available characters include "0 1 2 3 4 5 6 7 8 9 A B C D \$ + - . / :". A start/stop code is required; any one of A, B, C, and D is used.

## **CODE93**

This bar code, consisting of alphanumeric and control characters, has a variable length of columns. The HRI character string is preceded and followed by a "n" character. HRI characters for control characters (00H - 1FH, and 7FH) are each printed as a combination of a "n" character and an alphabetic character.

## **CODE128**

This bar code consists of 103 bar code characters and three code sets, enabling 128 ASCII code characters to be printed. It has a variable length of columns.

- Code set A ASCII characters 00H - 5FH can be represented.
- Code set B ASCII characters 20H - 7FH can be represented.
- Code set C Two-digit numbers 00 - 99 can each be represented by one character.

In addition to the above characters, special characters are available:

- Shift character (SHIFT)

When used in code set A, one character next to a Shift character is treated as a character of code set B. When used in code set B, one character next to a Shift character is treated as a character of code set A. The Shift character cannot be used in code set C.

- Code set select characters (CODE A, CODE B, CODE C):

The code set following a code set select character is switched to code set A, B, or C.

- Function characters (FNC1, FNC2, FNC3, FNC4):

How the function characters are used depends on each application. In code set C, only FNC1 is available.

When sending print data, note these points:

- (1) Each string of bar code data must begin with a code set select character (CODE A, CODE B, or CODE C), which selects the first code set to use.
- (2) Every special character is specified by a combination of two characters: a brace "{" followed by one character. A brace "{" itself is sent twice consecutively.

**SPECIAL CHARACTERS:**

Hex.	ASCII	Code set A	Code set B	Code set C
7B53	{ S	SHIFT	SHIFT	-N/A
7B41	{ A	-N/A	CODE A	CODE A
7B42	{ B	CODE B	-N/A	CODE B
7B43	{ C	CODE C	CODE C	-N/A
7B31	{ 1	FNC1	FNC1	FNC1
7B32	{ 2	FNC2	FNC2	-N/A
7B33	{ 3	FNC3	FNC3	-N/A
7B34	{ 4	FNC4	FNC4	-N/A
7B7B	{{	"{"	"{"	"{"

<Example>

To print "No." in code set B, followed by "123456" in code set C, send the following data string:

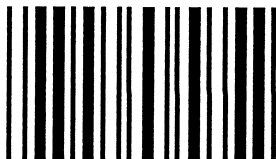
GS k <73><10><7Bh 42h> "No." <7Bh 43h><12><34><56>

- If the printer finds a string of bar code data that does not begin with a code set select character, it immediately aborts the command processing and handles the subsequent data as normal data.
- If the printer received a character that is not available in the currently selected code set, it immediately aborts the command processing and handles the subsequent data as normal data.
- An HRI character corresponding to either a Shift character or a code select character is not printed. An HRI character for either a function character or a control character is treated as a space character.

**[Sample Program]**

```
PRINT #1, CHR$(&H1D) + "H" + CHR$(2);
PRINT #1, CHR$(&H1D) + "k";
PRINT #1, CHR$(4);
PRINT #1, "123" + CHR$(0);
END
```

**[Print Results]**












\* 1 2 3 \*

When the data "123" is printed with the code 39

**[Description of Bar Codes]**

UPC-A, UPC-E, JAN-13 (EAN), JAN-8 (EAN), CODE39, ITF, CODABAR, CODE128

Type	Print Sample	Outline of Symbol	Max. Column
UPC-A		12-column fixed-length bar code consisting of numerals only.	—
UPC-E		8-column fixed-length bar code consisting of numerals only. Abbreviated version of UPC-A.	—
JAN-13		13-column fixed-length bar code consisting of numerals only.	—
JAN-8		8-column fixed-length bar code consisting of numerals only.	—
CODE39		Variable-length bar code consisting of alphabets and numerals. The start/stop code "*" is automatically added.	11
ITF		Even-column variable-length bar code consisting of numerals only.	22
CODABAR (NW-7)		Variable-length bar code consisting of alpha numerals. Any one of A, B, C, and D is required as the start/stop code.	15
CODE93			—
CODE128		Variable-length bar code consisting of all 128 ASCII code characters.	14

Printing is done depending on bar code specification type, number of print column, bar code height, width (Magnification), visible code presence, and bar code data specification.

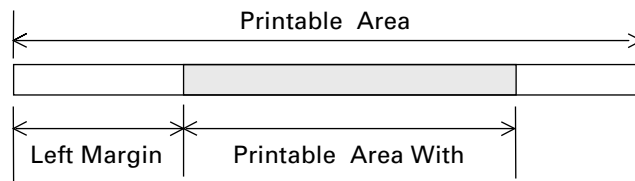
# GS w nL nH

**[Function]** Setting the print area width

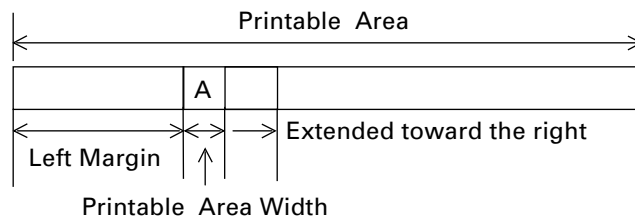
**[Code]** <1D>H<57>H<nL><nH>

**[Range]** {0 =< nL =< FF(Hex)}  
 {0 =< nH =< FF(Hex)}

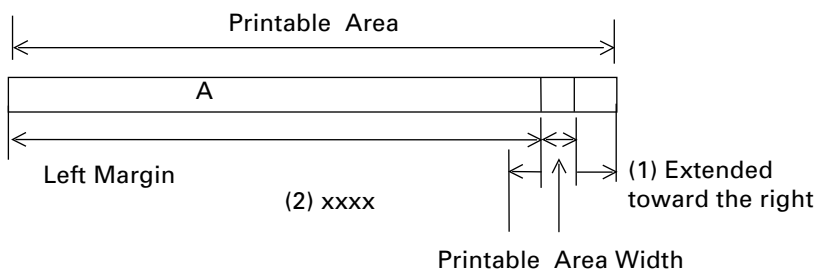
**[Outline]** Sets the print area width specified by nL and nH.  
 • The print area width will be  $[(nL + nH \times 256) \times 1/203]$  inches.



- [Caution]**
- This command only works when it is entered at the beginning of a line.
  - If the value entered with this command exceeds the printable area for one line, the entire area except the left margin is set as the print area width.
  - For calculation of print area width, the basic pitch 1/203 inch for horizontal direction is used.
- (1) The print area is extended toward the right to accommodate the first character, but not wider than the printable area.



- (2) When sufficient area cannot be secured even by executing the processing (1), the print area is extended to the left (decreasing left margin).



- (3) When sufficient area cannot be secured even by executing the processing (2), the right space is decreased.
- When bit image (or downloaded bit image) is developed, if the print area width is less than the minimum lateral width of the bit image (2 dots for single density, 1 dot for double density), the following processing is carried out only in the same line.
  - (1) Extend the print area to the left (decreases left margin) till it reaches the width equivalent to the minimum lateral width of bit image within the range of printable area.

**[Default]** nL = 40 (Hex), nH = 2 (Hex)

**[See Also]** GS L

### 3. CHARACTER CODES TABLE

WPC1252

	8	9	A	B	C	D	E	F
0	€			°	À	Ð	à	ð
1		‘	ı	±	Á	Ñ	á	ñ
2	,	’	¢	²	Â	Ò	â	ò
3	f	“	£	³	Ã	Ó	ã	ó
4	„	”	¤	´	Ä	Ô	ä	ô
5	...	•	¥	µ	Å	Ö	å	ö
6	†	–	¦	¶	Æ	Ö	æ	ö
7	‡	—	§	·	Ç	×	ç	÷
8	^	˘	¨	,	È	Ø	è	ø
9	‰	™	©	¹	É	Ù	é	ù
A	Š	š	ª	º	Ê	Ú	ê	ú
B	‹	›	«	»	Ë	Û	ë	û
C	Œ	œ	¬	¼	Ì	Ü	ì	ü
D			-	½	Í	Ý	í	ý
E			®	¾	Î	Þ	î	þ
F		ÿ	—	¿	Ï	ß	ï	ÿ

## 4. PROGRAMMING CONSIDERATIONS

### 4.1 Notes at Lowered Battery Voltage

When battery is low, highly loaded printing such as printing of graphics with much black area may cause power off during printing.

When executing high loaded printing, charging the battery before lowering of battery voltage is recommended while keeping checking the information on the battery by the use of relevant command.

### 4.2 Notes on Printing After Paper Cutting

When the paper is cut by manual cutter, paper is likely to be caught by the cutter.

Therefore, in printing the first line just after cutting, the printing may be compressed in vertical direction.

When starting printing, feed one line without printing to prevent compressed printing.