

DM Application Programming Guide

CONFIDENTIAL

EPSON

**Displays &
Commands**

**ESC/POS
Proprietary
Command
System**

**Character
Code Tables**

*[CLICK HERE](#) for
"About This Manual"*

*[CLICK HERE](#) for
"Confidentiality Agreement"*



ESC/POS® PROPRIETARY COMMAND SYSTEM

The market for store automation equipment is changing rapidly with the widespread introduction of POS (point of sale) terminals. These terminals are now appearing even in small retail stores and specialty shops.

As personal computers begin to be used as POS terminals, the demand for matching uniform peripheral devices is expected to rise. At present, however, many of the competing POS terminal printers on the market employ mutually incompatible command sets. This imposes limits on the expandability and range of applications possible with PC-based systems. There is a need for a new command set designed to provide the expandability and universal applicability demanded by the market.

ESC/POS

EPSON took the initiative by introducing ESC/POS, a proprietary POS printer command system including patented commands and enabling versatile POS system construction with high scalability. Compatible with all types of EPSON POS printers and displays, this proprietary control system also offers the flexibility to easily make future upgrades. Its popularity is worldwide.

ESC/POS is designed to reduce the processing load on the host computer in POS environments. It comprises a set of highly functional and efficient commands that enable the full realization of the potential of printers.

A command set designed for universal applicability

The commands that are supported by all EPSON POS printers and those that are specific to individual models are clearly described. This means that ESC/POS compatible software will work with any system and be suitable for a wide range of applications.

Superb expandability allowing the addition of new functions

New functions can be added and accommodated by the categories already provided in the command system.

Allows more effective use of software

Once a software application has been created for one printer in the TM series, it can be used as the basis for versions for the other printers in the series. Only a small portion of the program source code needs to be modified.



This page intentionally left blank.



COMMAND CLASSIFICATION

ESC/POS display commands are divided into two types: basic commands and extension commands.

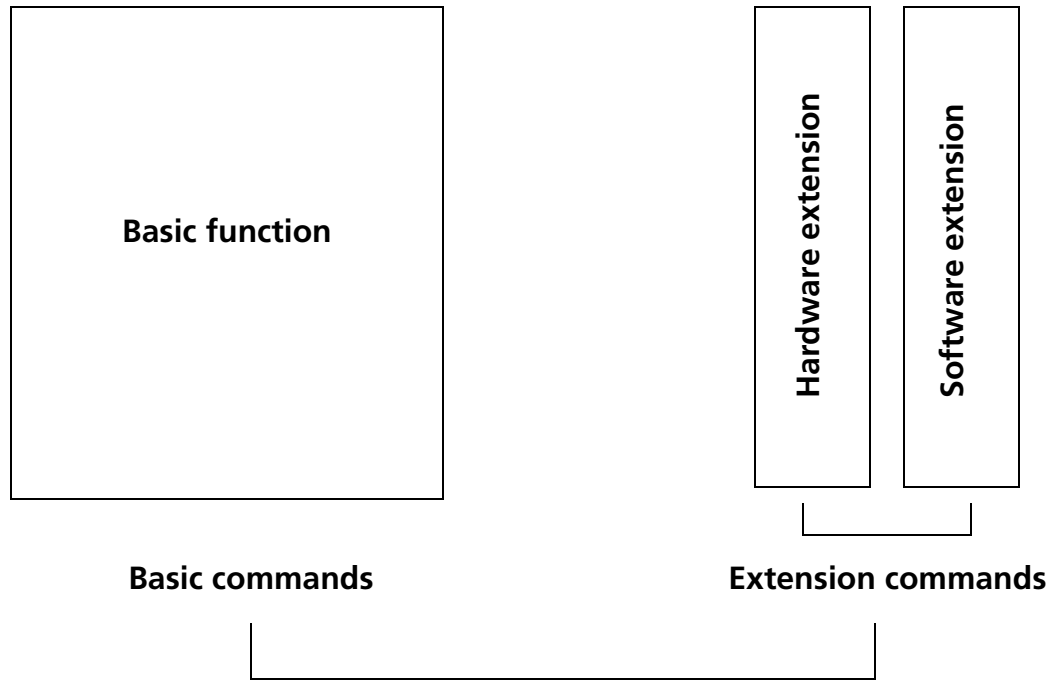
Basic commands are defined as the fundamental display control code, including display position specification and display screen control.

Extension commands are defined as control codes for functions specific to individual display. They are further divided into the following two categories.

- Hardware extension commands that relate to hardware functions such as brightness adjustment.
- Software extension commands that relate to software functions.

The following is an overview diagram of the matrix classification of ESC/POS.





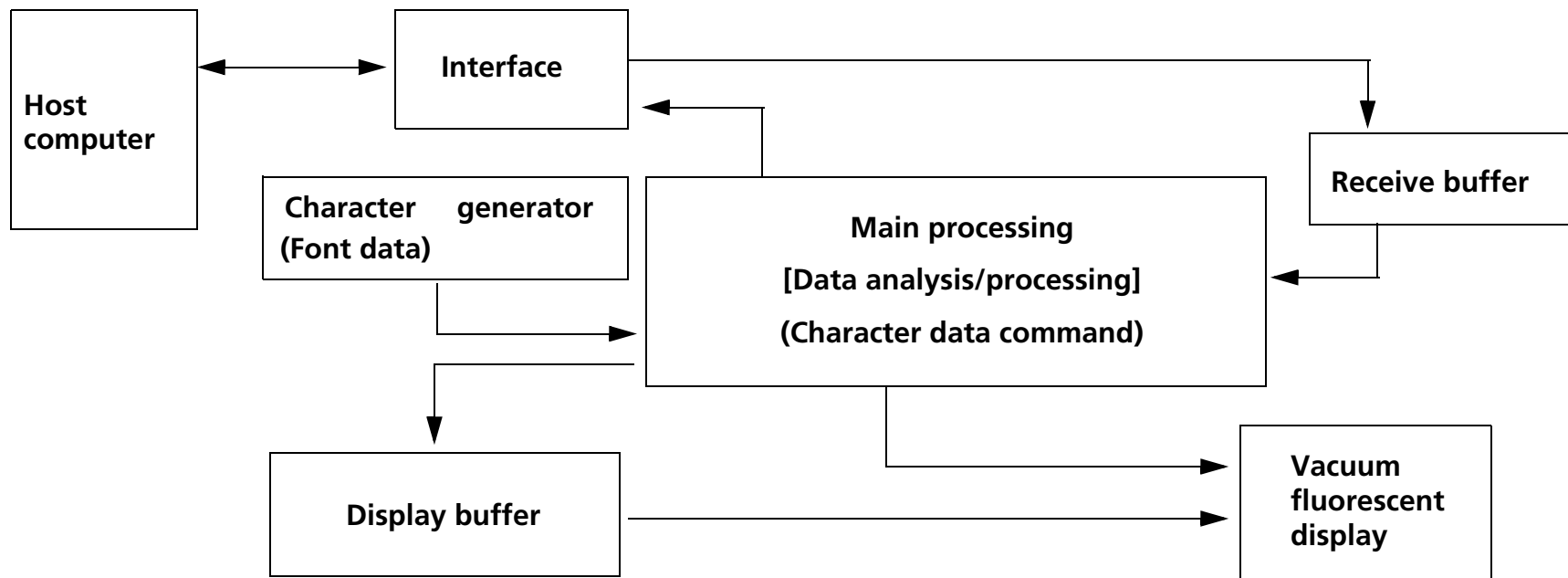
OVERVIEW OF DATA PROCESSING

Character Data Commands

The display stores data sent from the host computer in the receive buffer temporarily and then the display interprets the data and classifies them into commands or character data sequentially. If the data from the receive buffer is a command, the display processes the command corresponding to its function.

If the data from the receive buffer is character data, the display reads the appropriate font data from the resident character generator and displays image data.

Data processing diagram



DEFINITIONS

- (1) **Receive buffer**
The receive buffer is used to store data from the host computer. All received data is stored in this buffer and processed in the order received.
- (2) **Display buffer**
The display buffer is used to store image data for display.
- (3) **Ignoring a command**
This is the state in which the display does nothing after receiving all codes, including parameters.
- (4) **Setting commands**
The commands that change display status by processing a command and affect display operation and display results thereafter.
- (5) **Executing commands**
The commands that affect display operation and change the display status temporarily but do not affect the following display operation.
- (6) **MSB**
Most Significant Bit
- (7) **LSB**
Least Significant Bit



CHARACTER CODE TABLES

SP in a table represents space. See [Using the character code tables](#) for information on how to read these tables.

Page 0 (PC437: U.S.A., Standard Europe) (International character set: U.S.A.)

HEX	0	1	2	3	4	5	6	7
0	NUL	SP	O	Ⓞ	P		P	
1	MD1	!	1	A	Q	a	q	
2	MD2	"	2	B	R	b	r	
3	MD3	#	3	C	S	c	s	
4		\$	4	D	T	d	t	
5		%	5	E	U	e	u	
6		&	6	F	V	f	v	
7		'	7	G	W	g	w	
8	BS	CAN	(H	X	h	x	
9	HT)	9	I	Y	i	y	
A	LF	*	:	J	Z	j	z	
B	COM	ESC	+	;	K	[k	{
C	CLR	,	<	L	\	l	;	'
D	CR	-	=	M]	m	}	"
E		.	>	N	^	n	~	Ⓜ
F		US	/	?	O	o	SP	

HEX	8	9	A	B	C	D	E	F
0	Ç	È	É	Ⓜ	Ì	Í	Î	Ï
1	Ü	Æ	Ó	Ⓜ	Ť	Ŧ	ß	±
2	É	Æ	Ó	Ⓜ	Ť	Ŧ	Γ	≥
3	À	Ô	Ú	Ï	Ŧ	Ë	Ï	≤
4	Ä	Ö	Ñ	†	-	Ë	Σ	ƒ
5	À	Ò	Ñ	†	+	Ŧ	σ	ƒ
6	À	Û	À	†	†	Ŧ	μ	÷
7	Ç	Û	Ω	†	†	†	τ	≈
8	È	ÿ	Ë	†	†	†	φ	°
9	Ë	Ö	†	†	†	†	θ	•
A	È	Û	†	†	†	†	Ω	'
B	Ï	†	†	†	†	†	δ	√
C	Ï	€	†	†	†	†	∞	n
D	Ï	†	†	†	†	†	∅	ˆ
E	Ä	†	†	†	†	†	€	•
F	À	†	†	†	†	†	∩	SP

Note: Character codes from 00H (hexadecimal) to 7FH (hexadecimal) for each page are the same. Some characters indicated by character codes from 00H to 7FH are changed by selecting the international character set. See the **ESC R** command description.



	HEX	8	9	A	B	C	D	E	F
HEX	BIN	1000	1001	1010	1011	1100	1101	1110	1111
0	0000	■	■	SP	ー	タ	ミ	□	日
		128	144	160	176	192	208	224	240
1	0001	■	■	・	ア	チ	ム	■	月
		129	145	161	177	193	209	225	241
2	0010	■	■	「	イ	ツ	メ	■	火
		130	146	162	178	194	210	226	242
3	0011	■	■	」	ウ	テ	モ	○	水
		131	147	163	179	195	211	227	243
4	0100	■	■	、	エ	ト	ヤ	●	木
		132	148	164	180	196	212	228	244
5	0101	■	■	・	オ	ナ	ユ	◇	金
		133	149	165	181	197	213	229	245
6	0110	■	■	ヲ	カ	ニ	ヨ	◆	土
		134	150	166	182	198	214	230	246
7	0111	■	→	ア	キ	ヌ	ラ	◆	年
		135	151	167	183	199	215	231	247
8	1000	■	←	イ	ク	ネ	リ	▶	円
		136	152	168	184	200	216	232	248
9	1001	■	↑	ウ	ケ	ノ	ル	◀	分
		137	153	169	185	201	217	233	249
A	1010	■	↓	エ	コ	ハ	レ	▲	人
		138	154	170	186	202	218	234	250
B	1011	■	×	オ	サ	ヒ	ロ	▼	大
		139	155	171	187	203	219	235	251
C	1100	■	÷	ヤ	シ	フ	ワ	«	中
		140	156	172	188	204	220	236	252
D	1101	■	±	ユ	ス	ヘ	ン	»	小
		141	157	173	189	205	221	237	253
E	1110	■	≤	ヨ	セ	ホ	°	‡	〒
		142	158	174	190	206	222	238	254
F	1111	■	≥	ツ	ソ	マ	°	‡	°C
		143	159	175	191	207	223	239	255



	HEX	8	9	A	B	C	D	E	F
HEX	BIN	1000	1001	1010	1011	1100	1101	1110	1111
0	0000	Ç 128	É 144	á 160	⌘ 176	⌘ 192	ð 208	Ó 224	— 240
1	0001	ü 129	æ 145	í 161	⌘ 177	⌘ 193	Ð 209	ß 225	± 241
2	0010	é 130	Æ 146	ó 162	⌘ 178	⌘ 194	Ê 210	Ô 226	= 242
3	0011	â 131	ô 147	ú 163	 179	⌘ 195	Ë 211	Ò 227	³ / ₄ 243
4	0100	ä 132	ö 148	ñ 164	⌘ 180	— 196	È 212	õ 228	¶ 244
5	0101	à 133	ò 149	Ñ 165	Á 181	⌘ 197	ı 213	Õ 229	§ 245
6	0110	å 134	û 150	ä 166	Â 182	ã 198	í 214	μ 230	÷ 246
7	0111	ç 135	ù 151	ó 167	À 183	Ã 199	î 215	þ 231	· 247
8	1000	ê 136	ÿ 152	ı 168	© 184	⌘ 200	ï 216	þ 232	° 248
9	1001	ë 137	Ï 153	® 169	⌘ 185	⌘ 201	⌘ 217	Ú 233	¨ 249
A	1010	è 138	Û 154	¬ 170	⌘ 186	⌘ 202	⌘ 218	Û 234	· 250
B	1011	ï 139	ø 155	¹ / ₂ 171	⌘ 187	⌘ 203	■ 219	Ù 235	¹ 251
C	1100	î 140	£ 156	¹ / ₄ 172	⌘ 188	⌘ 204	■ 220	ý 236	³ 252
D	1101	ì 141	Ø 157	ı 173	¢ 189	= 205	ı 221	Ý 237	² 253
E	1110	Ä 142	× 158	« 174	¥ 190	⌘ 206	î 222	— 238	■ 254
F	1111	Å 143	f 159	» 175	⌘ 191	⌘ 207	■ 223	‘ 239	SP 255



	HEX	8	9	A	B	C	D	E	F
HEX	BIN	1000	1001	1010	1011	1100	1101	1110	1111
0	0000	Ç 128	É 144	á 160	☒ 176	Ł 192	ł 208	α 224	≡ 240
1	0001	ü 129	À 145	í 161	☒ 177	ł 193	ṽ 209	β 225	± 241
2	0010	é 130	È 146	ó 162	☒ 178	ṽ 194	ṽ 210	Γ 226	≥ 242
3	0011	ã 131	ô 147	ú 163	 179	ł 195	ł 211	π 227	≤ 243
4	0100	ã 132	õ 148	ñ 164	† 180	- 196	ł 212	Σ 228	f 244
5	0101	à 133	ò 149	Ñ 165	† 181	† 197	ṽ 213	σ 229	J 245
6	0110	Á 134	Ú 150	ã 166	† 182	ł 198	ṽ 214	μ 230	÷ 246
7	0111	ç 135	ù 151	Ω 167	ṽ 183	ł 199	ł 215	τ 231	≈ 247
8	1000	ê 136	Ï 152	¿ 168	ṽ 184	ł 200	ł 216	Φ 232	° 248
9	1001	Ê 137	Ö 153	Ò 169	† 185	ṽ 201	ł 217	θ 233	• 249
A	1010	è 138	Û 154	˘ 170	 186	ł 202	ṽ 218	Ω 234	· 250
B	1011	í 139	ç 155	‡ 171	ṽ 187	ṽ 203	■ 219	δ 235	√ 251
C	1100	ö 140	£ 156	‡ 172	ṽ 188	ł 204	■ 220	∞ 236	∞ 252
D	1101	ì 141	Û 157	i 173	ṽ 189	- 205	■ 221	ø 237	² 253
E	1110	Ā 142	Pt 158	« 174	ṽ 190	ł 206	■ 222	ε 238	■ 254
F	1111	Ā 143	Ó 159	» 175	ṽ 191	ł 207	■ 223	∩ 239	SP 255



	HEX	8	9	A	B	C	D	E	F
HEX	BIN	1000	1001	1010	1011	1100	1101	1110	1111
0	0000	Ç 128	É 144	Ï 160	Ï 176	Ł 192	Ł 208	α 224	≡ 240
1	0001	ü 129	È 145	Ï 161	Ï 177	Ł 193	Ł 209	β 225	± 241
2	0010	é 130	È 146	ó 162	Ï 178	Ł 194	Ł 210	Γ 226	≥ 242
3	0011	â 131	ô 147	ú 163	Ï 179	Ł 195	Ł 211	π 227	≤ 243
4	0100	Ë 132	È 148	Ï 164	Ï 180	Ł 196	Ł 212	Σ 228	f 244
5	0101	à 133	ÿ 149	Ï 165	Ï 181	Ł 197	Ł 213	σ 229	J 245
6	0110	ñ 134	û 150	Ï 166	Ï 182	Ł 198	Ł 214	μ 230	÷ 246
7	0111	ç 135	ù 151	Ï 167	Ï 183	Ł 199	Ł 215	τ 231	≈ 247
8	1000	ê 136	ϣ 152	Ï 168	Ï 184	Ł 200	Ł 216	Φ 232	° 248
9	1001	ë 137	ô 153	Ï 169	Ï 185	Ł 201	Ł 217	θ 233	• 249
A	1010	è 138	Û 154	Ï 170	Ï 186	Ł 202	Ł 218	Ω 234	· 250
B	1011	ï 139	φ 155	Ï 171	Ï 187	Ł 203	Ł 219	δ 235	√ 251
C	1100	î 140	£ 156	Ï 172	Ï 188	Ł 204	Ł 220	∞ 236	n 252
D	1101	— 141	Û 157	Ï 173	Ï 189	Ł 205	Ł 221	∅ 237	² 253
E	1110	À 142	Ó 158	« 174	Ï 190	Ł 206	Ł 222	€ 238	■ 254
F	1111	Š 143	f 159	» 175	Ï 191	Ł 207	Ł 223	∩ 239	SP 255



	HEX	8	9	A	B	C	D	E	F
HEX	BIN	1000	1001	1010	1011	1100	1101	1110	1111
0	0000	☺ 128	☻ 144	☼ 160	☽ 176	☾ 192	☿ 208	♁ 224	♂ 240
1	0001	ü 129	æ 145	í 161	☽ 177	☾ 193	☿ 209	♁ 225	♂ 241
2	0010	é 130	Æ 146	ó 162	☽ 178	☾ 194	☿ 210	♁ 226	♂ 242
3	0011	ã 131	ø 147	ú 163	☽ 179	☾ 195	☿ 211	♁ 227	♂ 243
4	0100	ä 132	ö 148	ñ 164	☽ 180	☾ 196	☿ 212	♁ 228	♂ 244
5	0101	à 133	ò 149	ñ 165	☽ 181	☾ 197	☿ 213	♁ 229	♂ 245
6	0110	å 134	û 150	ä 166	☽ 182	☾ 198	☿ 214	♁ 230	♂ 246
7	0111	☺ 135	ù 151	Ω 167	☽ 183	☾ 199	☿ 215	♁ 231	♂ 247
8	1000	ē 136	ÿ 152	ó 168	☽ 184	☾ 200	☿ 216	♁ 232	♂ 248
9	1001	ë 137	ö 153	☽ 169	☽ 185	☾ 201	☿ 217	♁ 233	♂ 249
A	1010	è 138	Û 154	☽ 170	☽ 186	☾ 202	☿ 218	♁ 234	♂ 250
B	1011	ï 139	ø 155	☽ 171	☽ 187	☾ 203	☿ 219	♁ 235	♂ 251
C	1100	î 140	£ 156	☽ 172	☽ 188	☾ 204	☿ 220	♁ 236	♂ 252
D	1101	ì 141	ø 157	☽ 173	☽ 189	☾ 205	☿ 221	♁ 237	♂ 253
E	1110	Ä 142	pt 158	« 174	☽ 190	☾ 206	☿ 222	♁ 238	♂ 254
F	1111	Å 143	f 159	☽ 175	☽ 191	☾ 207	☿ 223	♁ 239	♂ 255



	HEX	8	9	A	B	C	D	E	F
HEX	BIN	1000	1001	1010	1011	1100	1101	1110	1111
0	0000	€		NBSP	°	À	Ð	à	ð
1	0001		'	ı	±	Á	Ñ	á	ñ
2	0010	,	´	¢	”	Â	Ò	â	ò
3	0011	/	“	£	”	Ã	Ó	ã	ó
4	0100	”	”	¤	”	Ä	Ô	ä	ô
5	0101	…	•	¥	µ	Å	Õ	å	õ
6	0110	†	-	¦	¶	Æ	Ö	æ	ö
7	0111	‡	—	§	·	Ç	×	ç	+
8	1000	ˆ	ˆ	ˆ	ˆ	È	Ø	è	ø
9	1001	%	™	©	ı	É	Ù	é	ù
A	1010	Š	š	•	•	Ê	Ú	ê	ú
B	1011	‘	’	“	”	Ë	Û	ë	û
C	1100	œ	œ	ˆ	¼	Ì	Ü	ì	ü
D	1101			ˆ	½	Í	Ý	í	ý
E	1110	Ž	ž	®	¾	Î	Þ	î	þ
F	1111		ÿ	—	¿	Ï	ß	ï	ÿ



	HEX	8	9	A	B	C	D	E	F
HEX	BIN	1000	1001	1010	1011	1100	1101	1110	1111
0	0000	А 128	Р 144	а 160	Ѡ 176	Ѳ 192	Ѵ 208	Р 224	Ѹ 240
1	0001	Б 129	С 145	б 161	ѡ 177	ѳ 193	ѵ 209	с 225	ѹ 241
2	0010	В 130	Т 146	в 162	Ѣ 178	Ѵ 194	Ѷ 210	т 226	Ѻ 242
3	0011	Г 131	У 147	г 163	ѣ 179	Ѷ 195	ѷ 211	у 227	ѻ 243
4	0100	Д 132	Ф 148	д 164	Ѥ 180	Ѹ 196	ѹ 212	ф 228	Ѽ 244
5	0101	Е 133	Х 149	е 165	ѥ 181	ѹ 197	Ѻ 213	х 229	ѽ 245
6	0110	Ж 134	Ц 150	ж 166	Ѧ 182	Ѻ 198	ѻ 214	ц 230	Ѿ 246
7	0111	З 135	Ч 151	з 167	ѧ 183	ѻ 199	Ѽ 215	ч 231	ѿ 247
8	1000	И 136	Ш 152	и 168	Ѩ 184	Ѽ 200	ѽ 216	ш 232	ѿ 248
9	1001	Й 137	Щ 153	й 169	ѩ 185	ѽ 201	Ѿ 217	щ 233	ѿ 249
A	1010	К 138	Ъ 154	к 170	Ѫ 186	Ѿ 202	ѿ 218	ъ 234	ѿ 250
B	1011	Л 139	Ы 155	л 171	ѫ 187	ѿ 203	Ѽ 219	ы 235	ѿ 251
C	1100	М 140	Ь 156	м 172	Ѭ 188	ѿ 204	ѽ 220	ь 236	ѿ 252
D	1101	Н 141	Э 157	н 173	ѭ 189	ѿ 205	Ѽ 221	э 237	ѿ 253
E	1110	О 142	Ю 158	о 174	Ѯ 190	ѿ 206	ѽ 222	ю 238	ѿ 254
F	1111	П 143	Я 159	п 175	ѯ 191	ѿ 207	Ѽ 223	я 239	ѿ 255



	HEX	8	9	A	B	C	D	E	F
HEX	BIN	1000	1001	1010	1011	1100	1101	1110	1111
0	0000	Ç 128	É 144	á 160	⌘ 176	⌘ 192	đ 208	Ó 224	␣ 240
1	0001	ü 129	Ł 145	í 161	␣ 177	⌘ 193	Đ 209	ß 225	” 241
2	0010	é 130	í 146	ó 162	␣ 178	⌘ 194	Ď 210	Ô 226	˘ 242
3	0011	ã 131	ô 147	ú 163	⌘ 179	⌘ 195	Ë 211	Ń 227	˘ 243
4	0100	ä 132	ö 148	À 164	⌘ 180	⌘ 196	đ 212	ń 228	˘ 244
5	0101	ù 133	Ł 149	ą 165	Á 181	⌘ 197	Ń 213	ň 229	§ 245
6	0110	ć 134	Ī 150	Ż 166	Â 182	Ă 198	İ 214	Š 230	÷ 246
7	0111	ç 135	Ś 151	ż 167	Ë 183	ā 199	İ 215	š 231	· 247
8	1000	ł 136	ś 152	Ɔ 168	Ş 184	⌘ 200	ž 216	Ŕ 232	° 248
9	1001	ë 137	Ö 153	ę 169	⌘ 185	⌘ 201	⌘ 217	Ú 233	˘ 249
A	1010	Û 138	Û 154	⌘ 170	⌘ 186	⌘ 202	⌘ 218	ı 234	· 250
B	1011	õ 139	Ŧ 155	ź 171	⌘ 187	⌘ 203	␣ 219	Û 235	ú 251
C	1100	ï 140	ı 156	Č 172	⌘ 188	⌘ 204	␣ 220	ý 236	Ř 252
D	1101	Ż 141	Ł 157	š 173	Ž 189	⌘ 205	Ŧ 221	Ý 237	ř 253
E	1110	Ä 142	× 158	« 174	ž 190	⌘ 206	Û 222	ı 238	␣ 254
F	1111	Ć 143	č 159	» 175	⌘ 191	⌘ 207	␣ 223	˘ 239	␣ 255



CONFIDENTIAL

Page 19 (PC858: Euro)

HEX	8	9	A	B	C	D	E	F
HEX BIN	1000	1001	1010	1011	1100	1101	1110	1111
0	0000	Ç	É	á	⌘	Ł	ð	ó
	128	144	160	176	192	208	224	240
1	0001	ü	æ	í	⌘	Ð	ß	±
	129	145	161	177	193	209	225	241
2	0010	é	Æ	ó	⌘	Τ	È	Ô
	130	146	162	178	194	210	226	242
3	0011	â	ô	ú		†	Ë	Ö
	131	147	163	179	195	211	227	243
4	0100	ä	ö	ñ	†	-	È	ö
	132	148	164	180	196	212	228	244
5	0101	à	ò	Ñ	À	+	€	ö
	133	149	165	181	197	213	229	245
6	0110	á	û	â	À	ä	í	μ
	134	150	166	182	198	214	230	246
7	0111	ç	ù	ó	À	À	ï	þ
	135	151	167	183	199	215	231	247
8	1000	ê	ÿ	ô	é	Ł	ÿ	þ
	136	152	168	184	200	216	232	248
9	1001	ë	ö	•	†	ŗ	Ĵ	Ů
	137	153	169	185	201	217	233	249
A	1010	è	Ů	˘		Ł	ŗ	Ů
	138	154	170	186	202	218	234	250
B	1011	ÿ	ø	†	ŗ	Ł	Ů	ı
	139	155	171	187	203	219	235	251
C	1100	ï	£	†	Ĵ	†	Ÿ	ˆ
	140	156	172	188	204	220	236	252
D	1101	ì	ø	ı	φ	-	ı	Ÿ
	141	157	173	189	205	221	237	253
E	1110	À	×	«	¥	+	ı	■
	142	158	174	190	206	222	238	254
F	1111	Á	ƒ	»	ŗ	■	'	SP
	143	159	175	191	207	223	239	255



USING THE CHARACTER CODE TABLES

The example below uses Page 0 (PC437) to illustrate the use of the character code tables.

You can find the character "A" in Page 0 as follows:

The decimal value for the character "A" is 65.

Follow its column straight up to find the digits.

Hexadecimal 4

Binary 0100

These numbers are the most significant bits of the ASCII code.

Follow its row to the left to find the digits.

Hexadecimal 1

Binary 0001

These numbers are the least significant bits of the ASCII code.

The combination of the numbers above is the ASCII code for character "A."

Decimal 65

Hexadecimal 41

Binary 01000001



COMMAND LIST

- **Commands in alphanumeric order**
- **Text command set**
- **Graphics command set**
- **DM-D110 supported commands**
- **DM-D210 supported commands**
- **DM-D500 supported commands**



COMMANDS IN ALPHANUMERIC ORDER

In this table, click any name to see the command description with program examples and display samples.

The display samples are images of the display results of the program examples; they do not represent actual displays.

Command	Name	Function type	Page
BS	Backspace	TEXT COMMAND SET: DISPLAY POSITION COMMANDS	37
HT	Horizontal tab	TEXT COMMAND SET: DISPLAY POSITION COMMANDS	40
LF	Move cursor down	TEXT COMMAND SET: DISPLAY POSITION COMMANDS	43
US LF	Move cursor up	TEXT COMMAND SET: DISPLAY POSITION COMMANDS	46
HOM	Move cursor to home position	TEXT COMMAND SET: DISPLAY POSITION COMMANDS	49
CR	Move cursor to left-most position	TEXT COMMAND SET: DISPLAY POSITION COMMANDS	51
US CR	Move cursor to right-most position	TEXT COMMAND SET: DISPLAY POSITION COMMANDS	53
US B	Move cursor to bottom position	TEXT COMMAND SET: DISPLAY POSITION COMMANDS	55
US \$	Move cursor to the specified position	TEXT COMMAND SET: DISPLAY POSITION COMMANDS	57
CLR	Clear display screen	TEXT COMMAND SET: CHARACTER COMMANDS	60
CAN	Clear cursor line	TEXT COMMAND SET: CHARACTER COMMANDS	62
ESC =	Select peripheral device	TEXT COMMAND SET: MISCELLANEOUS COMMANDS	64
ESC @	Initialize display	TEXT COMMAND SET: MISCELLANEOUS COMMANDS	67
ESC %	Select/cancel user-defined character set	TEXT COMMAND SET: CHARACTER COMMANDS	70



Command	Name	Function type	Page
ESC &	Define user-defined characters	TEXT COMMAND SET: CHARACTER COMMANDS	73
ESC ?	Cancel user-defined characters	TEXT COMMAND SET: CHARACTER COMMANDS	77
ESC R	Select an international character set	TEXT COMMAND SET: CHARACTER COMMANDS	80
ESC t	Select character code table	TEXT COMMAND SET: CHARACTER COMMANDS	84
ESC W	Set/cancel window range	TEXT COMMAND SET: DISPLAY POSITION COMMANDS	88
US MD1	Select overwrite mode	TEXT COMMAND SET: CHARACTER COMMANDS	93
US MD2	Select vertical scroll mode	TEXT COMMAND SET: CHARACTER COMMANDS	95
US MD3	Select horizontal scroll mode	TEXT COMMAND SET: CHARACTER COMMANDS	97
US C	Turn cursor display mode on/off	TEXT COMMAND SET: DISPLAY COMMANDS	99
US E	Sets display screen blank interval	TEXT COMMAND SET: DISPLAY COMMANDS	102
US T	Set and display counter time	TEXT COMMAND SET: MISCELLANEOUS COMMANDS	103
US U	Display time counter	TEXT COMMAND SET: MISCELLANEOUS COMMANDS	105
US X	Set brightness	TEXT COMMAND SET: DISPLAY COMMANDS	108
US r	Select/cancel reverse characters	TEXT COMMAND SET: DISPLAY COMMANDS	109
US v	Set status confirmation for DTR signal	TEXT COMMAND SET: STATUS COMMANDS	110
US @	Execute self test	TEXT COMMAND SET: MISCELLANEOUS COMMANDS	111
US :	Start/end macro definition	TEXT COMMAND SET: MISCELLANEOUS COMMANDS	112
US ^	Execute macro	TEXT COMMAND SET: MISCELLANEOUS COMMANDS	114



CONFIDENTIAL

Command	Name	Function type	Page
US .	Display period	TEXT COMMAND SET: DISPLAY COMMANDS	117
US ,	Display comma	TEXT COMMAND SET: DISPLAY COMMANDS	120
US ;	Display semicolon	TEXT COMMAND SET: DISPLAY COMMANDS	123
US #	Turn annunciator on/off	TEXT COMMAND SET: DISPLAY COMMANDS	126
US (A	Select display(s)	TEXT COMMAND SET: MISCELLANEOUS COMMANDS	129
US (B	Transmit display information	GRAPHICS COMMAND SET: STATUS COMMANDS	130
US (C	Edit NV user memory	GRAPHICS COMMAND SET: MISCELLANEOUS COMMANDS	132
US (D	Select window control	GRAPHICS COMMAND SET: DISPLAY COMMANDS	148
US (E	User setting commands	TEXT COMMAND SET: MISCELLANEOUS COMMANDS	164
US (F	Display bit image	GRAPHICS COMMAND SET: BIT IMAGE COMMANDS	183
US (G	Select character style	GRAPHICS COMMAND SET: CHARACTER COMMANDS	195
US (H	Set display layout	GRAPHICS COMMAND SET: DISPLAY COMMANDS	209



TEXT COMMAND SET

Command	Classification	Name	Function type	Page
BS	EXECUTING COMMAND	Backspace	TEXT COMMAND SET: DISPLAY POSITION COMMANDS	37
HT	EXECUTING COMMAND	Horizontal tab	TEXT COMMAND SET: DISPLAY POSITION COMMANDS	40
LF	EXECUTING COMMAND	Move cursor down	TEXT COMMAND SET: DISPLAY POSITION COMMANDS	43
US LF	EXECUTING COMMAND	Move cursor up	TEXT COMMAND SET: DISPLAY POSITION COMMANDS	46
HOM	EXECUTING COMMAND	Move cursor to home position	TEXT COMMAND SET: DISPLAY POSITION COMMANDS	49
CR	EXECUTING COMMAND	Move cursor to left-most position	TEXT COMMAND SET: DISPLAY POSITION COMMANDS	51
US CR	EXECUTING COMMAND	Move cursor to right-most position	TEXT COMMAND SET: DISPLAY POSITION COMMANDS	53
US B	EXECUTING COMMAND	Move cursor to bottom position	TEXT COMMAND SET: DISPLAY POSITION COMMANDS	55
US \$	EXECUTING COMMAND	Move cursor to the specified position	TEXT COMMAND SET: DISPLAY POSITION COMMANDS	57
CLR	EXECUTING COMMAND	Clear display screen	TEXT COMMAND SET: CHARACTER COMMANDS	60
CAN	EXECUTING COMMAND	Clear cursor line	TEXT COMMAND SET: CHARACTER COMMANDS	62
ESC =	SETTING COMMAND	Select peripheral device	TEXT COMMAND SET: MISCELLANEOUS COMMANDS	64
ESC @	EXECUTING COMMAND SETTING COMMAND	Initialize display	TEXT COMMAND SET: MISCELLANEOUS COMMANDS	67
ESC %	SETTING COMMAND	Select/cancel user-defined character set	TEXT COMMAND SET: CHARACTER COMMANDS	70
ESC &	SETTING COMMAND	Define user-defined characters	TEXT COMMAND SET: CHARACTER COMMANDS	73
ESC ?	SETTING COMMAND	Cancel user-defined characters	TEXT COMMAND SET: CHARACTER COMMANDS	77



Command	Classification	Name	Function type	Page
ESC R	SETTING COMMAND	Select an international character set	TEXT COMMAND SET: CHARACTER COMMANDS	80
ESC t	SETTING COMMAND	Select character code table	TEXT COMMAND SET: CHARACTER COMMANDS	84
ESC W	SETTING COMMAND	Set/cancel window range	TEXT COMMAND SET: DISPLAY POSITION COMMANDS	88
US MD1	EXECUTING COMMAND	Select overwrite mode	TEXT COMMAND SET: CHARACTER COMMANDS	93
US MD2	EXECUTING COMMAND	Select vertical scroll mode	TEXT COMMAND SET: CHARACTER COMMANDS	95
US MD3	EXECUTING COMMAND	Select horizontal scroll mode	TEXT COMMAND SET: CHARACTER COMMANDS	97
US C	EXECUTING COMMAND	Turn cursor display mode on/off	TEXT COMMAND SET: DISPLAY COMMANDS	99
US E	EXECUTING COMMAND	Sets display screen blank interval	TEXT COMMAND SET: DISPLAY COMMANDS	102
US T	EXECUTING COMMAND	Set and display counter time	TEXT COMMAND SET: MISCELLANEOUS COMMANDS	103
US U	EXECUTING COMMAND	Display time counter	TEXT COMMAND SET: MISCELLANEOUS COMMANDS	105
US X	EXECUTING COMMAND	Set brightness	TEXT COMMAND SET: DISPLAY COMMANDS	108
US r	EXECUTING COMMAND	Select/cancel reverse characters	TEXT COMMAND SET: DISPLAY COMMANDS	109
US v	EXECUTING COMMAND	Set status confirmation for DTR signal	TEXT COMMAND SET: STATUS COMMANDS	110
US @	EXECUTING COMMAND	Execute self test	TEXT COMMAND SET: MISCELLANEOUS COMMANDS	111
US :	EXECUTING COMMAND	Start/end macro definition	TEXT COMMAND SET: MISCELLANEOUS COMMANDS	112
US ^	EXECUTING COMMAND	Execute macro	TEXT COMMAND SET: MISCELLANEOUS COMMANDS	114
US .	EXECUTING COMMAND	Display period	TEXT COMMAND SET: DISPLAY COMMANDS	117
US ,	EXECUTING COMMAND	Display comma	TEXT COMMAND SET: DISPLAY COMMANDS	120



CONFIDENTIAL

Command	Classification	Name	Function type	Page
US ;	EXECUTING COMMAND	Display semicolon	TEXT COMMAND SET: DISPLAY COMMANDS	123
US #	EXECUTING COMMAND	Turn annunciator on/off	TEXT COMMAND SET: DISPLAY COMMANDS	126
US (A	EXECUTING COMMAND	Select display(s)	TEXT COMMAND SET: MISCELLANEOUS COMMANDS	129
US (E	EXECUTING COMMAND	User setting commands	TEXT COMMAND SET: MISCELLANEOUS COMMANDS	164



GRAPHICS COMMAND SET

Command	Classification	Name	Function type	Page
US (B	EXECUTING COMMAND	Transmit display information	GRAPHICS COMMAND SET: STATUS COMMANDS	130
US (C	EXECUTING COMMAND	Edit NV user memory	GRAPHICS COMMAND SET: MISCELLANEOUS COMMANDS	132
US (D	EXECUTING COMMAND	Select window control	GRAPHICS COMMAND SET: DISPLAY COMMANDS	148
US (F	EXECUTING COMMAND	Display bit image	GRAPHICS COMMAND SET: BIT IMAGE COMMANDS	183
US (G	EXECUTING COMMAND	Select character style	GRAPHICS COMMAND SET: CHARACTER COMMANDS	195
US (H	EXECUTING COMMAND	Set display layout	GRAPHICS COMMAND SET: DISPLAY COMMANDS	209



DM-D110 SUPPORTED COMMANDS

Command	Classification	Name	Function type	Page
BS	EXECUTING COMMAND	Backspace	TEXT COMMAND SET: DISPLAY POSITION COMMANDS	37
HT	EXECUTING COMMAND	Horizontal tab	TEXT COMMAND SET: DISPLAY POSITION COMMANDS	40
LF	EXECUTING COMMAND	Move cursor down	TEXT COMMAND SET: DISPLAY POSITION COMMANDS	43
US LF	EXECUTING COMMAND	Move cursor up	TEXT COMMAND SET: DISPLAY POSITION COMMANDS	46
HOM	EXECUTING COMMAND	Move cursor to home position	TEXT COMMAND SET: DISPLAY POSITION COMMANDS	49
CR	EXECUTING COMMAND	Move cursor to left-most position	TEXT COMMAND SET: DISPLAY POSITION COMMANDS	51
US CR	EXECUTING COMMAND	Move cursor to right-most position	TEXT COMMAND SET: DISPLAY POSITION COMMANDS	53
US B	EXECUTING COMMAND	Move cursor to bottom position	TEXT COMMAND SET: DISPLAY POSITION COMMANDS	55
US \$	EXECUTING COMMAND	Move cursor to the specified position	TEXT COMMAND SET: DISPLAY POSITION COMMANDS	57
CLR	EXECUTING COMMAND	Clear display screen	TEXT COMMAND SET: CHARACTER COMMANDS	60
CAN	EXECUTING COMMAND	Clear cursor line	TEXT COMMAND SET: CHARACTER COMMANDS	62
ESC =	SETTING COMMAND	Select peripheral device	TEXT COMMAND SET: MISCELLANEOUS COMMANDS	64
ESC @	EXECUTING COMMAND SETTING COMMAND	Initialize display	TEXT COMMAND SET: MISCELLANEOUS COMMANDS	67
ESC %	SETTING COMMAND	Select/cancel user-defined character set	TEXT COMMAND SET: CHARACTER COMMANDS	70
ESC &	SETTING COMMAND	Define user-defined characters	TEXT COMMAND SET: CHARACTER COMMANDS	73
ESC ?	SETTING COMMAND	Cancel user-defined characters	TEXT COMMAND SET: CHARACTER COMMANDS	77



CONFIDENTIAL

Command	Classification	Name	Function type	Page
ESC R	SETTING COMMAND	Select an international character set	TEXT COMMAND SET: CHARACTER COMMANDS	80
ESC t	SETTING COMMAND	Select character code table	TEXT COMMAND SET: CHARACTER COMMANDS	84
ESC W	SETTING COMMAND	Set/cancel window range	TEXT COMMAND SET: DISPLAY POSITION COMMANDS	88
US MD1	EXECUTING COMMAND	Select overwrite mode	TEXT COMMAND SET: CHARACTER COMMANDS	93
US MD2	EXECUTING COMMAND	Select vertical scroll mode	TEXT COMMAND SET: CHARACTER COMMANDS	95
US MD3	EXECUTING COMMAND	Select horizontal scroll mode	TEXT COMMAND SET: CHARACTER COMMANDS	97
US C	EXECUTING COMMAND	Turn cursor display mode on/off	TEXT COMMAND SET: DISPLAY COMMANDS	99
US E	EXECUTING COMMAND	Sets display screen blank interval	TEXT COMMAND SET: DISPLAY COMMANDS	102
US T	EXECUTING COMMAND	Set and display counter time	TEXT COMMAND SET: MISCELLANEOUS COMMANDS	103
US U	EXECUTING COMMAND	Display time counter	TEXT COMMAND SET: MISCELLANEOUS COMMANDS	105
US X	EXECUTING COMMAND	Set brightness	TEXT COMMAND SET: DISPLAY COMMANDS	108
US r	EXECUTING COMMAND	Select/cancel reverse characters	TEXT COMMAND SET: DISPLAY COMMANDS	109
US v	EXECUTING COMMAND	Set status confirmation for DTR signal	TEXT COMMAND SET: STATUS COMMANDS	110
US @	EXECUTING COMMAND	Execute self test	TEXT COMMAND SET: MISCELLANEOUS COMMANDS	111
US :	EXECUTING COMMAND	Start/end macro definition	TEXT COMMAND SET: MISCELLANEOUS COMMANDS	112
US ^	EXECUTING COMMAND	Execute macro	TEXT COMMAND SET: MISCELLANEOUS COMMANDS	114
US (A	EXECUTING COMMAND	Select display(s)	TEXT COMMAND SET: MISCELLANEOUS COMMANDS	129
US (E	EXECUTING COMMAND	User setting commands	TEXT COMMAND SET: MISCELLANEOUS COMMANDS	164



DM-D210 SUPPORTED COMMANDS

Command	Classification	Name	Function type	Page
BS	EXECUTING COMMAND	Backspace	TEXT COMMAND SET: DISPLAY POSITION COMMANDS	37
HT	EXECUTING COMMAND	Horizontal tab	TEXT COMMAND SET: DISPLAY POSITION COMMANDS	40
LF	EXECUTING COMMAND	Move cursor down	TEXT COMMAND SET: DISPLAY POSITION COMMANDS	43
US LF	EXECUTING COMMAND	Move cursor up	TEXT COMMAND SET: DISPLAY POSITION COMMANDS	46
HOM	EXECUTING COMMAND	Move cursor to home position	TEXT COMMAND SET: DISPLAY POSITION COMMANDS	49
CR	EXECUTING COMMAND	Move cursor to left-most position	TEXT COMMAND SET: DISPLAY POSITION COMMANDS	51
US CR	EXECUTING COMMAND	Move cursor to right-most position	TEXT COMMAND SET: DISPLAY POSITION COMMANDS	53
US B	EXECUTING COMMAND	Move cursor to bottom position	TEXT COMMAND SET: DISPLAY POSITION COMMANDS	55
US \$	EXECUTING COMMAND	Move cursor to the specified position	TEXT COMMAND SET: DISPLAY POSITION COMMANDS	57
CLR	EXECUTING COMMAND	Clear display screen	TEXT COMMAND SET: CHARACTER COMMANDS	60
CAN	EXECUTING COMMAND	Clear cursor line	TEXT COMMAND SET: CHARACTER COMMANDS	62
ESC =	SETTING COMMAND	Select peripheral device	TEXT COMMAND SET: MISCELLANEOUS COMMANDS	64
ESC @	EXECUTING COMMAND SETTING COMMAND	Initialize display	TEXT COMMAND SET: MISCELLANEOUS COMMANDS	67
ESC %	SETTING COMMAND	Select/cancel user-defined character set	TEXT COMMAND SET: CHARACTER COMMANDS	70
ESC &	SETTING COMMAND	Define user-defined characters	TEXT COMMAND SET: CHARACTER COMMANDS	73
ESC ?	SETTING COMMAND	Cancel user-defined characters	TEXT COMMAND SET: CHARACTER COMMANDS	77



Command	Classification	Name	Function type	Page
ESC R	SETTING COMMAND	Select an international character set	TEXT COMMAND SET: CHARACTER COMMANDS	80
ESC t	SETTING COMMAND	Select character code table	TEXT COMMAND SET: CHARACTER COMMANDS	84
ESC W	SETTING COMMAND	Set/cancel window range	TEXT COMMAND SET: DISPLAY POSITION COMMANDS	88
US MD1	EXECUTING COMMAND	Select overwrite mode	TEXT COMMAND SET: CHARACTER COMMANDS	93
US MD2	EXECUTING COMMAND	Select vertical scroll mode	TEXT COMMAND SET: CHARACTER COMMANDS	95
US MD3	EXECUTING COMMAND	Select horizontal scroll mode	TEXT COMMAND SET: CHARACTER COMMANDS	97
US E	EXECUTING COMMAND	Sets display screen blank interval	TEXT COMMAND SET: DISPLAY COMMANDS	102
US T	EXECUTING COMMAND	Set and display counter time	TEXT COMMAND SET: MISCELLANEOUS COMMANDS	103
US U	EXECUTING COMMAND	Display time counter	TEXT COMMAND SET: MISCELLANEOUS COMMANDS	105
US X	EXECUTING COMMAND	Set brightness	TEXT COMMAND SET: DISPLAY COMMANDS	108
US r	EXECUTING COMMAND	Select/cancel reverse characters	TEXT COMMAND SET: DISPLAY COMMANDS	109
US v	EXECUTING COMMAND	Set status confirmation for DTR signal	TEXT COMMAND SET: STATUS COMMANDS	110
US @	EXECUTING COMMAND	Execute self test	TEXT COMMAND SET: MISCELLANEOUS COMMANDS	111
US :	EXECUTING COMMAND	Start/end macro definition	TEXT COMMAND SET: MISCELLANEOUS COMMANDS	112
US ^	EXECUTING COMMAND	Execute macro	TEXT COMMAND SET: MISCELLANEOUS COMMANDS	114
US .	EXECUTING COMMAND	Display period	TEXT COMMAND SET: DISPLAY COMMANDS	117
US ,	EXECUTING COMMAND	Display comma	TEXT COMMAND SET: DISPLAY COMMANDS	120
US ;	EXECUTING COMMAND	Display semicolon	TEXT COMMAND SET: DISPLAY COMMANDS	123



CONFIDENTIAL

Command	Classification	Name	Function type	Page
US #	EXECUTING COMMAND	Turn annunciator on/off	TEXT COMMAND SET: DISPLAY COMMANDS	126
US (A	EXECUTING COMMAND	Select display(s)	TEXT COMMAND SET: MISCELLANEOUS COMMANDS	129
US (E	EXECUTING COMMAND	User setting commands	TEXT COMMAND SET: MISCELLANEOUS COMMANDS	164



DM-D500 SUPPORTED COMMANDS

Command	Classification	Name	Function type	Page
BS	EXECUTING COMMAND	Backspace	TEXT COMMAND SET: DISPLAY POSITION COMMANDS	37
HT	EXECUTING COMMAND	Horizontal tab	TEXT COMMAND SET: DISPLAY POSITION COMMANDS	40
LF	EXECUTING COMMAND	Move cursor down	TEXT COMMAND SET: DISPLAY POSITION COMMANDS	43
US LF	EXECUTING COMMAND	Move cursor up	TEXT COMMAND SET: DISPLAY POSITION COMMANDS	46
HOM	EXECUTING COMMAND	Move cursor to home position	TEXT COMMAND SET: DISPLAY POSITION COMMANDS	49
CR	EXECUTING COMMAND	Move cursor to left-most position	TEXT COMMAND SET: DISPLAY POSITION COMMANDS	51
US CR	EXECUTING COMMAND	Move cursor to right-most position	TEXT COMMAND SET: DISPLAY POSITION COMMANDS	53
US B	EXECUTING COMMAND	Move cursor to bottom position	TEXT COMMAND SET: DISPLAY POSITION COMMANDS	55
US \$	EXECUTING COMMAND	Move cursor to the specified position	TEXT COMMAND SET: DISPLAY POSITION COMMANDS	57
CLR	EXECUTING COMMAND	Clear display screen	TEXT COMMAND SET: CHARACTER COMMANDS	60
CAN	EXECUTING COMMAND	Clear cursor line	TEXT COMMAND SET: CHARACTER COMMANDS	62
ESC =	SETTING COMMAND	Select peripheral device	TEXT COMMAND SET: MISCELLANEOUS COMMANDS	64
ESC @	EXECUTING COMMAND SETTING COMMAND	Initialize display	TEXT COMMAND SET: MISCELLANEOUS COMMANDS	67
ESC %	SETTING COMMAND	Select/cancel user-defined character set	TEXT COMMAND SET: CHARACTER COMMANDS	70
ESC &	SETTING COMMAND	Define user-defined characters	TEXT COMMAND SET: CHARACTER COMMANDS	73
ESC ?	SETTING COMMAND	Cancel user-defined characters	TEXT COMMAND SET: CHARACTER COMMANDS	77



Command	Classification	Name	Function type	Page
ESC R	SETTING COMMAND	Select an international character set	TEXT COMMAND SET: CHARACTER COMMANDS	80
ESC t	SETTING COMMAND	Select character code table	TEXT COMMAND SET: CHARACTER COMMANDS	84
ESC W	SETTING COMMAND	Set/cancel window range	TEXT COMMAND SET: DISPLAY POSITION COMMANDS	88
US MD1	EXECUTING COMMAND	Select overwrite mode	TEXT COMMAND SET: CHARACTER COMMANDS	93
US MD2	EXECUTING COMMAND	Select vertical scroll mode	TEXT COMMAND SET: CHARACTER COMMANDS	95
US MD3	EXECUTING COMMAND	Select horizontal scroll mode	TEXT COMMAND SET: CHARACTER COMMANDS	97
US C	EXECUTING COMMAND	Turn cursor display mode on/off	TEXT COMMAND SET: DISPLAY COMMANDS	99
US E	EXECUTING COMMAND	Sets display screen blank interval	TEXT COMMAND SET: DISPLAY COMMANDS	102
US T	EXECUTING COMMAND	Set and display counter time	TEXT COMMAND SET: MISCELLANEOUS COMMANDS	103
US U	EXECUTING COMMAND	Display time counter	TEXT COMMAND SET: MISCELLANEOUS COMMANDS	105
US X	EXECUTING COMMAND	Set brightness	TEXT COMMAND SET: DISPLAY COMMANDS	108
US r	EXECUTING COMMAND	Select/cancel reverse characters	TEXT COMMAND SET: DISPLAY COMMANDS	109
US v	EXECUTING COMMAND	Set status confirmation for DTR signal	TEXT COMMAND SET: STATUS COMMANDS	110
US @	EXECUTING COMMAND	Execute self test	TEXT COMMAND SET: MISCELLANEOUS COMMANDS	111
US :	EXECUTING COMMAND	Start/end macro definition	TEXT COMMAND SET: MISCELLANEOUS COMMANDS	112
US ^	EXECUTING COMMAND	Execute macro	TEXT COMMAND SET: MISCELLANEOUS COMMANDS	114
US .	EXECUTING COMMAND	Display period	TEXT COMMAND SET: DISPLAY COMMANDS	117
US ,	EXECUTING COMMAND	Display comma	TEXT COMMAND SET: DISPLAY COMMANDS	120



CONFIDENTIAL

Command	Classification	Name	Function type	Page
US ;	EXECUTING COMMAND	Display semicolon	TEXT COMMAND SET: DISPLAY COMMANDS	123
US #	EXECUTING COMMAND	Turn annunciator on/off	TEXT COMMAND SET: DISPLAY COMMANDS	126
US (A	EXECUTING COMMAND	Select display(s)	TEXT COMMAND SET: MISCELLANEOUS COMMANDS	129
US (B	EXECUTING COMMAND	Transmit display information	GRAPHICS COMMAND SET: STATUS COMMANDS	130
US (C	EXECUTING COMMAND	Edit NV user memory	GRAPHICS COMMAND SET: MISCELLANEOUS COMMANDS	132
US (D	EXECUTING COMMAND	Select window control	GRAPHICS COMMAND SET: DISPLAY COMMANDS	148
US (E	EXECUTING COMMAND	User setting commands	TEXT COMMAND SET: MISCELLANEOUS COMMANDS	164
US (F	EXECUTING COMMAND	Display bit image	GRAPHICS COMMAND SET: BIT IMAGE COMMANDS	183
US (G	EXECUTING COMMAND	Select character style	GRAPHICS COMMAND SET: CHARACTER COMMANDS	195
US (H	EXECUTING COMMAND	Set display layout	GRAPHICS COMMAND SET: DISPLAY COMMANDS	209



BS

EXECUTING COMMAND

[Name]	Backspace	
[Format]	ASCII	BS
	Hex	08
	Decimal	8


[Displays featuring this command] **DM-D110**, **DM-D210**, and **DM-D500**.

[Description] Moves the cursor one character position to the left.





- [Notes]
- When a window is defined, the cursor moves only within the current window.
 - When the cursor is at the left end of a line, the operation of this command depends on the display mode (overwrite mode, vertical scroll mode, or horizontal scroll mode).
 - Overwrite mode: When the cursor is at the left end of the lower line, it is moved to the right end of the upper line. When it is at the left end of the upper line, it is moved to the right end of the lower line.
 - Vertical scroll mode: When the cursor is at the left end of a lower line, it is moved to the right end of the upper line. When it is at the left end of the upper line, the display on the upper line is scrolled to the lower line, and the upper line is cleared. At this time, the cursor is moved to the right end of the upper line.
 - Horizontal scroll mode: All characters in the current line are scrolled one character to the right. The cursor is not moved, but the character area at the left end is cleared.

Example	Display Sample
Normal	
Before BS is sent	
Send BS command	PRINT #1, CHR\$(8);








Overwrite Mode	
Vertical Scroll Mode	
Horizontal Scroll Mode	

Each scroll mode

Before BS is sent	
Send BS command	PRINT #1, CHR\$(8);
Overwrite Mode	
Vertical Scroll Mode	
Horizontal Scroll Mode	



Programming Example	
<pre>PRINT #1, CHR\$(27); PRINT #1, "@";</pre>	
<pre>PRINT #1, CHR\$(31); "(D"; PRINT #1, CHR\$(4); CHR\$(0); PRINT #1, CHR\$(3); CHR\$(97); PRINT #1, CHR\$(1); CHR\$(2);</pre>	
<pre>PRINT #1, "EPSON";</pre>	
<pre>PRINT #1, CHR\$(8);</pre>	
<pre>PRINT #1, "DM-D500";</pre>	

[References]

US MD1, US MD2, US MD3, ESC W



HT

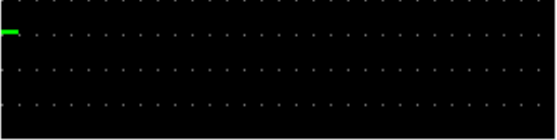
EXECUTING COMMAND

[Name] Horizontal tab
 [Format] ASCII HT
 Hex 09
 Decimal 9


[Displays featuring this command] **DM-D110**, **DM-D210**, and **DM-D500**.

[Description] Moves the cursor one character position to the right.





- [Notes]
- When a window is defined, the cursor moves only within the current window.
 - When the cursor is at the right end of a line, the operation of this command depends on the display mode (overwrite mode, vertical scroll mode, or horizontal scroll mode).
 - Overwrite mode: When the cursor is at the right end of the upper line, it is moved to the left end of the lower line. When it is at the right end of the lower line, it is moved to the left end of the upper line.
 - Vertical scroll mode: When the cursor is at the right end of a upper line, it is moved to the left end of the lower line. When it is at the right end of the lower line, the display on the lower line is scrolled to the upper line, and the lower line is cleared. At this time, the cursor is moved to the left end of the lower line.
 - Horizontal scroll mode: All characters in the current line are scrolled one character to the left. The cursor is not moved, but the character area at the left end is cleared.

Example	Display Sample
Normal	
Before HT is sent	
Send HT command	PRINT #1, CHR\$(9);








Overwrite Mode	
Vertical Scroll Mode	
Horizontal Scroll Mode	

Each scroll mode

Before HT is sent	
Send HT command	PRINT #1, CHR\$(9);
Overwrite Mode	
Vertical Scroll Mode	
Horizontal Scroll Mode	



Programming Example	
<pre>PRINT #1, CHR\$(27); PRINT #1, "@";</pre>	
<pre>PRINT #1, CHR\$(31); "(D"; PRINT #1, CHR\$(4); CHR\$(0); PRINT #1, CHR\$(3); CHR\$(97); PRINT #1, CHR\$(1); CHR\$(2);</pre>	
<pre>PRINT #1, "EPSON";</pre>	
<pre>PRINT #1, CHR\$(9);</pre>	
<pre>PRINT #1, "DM-D500";</pre>	

[References]

US MD1, US MD2, US MD3, ESC W



LF

EXECUTING COMMAND

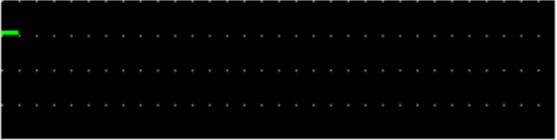
[Name] Move cursor down

[Format] ASCII LF
Hex 0A
Decimal 10


[Displays featuring this command] **DM-D110**, **DM-D210**, and **DM-D500**.

[Description] Moves the cursor down one line





- [Notes]
- When a window is defined, the cursor moves only within the current window.
 - When the cursor is on the lower line, the operation of this command depends on the display mode (overwrite mode, vertical scroll mode, or horizontal scroll mode).
 - Overwrite mode: The cursor moves to the same column on the upper line.
 - Vertical scroll mode: The characters displayed on the lower line are scrolled to the upper line, and the lower line is cleared. The cursor remains in the same position.
 - Horizontal scroll mode: The cursor does not move.

Example	Display Sample
Normal	
Before LF is sent	
Send LF command	PRINT #1, CHR\$(10);





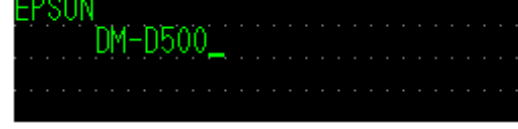


Overwrite Mode	
Vertical Scroll Mode	
Horizontal Scroll Mode	

Each scroll mode

Before LF is sent	
Send LF command	PRINT #1, CHR\$(10);
Overwrite Mode	
Vertical Scroll Mode	
Horizontal Scroll Mode	



Programming Example	
<pre>PRINT #1, CHR\$(27); PRINT #1, "@";</pre>	
<pre>PRINT #1, CHR\$(31); "(D"; PRINT #1, CHR\$(4); CHR\$(0); PRINT #1, CHR\$(3); CHR\$(97); PRINT #1, CHR\$(1); CHR\$(2);</pre>	
<pre>PRINT #1, "EPSON";</pre>	
<pre>PRINT #1, CHR\$(10);</pre>	
<pre>PRINT #1, "DM-D500";</pre>	

[References]

US MD1, US MD2, US MD3, ESC W



US LF


EXECUTING COMMAND

[Name]	Move cursor up		
[Format]	ASCII	US	LF
	Hex	1F	0A
	Decimal	31	10


[Displays featuring this command] **DM-D110**, **DM-D210**, and **DM-D500**.

[Description] Moves the cursor up one line.





- [Notes]
- When a window is defined, the cursor moves only within the current window.
 - When the cursor is on the upper line, the operation of this command depends on the display mode (overwrite mode, vertical scroll mode, or horizontal scroll mode).
 - Overwrite mode: The cursor moves to the same column on the lower line.
 - Vertical scroll mode: The characters displayed on the upper line are scrolled to the lower line, and the upper line is cleared. The cursor remains in the same position.
 - Horizontal scroll mode: The cursor does not move.

Example	Display Sample
Normal	
Before US LF is sent	
Send US LF command	PRINT #1, CHR\$(31); CHR\$(10);








Overwrite Mode	
Vertical Scroll Mode	
Horizontal Scroll Mode	

Each scroll mode

Before US LF is sent	
Send US LF command	PRINT #1, CHR\$(31); CHR\$(10);
Overwrite Mode	
Vertical Scroll Mode	
Horizontal Scroll Mode	



Programming Example	
<pre>PRINT #1, CHR\$(27); PRINT #1, "@";</pre>	
<pre>PRINT #1, CHR\$(31); "(D"; PRINT #1, CHR\$(4); CHR\$(0); PRINT #1, CHR\$(3); CHR\$(97); PRINT #1, CHR\$(1); CHR\$(2);</pre>	
<pre>PRINT #1, "EPSON";</pre>	
<pre>PRINT #1, CHR\$(31); CHR\$(10);</pre>	
<pre>PRINT #1, "DM-D500";</pre>	

[References]

US MD1, US MD2, US MD3, ESC W



HOM

EXECUTING COMMAND



[Name] Move cursor to home position

[Format] ASCII HOM
Hex 0B
Decimal 11

[Displays featuring this command] **DM-D110**, **DM-D210**, and **DM-D500**.

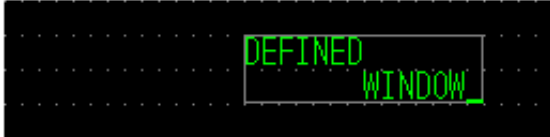

[Description] Moves the cursor to the left-most position on the upper line (home position).

- [Notes]
- The home position indicates the first column of the upper line.
 - When a window is defined, the home position is the upper left corner of the window.

Example	Display Sample
Normal	
Before HOM is sent	
Send HOM command	PRINT #1, CHR\$(11);
Overwrite Mode	
Vertical Scroll Mode	
Horizontal Scroll Mode	



Window condition

<p>Before HOM is sent</p>	
<p>Send HOM command</p>	<p>PRINT #1, CHR\$(11);</p>
<p>Moves the cursor to the home position of the window</p>	

[References]

BS, HT, LF, US LF, CR, US CR, US B, US \$, ESC W



CR

EXECUTING COMMAND



[Name] Move cursor to left-most position

[Format] ASCII CR
 Hex 0D
 Decimal 13

[Displays featuring this command] **DM-D110**, **DM-D210**, and **DM-D500**.



[Description] Moves the cursor to the left-most position on the current line.

[Note] ■ The cursor moves only within the current window.

Example	Display Sample
Normal	
Before CR is sent	
Send CR command	PRINT #1, CHR\$(13);
Overwrite Mode	
Vertical Scroll Mode	
Horizontal Scroll Mode	



Window condition

Before CR is sent	
Send CR command	PRINT #1, CHR\$(13);
Moves the cursor to the left-most position on the current line of window	

[References]

BS, HT, LF, US LF, HOM, US CR, US B, US \$, ESC W



US CR

EXECUTING COMMAND



[Name] Move cursor to right-most position

[Format] ASCII US CR
 Hex 1F 0D
 Decimal 31 13

[Displays featuring this command] **DM-D110**, **DM-D210**, and **DM-D500**.



[Description] Moves the cursor to the right-most position on the current line.

[Note] ■ The cursor is moved only within the current window.

Example	Display Sample
Normal	
Before US CR is sent	
Send US CR command	PRINT #1, CHR\$(31); CHR\$(13);
Overwrite Mode	
Vertical Scroll Mode	
Horizontal Scroll Mode	



Window condition

Before US CR is sent	
Send US CR command	PRINT #1, CHR\$(31); CHR\$(13);
Moves the cursor to the right-most position on the current line of window	

[References]

BS, HT, LF, US LF, HOM, CR, US B, US \$, ESC W



US B

EXECUTING COMMAND

[Name] Move cursor to bottom position

[Format] ASCII US B
 Hex 1F 42
 Decimal 31 66


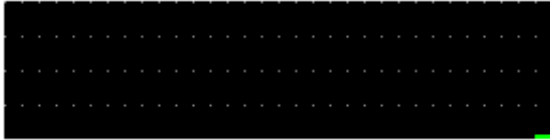
[Displays featuring this command] **DM-D110**, **DM-D210**, and **DM-D500**.

[Description] Moves the cursor to the bottom position.

- [Notes]
- When a window is defined, the bottom position is the lower right corner of the window.
 - The cursor moves only within the current window.

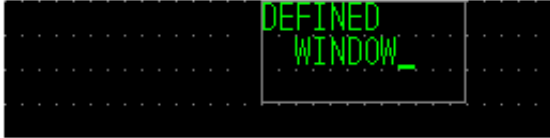
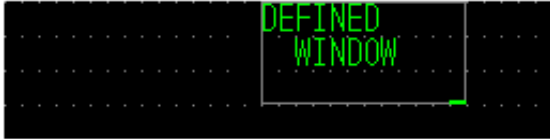
[Model-dependent variations]

For the **DM-D110** and the **DM-D210** (and the **DM-D500** in 20-column/2-line mode), the bottom position indicates the 20th column of the lower line.

Example	Display Sample
Normal	
Before US B is sent	
Send US B command	PRINT #1, CHR\$(31); CHR\$(66);
Overwrite Mode	
Vertical Scroll Mode	
Horizontal Scroll Mode	



Window condition

Before US B is sent	
Send US B command	PRINT #1, CHR\$(31); CHR\$(66);
Moves the cursor to the bottom position of window	

[References]

BS, HT, LF, US LF, HOM, CR, US CR, US \$, ESC W



US \$

EXECUTING COMMAND

[Name] Move cursor to the specified position

[Format]	ASCII	US	\$	<i>n</i>	<i>m</i>
	Hex	1F	24	<i>n</i>	<i>m</i>
	Decimal	31	36	<i>n</i>	<i>m</i>

[Range] **DM-D110** and **DM-D210**

$1 \leq n \leq 20$

m = 1 or 2

DM-D500:

256 × 64 mode:

$1 \leq n \leq$ (The maximum value varies, depending on the size of the window.)

m ≤ (The maximum value varies, depending on the size of the window.)

20-column × 2-line mode:

$1 \leq n \leq 20$

m = 1 or 2

[Displays featuring this command] **DM-D110**, **DM-D210**, and **DM-D500**.

[Description] Moves the cursor to the *n*th position on the *m*th line.





[Note] ■ If the command specifies a value for *n* or *m* that exceeds the range, this command is ignored, and the cursor is not moved.

[Model-dependent variations] **DM-D500**

DM-D500




When the **DM-D500** is in 256 × 64 mode, the size of the window can vary.



Example	Range for n	Range for m	PRINT #1, CHR\$(31); "\$"; CHR\$(10); CHR\$(2);
Each graphic mode			
256 × 64 dots (font A)	$1 \leq n \leq 32$	$1 \leq m \leq 4$	
256 × 64 dots (font B)	$1 \leq n \leq 42$	$1 \leq m \leq 8$	
32 columns × 4 lines	$1 \leq n \leq 32$	$1 \leq m \leq 4$	
42 columns × 8 lines	$1 \leq n \leq 42$	$1 \leq m \leq 8$	



CONFIDENTIAL

32 columns × 3 lines	$1 \leq n \leq 32$	$1 \leq m \leq 4$	
32 columns × 2 lines	$1 \leq n \leq 32$	$m = 1, 2$	
20 columns × 2 lines	$1 \leq n \leq 20$	$m = 1, 2$	

[References]

BS, HT, LF, US LF, HOM, CR, US CR, US \$, ESC W



CLR

EXECUTING COMMAND

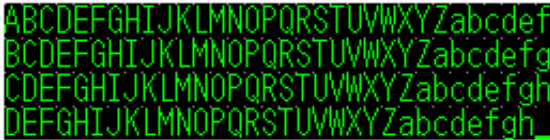

[Name] Clear display screen

[Format] ASCII CLR
Hex 0C
Decimal 12

[Displays featuring this command] **DM-D110**, **DM-D210**, and **DM-D500**.

[Description] Clears all displayed characters.



- [Notes]
- After this command is executed, the cursor moves to the home position.
 - If a window is defined, the cursor is moved only within the current window.

Example	Display Sample
Normal	
Before CLR is sent	
Send CLR command	PRINT #1, CHR\$(12);
Overwrite Mode	
Vertical Scroll Mode	
Horizontal Scroll Mode	



CONFIDENTIAL

Window condition

Before CLR is sent	
Send CLR command	PRINT #1, CHR\$(12);
Clears current window, and the cursor moves to the home position of window	

[References]

CAN, ESC W



CAN

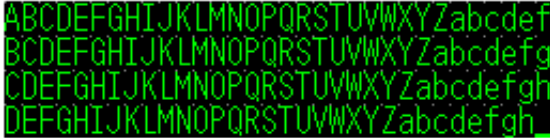
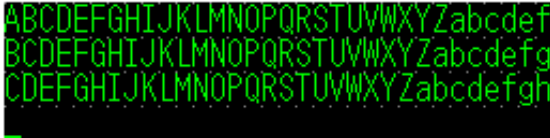
EXECUTING COMMAND

[Name] Clear cursor line
 [Format] ASCII CAN
 Hex 18
 Decimal 24

[Displays featuring this command] **DM-D110**, **DM-D210**, and **DM-D500**.

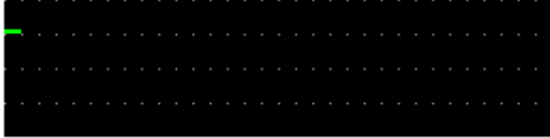
[Description] Clears the line containing the cursor.

- [Notes]
- After this command is executed, the cursor moves to the left-most position on the current line.
 - If a window is defined, the cursor moves only within the current window.

Example	Display Sample
Normal	
Before CAN is sent	
Send CAN command	PRINT #1, CHR\$(24);
Overwrite Mode	
Vertical Scroll Mode	
Horizontal Scroll Mode	



CONFIDENTIAL

Window condition	
Graphic (256 × 64) mode	
Before CAN is sent	<pre>ABCDEFGHIJKLMN OPQRSTUVWXYZabcdef BCDEFGHIJKLMN ODEFINED defg CDEFGHIJKLMN OP WINDOW_efgh DEFGHIJKLMN OPQRSTUVWXYZabcdefgh</pre>
Send CAN command	PRINT #1, CHR\$(24);
Clears current window, and the cursor moves to the home position of window	<pre>ABCDEFGHIJKLMN OPQRSTUVWXYZabcdef BCDEFGHIJKLMN ODEFINED defg CDEFGHIJKLMN OP_ efgh DEFGHIJKLMN OPQRSTUVWXYZabcdefgh</pre>

[References]

CLR, ESC W



ESC =

SETTING COMMAND

[Name] Select peripheral device

[Format] ASCII ESC = *n*
 Hex 1B 3D *n*
 Decimal 27 61 *n*

[Range] 1 ≤ *n* ≤ 3

[Default] **DM-D110, DM-D210:** *n* = 2.

DM-D500: When DIP switch 2-4 is OFF, *n* = 2.
 When DIP switch 2-4 is ON, *n* = 1.

[Displays featuring this command] **DM-D110, DM-D210,** and **DM-D500.**

[Description] Selects the device the host computer sends data to, using *n* as follows:

<i>n</i>	Peripheral Device Status
0	Printer canceled and display canceled
1	Only printer selected (customer display is disabled)
2	Only customer display selected (printer is disabled)
3	Both the printer and display are selected
4	Undefined

- [Notes]
- When the printer is selected (*n* = 1), the printer processes data from the host, and the display ignores this data.
 - When the customer display is selected (*n* = 2), the display processes the data from the host, and the printer ignores data from the host.
 - When both the printer and customer displays are selected (*n* = 3), all the data from the host computer is processed simultaneously by the display and by the printer.
 - If the printer receives **ESC = 2** when it is selected by *n* = 1 or *n* = 3, the host sends 1BH (27) 3DH (61) 02H (2) to the printer and stops transmitting data for the printer.





- If **ESC = 1** is received when the customer display is selected by **n = 2**, the host sends 1BH (27) 3DH (61) 01H (1) to the display and starts transmitting data to the printer.
- If **ESC = 3** is received when the customer display is selected by **n = 2**, the host sends 1BH (27) 3DH (61) 03H (3) to the printer and starts transmitting data to the printer and display.
- If **ESC = 2** is received again after the customer display is selected by **n = 2**, the three bytes of data are executed inside the display only, and nothing is sent to the printer.

[Example]

Program example for all displays	
Print #1;CHR\$(&H1B);CHR\$(&H3D);CHR\$(&H1);	①
Print #1,"SELECT PRINTER";	②
Print #1,CHR\$(&H1B);CHR\$(&H3D);CHR\$(&H2);	③
Print #1,"SELECT DISPLAY";	④

- Data in lines ① and ③ is processed internally in the display and sent to the printer simultaneously.
- Data in line ② is sent to the printer regardless of display execution.
- Data in line ④ only appears on the display screen, and nothing is sent to the printer.

Command	Action	Example
PRINT #1, CHR\$(27) PRINT #1, "@";	Display action (initialization)	
	Printer action (initialization)	



CONFIDENTIAL

<pre>PRINT #1, CHR\$(27); "="; CHR\$(1); PRINT #1, "PRINTER SELECT"</pre>	Display action (none: printer selected)	
	Printer action (printer selected; text printed)	
<pre>PRINT #1, CHR\$(27); "="; CHR\$(2); PRINT #1, "DISPLAY SELECT"</pre>	Display action (display selected; text displayed)	
	Printer action (none; display selected)	
<pre>PRINT #1, CHR\$(27); "="; CHR\$(3); PRINT #1, "PRINTER AND DISPLAY SELECT"</pre>	Display action (display selected; text displayed)	
	Printer action (printer selected; text printed)	

[Reference]

US (A



ESC @

EXECUTING COMMAND

SETTING COMMAND

[Name] Initialize display
 [Format] ASCII ESC @
 Hex 1B 40
 Decimal 27 64

[Displays featuring this command] **DM-D110**, **DM-D210**, and **DM-D500**.

[Description] Resets display settings to their initial values.

- [Notes]
- Software settings are reset to their power on values.
 - DIP switches are not checked again.
 - Data in the receive buffer is not cleared.
 - After the customer display is initialized, the screen is cleared, and the cursor is moved to the home position.
 - The initial display settings for the **DM-D110** and **DM-D210** are as follows:

Settings	Values
Display mode	Overwrite mode
Position	Home position (upper left corner of window)
Screen	Clear
Window	Not defined
Character code table	Page 0
International character set	U.S.A.
User-defined characters	Not defined
Macro definition	Not defined
Reverse characters	Canceled



Settings	Values
Display blinking	Canceled
Brightness adjustment	100%
Peripheral device selection	Display
Set-up time	00:00
Cursor display	Selected

■ The initial display settings for the **DM-D500** are as follows:

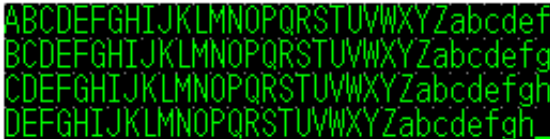
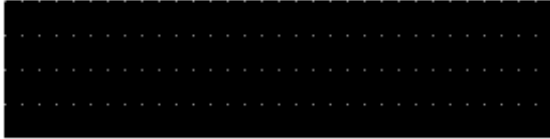
Settings	Values
Display mode	Overwrite mode
Position	Home position (upper left corner of window)
Screen	Clear
Window	Not defined
Character code table	Page 0
International character set	U.S.A.
User-defined characters	Not defined
Macro definition	Not defined
Reverse characters	Canceled
Display blinking	Canceled
Brightness adjustment	100%



CONFIDENTIAL

Settings	Values
Peripheral device selection	Display
Set-up time	00:00
Cursor display	Selected

[Example]

Example	Display Sample
Normal	
Before ESC @ is sent	
Send ESC @ command	PRINT #1, CHR\$(27); CHR\$(64);
After ESC @ is sent	

[References]

ESC =, ESC %, ESC &, ESC R, ESC t, ESC W, US MD1, US MD2, US MD3, US C, US E, US T, US U, US X, US r, US :, US (D Function 1, US (F Function 2, US (G Functions A1, A2, A3, B1, C1, C2, C3, C4, US (H Functions A1, B1, C1



ESC %

SETTING COMMAND

[Name] Select/cancel user-defined character set

[Format] ASCII ESC % *n*
 Hex 1B 25 *n*
 Decimal 27 37 *n*

[Range] $0 \leq n \leq 255$

[Default] *n* = 0


[Displays featuring this command] **DM-D110**, **DM-D210**, and **DM-D500**.

[Description] Selects or cancels the user-defined character set.



- When the least significant bit (LSB) of *n* is 0, the user-defined character set is canceled.
- When the LSB of *n* is 1, the user-defined character set is selected.

- [Notes]
- When the user-defined character set is canceled, the resident character set is automatically selected.
 - When the user-defined character set is not defined using the **ESC &** command, the internal character set is displayed.
 - Canceling the display of the user-defined character set with *n* = 0 has no effect on the characters defined with the **ESC &** command.

[Example]


Action	Command/Example
Send ESC @ command to initialize display	PRINT #1, CHR\$(27); "@";
	



<p>Define user-defined character set "0"</p>	<pre>PRINT #1, CHR\$(27); "&"; CHR\$(2); PRINT #1, "00"; CHR\$(8); PRINT #1, CHR\$(&HFF); CHR\$(&HFF); PRINT #1, CHR\$(&HFF); CHR\$(&HFF); PRINT #1, CHR\$(&HFE); CHR\$(&H7F); PRINT #1, CHR\$(&HFD); CHR\$(&HBF); PRINT #1, CHR\$(&HFD); CHR\$(&HBF); PRINT #1, CHR\$(&HFE); CHR\$(&H7F); PRINT #1, CHR\$(&HFF); CHR\$(&HFF); PRINT #1, CHR\$(&HFF); CHR\$(&HFF);</pre>
<p>Display "012"</p>	<pre>PRINT #1, "012";</pre> 
<p>Select user-defined character set</p>	<pre>PRINT #1, CHR\$(27); CHR\$(1);</pre>
<p>Display "012"</p>	<pre>PRINT #1, "012";</pre> 
<p>Cancel user-defined character set "0"</p>	<pre>PRINT #1, CHR\$(27); "?0";</pre>



CONFIDENTIAL

Display "012"	PRINT #1, "012";
	

[References]

ESC &, ESC ?, ESC R, ESC t



ESC &

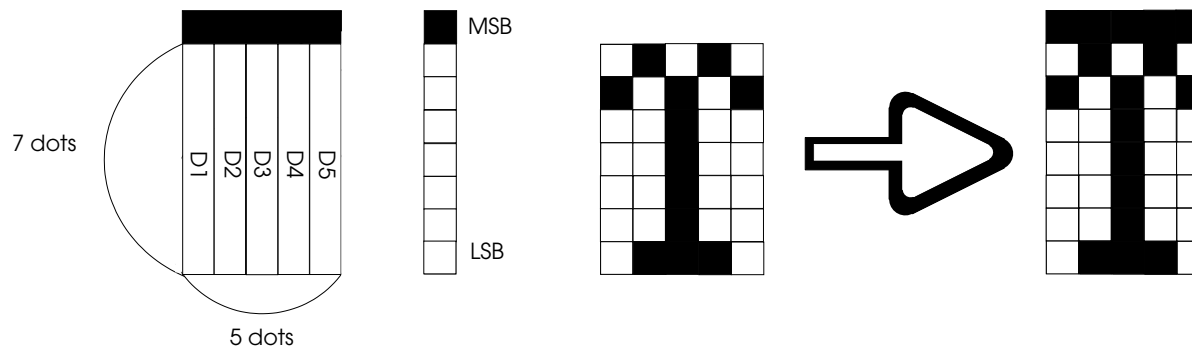
SETTING COMMAND

[Name]	Define user-defined characters			
[Format]	ASCII	ESC	&	y c1 c2 [x1 d1 ... d(y × x1)] ... [xk d1 ... d(y × xk)]
	Hex	1B	26	y c1 c2 [x1 d1 ... d(y × x1)] ... [xk d1 ... d(y × xk)]
	Decimal	27	38	y c1 c2 [x1 d1 ... d(y × x1)] ... [xk d1 ... d(y × xk)]
[Range]	<p>DM-D500 only — in 256 × 64 mode: y = 2 (when font A is selected) y = 1 (when font B is selected) 32 ≤ c1 ≤ c2 ≤ 126 0 ≤ x ≤ 8 (when font A is selected) 0 ≤ x ≤ 5 (when font B is selected) 0 ≤ d ≤ 255 k = c1 - c2 + 1</p> <p>DM-D110, DM-D210, and DM-D500 — in 20-column/2-line mode: y = 1 32 ≤ c1 ≤ c2 ≤ 126 0 ≤ x ≤ 5 0 ≤ d1 ... dy × x ≤ 255 k = c1 - c2 + 1</p>			
[Displays featuring this command]	DM-D110, DM-D210, and DM-D500.			
[Description]	<p>Defines user-defined characters.</p> <ul style="list-style-type: none"> • y specifies the number of bytes in the vertical direction. • x specifies the number of dots in the horizontal direction. When x < 5, the remaining dots on the right side of the user-defined characters are padded with spaces. • c1 specifies the beginning character code for definition, and c2 specifies the final character code. When only one character is defined, use c1 = c2. • d1 ... dk is the dot data for the user-defined characters. 			
[Notes]	<ul style="list-style-type: none"> ■ You can define characters between character codes 20H (32) to 7EH (126) in the character code table. ■ The data to define a user-defined character is (y × x) bytes. 			



- When eight bits are specified for the communication word length, the most significant bit is ignored.
- Once the user-defined characters have been defined, they remain available until they are redefined, **ESC @** is executed, or the power is turned off.
- Data (**d**) specifies a bit printed as 1 and not printed as 0. The dot pattern is in the horizontal direction from the left side. Any remaining dots on the right side are blank.
- After you have defined the user-defined characters, they will not be displayed until the user defined set is selected with **ESC % n**.
- In the default settings, the user-defined characters are not defined, and the resident characters are displayed.

[Examples]



When communication word length is specified as seven bits or when the word length is specified as eight bits, the most significant bit (MSB) is processed as "0." The user-defined character is defined as shown below.


```

Program example for all displays
Print #1;CHR$(&H1B);CHR$(&H26);CHR$(&H1);
Print #1,CHR$(&H20);CHR$(&H20);CHR$(&H5);
Print #1,CHR$(&H20);CHR$(&H41);CHR$(&H3F);CHR$(&H41);CHR$(&H20);
    
```



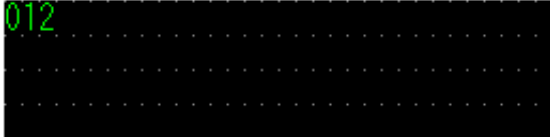
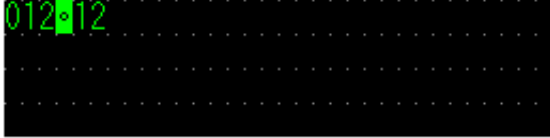

When the word length is specified as eight bits, the most significant bit (MSB) is processed as "1." The user-defined character is defined as shown below.

Program example for all displays
<pre>Print #1,CHR\$(&H1B);CHR\$(&H26);CHR\$(&H1); Print #1,CHR\$(&H20);CHR\$(&H20);CHR\$(&H5); Print #1,CHR\$(&HA0);CHR\$(&HC1);CHR\$(&HBF);CHR\$(&HC1);CHR\$(&HA0);</pre>

Action	Command/Example
Send ESC @ command to initialize display	PRINT #1, CHR\$(27); "@"; 
Define user-defined character set "0"	PRINT #1, CHR\$(27); "&"; CHR\$(2); PRINT #1, "00"; CHR\$(8); PRINT #1, CHR\$(&HFF); CHR\$(&HFF); PRINT #1, CHR\$(&HFF); CHR\$(&HFF); PRINT #1, CHR\$(&HFE); CHR\$(&H7F); PRINT #1, CHR\$(&HFD); CHR\$(&HBF); PRINT #1, CHR\$(&HFD); CHR\$(&HBF); PRINT #1, CHR\$(&HFE); CHR\$(&H7F); PRINT #1, CHR\$(&HFF); CHR\$(&HFF); PRINT #1, CHR\$(&HFF); CHR\$(&HFF);



CONFIDENTIAL

Display "012"	PRINT #1, "012";
	
Select user-defined character set	PRINT #1, CHR\$(27); CHR\$(1);
Display "012"	PRINT #1, "012";
	
Cancel user-defined character set "0"	PRINT #1, CHR\$(27); "?0";
Display "012"	PRINT #1, "012";
	

[References]

ESC %, **ESC ?**, **ESC R**, **ESC t**



ESC ?

SETTING COMMAND

[Name] Cancel user-defined characters

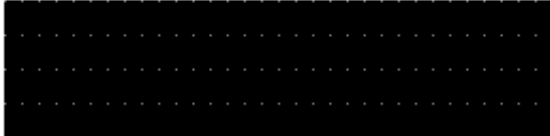
[Format] ASCII ESC ? *n*
 Hex 1B 3F *n*
 Decimal 27 63 *n*

[Range] $32 \leq n \leq 126$



[Displays featuring this command] **DM-D110**, **DM-D210**, and **DM-D500**.

[Description] Cancels the user-defined characters defined for character code *n*.

- [Notes]
- After the user-defined characters are canceled, the resident character set is printed.
 - This command has no effect on characters already displayed.
 - If no user-defined characters are defined, this command is ignored.


Action	Command/Example
Send ESC @ command to initialize display	PRINT #1, CHR\$(27); "@";
	



<p>Define user-defined character set "0"</p>	<pre>PRINT #1, CHR\$(27); "&"; CHR\$(2); PRINT #1, "00"; CHR\$(8); PRINT #1, CHR\$(&HFF); CHR\$(&HFF); PRINT #1, CHR\$(&HFF); CHR\$(&HFF); PRINT #1, CHR\$(&HFE); CHR\$(&H7F); PRINT #1, CHR\$(&HFD); CHR\$(&HBF); PRINT #1, CHR\$(&HFD); CHR\$(&HBF); PRINT #1, CHR\$(&HFE); CHR\$(&H7F); PRINT #1, CHR\$(&HFF); CHR\$(&HFF); PRINT #1, CHR\$(&HFF); CHR\$(&HFF);</pre>
<p>Display "012"</p>	<pre>PRINT #1, "012";</pre> 
<p>Select user-defined character set</p>	<pre>PRINT #1, CHR\$(27); CHR\$(1);</pre>
<p>Display "012"</p>	<pre>PRINT #1, "012";</pre> 
<p>Cancel user-defined character set "0"</p>	<pre>PRINT #1, CHR\$(27); "?0";</pre>



CONFIDENTIAL

Display "012"	PRINT #1, "012";
	

[References]

ESC %, **ESC &**, **ESC R**, **ESC t**



ESC R

SETTING COMMAND

[Name] Select an international character set

[Format]

ASCII	ESC	R	<i>n</i>
Hex	1B	52	<i>n</i>
Decimal	27	82	<i>n</i>

[Range] $0 \leq n \leq 13$

[Default] ***n*** = 0

[Displays featuring this command] **DM-D110**, **DM-D210**, and **DM-D500**.




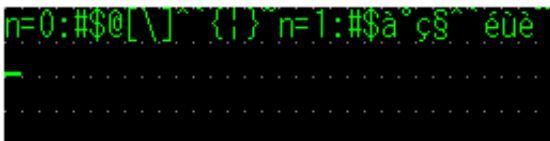
[Description] Selects an international character set ***n*** as follows:

<i>n</i>	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
10	Denmark II



<i>n</i>	Character Set
11	Spain II
12	Latin America
13	Korea

[Example]

Command	Display
PRINT #1, CHR\$(27); "@";	
PRINT #1, CHR\$(31); "(D"; PRINT #1, CHR\$(4); CHR\$(0); PRINT #1, CHR\$(3); CHR\$(97); PRINT #1, CHR\$(1); CHR\$(2); PRINT #1, CHR\$(31); CHR\$(2);	
PRINT #1, CHR\$(27); "R"; CHR\$(0); PRINT #1, "n= 0:##\$@[\\]^`{ }~";	
PRINT #1, CHR\$(27); "R"; CHR\$(1); PRINT #1, "n= 1:##\$@[\\]^`{ }~";	



<pre>PRINT #1, CHR\$(27); "R"; CHR\$(2); PRINT #1, "n= 2:\$\$@[N]^{}~";</pre>	
<pre>PRINT #1, CHR\$(27); "R"; CHR\$(3); PRINT #1, "n= 3:\$\$@[N]^{}~";</pre>	
<pre>PRINT #1, CHR\$(27); "R"; CHR\$(4); PRINT #1, "n= 4:\$\$@[N]^{}~";</pre>	
<pre>PRINT #1, CHR\$(27); "R"; CHR\$(5); PRINT #1, "n= 5:\$\$@[N]^{}~";</pre>	
<pre>PRINT #1, CHR\$(27); "R"; CHR\$(6); PRINT #1, "n= 6:\$\$@[N]^{}~";</pre>	
<pre>PRINT #1, CHR\$(27); "R"; CHR\$(7); PRINT #1, "n= 7:\$\$@[N]^{}~";</pre>	



CONFIDENTIAL

<pre>PRINT #1, CHR\$(27); "R"; CHR\$(8); PRINT #1, "n= 8:\$\$@[N]^{}~";</pre>	<pre>n=2:##\$SAOU` aouBn=3:£\$@[\]` { } n=4:##\$@A0A` xoa n=5:##EAOAUéaoau n=6:##\$@`é` uadoei n=7:##\$@iNz` n} n=8:##\$@[¥]` { }`</pre>
<pre>PRINT #1, CHR\$(27); "R"; CHR\$(9); PRINT #1, "n= 9:\$\$@[N]^{}~";</pre>	<pre>n=4:##\$@A0A` xoa n=5:##EAOAUéaoau n=6:##\$@`é` uadoei n=7:##\$@iNz` n} n=8:##\$@[¥]` { }` n=9:##EAOAUéaoau</pre>
<pre>PRINT #1, CHR\$(27); "R"; CHR\$(10); PRINT #1, "n10:\$\$@[N]^{}~";</pre>	<pre>n=4:##\$@A0A` xoa n=5:##EAOAUéaoau n=6:##\$@`é` uadoei n=7:##\$@iNz` n} n=8:##\$@[¥]` { }` n=9:##EAOAUéaoau n10:##\$EAOAUéaoau`</pre>
<pre>PRINT #1, CHR\$(27); "R"; CHR\$(11); PRINT #1, "n11:\$\$@[N]^{}~";</pre>	<pre>n=6:##\$@`é` uadoei n=7:##\$@iNz` n} n=8:##\$@[¥]` { }` n=9:##EAOAUéaoau n10:##\$EAOAUéaoau n11:##\$aiNzé` inou`</pre>
<pre>PRINT #1, CHR\$(27); "R"; CHR\$(12); PRINT #1, "n12:\$\$@[N]^{}~";</pre>	<pre>n=6:##\$@`é` uadoei n=7:##\$@iNz` n} n=8:##\$@[¥]` { }` n=9:##EAOAUéaoau n10:##\$EAOAUéaoau n11:##\$aiNzé` inou` n12:##\$aiNzé` uinou`</pre>
<pre>PRINT #1, CHR\$(27); "R"; CHR\$(13); PRINT #1, "n13:\$\$@[N]^{}~";</pre>	<pre>n=8:##\$@[¥]` { }` n=9:##EAOAUéaoau n10:##\$EAOAUéaoau n11:##\$aiNzé` inou` n12:##\$aiNzé` uinou` n13:##\$@[#]` { }`</pre>

[References]

ESC %, **ESC &**, **ESC ?**, **ESC t**



ESC t

[Name] Select character code table

[Format] ASCII ESC t ***n***
 Hex 1B 74 ***n***
 Decimal 27 116 ***n***

[Range] **DM-D110** and **DM-D210**: $0 \leq n \leq 5, 16, 17, 18, 19, 254, 255$
DM-D500 $n = 0, 1, 2, 3, 4, 5, 16, 17, 18, 19, 254, 255$

[Default] ***n*** = 0

[Displays featuring this command] **DM-D110**, **DM-D210**, and **DM-D500**.

[Description] Selects page ***n*** from the character code tables, as follows:






<i>n</i>	Character Code Table
0	Page 0 [PC437 (U.S.A., Standard Europe)]
1	Page 1 [Katakana]
2	Page 2 [PC850 (Multilingual)]
3	Page 3 [PC860 (Portuguese)]
4	Page 4 [PC863 (Canadian-French)]
5	Page 5 [PC865 (Norwegian)]
16	WPC1252
17	PC866 [Cyrillic #2]
18	PC852 [Latin 2]
19	Page 19 [PC858]
254	Page 254 (user-defined code page)
255	Page 255 (user-defined code page)









[Notes]

- This command has no effect on data displayed before this command was sent.
- This command has no effect on the selected international character set.

[Examples]


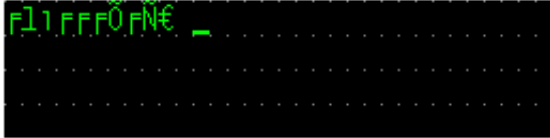
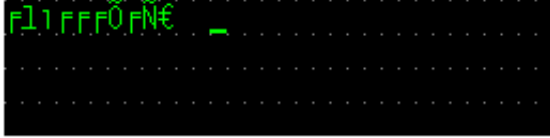
Command	Display
<pre>PRINT #1, CHR\$(27); "@";</pre>	
<pre>PRINT #1, CHR\$(31); "(D"; PRINT #1, CHR\$(4); CHR\$(0); PRINT #1, CHR\$(3); CHR\$(97); PRINT #1, CHR\$(1); CHR\$(2); PRINT #1, CHR\$(31); CHR\$(2);</pre>	
<pre>PRINT #1, CHR\$(27); "t"; CHR\$(0); CHR\$(213);</pre>	
<pre>PRINT #1, CHR\$(27); "t"; CHR\$(1); CHR\$(213);</pre>	
<pre>PRINT #1, CHR\$(27); "t"; CHR\$(2); CHR\$(213);</pre>	



PRINT #1, CHR\$(27); "t"; CHR\$(3); CHR\$(213);	
PRINT #1, CHR\$(27); "t"; CHR\$(4); CHR\$(213);	
PRINT #1, CHR\$(27); "t"; CHR\$(5); CHR\$(213);	
PRINT #1, CHR\$(27); "t"; CHR\$(16); CHR\$(213);	
PRINT #1, CHR\$(27); "t"; CHR\$(17); CHR\$(213);	
PRINT #1, CHR\$(27); "t"; CHR\$(18); CHR\$(213);	



CONFIDENTIAL

<pre>PRINT #1, CHR\$(27); "t"; CHR\$(19); CHR\$(213);</pre>	
<pre>PRINT #1, CHR\$(27); "t"; CHR\$(254); CHR\$(213);</pre>	
<pre>PRINT #1, CHR\$(27); "t"; CHR\$(255); CHR\$(213);</pre>	

[References]

ESC %, **ESC &**, **ESC ?**, **ESC R**



ESC W

SETTING COMMAND

[Name]	Set/cancel window range								
[Format]	ASCII	ESC	W	<i>n</i>	<i>m</i>	<i>x1</i>	<i>y1</i>	<i>x2</i>	<i>y2</i>
	Hex	1B	57	<i>n</i>	<i>m</i>	<i>x1</i>	<i>y1</i>	<i>x2</i>	<i>y2</i>
	Decimal	27	87	<i>n</i>	<i>m</i>	<i>x1</i>	<i>y1</i>	<i>x2</i>	<i>y2</i>
[Range]	$1 \leq n \leq 4$ $m = 0, 1, 48, 49$ $1 \leq x1 \leq y2 \leq 20$ $1 \leq y1 \leq y2 \leq 2$								
[Default]	Not defined.								
[Displays featuring this command]	DM-D110 , DM-D210 , and DM-D500 (in 20-character/2-line mode only) .								
[Description]	<p>Selects or cancels a single window on the display screen.</p> <ul style="list-style-type: none"> • <i>n</i> specifies the number of the window to be selected or canceled. • <i>m</i> specifies selection or cancellation. <ul style="list-style-type: none"> When <i>m</i> = 1 or 49, a window is selected. (Values <i>x1</i>, <i>y1</i>, <i>x2</i>, and <i>y2</i> are required.) When <i>m</i> = 0 or 48, a window is canceled. (Values <i>x1</i>, <i>y1</i>, <i>x2</i>, and <i>y2</i> are not required.) • <i>x1</i>, <i>y1</i> set the positions of the upper left column and line of the window, respectively. • <i>x2</i>, <i>y2</i> set the positions of the lower right column and line of the window, respectively. 								
[Notes]	<ul style="list-style-type: none"> ■ Up to four windows can be selected simultaneously on the display screen; however, the window ranges cannot overlap. ■ To cancel a window, send the command without arguments for <i>x1</i>, <i>y1</i>, <i>x2</i>, and <i>y2</i>. ■ If a value is set outside the display screen or overlapping another window, this command is ignored. 								

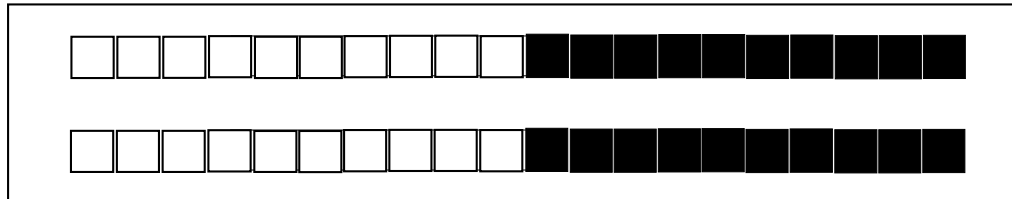


[Examples]

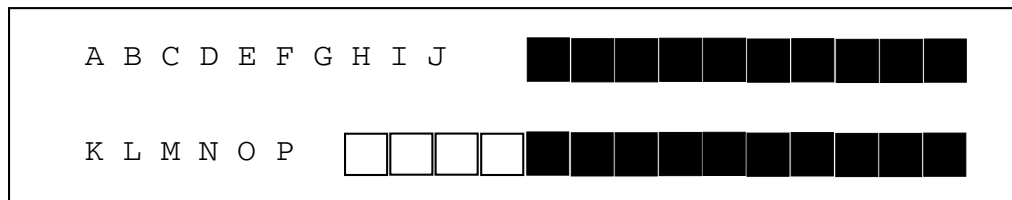
Program Example for all displays




```
PRINT #1, CHR$ (&H1B);CHR$ (&H57);CHR$ (&H1)CHR$ (&H1);①  
PRINT #1, CHR$ (1);CHR$ (1);CHR$ (10);CHR$ (2);②  
PRINT #1, CHR$ (&H1B);CHR$ (&H24);CHR$ (1);CHR$ (1);③  
PRINT #1, "ABCDEFGH IJKLMN O P"; ④
```

- The left half of the range shown in the figure below is specified as Window 1 by executing lines ① and ② in the program above.


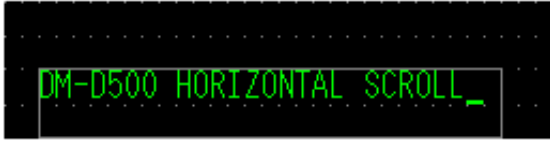
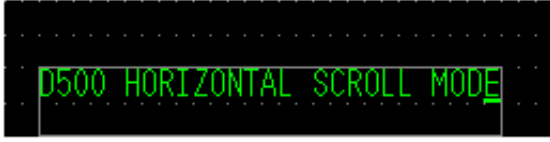


- Executing ③ displays only the inside of the window, and executing ④ results in the display shown in the figure below.



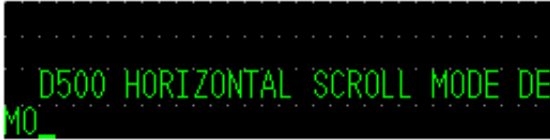
Action	Command/Example
Send ESC @ command to initialize display	PRINT #1, CHR\$(27); "@";
	
Change to 32-column × 4-line mode	PRINT #1, CHR\$(31); "(D"; PRINT #1, CHR\$(4); CHR\$(0); PRINT #1, CHR\$(3); CHR\$(97); PRINT #1, CHR\$(1); CHR\$(2);
	
Define window	PRINT #1, CHR\$(27); "W"; PRINT #1, CHR\$(1); CHR\$(1); PRINT #1, CHR\$(3); CHR\$(3); PRINT #1, CHR\$(29); CHR\$(4);
	



<p>Move to created window and change scroll mode to horizontal scroll mode</p>	<pre>PRINT #1, CHR\$(31); "\$"; PRINT #1, CHR\$(3); CHR\$(3); PRINT #1, CHR\$(31); CHR\$(3);</pre>
<p>Display text in horizontal scroll mode</p>	
	<pre>PRINT #1, "DM-D500 HORIZONTAL SCROLL";</pre>
	
	<pre>PRINT #1, " MODE";</pre>
	



CONFIDENTIAL

Delete window and display text in overwrite mode	<pre>PRINT #1, CHR\$(27); "W"; PRINT #1, CHR\$(1); CHR\$(0); PRINT #1, CHR\$(9); PRINT #1, " DEMO";</pre>
	

[Model-dependent variations] **DM-D500**

DM-D500

This command is effective only when the 20-column/2-line mode is selected. Use the **US (D command when the 256 × 64 mode is selected.**

[Reference] **US (D** Functions 1, 2, 3, 4, and 6



US MD1

EXECUTING COMMAND


[Name] Select overwrite mode

[Format] ASCII US MD1
 Hex 1F 01
 Decimal 31 1

[Displays featuring this command] **DM-D110**, **DM-D210**, and **DM-D500**.

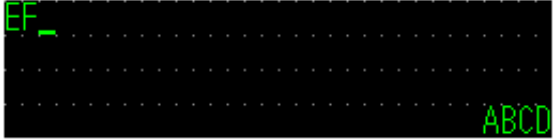
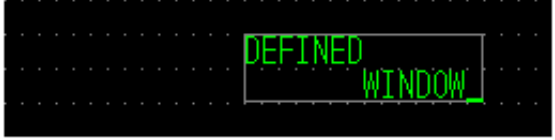

[Description] Selects overwrite mode as the screen display mode.

- [Notes]
- This mode is selected at power on.
 - Entering a character code displays the character and then moves the cursor one character space to the right, unless the cursor is at the right end of the line.
 - If the cursor is at the right end of the upper line in overwrite mode, entering a character code moves the cursor to the left end of the lower line.
 - If the cursor is at the right end of the lower line in overwrite mode, entering a character code moves the cursor to the left end of the upper line.
 - Selecting overwrite mode cancels horizontal or vertical scroll mode.

Example	Display Sample
Normal	
Before US MD1 is sent	
Send US MD1 command	PRINT #1, CHR\$(31); CHR\$(1);



CONFIDENTIAL

Send more data	PRINT #1, "DEF";
	
Window condition	
Before US MD1 is sent	
Send US MD1 command	PRINT #1, CHR\$(31); CHR\$(1);
Send more data	PRINT #1, "ABCDEF";
	

[References]

US MD2, US MD3, BS, HT, LF, US LF



US MD2

EXECUTING COMMAND

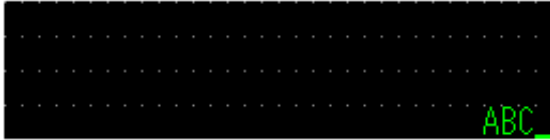
[Name] Select vertical scroll mode

[Format] ASCII US MD2
 Hex 1F 02
 Decimal 31 2

[Displays featuring this command] **DM-D110**, **DM-D210**, and **DM-D500**.

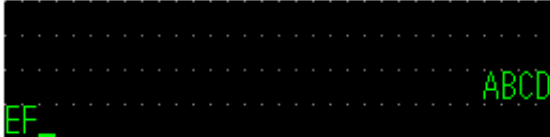
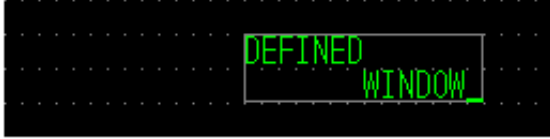

[Description] Selects vertical scroll mode as the screen display mode.

- [Notes]
- Entering a character code displays the character and then moves the cursor one character space to the right, unless the cursor is at the right end of the line.
 - If the cursor is at the right end of the upper line in vertical scroll mode, entering a character code moves the cursor to the left end of the lower line, scrolls the characters on the lower line to the upper line, and clears the lower line.
 - Selecting vertical scroll mode cancels overwrite or horizontal scroll mode.

Example	Display Sample
Normal	
Before US MD2 is sent	
Send US MD2 command	PRINT #1, CHR\$(31); CHR\$(2);



CONFIDENTIAL

Send more data	PRINT #1, "DEF";
	
Window condition	
Before US MD2 is sent	
Send US MD2 command	PRINT #1, CHR\$(31); CHR\$(2);
Send more data	PRINT #1, "ABCDEF";
	

[References]

US MD1, US MD3, BS, HT, LF, US LF



US MD3

EXECUTING COMMAND

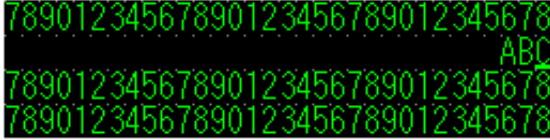
[Name] Select horizontal scroll mode

[Format] ASCII US MD3
 Hex 1F 03
 Decimal 31 3

[Displays featuring this command] **DM-D110**, **DM-D210**, and **DM-D500**.

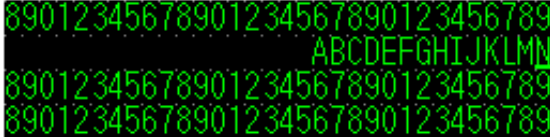
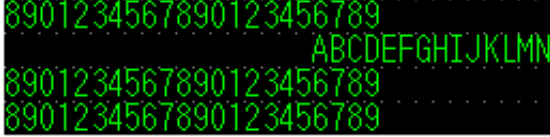
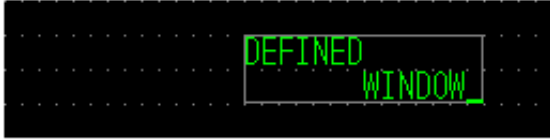

[Description] Selects horizontal scroll mode as the display screen mode.

- [Notes]
- Entering a character code displays the character and then moves the cursor one character space to the right, unless the cursor is at the right end of the line.
 - When the cursor is at the right end of either line in horizontal scroll mode, entering a character code scrolls all displayed characters, including commas and periods, one character position to the left and then displays the new character at the right end of the line.
 - Selecting horizontal scroll mode cancels overwrite mode and vertical scroll mode.

Example	Display Sample
Normal	
Before US MD3 is sent	
Send US MD3 command	PRINT #1, CHR\$(31); CHR\$(3);
Send more data	PRINT #1, "DEFGHIJKLMN";



CONFIDENTIAL

Character mode	
Graphics mode (256 × 64 dots)	
Window condition	
Before US MD3 is sent	
Send US MD3 command	PRINT #1, CHR\$(31); CHR\$(3);
Send more data	PRINT #1, "ABCDEF";
	

[References]

US MD1, US MD2, BS, HT, LF, US LF



US C

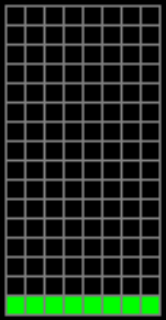
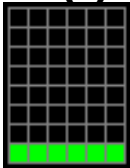
EXECUTING COMMAND

[Name]	Turn cursor display mode on/off									
[Format]	<table border="0"> <tr> <td>ASCII</td> <td>US</td> <td>C</td> </tr> <tr> <td>Hex</td> <td>1F</td> <td>43</td> </tr> <tr> <td>Decimal</td> <td>31</td> <td>67</td> </tr> </table>	ASCII	US	C	Hex	1F	43	Decimal	31	67
ASCII	US	C								
Hex	1F	43								
Decimal	31	67								
[Range]	<i>n</i> = 0, 1, 48, 49									
[Default]	<i>n</i> = 1 DM-D110 only: <i>n</i> = 1 or the value set by memory switch 14. (See US (E Function 3.)									
[Displays featuring this command]	DM-D110 and DM-D500 .									
[Displays not featuring this command]	DM-D210 .									
[Description]	<p>Selects or cancels the cursor display.</p> <p>When <i>n</i> = 1 or 49, the cursor is displayed.</p> <p>When <i>n</i> = 0 or 48, the cursor is not displayed.</p>									
[Notes]	<ul style="list-style-type: none"> ■ This command has no effect on characters already displayed. ■ This command is effective only when the 20-column/2-line mode (DM-D1xx emulation mode) or 256 × 64 mode is selected. 									
[Model-dependent variations]	DM-D500									

DM-D500

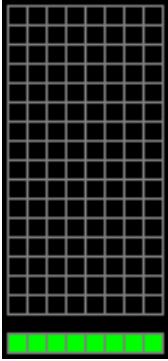
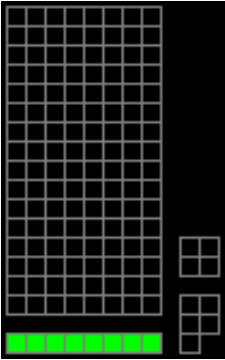
- **In 256 × 64 mode, this command is specified for each window. The cursor is displayed only in the current window.**



Mode	Character display area and cursor
Graphics mode 1 (256 × 64 dots)	Cannot use.
Character mode 1 (32-column × 4-line mode)	<p style="text-align: center;">8</p>  <p style="text-align: right;">16</p>
Character mode 2 (42-column × 8-line mode)	<p style="text-align: center;">5 (6)</p>  <p style="text-align: right;">7 (8)</p>



CONFIDENTIAL

<p>Character mode 3 (32-column × 3-line mode) and Character mode 4 (32-column × 2-line mode)</p>	 <p>8 16</p>
<p>Character mode 5 (20-column × 2-line mode)</p>	 <p>8 16</p>

[References]

US E, US X, US r, US X, US ,, US ;, US #, US (G Functions A1, A2, A3, B1



US E

[Name] Sets display screen blank interval

[Format] ASCII US E ***n***
Hex 1F 45 ***n***
Decimal 31 69 ***n***

[Range] $0 \leq n \leq 255$

[Default] ***n*** = 0

[Displays featuring this command] **DM-D110**, **DM-D210**, and **DM-D500**.

[Description] Sets or cancels the blink interval of the display screen.

- ***n*** specifies the blink interval. The interval [ON= ***n*** × 50 ms] / [OFF = ***n*** × 50 ms] is repeated.
- When ***n*** = 0, blinking is canceled. (The screen remains ON.)
- When ***n*** = 255, the display is turned OFF, but the contents of the display are maintained.

[Note] ■ This command does not affect the brightness of the vacuum fluorescent display.

[References] **US C**, **US X**, **US r**, **US .**, **US ,**, **US ;**, **US #**, **US (G** Functions A1, A2, A3, B1



US T

EXECUTING COMMAND


- [Name] Set and display counter time
- [Format]

ASCII	US	T	<i>h</i>	<i>m</i>
Hex	1F	54	<i>h</i>	<i>m</i>
Decimal	31	84	<i>h</i>	<i>m</i>
- [Range]
 - 0 ≤ ***h*** ≤ 23
 - 0 ≤ ***m*** ≤ 59
- [Default] ***h*** = 0, ***m*** = 0
- [Displays featuring this command] **DM-D110**, **DM-D210**, and **DM-D500 (when 20-character/2-line mode is selected)**.
- [Description] Sets the counter time and displays it in the bottom right of the screen.
- [Notes]
 - When this command is entered, the screen is cleared, and the time is displayed in 24-hour mode at the right side of the bottom line.
 - The time counter starts after receipt of the transmitted code ***h:m:00***.
 - After the time is displayed, the cursor moves to the home position.
 - The counter display disappears under any of the following conditions:
 - The cursor moves to the bottom line.
 - Display characters move to the bottom line.
 - The **CLR** command is received.
 - Even if the time counter is cleared, it continues to be updated in the display.

Mode	Display after US T is sent
256 × 64 dots (not supported)	Cannot use.
32 columns × 4 lines (not supported)	Cannot use.
42 columns × 8 lines (not supported)	Cannot use.



CONFIDENTIAL

32 columns × 3 lines (not supported)	Cannot use.
32 columns × 2 lines (not supported)	Cannot use.
20 columns × 2 lines	

[References]

US U, US (D Function 3



US U

[Name] Display time counter

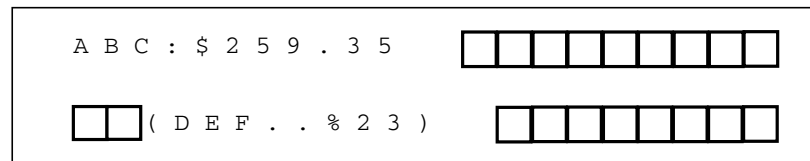
[Format]	ASCII	US	U
	Hex	1F	55
	Decimal	31	85

[[Displays featuring this command] **DM-D110**, **DM-D210**, and **DM-D500 (when 20-character/2-line mode is selected)**.

[Description] Displays the time counter at the right side of the bottom line.

- [Notes]
- If the time has not been set, the display shows the elapsed time from when the counter was initialized at power on or by an **ESC @** command. The display uses the time format "hours : minutes : seconds."
 - If the time has already been set using the **US T h m** command, the elapsed time is displayed in the time format "hours : minutes : seconds."
 - After the counter is displayed, the cursor moves to the home position.
 - The counter display is cleared when any of the following occurs:
 - The cursor moves to the bottom line.
 - Display characters move to the bottom line.
 - The **CLR** command is received.
 - Even if the time counter is cleared, it continues to be updated in the display.

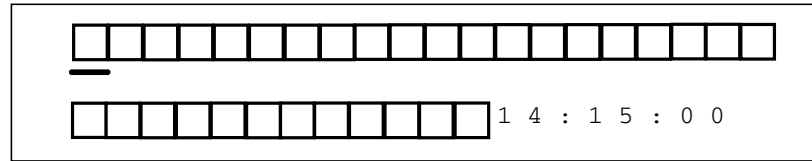
[Examples] Below is the customer display before it receives the **US T h m** command to set the counter. (The empty boxes represent blank character positions on the display.)



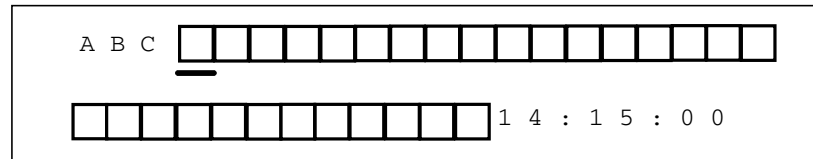
The display receives the command **US T h m**:

```
Program Example  
  
PRINT #1, CHR$( &H1F );CHR$( &H54 );CHR$( 14)CHR$( 15 );
```

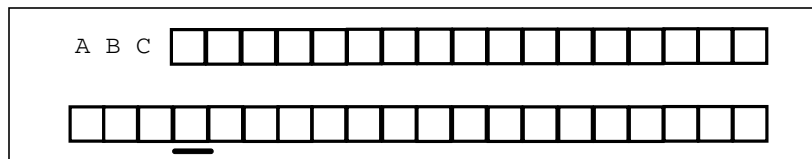
The screen is cleared, and the input time is displayed at the right side on the lower line. Counting begins at 14:15:00. The cursor moves to the top left character position.




In the screen below, the host has sent the characters A, B, C to the display, and the cursor moves to the fourth character position.



In the screen below, the display has received an **LF** code in overwrite mode. The cursor moves to the bottom line of the display, and the time display is cleared. The display continues counting the time internally.



Mode	Display after US T is sent
256 × 64 dots (not supported)	Cannot use.
32 columns × 4 lines (not supported)	Cannot use.
42 columns × 8 lines (not supported)	Cannot use.
32 columns × 3 lines (not supported)	Cannot use.
32 columns × 2 lines (not supported)	Cannot use.
20 columns × 2 lines	

[Model-dependent variations] **DM-D500**

[References] **US T** , **US (D** Function 3

DM-D500

- This command is effective only when the 20-column/2-line mode is selected.



US X

[Name] Set brightness

[Format] ASCII US X
 Hex 1F 58
 Decimal 31 88

[Range] $1 \leq n \leq 4$

[Default] $n = 4$

[Displays featuring this command] **DM-D110**, **DM-D210**, and **DM-D500**.

[Description] Sets the brightness of the fluorescent character display tube. n selects the percentage of brightness as follows:

n	Brightness
1	20%
2	40%
3	60%
4	100%

[References] **US C, US E, US r, US ., US ,, US ;, US #, US (G** Functions A1, A2, A3, B1



US r

[Name] Select/cancel reverse characters

[Format] ASCII US r **n**
 Hex 1F 72 **n**
 Decimal 31 114 **n**

[Range] **n** = 0, 1, 48, 49


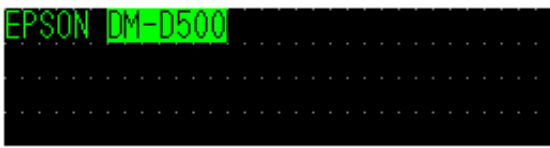
[Default] **n** = 0

[Displays featuring this command] **DM-D110**, **DM-D210**, and **DM-D500**.

[Description] Selects or cancels reverse display of the characters received after this command.

- When **n** = 1 or 49, reverse characters are selected.
- When **n** = 0 or 48, reverse characters are canceled.

[Example]

Command	Display
Send data	PRINT #1, "EPSON ";
	
Turn on reverse mode and send data	PRINT #1, CHR\$(31); "r"; PRINT #1, CHR\$(1); PRINT #1, "DM-D500";
	

[Reference] **US (G** Functions A2, A3



US v

[Name] Set status confirmation for DTR signal

[Format]	ASCII	US	v	n
	Hex	1F	76	n
	Decimal	31	118	n

[Range] **n** = 0, 1, 48, 49

[Displays featuring this command] **DM-D110**, **DM-D210**, and **DM-D500**.

[Description] Sets the DTR signal in the host interface to the MARK or SPACE state.

- When **n** = 1 or 49, the DTR signal goes to the MARK state. If the state is already MARK, the signal does not change.
- When **n** = 0 or 48, the DTR signal goes to the SPACE state, under the following conditions:
 - The receive buffer is not in the buffer-full state.
 - The self test is not being executed.

If the state is already SPACE, the signal does not change.

[Notes] ■ If any data is received during status confirmation using the DTR signal, normal interface timing control is re-entered immediately.

DM-D110 and **DM-D210** only:

- This command is effective only when the display alone is selected with **ESC = 2**; therefore, if the printer is already selected, this command is ignored (three bytes) and is processed as normal data (transmitted to the printer for printing).

[Reference] **ESC =**



US @

EXECUTING COMMAND

[Name]	Execute self test		
[Format]	ASCII	US	@
	Hex	1F	40
	Decimal	31	64

[Displays featuring this command] **DM-D110**, **DM-D210**, and **DM-D500**.

[Description] Runs the self test.

- [Notes]
- A series of self tests is displayed. All set values except those listed below are initialized:
 - User-defined character definitions.
 - Macro definitions.
 - Time counter value.
 - After completion of the self tests, the screen is cleared, and the cursor is moved to the home position.

[Reference] **ESC @**



US :

EXECUTING COMMAND

[Name] Start/end macro definition

[Format] ASCII US :
 Hex 1F 3A
 Decimal 31 58

[Displays featuring this command] **DM-D110**, **DM-D210**, and **DM-D500**.

[Description] Starts or ends a macro definition.

- [Notes]
- **DM-D110, DM-D210:**
 Up to 80 bytes can be defined for macro processing (one byte per character).
 - **DM-D500:**
Up to 2 KB can be defined for macro processing (one byte per character).
 - Macro definition processing starts with the first **US :** command and ends with the second **US :** command.
 - To delete a macro, send a **US :** immediately followed by a **US :**.
 - When the display encounters a macro definition error, it stops macro definition processing and processes any data that follows as normal display data.
 - Either of the two conditions below is considered a macro definition error:
 - Receipt of the **US ^** command during macro definition.
 - Receipt of a macro that exceeds the maximum number of bytes that can be defined.

[Example]

Program example for all displays

```

PRINT #1, CHR$( &H1F );CHR$( &H3A );①
PRINT #1, CHR$( &HC );②
PRINT #1, CHR$( &H1F );CHR$( &H45 );CHR$( 0 );③
PRINT #1, "Execute Macro!"; ④
PRINT #1, CHR$( &H1F );CHR$( &H45 );CHR$( 10 );⑤
PRINT #1, CHR$( &H1F );CHR$( &H3A );⑥
    
```



CONFIDENTIAL

- ① is the **US :** command that start macro definition. ⑥ is the **US :** command that ends macro definition.
- ② is the clear screen command **CLR**.
- ③ and ⑤ are blinking commands (**US E**).

[Reference] **US ^**



US ^

EXECUTING COMMAND

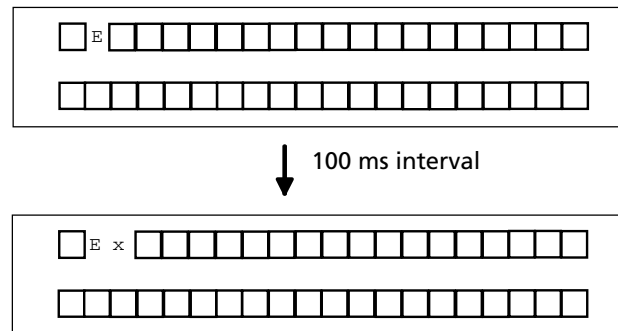
[Name]	Execute macro				
[Format]	ASCII	US	^	<i>n</i>	<i>m</i>
	Hex	1F	5E	<i>n</i>	<i>m</i>
	Decimal	31	94	<i>n</i>	<i>m</i>
[Range]	0 ≤ <i>n</i> ≤ 255				
	0 ≤ <i>m</i> ≤ 255				
[Displays featuring this command]	DM-D110 , DM-D210 , and DM-D500 .				
[Description]	<p>Executes the process defined in the macro.</p> <ul style="list-style-type: none"> • <i>n</i> specifies the time interval in units of [<i>n</i> × 20 ms] for displaying characters when a macro is executed. Although this affects the time interval between displaying each successive character, it does not affect the processing speed of command codes. • <i>m</i> specifies the interval between macro executions when the macro is run more than once. When macro processing is repeated, it is held for [<i>m</i> × 50 ms] after completion of one macro before repeating the macro. 				
[Notes]	<ul style="list-style-type: none"> ■ If data is received from the host during macro processing, macro processing is terminated. ■ After macro processing is completed, the current window is cleared, and the cursor is moved to the home position in the current window. ■ Display settings at the completion of macro processing remain valid. ■ If no macro is defined, this command is invalid, and the display content is not affected. ■ If macro definition includes ESC =, ESC @, or US @ within the macro, these commands are ignored during execution of the macro. ■ After macro processing is started, even if the printer has been selected by a peripheral device selection command, no data is transmitted to the printer during macro processing. 				



[Example]

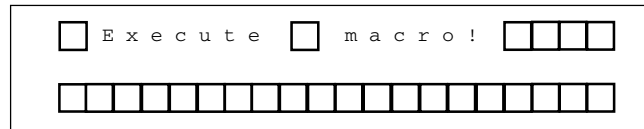
```
Program Example for all displays  
  
PRINT #1, CHR$( &H1F );CHR$( &H3A );  
PRINT #1, CHR$( &HC );  
PRINT #1, CHR$( &H1F );CHR$( &H45 );CHR$( 0 );①  
PRINT #1, "Execute macro!";  
PRINT #1, CHR$( &H1F );CHR$( &H45 );CHR$( 10 );  
PRINT #1, CHR$( &H1F );CHR$( &H3A );  
PRINT #1, CHR$( &H1F );CHR$( &H5E );CHR$( 5 ); CHR$( 100 );②
```

- Macro definition is started by ①.
- Macro execution is started by ②. In this case, the time interval for displaying the characters is (5 × 20) or 100 ms. After 100 ms has elapsed after the character "E" is displayed, the next character, "x," is displayed.



CONFIDENTIAL

- The macro execution interval is (100 × 50 ms). After the blinking display shown in the figure below is held for 5 seconds, the screen is cleared, and macro processing is repeated.



Held for 5 seconds

[Reference]

US :



US .

EXECUTING COMMAND

[Name] Display period

[Format]	ASCII	US	.	<i>n</i>
	Hex	1F	2E	<i>n</i>
	Decimal	31	46	<i>n</i>

[Range] $32 \leq n \leq 126$ and $128 \leq n \leq 255$

[Displays featuring this command] **DM-D210** and **DM-D500**.

[Displays not featuring this command] **DM-D110**.

[Description] Displays the specified character *n* and a period to the right of the character.

- [Notes]
- *n* indicates a character code that can be displayed.
 - The period is displayed only once for the specified character *n* and is not displayed for subsequent characters.
 - The cursor moves one character position to the right after displaying the period.
 - In overwrite mode: If any other character is overwritten in the character position of character *n*, the period is cleared.
 - In vertical scroll mode: If the display position of the character *n* moves, the period moves with the character.
 - In horizontal scroll mode: If the display position of the character *n* moves, the period moves with the character.

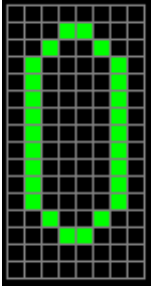
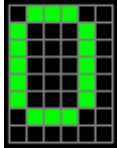
[Model-dependent variations] **DM-D500**

DM-D500

Available only when 20-column/2-line mode (DM-D2xx emulation) is selected.

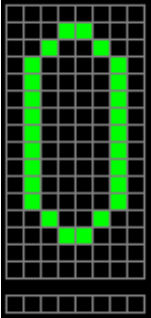
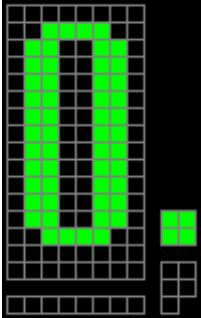


[Examples]

Graphic mode	Character display area and period
Send US . command	PRINT #1, CHR\$(31); ".0";
Graphic mode 1 font A (256 × 4 dots) and character mode 1 (32-column × 4-line mode)	 <p>Not supported</p>
Graphic mode 1 font B (256 × 64 dots) and character mode 2 (42-column × 8-line mode)	 <p>Not supported</p>



CONFIDENTIAL

<p>Character mode 3 (32-column × 3-line mode) and character mode 4 (32-column × 2-line mode)</p>	 <p>Not supported</p>
<p>Character mode 5 (20-column × 2-line mode)</p>	 <p>Supported</p>

[References]

US C, US E, US X, US r, US ,, US ;, US #, US (G Functions A1, A2, A3, B1



US ,

EXECUTING COMMAND

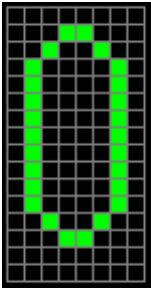
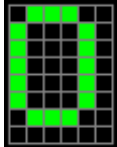
[Name]	Display comma			
[Format]	ASCII	US	,	<i>n</i>
	Hex	1F	2C	<i>n</i>
	Decimal	31	44	<i>n</i>
[Range]	32 ≤ <i>n</i> ≤ 126 and 128 ≤ <i>n</i> ≤ 255			
[Displays featuring this command]	DM-D210 and DM-D500 .			
[Displays not featuring this command]	DM-D110 .			
[Description]	Displays the specified character <i>n</i> and a comma to the right of the character.			
[Notes]	<ul style="list-style-type: none"> ■ <i>n</i> indicates a character code that can be displayed. ■ The comma is displayed only once for the specified character <i>n</i> and is not displayed for subsequent characters. ■ The cursor moves one character position to the right after displaying the comma. ■ In overwrite mode: If any other character is overwritten in the character position of character <i>n</i>, the comma is cleared. ■ In vertical scroll mode: If the display position of the character <i>n</i> moves, the comma moves with the character. ■ In horizontal scroll mode: If the display position of the character <i>n</i> moves, the comma moves with the character. ■ This command is not valid for user-defined characters. 			
[Model-dependent variations]	DM-D500			

DM-D500

Available only when 20-column/2-line mode (DM-D2xx emulation) is selected.

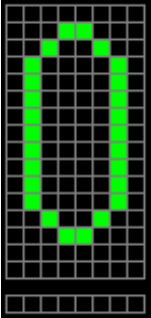
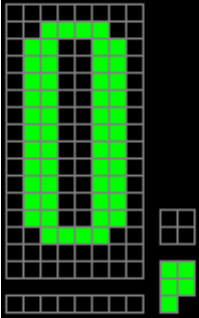


[Examples]

Graphic mode	Character display area and period
Send US , command	PRINT #1, CHR\$(31); ".0";
Graphic mode 1 font A (256 × 4 dots) and character mode 1 (32-column × 4-line mode)	 <p>Not supported</p>
Graphic mode 1 font B (256 × 64 dots) and character mode 2 (42-column × 8-line mode)	 <p>Not supported</p>



CONFIDENTIAL

<p>Character mode 3 (32-column × 3-line mode) and character mode 4 (32-column × 2-line mode)</p>	 <p>Not supported</p>
<p>Character mode 5 (20-column × 2-line mode)</p>	 <p>Supported</p>

[References]

US C, US E, US X, US r, US ., US ;, US #, US (G Functions A1, A2, A3, B1



US ;

EXECUTING COMMAND

[Name] Display semicolon

[Format]	ASCII	US	;	<i>n</i>
	Hex	1F	2B	<i>n</i>
	Decimal	31	59	<i>n</i>

[Range] $32 \leq n \leq 126$ and $128 \leq n \leq 255$

[Displays featuring this command] **DM-D210** and **DM-D500**.

[Displays not featuring this command] **DM-D110**.

[Description] Displays the specified character *n* and a semicolon to the right of the character.

- [Notes]
- *n* indicates a character code that can be displayed.
 - The semicolon is displayed only once for the specified character *n* and is not displayed for subsequent characters.
 - The cursor moves one character position to the right after displaying the semicolon.
 - In overwrite mode: If any other character is overwritten in the character position of character *n*, the semicolon is cleared.
 - In vertical scroll mode: If the display position of the character *n* moves, the semicolon moves with the character.
 - In horizontal scroll mode: If the display position of the character *n* moves, the semicolon moves with the character.
 - This command is not valid for user-defined characters.

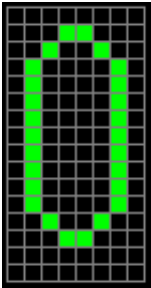
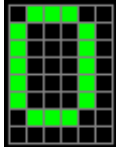
[Model-dependent variations] **DM-D500**

DM-D500

Available only when 20-column/2-line mode (DM-D2xx emulation) is selected.

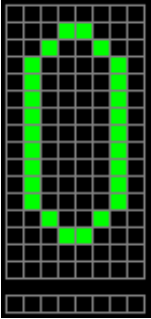
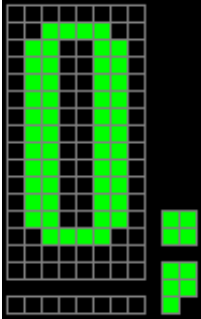


[Examples]

Graphic mode	Character display area and period
Send US ; command	PRINT #1, CHR\$(31); ".0";
Graphic mode 1 font A (256 × 4 dots) and character mode 1 (32-column × 4-line mode)	 <p>Not supported</p>
Graphic mode 1 font B (256 × 64 dots) and character mode 2 (42-column × 8-line mode)	 <p>Not supported</p>



CONFIDENTIAL

<p>Character mode 3 (32-column × 3-line mode) and character mode 4 (32-column × 2-line mode)</p>	 <p>Not supported</p>
<p>Character mode 5 (20-column × 2-line mode)</p>	 <p>Supported</p>

[References]

US C, US E, US X, US r, US ., US ,, US #, US (G Functions A1, A2, A3, B1



US

EXECUTING COMMAND

[Name] Turn annunciator on/off

[Format]	ASCII	US	#	<i>m</i>	<i>n</i>
	Hex	1F	23	<i>m</i>	<i>n</i>
	Decimal	31	35	<i>m</i>	<i>n</i>

[Range] ***m*** = 0, 1, 48, 49
 $0 \leq \mathbf{n} \leq 20$

[Default] ***n*** = 0, ***m*** = 0

[Displays featuring this command] **DM-D210** and **DM-D500**.

[Displays not featuring this command] **DM-D110**.

[Description] The setting of ***m*** turns annunciator at column ***n*** on or off.

- When ***m*** = 0 or 48, the annunciator at column ***n*** is turned off.
- When ***m*** = 1 or 49, the annunciator at column ***n*** is turned on.
- When ***n*** = 0, all the annunciators are either turned on or off, depending on the value of ***m***.

[Notes]

- ***m*** specifies the column number to be turned on or off. The left-most column is column 1.
- After you have turned an annunciator on, it remains on until any of the following occur:
 - The annunciator is turned off, using this command with ***m*** = 0.
 - The **ESC @**, **US @**, or **CLR** command is encountered.
 - The power is turned off.

[Examples]

Program example	
Print #1;CHR\$(&H1F);CHR\$(&H23);CHR\$(&1);CHR\$(3);	①
Print #1;CHR\$(&H1F);CHR\$(&H23);CHR\$(&0);CHR\$(0);	②



- The annunciator is turned on by executing ①.
- All the annunciators are turned off by executing ②.

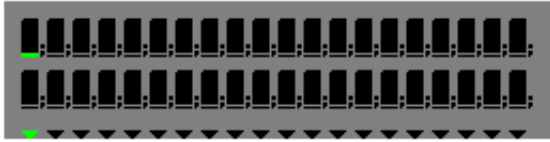
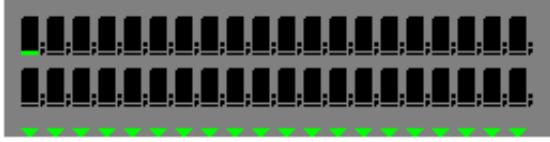
Program example for all displays	
Print #1;CHR\$(&H1F);CHR\$(&H23);CHR\$(&1);CHR\$(0);	③
Print #1;CHR\$(&H1F);CHR\$(&H23);CHR\$(&0);CHR\$(3);	④

- All the annunciators are turned on by executing ③.
- The annunciator at the third column is turned off by executing ④.

Action	Command and display sample
Before US # command is sent	
Send US # command to turn annunciator on at column 1	<p>PRINT #1, CHR\$(31); "#"; CHR\$(1); CHR\$(1);</p>
Send US # command to turn annunciator on at column 1 and column 19	<p>PRINT #1, CHR\$(31); "#"; CHR\$(1); CHR\$(19);</p>



CONFIDENTIAL

Send US # command to turn annunciator off at column 19	<pre>PRINT #1, CHR\$(31); "#"; CHR\$(0); CHR\$(19);</pre> 
Send US # command to turn all annunciators on	<pre>PRINT #1, CHR\$(31); "#"; CHR\$(1); CHR\$(0);</pre> 

[Model-dependent variations] **DM-D500**

[References] **ESC @, US @, CLR, US C, US E, US X, US r, US ., US ,, US ;, US (G** Functions A1, A2, A3, B1

DM-D500

This command is effective only when the 20-column/2-line mode is selected.



US (A

[Name]	Select display(s)											
[Format]	ASCII	US	(A	<i>pL</i>	<i>pH</i>	<i>a</i>	<i>[n</i>	<i>m]1</i>	<i>...</i>	<i>[n</i>	<i>m]k</i>
	Hex	1F	28	41	<i>pL</i>	<i>pH</i>	<i>a</i>	<i>[n</i>	<i>m]1</i>	<i>...</i>	<i>[n</i>	<i>m]k</i>
	Decimal	31	40	65	<i>pL</i>	<i>pH</i>	<i>a</i>	<i>[n</i>	<i>m]1</i>	<i>...</i>	<i>[n</i>	<i>m]k</i>
[Range]	$3 \leq (pL + pH \times 256) \leq 65535$ ($0 \leq pL \leq 255, 0 \leq pH \leq 255$) <i>a</i> = 48 <i>n</i> = 48, 49 $0 \leq m \leq 255$ $1 \leq k \leq 32767$											
[Default]	DM-D110 and DM-D210 : Display number 0 or the value set for memory switch 15 DM-D500: The device can send or receive data											
[Displays featuring this command]	DM-D110 , DM-D210 , and DM-D500 .											
[Description]	Selects the display to receive data from the host computer. <ul style="list-style-type: none"> <i>n</i> specifies whether the display is enabled or disabled. When <i>n</i> = 48, the display with the device number specified by <i>m</i> is disabled from receiving data from the host. When <i>n</i> = 49, the display with the device number specified by <i>m</i> is enabled for receiving data from the host. <i>m</i> specifies the display device number. When <i>m</i> = 0, all the displays are selected regardless of the previous-set value. When <i>m</i> ≠ 0, the display of the device number specified with <i>m</i> is selected. 											
[Reference]	ESC =											



US (B

[Name] Transmit display information

[Format]

ASCII	US	(B	<i>pL</i>	<i>pH</i>	<i>a</i>	<i>n</i>
Hex	1F	28	42	<i>pL</i>	<i>pH</i>	<i>a</i>	<i>n</i>
Decimal	31	40	66	<i>pL</i>	<i>pH</i>	<i>a</i>	<i>n</i>

[Range] $(pL + pH \times 256) = 2$ ($pL = 2, pH = 0$)
a = 48
 $0 \leq n \leq 255$

[Displays featuring this command] **DM-D500**.

[Displays not featuring this command] **DM-D110** and **DM-D210**.

[Description] Transmits the display ID specified by ***n*** as follows:

n specifies the types of display ID

<i>n</i>	Display ID type	ID
49	Display model ID	40H (64 Decimal) = DM-D500
50	Type ID	See table on the next page

n specifies the display information

<i>n</i>	Display ID type	ID
64	Boot software version	Depends on boot software version
65	Firmware version	Depends on firmware version
66	Manufacturer	"EPSON"
67	Product name	" DM-D500 "
68	Serial number	Serial number
69	Supported Kanji type	Depends on built-in Kanji type



CONFIDENTIAL

Type ID [$n = 50$]

	On/Off	Hex	Decimal	Function
0	Off	00	0	Selection of mode: 256 × 64
	On	01	1	Selection of mode: 20-column/2-line
1	Off	00	0	Default with ESC = : Display
	On	02	2	Default with ESC = : Printer
2, 3	Off	00	0	Reserved
4	Off	00	0	Fixed
5	Off	00	0	Reserved
6	On	40	64	Fixed
7	Off	00	0	Fixed

ID information and character string information is transmitted as a data block as shown below.

Transmitted data	Hexadecimal	Decimal	Number of bytes
Header	57H	87	1 byte
Flag	20H	32	1 byte
Display number	30H – 39H	48 – 57	0 – 2 bytes
Separate code	1FH	31	1 byte
Data	ID information or character strings		1 – 80 bytes
NUL	00H	0	1 byte

[Note] ■ The display can transmit information to the host only when it is connected as a stand-alone device.

[Reference] **US (E** Functions 1, 2, 3, 4



US (C

EXECUTING COMMAND

[Name]	Edit NV user memory										
[Format]	ASCII	US	(C	<i>pl</i>	<i>ph</i>	<i>m</i>	<i>fn</i>	<i>b</i>	[c1, c2]	[d1...dk]
	Hex	1F	28	43	<i>pl</i>	<i>ph</i>	<i>m</i>	<i>fn</i>	<i>b</i>	[c1, c2]	[d1...dk]
	Decimal	31	40	67	<i>pl</i>	<i>ph</i>	<i>m</i>	<i>fn</i>	<i>b</i>	[c1, c2]	[d1...dk]

[Displays featuring this command] **DM-D500**.

[Displays not featuring this command] **DM-D110** and **DM-D210**.

[Description] Deletes, stores, and transmits data in the NV user memory area, based on the functions in the table below. Also sends status of the amount of space used in NV RAM and the amount of space still available.

- The value of *fn* specifies the function.

<i>fn</i>	Function	Description
0, 48	Function 0	Deletes specified record
1, 49	Function 1	Stores data in specified record
2, 50	Function 2	Sends data stored in specified record
3, 51	Function 3	Sends number of bytes in NV user memory currently being used
4, 52	Function 4	Sends number of bytes available in NV user memory
5, 53	Function 5	Transmits the list of key codes of records stored in the NV user memory
6, 54	Function 6	Cancels all records stored in the NV user memory

- *pl* and *ph* specify the bytes following parameter *ph* (*m* and [a1 b1] ... [ak bk]) as (*pl* + (*ph* × 256)).
- The other parameters are explained under each of the functions.

- [Notes]
- The command function is defined by the value of *fn*. The operation differs, depending on the function.
 - The host can receive data from the display only if it is connected as a stand-alone device.
 - NV user memory data records are configured as follows: key code + data + terminator.



CONFIDENTIAL

- A record is one data-processing unit in the NV user memory. It is identified and specified by a key code. One record consists of a group of 4 bytes of data or more.
 - The key code is a two-byte ID code used to identify records and is created with parameters **c1**, **c2** in the command. You can specify any desired character code from 32H – 7EH.
 - Data is the character string specified by **d1... dk** in **Function 1** of the **US (C** command. You cannot specify control codes 00H – 1FH, 7FH as character string data.
 - The terminator is a one-byte code, automatically assigned when the display stores data.
- NV user memory data remains valid until the host sends a deletion or storage function command.
 - You can use the NV user memory as an index of the NV bit image definition contents.
 - The display is BUSY during processing of this command and remains BUSY while writing data into memory. During this time, do not send data to the display.
 - You cannot use ASB status and real-time commands while “header to NUL” data is being sent.
 - Frequent write commands can damage the NV memory; therefore, do not use any combination of the following commands more than 10 times per day: **US (C** and **US (E**.



CONFIDENTIAL

US (C *pL* *pH* *m* *fn* *b* *c1* *c2* <Function 0>

[Format]	ASCII	US	(C	<i>pL</i>	<i>pH</i>	<i>m</i>	<i>fn</i>	<i>b</i>	<i>c1</i>	<i>c2</i>
	Hex	1F	28	43	05	00	00	<i>fn</i>	00	<i>c1</i>	<i>c2</i>
	Decimal	31	40	67	5	0	0	<i>fn</i>	0	<i>c1</i>	<i>c2</i>

[Range] (***pL*** + ***pH*** × 256) = 5 (***pL*** = 5, ***pH*** = 0)
m = 0
fn = 0, 48
b = 0
32 ≤ ***c1*** ≤ 126
32 ≤ ***c2*** ≤ 126

[Description] Deletes the specified record stored in the NV user memory. The deleted area becomes an “unused” area available for storage.

- [Notes]
- In standard mode, this command is valid only at the beginning of the line.
 - In page mode, this command is ignored.
 - You cannot include macros with this command, so do not use this command while defining macros.



CONFIDENTIAL

US (C *p_L* *p_H* *m* *fn* *b* *c1* *c2* *d1...dk* <Function 1>

[Format]	ASCII	US	(C	<i>p_L</i>	<i>p_H</i>	<i>m</i>	<i>fn</i>	<i>b</i>	<i>c1</i>	<i>c2</i>	<i>d1...dk</i>
	Hex	1F	28	43	<i>p_L</i>	<i>p_H</i>	00	<i>fn</i>	00	<i>c1</i>	<i>c2</i>	<i>d1...dk</i>
	Decimal	31	40	67	<i>p_L</i>	<i>p_H</i>	0	<i>fn</i>	0	<i>c1</i>	<i>c2</i>	<i>d1...dk</i>
[Range]	$6 \leq (p_L + p_H \times 256) \leq 65535$ $(0 \leq p_L \leq 255, 0 \leq p_H \leq 255)$ <i>m</i> = 0 <i>fn</i> = 1, 49 <i>b</i> = 0 $32 \leq c1 \leq 126$ $32 \leq c2 \leq 126$ $32 \leq d \leq 254$ $k = (p_L + p_H \times 256) - 5$											
[Description]	Stores data (<i>d1... dk</i>) in the record specified by parameters <i>c1</i> and <i>c2</i> (the key code ID numbers). <ul style="list-style-type: none">• When the specified record already exists, the data is overwritten.• A terminator is automatically assigned.											
[Notes]	<ul style="list-style-type: none">■ In standard mode, this command is valid only at the beginning of the line.■ In page mode, this command is ignored.■ You cannot include macros with this command, so do not use this command while defining macros.■ This function uses the "data quantity (<i>k</i>) + 3 byte" area for execution.■ NV user memory data remains valid until the host sends a deletion or another storage function command.											



US (C *pL* *pH* *m* *fn* *b* *c1* *c2* <Function 2>

[Format]	ASCII	US	(C	<i>pL</i>	<i>pH</i>	<i>m</i>	<i>fn</i>	<i>b</i>	<i>c1</i>	<i>c2</i>
	Hex	1F	28	43	05	00	00	<i>fn</i>	00	<i>c1</i>	<i>c2</i>
	Decimal	31	40	67	5	0	0	<i>fn</i>	0	<i>c1</i>	<i>c2</i>

[Range] **(*pL* + *pH* × 256) = 5 (*pL* = 5, *pH* = 0)**
***m* = 0**
***fn* = 2, 50**
***b* = 0**
32 ≤ *c1* ≤ 126
32 ≤ *c2* ≤ 126

[Description] Transmits the data for the record with the ID code specified by parameters ***c1***, ***c2*** in the NV user memory.
 • Handshake control is required for this function.

[Notes] ■ When the specified record exists, the printer sends the “header to NUL” data shown below:

Send Data	Hex	Decimal	Data Quantity
Header	57H	87	1 byte
Identifier	70H	112	1 byte
Display number	30 – 39H	48 – 57	0 – 2 bytes
Separate code	1F	31	1 byte
Status ^(*1) ^(*2)	40H or 41H	64 or 65	1 byte
Data ^(*3)	20H – FEH	32 – 254	1 – 80 bytes
NUL	00H	0	1 byte

(*1) If there are more than 80 bytes of stored data, the display performs partial processing as follows:
 • If there is unsent data, the status of the third byte is 41H or 65 decimal.
 • If there is no unsent data, the status of the third byte is 40H or 64 decimal.



- (*2) The display performs batch processing if data to be stored in the specified record is 80 bytes or less.
The status of the third byte is 40H or 64 decimal.
- (*3) The key code and terminator are not included in the data.
- If the data in the specified record is abnormal, the display transmits the following data:

Send Data	Hex	Decimal	Data Quantity
Header	57H	87	1 byte
Identifier	70H	112	1 byte
Display number	30 – 39H	48 – 57	0 – 2 bytes
Separate code	1FH	31	1 byte
Status	40H	64	1 byte
Data	FFH	255	1 byte
NUL	00H	0	1 byte

- If memory data is abnormal, the display transmits the following data:

Send Data	Hex	Decimal	Data Quantity
Header	57H	87	1 byte
Identifier	70H	112	1 byte
Display number	30 – 39H	48 – 57	0 – 2 bytes
Separate code	1FH	31	1 byte
Status	40H	64	1 byte
NUL	00H	0	1 byte

- The host responds, and the display sends one of the following replies, depending on the host's response.



CONFIDENTIAL

If the status is 41H or 65 decimal:

Host response		Display action
ASCII	Decimal	
ACK	6	Transmits next data block
NAK	21	Transmits previous data block again
CAN	24	Quits processing the command

If the status is 40H or 64 decimal:

Host response		Display action
ASCII	Decimal	
ACK	6	Ends command processing
NAK	21	Transmits previous data block again
CAN	24	Quits processing the command



US (C *pL* *pH* *m* *fn* *b* <Function 3>

[Format]	ASCII	US	(C	<i>pL</i>	<i>pH</i>	<i>m</i>	<i>fn</i>	<i>b</i>
	Hex	1F	28	43	03	00	00	<i>fn</i>	00
	Decimal	31	40	67	3	0	0	<i>fn</i>	0

[Range] **(*pL* + *pH* × 256) = 3** (***pL* = 3, *pH* = 0**)
***m* = 0**
***fn* = 3, 51**
***b* = 0**

[Description] The display sends the host the number of data bytes currently used in the NV memory area.
 • Handshake control is not required for this function.

[Notes] ■ With this function, the printer sends the “Header to NUL” data shown below:

Send Data	Hex	Decimal	Data Quantity
Header	57H	87	1 byte
Identifier	21H	33	1 byte
Display number	30 – 39H	48 – 57	0 – 2 bytes
Separate code	1F	31	1 byte
Capacity Used ^(*1)	30H – 39H	48 – 57	1 – 8 bytes
NUL	00H	0	1 byte

(*1) The quantity of stored data bytes, plus the key code and terminator, equal the capacity being used.

■ The decimal value expressing the capacity being used is converted to ASCII character data and is sent from the most significant digit.

Example: When 120 bytes are used, the number 120 is expressed with three bytes of data (decimal numbers 49, 50, 48).
 If no memory area is used, the number 0 is expressed with 1 byte of data (decimal number 48).



CONFIDENTIAL

- The host can differentiate the numbers for NV user memory “capacity being used” from other data by the specific information in the send data block. When the header sent from the display is 57H or 87 decimal, the data up to NUL, 00H, or 0 decimal, is handled as one group and can be identified by the following data:

Send Data	Hex	Decimal
1st byte (header)	57H	87
2nd byte (identifier)	21H	33



US (C *pL* *pH* *m* *fn* *b* <Function 4>

[Format]	ASCII	US	(C	<i>pL</i>	<i>pH</i>	<i>m</i>	<i>fn</i>	<i>b</i>
	Hex	1F	28	43	03	00	00	<i>fn</i>	00
	Decimal	31	40	67	3	0	0	<i>fn</i>	0

[Range] **(*pL* + *pH* × 256) = 3 (*pL* = 3, *pH* = 0)**
***m* = 0**
***fn* = 4, 52**
***b* = 0**

[Description] The display sends the available NV user memory.

- Handshake control is not required for this function.

[Notes] ■ With this function, the printer sends the “Header to NUL” data shown below:

Send Data	Hex	Decimal	Data Quantity
Header	57H	87	1 byte
Identifier	22H	34	1 byte
Display number	30 – 39H	48 – 57	0 – 2 bytes
Separate code	1F	31	1 byte
Capacity Used ^(*1)	30H – 39H	48 – 57	1 – 8 bytes
NUL	00H	0	1 byte

(*1) The available capacity indicates the number of bytes not being used.

- The decimal value for the available capacity is converted to ASCII character data and is sent from the most significant digit.

Example: When 120 bytes are available (not being used), the number 120 is expressed with three bytes of data (decimal numbers = 49, 50, 48).



CONFIDENTIAL

- The host can differentiate the numbers for available NV user memory capacity from other sent data by the specific information in the data block. When the header sent from the display is 37H or 55 decimal, the data up to NUL (00H or 0 decimal) is handled as one group and can be identified using the following data:

Send Data	Hex	Decimal
1st byte (header)	57H	87
2nd byte (identifier)	22H	34



US (C *p_L* *p_H* *m* *fn* *b* <Function 5>

[Format]	ASCII	US	(C	<i>p_L</i>	<i>p_H</i>	<i>m</i>	<i>fn</i>	<i>b</i>
	Hex	1F	28	43	03	00	00	<i>fn</i>	00
	Decimal	31	40	67	3	0	0	<i>fn</i>	0

[Range] **(*p_L* + *p_H* × 256) = 3** (***p_L* = 3, *p_H* = 0**)
***m* = 0**
***fn* = 5, 53**
***b* = 0**

[Description] Transmits the key code ID list for all records stored in NV user memory.

[Notes] ■ When stored NV data records exist, the display sends the “header to NUL” data shown below:

Send Data	Hex	Decimal	Data Quantity
Header	57H	87	1 byte
Identifier	22H	34	1 byte
Display number	30 – 39H	48 – 57	0 – 2 bytes
Separate code	1F	31	1 byte
Status	40H or 41H	64 or 65	1 byte
Data ^(*1)	20H – FEH	32 – 254	2 – 80 bytes
NUL	00H	0	1 byte

(*1) Data is the key code data string.



- If no stored records exist, the display sends the “Header to NUL” data shown below:

Send Data	Hex	Decimal	Data Quantity
Header	57H	87	1 byte
Identifier	71H	113	1 byte
Display number	30 – 39H	48 – 57	0 – 2 bytes
Separate code	1FH	31	1 byte
Status	40H	64	1 byte
NUL	00H	0	1 byte

- The host responds, and the display sends one of the following replies, depending on the host’s response.
If the status is 41H or 65 decimal:

Host response		Display action
ASCII	Decimal	
ACK	6	Transmits next data block
NAK	21	Transmits previous data block again
CAN	24	Quits processing the command

If the status is 40H or 64 decimal:

Host response		Display action
ASCII	Decimal	
ACK	6	Ends command processing
NAK	21	Transmits previous data block again
CAN	24	Quits processing the command



US (C *pL* *pH* *m* *fn* *b* *d1* *d2* *d3* <Function 6>

[Format]	ASCII	US	(C	<i>pL</i>	<i>pH</i>	<i>m</i>	<i>fn</i>	<i>b</i>	<i>d1</i>	<i>d2</i>	<i>d3</i>
	Hex	1F	28	43	06	00	00	<i>fn</i>	00	<i>d1</i>	<i>d2</i>	<i>d3</i>
	Decimal	31	40	67	6	0	0	<i>fn</i>	0	<i>d1</i>	<i>d2</i>	<i>d3</i>



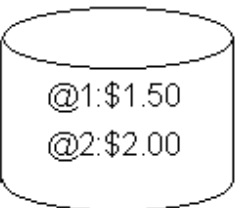
[Range] **(*pL* + *pH* × 256) = 6** **(*pL* = 6, *pH* = 0)**
***m* = 0**
***fn* = 6, 54**
***b* = 0**
***d1* = 67**
***d2* = 76**
***d3* = 82**

[Description] The display deletes all records stored in the NV user memory.
 • The entire area is changed to unused area.

[Note] ■ This command cannot be included in a macro, so do not use this command when defining a macro.

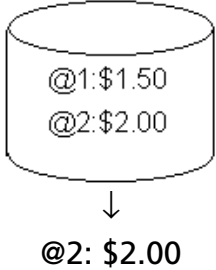




[Examples of using functions]

Action	Commands	NV User Memory
<p>Function 1 Store data as a record with the specified key code name</p>	<pre>PRINT #1, CHR\$(31); "(C"; 'US(C Command PRINT #1, CHR\$(10); CHR\$(0); 'pL pH PRINT #1, CHR\$(0); 'm PRINT #1, CHR\$(1); 'fn PRINT #1, CHR\$(0); 'b PRINT #1, "@1"; 'c1 c2 (key code) PRINT #1, "\$1.50"; 'data</pre>	<p>@1: \$1.50 ↓ </p>
<p>Function 1 Store data as a record with the specified key code name</p>	<pre>PRINT #1, CHR\$(31); "(C"; 'US(C Command PRINT #1, CHR\$(10); CHR\$(0); 'pL pH PRINT #1, CHR\$(0); 'm PRINT #1, CHR\$(1); 'fn PRINT #1, CHR\$(0); 'b PRINT #1, "@2"; 'c1 c2 (key code) PRINT #1, "\$2.00"; 'data</pre>	<p>@2: \$@.00 ↓ </p>
<p>Function 2 Transmit the data stored as the record with the specified key code name</p>	<pre>PRINT #1, CHR\$(31); "(C"; 'US(C Command PRINT #1, CHR\$(5); CHR\$(0); 'pL pH PRINT #1, CHR\$(0); 'm PRINT #1, CHR\$(2); 'fn PRINT #1, CHR\$(0); 'b PRINT #1, "@1"; 'c1 c2 (key code)</pre>	<p> ↓ @1: \$1.50</p>
<p>Display sends the record with the key code @1 back to the host</p>		



CONFIDENTIAL

Action	Commands	NV User Memory
<p>Function 2 To get the next block of data from the display, the host must send it an ACK</p>	<p>PRINT #1, CHR\$(6);</p>	
<p>Display sends the record with the key code @2 back to the host.</p>		
<p>Function 2 If you do not need any further data from the display, the host can send the display a CAN</p>	<p>PRINT #1, CHR\$(24);</p>	
<p>Function 0 Delete specified record</p>	<p>PRINT #1, CHR\$(31); "(C"; 'US(C Command PRINT #1, CHR\$(5); CHR\$(0); 'pL pH PRINT #1, CHR\$(0); 'm PRINT #1, CHR\$(0); 'fn PRINT #1, CHR\$(0); 'b PRINT #1, "@1"; 'c1 c2 (key code)</p>	
<p>Function 6 Delete all data in the NV user memory</p>	<p>PRINT #1, CHR\$(31); "(C"; 'US(C Command PRINT #1, CHR\$(6); CHR\$(0); 'pL pH PRINT #1, CHR\$(0); 'm PRINT #1, CHR\$(6); 'a PRINT #1, CHR\$(0); 'b PRINT #1, "CLR"; 'd1 d2 d3</p>	

[References]

US (A, US (E Functions 3 and 4



[Name]	Select window control									
[Format]	ASCII	US	(D	PL	PH	fn	wno	[parameters]	
	Hex	1F	28	44	PL	PH	fn	wno	[parameters]	
	Decimal	31	40	68	PL	PH	fn	wno	[parameters]	

[Displays featuring this command] **DM-D500**.

[Displays not featuring this command] **DM-D110** and **DM-D210**.

[Description] Creates and controls windows on the display.

- The value of **fn** specifies the function.

fn	Function	Description
1, 49	Function 1	Defines a window
2, 50	Function 2	Deletes a window
3, 51	Function 3	Selects the display mode for the current window
4, 52	Function 4	Selects the current window
6, 54	Function 6	Moves the display location

- PL** and **PH** specify the bytes following parameter **PH** (**fn**, **wno** and **[parameters]**) as $(PL + (PH \times 256))$.
- The other parameters are explained under each of the functions.

[Note] ■ The command function is defined by the value of **fn**. The operation differs, depending on the function.



US (D *p_L* *p_H* *fn* *wno* *m1* *m2* *m3* *x_L* *x_H* *y_L* *y_H* *dx_L* *dx_H* *dy_L* *dy_H* <Function 1>

[Format]	ASCII	US (D	<i>p_L</i>	<i>p_H</i>	<i>fn</i>	<i>wno</i>	<i>m1</i>	<i>m2</i>	<i>m3</i>	<i>x_L</i>	<i>x_H</i>	<i>y_L</i>	<i>y_H</i>	<i>dx_L</i>	<i>dx_H</i>	<i>dy_L</i>	<i>dy_H</i>
	Hex	1F 28 44	0D	00	01	<i>wno</i>	<i>m1</i>	<i>m2</i>	<i>m3</i>	<i>x_L</i>	<i>x_H</i>	<i>y_L</i>	<i>y_H</i>	<i>dx_L</i>	<i>dx_H</i>	<i>dy_L</i>	<i>dy_H</i>
	Decimal	31 40 68	13	0	1	<i>wno</i>	<i>m1</i>	<i>m2</i>	<i>m3</i>	<i>x_L</i>	<i>x_H</i>	<i>y_L</i>	<i>y_H</i>	<i>dx_L</i>	<i>dx_H</i>	<i>dy_L</i>	<i>dy_H</i>

[Range] **(*p_L* + *p_H* × 256) = 13** (***p_L* = 13, *p_H* = 0**)
***fn* = 1**
1 ≤ *wno* ≤ 4
0 ≤ *m1* ≤ 255
***m2* = 0, 1**
***m3* = 2**
0 ≤ (*x_L* + *x_H* × 256) ≤ 65534 (0 ≤ *x_L* ≤ 255, 0 ≤ *x_H* ≤ 255)
0 ≤ (*y_L* + *y_H* × 256) ≤ 65534 (0 ≤ *y_L* ≤ 255, 0 ≤ *y_H* ≤ 255)
1 ≤ (*dx_L* + *dx_H* × 256) ≤ 65535 (0 ≤ *dx_L* ≤ 255, 0 ≤ *dx_H* ≤ 255)
1 ≤ (*dy_L* + *dy_H* × 256) ≤ 65535 (0 ≤ *dy_L* ≤ 255, 0 ≤ *dy_H* ≤ 255)

[Default] Window is undefined.

[Description] Defines a new window with the window number, position, and size specified.

- ***m1*** specifies the window display mode.

<i>m1</i>	Mode	Display	Usable fonts
65	Graphic mode 1	256 × 64 dots	All supported fonts
97	Character mode 1	32 columns × 4 lines	Font A, Kanji
98	Character mode 2	42 columns × 8 lines	Font B
99	Character mode 3	32 columns × 3 lines	Font A, Kanji
100	Character mode 4	32 columns × 2 lines	Font A, Kanji
101	Character mode 5	20 columns × 2 lines	Same as 20-column/2-line mode



CONFIDENTIAL

- **m2** specifies the window background.
0 = transparent (no background)
1 = nontransparent (background)
- **XL, XH, YL, YH** specify the origin of the window.
In graphics mode, this is (**XL** + **XH** × 256) dots over horizontally and (**YL** + **YH** × 256) dots down vertically from the absolute origin of the display.
In character mode, this is (**XL** + **XH** × 256) columns over and (**YL** + **YH** × 256) rows down from the absolute origin of the display.
- **dxL, dxH, dyL, dyH** specify the size of the window.
In graphics mode, the four parameters above specify a point (**dxL** + **dxH** × 256) dots over from the window origin and (**dyL** + **dyH** × 256) dots down from the window origin. The diagonal line from window origin to this point defines the size of the window.
In character mode, the four parameters above specify a point (**dxL** + **dxH** × 256) columns over from the window origin and (**dyL** + **dyH** × 256) rows down from the window origin. The diagonal line from window origin to this point defines the size of the window.
- The origin is the starting point of the window.
- The cursor moves to the origin of the window.



CONFIDENTIAL

US (D *p_L* *p_H* *fn* *wno* <Function 2>

[Format]	ASCII	US	(D	<i>p_L</i>	<i>p_H</i>	<i>fn</i>	<i>wno</i>
	Hex	1F	28	44	02	00	02	<i>wno</i>
	Decimal	31	40	68	2	0	2	<i>wno</i>
[Range]	$(p_L + p_H \times 256) = 2$ $(p_L = 2, p_H = 0)$ <i>fn</i> = 2 $1 \leq wno \leq 4$							
[Description]	Deletes the window number specified with <i>wno</i> .							



US (D *p_L* *p_H* *fn* *wno* *m1* *m2* *m3* <Function 3>

[Format]	ASCII	US	(D	<i>p_L</i>	<i>p_H</i>	<i>fn</i>	<i>wno</i>	<i>m1</i>	<i>m2</i>	<i>m3</i>
	Hex	1F	28	44	04	00	03	<i>wno</i>	<i>m1</i>	<i>m2</i>	<i>m3</i>
	Decimal	31	40	68	4	0	3	<i>wno</i>	<i>m1</i>	<i>m2</i>	<i>m3</i>

[Range] **(*p_L* + *p_H* × 256) = 4** **(*p_L* = 4, *p_H* = 0)**
***fn* = 3**
1 ≤ *wno* ≤ 4
0 ≤ *m1* ≤ 255
***m2* = 0, 1**
***m3* = 2**

[Default] Base window: ***m1* = 65, *m2* = 1**
 Other windows: State defined with **Function 1**.

[Description] Specifies the display mode for the current window.

- If any window except the base window is selected, only ***m1* = 0** (does not change the display mode) is effective.

<i>m1</i>	Mode	Display
65	Graphic mode 1	256 × 64 dots
97	Character mode 1	32 columns × 4 lines
98	Character mode 2	42 columns × 8 lines
99	Character mode 3	32 columns × 3 lines
100	Character mode 4	32 columns × 2 lines
101	Character mode 5	20 columns × 2 lines

- ***m2*** specifies the window background.
 0 = transparent (no background)
 1 = nontransparent (background)



CONFIDENTIAL

- If display mode is changed with **m1**, the display performs the following operations:
Clears the data in the current window.
Specifies the home position point as the origin point of the window.
Moves the cursor to the origin point of the window.



US (D *p_L* *p_H* *fn* *wno* *m* <Function 4>

[Format]	ASCII	US	(D	<i>p_L</i>	<i>p_H</i>	<i>fn</i>	<i>wno</i>	<i>m</i>
	Hex	1F	28	44	03	00	04	<i>wno</i>	<i>m</i>
	Decimal	31	40	68	3	0	4	<i>wno</i>	<i>m</i>

[Range] **(*p_L* + *p_H* × 256) = 3** (***p_L* = 3, *p_H* = 0**)
***fn* = 4**
1 ≤ *wno* ≤ 4
***m* = 0, 1, 48, 49**

[Default] ***wno* = 0, *m* = 0**

- [Description]
- Specifies the window named with ***wno*** as the current window.
 When ***wno* ≠ 0**, the window number specified by ***wno*** is selected.
 When ***wno* = 0**, the base window is selected.
 - ***m*** specifies the process for the current window, as follows:
 When ***m* = 0** or **48**, the current window is specified to receive subsequent data.
 When ***m* = 1** or **49**, the current window is specified to receive and display subsequent data, and it becomes the front-most window.
 - The cursor moves to the current position in the current window.



US (D *p_L* *p_H* *fn* *m* *nL* *nH* <Function 6>

[Format]	ASCII	US	(D	<i>p_L</i>	<i>p_H</i>	<i>fn</i>	<i>m</i>	<i>nL</i>	<i>nH</i>
	Hex	1F	28	44	04	00	06	m	nL	nH
	Decimal	31	40	68	4	0	6	m	nL	nH

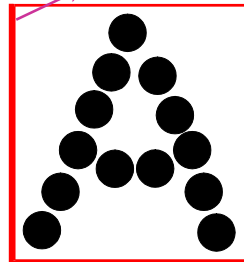
[Range] **(*p_L* + *p_H* × 256) = 4 (*p_L* = 4, *p_H* = 0)**
***fn* = 6**
***m* = 48, 49, 50, 51**
 When ***m* = 48 or 49:**
 $0 \leq (*nL* + *nH* \times 256) \leq 65535$ ($0 \leq *nL* \leq 255, 0 \leq *nH* \leq 255$)
 When ***m* = 50 or 51:**
 $-32768 \leq (*nL* + *nH* \times 256) \leq 32767$ ($0 \leq *nL* \leq 255, 0 \leq *nH* \leq 255$)

[Description] Moves the current display position.

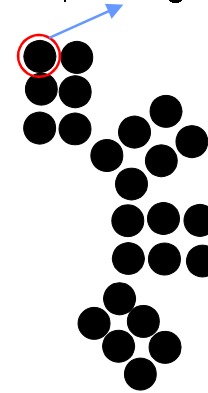
<i>m</i>	Type of movement	Basic position	Direction of movement
48	Absolute move	Starting point	Horizontal to left side of character or top left dot of graphic
49			Vertical from left side of character or top left dot of graphic
50	Relative move	Current display point	Horizontal to current character or current dot
51			Vertical to current character or current dot



Horizontal from left edge
of character unit.



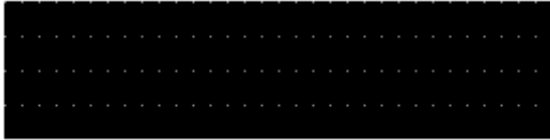

Horizontal or vertical from
this point in graphic.



- A positive number specifies movement downward, and a negative number specifies movement upward; for example, using **y** to represent units moved downward, $(nL + nH \times 256) = y$. To set movement upward, use the complement of **y**: $(nL + nH \times 256) = 65536 - y$.
- If the current window is in graphics mode, the command moves the cursor $(nL + nH \times 256)$ dots.
- If the current window is in character mode, the command moves the cursor $(nL + nH \times 256)$ columns or $(nL + nH \times 256)$ lines.




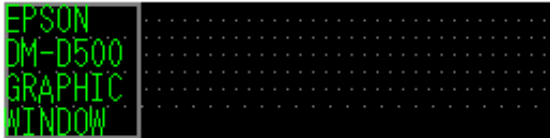
[Examples]

Action	Command/Example
Send ESC @ command to initialize display	<pre data-bbox="1150 251 1528 289">PRINT #1, CHR\$(27); "@";</pre> 
<p data-bbox="388 781 562 813">Function 1</p> <p data-bbox="388 824 884 898">Define window in graphics mode (Window 1)</p>	<pre data-bbox="1150 505 1724 987">PRINT #1, CHR\$(31); "(D"; PRINT #1, CHR\$(13); CHR\$(0); 'pL pH PRINT #1, CHR\$(1); 'fn PRINT #1, CHR\$(1); 'wno PRINT #1, CHR\$(65); 'm1 PRINT #1, CHR\$(1); 'm2 PRINT #1, CHR\$(2); 'm3 PRINT #1, CHR\$(0); CHR\$(0); 'xL xH PRINT #1, CHR\$(0); CHR\$(0); 'yL yH PRINT #1, CHR\$(64); CHR\$(0); 'dxL dxH PRINT #1, CHR\$(64); CHR\$(0); 'dyL dyH</pre> 

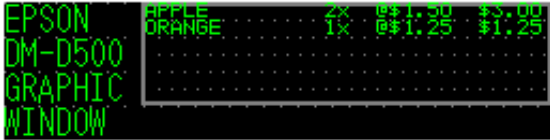


<p>Function 1 Define window in character mode (Window 2)</p>	<pre>PRINT #1, CHR\$(31); "(D"; PRINT #1, CHR\$(13); CHR\$(0); 'pL pH PRINT #1, CHR\$(1); 'fn PRINT #1, CHR\$(2); 'wno PRINT #1, CHR\$(98); 'm1 PRINT #1, CHR\$(1); 'm2 PRINT #1, CHR\$(2); 'm3 PRINT #1, CHR\$(12); CHR\$(0); 'xL xH PRINT #1, CHR\$(1); CHR\$(0); 'yL yH PRINT #1, CHR\$(31); CHR\$(0); 'dxL dxH PRINT #1, CHR\$(6); CHR\$(0); 'dyL dyH</pre>
<p>Function 1 Define window in character mode (Window 3)</p>	<div data-bbox="1276 691 1822 829" data-label="Image"> </div> <pre>PRINT #1, CHR\$(31); "(D"; PRINT #1, CHR\$(13); CHR\$(0); 'pL pH PRINT #1, CHR\$(1); 'fn PRINT #1, CHR\$(3); 'wno PRINT #1, CHR\$(97); 'm1 PRINT #1, CHR\$(1); 'm2 PRINT #1, CHR\$(2); 'm3 PRINT #1, CHR\$(9); CHR\$(0); 'xL xH PRINT #1, CHR\$(4); CHR\$(0); 'yL yH PRINT #1, CHR\$(24); CHR\$(0); 'dxL dxH PRINT #1, CHR\$(1); CHR\$(0); 'dyL dyH</pre>

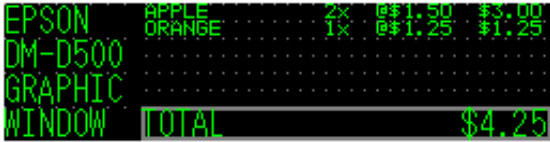
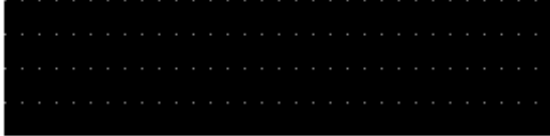


<p>Function 1 Define window in character mode (Window 3) (continued)</p>	
<p>Function 4 Select Window 1 (wno = 1) and display text</p>	<pre>PRINT #1, CHR\$(31); "(D"; PRINT #1, CHR\$(3); CHR\$(0); 'pL pH PRINT #1, CHR\$(4); 'fn PRINT #1, CHR\$(1); 'wno PRINT #1, CHR\$(1); 'm PRINT #1, "EPSON" PRINT #1, "DM-D500 GRAPHIC WINDOW";</pre> 
<p>Function 4 Select Window 2 (wno = 2) and display text</p>	<pre>PRINT #1, CHR\$(31); "(D"; PRINT #1, CHR\$(3); CHR\$(0); 'pL pH PRINT #1, CHR\$(4); 'fn PRINT #1, CHR\$(2); 'wno PRINT #1, CHR\$(1); 'm PRINT #1, CHR\$(31); "\$"; CHR\$(1); CHR\$(1); PRINT #1, "APPLE"; PRINT #1, CHR\$(31); "\$";</pre>

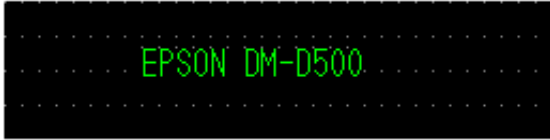



<p>Function 4 Select Window 2 (wno = 2) and display text (continued)</p>	<pre>CHR\$(15); CHR\$(1); PRINT #1, "2x @\$1.50 \$3.00"; PRINT #1, CHR\$(31); "\$"; CHR\$(1); CHR\$(2); PRINT #1, "ORANGE"; PRINT #1, CHR\$(31); "\$"; CHR\$(15); CHR\$(2); PRINT #1, "1x @\$1.25 \$1.25";</pre> 
<p>Function 4 Select Window 3 (wno = 3) and display text</p>	<pre>PRINT #1, CHR\$(31); "(D"; PRINT #1, CHR\$(3); CHR\$(0); 'pL pH PRINT #1, CHR\$(4); 'fn PRINT #1, CHR\$(3); 'wno PRINT #1, CHR\$(1); 'm PRINT #1, CHR\$(31); "\$"; CHR\$(1); CHR\$(1); PRINT #1, "TOTAL"; PRINT #1, CHR\$(31); "\$"; CHR\$(20); CHR\$(1); PRINT #1, "\$4.25";</pre>





<p>Function 4 Select Window 3 (<i>wno</i> = 3) and display text (continued)</p>	
<p>Function 4 Select the base window (Window 0)</p>	<pre>PRINT #1, CHR\$(31); "(D"; PRINT #1, CHR\$(3); CHR\$(0); 'pL pH PRINT #1, CHR\$(4); 'fn PRINT #1, CHR\$(0); 'wno PRINT #1, CHR\$(1); 'm</pre> 
<p>Function 6 Move the display location and display text</p>	<pre>PRINT #1, CHR\$(31); "(D"; PRINT #1, CHR\$(4); CHR\$(0); 'pL pH PRINT #1, CHR\$(6); 'fn PRINT #1, CHR\$(48); 'm PRINT #1, CHR\$(64); CHR\$(0); 'nL nH PRINT #1, CHR\$(31); "(D"; PRINT #1, CHR\$(4); CHR\$(0); 'pL pH PRINT #1, CHR\$(6); 'fn PRINT #1, CHR\$(49); 'm PRINT #1, CHR\$(20); CHR\$(0); 'nL nH PRINT #1, "EPSON DM-D500";</pre>



<p>Function 6 Move the display location and display text (continued)</p>	
<p>Function 3 Select the display mode for the current window (Window 0, the base window)</p>	<pre>PRINT #1, CHR\$(31); "(D"; PRINT #1, CHR\$(4); CHR\$(0); 'pL pH PRINT #1, CHR\$(3); 'fn PRINT #1, CHR\$(101); 'm1 PRINT #1, CHR\$(1); 'm2 PRINT #1, CHR\$(2); 'm3 PRINT #1, " EPSON DM-D500";</pre>
	
<p>Function 4 Select Window 2 (wno = 2) and Function 3 make background transparent</p>	<pre>PRINT #1, CHR\$(31); "(D"; PRINT #1, CHR\$(3); CHR\$(0); 'pL pH PRINT #1, CHR\$(4); 'fn PRINT #1, CHR\$(2); 'wno PRINT #1, CHR\$(1); 'm PRINT #1, CHR\$(31); "(D"; PRINT #1, CHR\$(4); CHR\$(0); 'pL pH PRINT #1, CHR\$(3); 'fn PRINT #1, CHR\$(0); 'm1</pre>



CONFIDENTIAL

<p>Function 4 Select Window 2 (wno = 2) and Function 3 make background transparent (continued)</p>	<pre>PRINT #1, CHR\$(0); 'm2 PRINT #1, CHR\$(2); 'm3</pre> 
<p>Function 4 Select Window 1 (wno = 1)</p>	<pre>PRINT #1, CHR\$(31); "(D"; PRINT #1, CHR\$(3); CHR\$(0); 'pL pH PRINT #1, CHR\$(4); 'fn PRINT #1, CHR\$(1); 'wno PRINT #1, CHR\$(1); 'm</pre> 

[References]

CLR, CAN, US \$, ESC W, US MD1, US MD2, US MD3, US C, US T, US U, US r, US (G Functions A1, A2, A3, B1, **US (H** Functions A1, A2, B1, C1



US (E

EXECUTING COMMAND

[Name]	User setting commands							
[Format]	ASCII	US	(E	PL	PH	fn	[parameters]
	Hex	1F	28	45	PL	PH	fn	[parameters]
	Decimal	31	40	69	PL	PH	fn	[parameters]

[Displays featuring this command] **DM-D110**, **DM-D210**, and **DM-D500**.

[Description] Executes user setting commands.

- The value of **fn** specifies the function.

fn	Function	Description
1	Function 1	Changes into user setting mode
2	Function 2	Ends user setting mode and performs a software reset
3	Function 3	Sets values for the memory switches
4	Function 4	Transmits settings of the memory switches back to the host
7	Function 7	DM-D500 only: Copies the data stored in the user-defined code page
8	Function 8	DM-D500 only: Defines the character pattern of the user-defined code page in the work area with column type
9	Function 9	DM-D500 only: Defines the character pattern of the user-defined code page in the work area with raster type
10	Function 10	DM-D500 only: Deletes the data in the character code table in the work area.
11	Function 11	DM-D500 only: Defines NV bit image

- **PL** and **PH** specify the number of bytes following parameter **fn** as $(PL + (PH \times 256))$.
- The other parameters are explained under each of the functions.

[Notes] ■ The command function is defined by the value of **fn**. The operation differs, depending on the function.



CONFIDENTIAL

- The customer display must be in the user setting mode before this command can change the values in NV memory.
- In **Function 2**, the customer display performs a software reset; therefore, the customer display clears the receive buffer and resets all settings (user-defined characters, macros, windows, etc.), and the display returns to the mode in effect at power on.
- Using **Function 4**, the customer display can read back all customized values set for the memory switches with this command, even when it is not changed into the user setting mode.
- The display can transmit data back to the host PC only when it is connected as a stand-alone device.
- During processing of this command, the customer display is BUSY while writing data to NV memory, and it stops receiving data; therefore, do not transmit data to the display while the display is BUSY.
- Frequent write commands to NV memory may damage the NV memory.
For the **DM-D500**, do not write to NV memory more than 10 times per day.
For the **DM-D110** and **DM-D210**, do not write to NV memory more than 50 times per day.



CONFIDENTIAL

US (E *p_L* *p_H* *fn* *d1* *d2* <Function 1>

[Format]	ASCII	US	(E	<i>p_L</i>	<i>p_H</i>	<i>fn</i>	<i>d1</i>	<i>d2</i>
	Hex	1F	28	45	03	00	01	d1	d2
	Decimal	31	40	69	3	0	1	d1	d2

[Range] $(p_L + p_H \times 256) = 3$ ($p_L = 3, p_H = 0$)
fn = 1
d1 = 73 (ASCII I)
d2 = 78 (ASCII N)

[Description] Changes the display into the user setting mode and sends the following data.

Transmitted data	Hexadecimal	Decimal	Number of bytes
Header	57H	87	1 byte
Flag	23H	35	1 byte
Display number	30H – 39H	48 – 57	0 – 2 bytes
Separate code	1FH	31	1 byte
NUL	00H	0	1 byte



CONFIDENTIAL

US (E *p_L* *p_H* *fn* *d1* *d2* *d3* <Function 2>

[Format]	ASCII	US	(E	<i>p_L</i>	<i>p_H</i>	<i>fn</i>	<i>d1</i>	<i>d2</i>	<i>d3</i>
	Hex	1F	28	45	04	00	02	d1	d2	d3
	Decimal	31	40	69	4	0	2	d1	d2	d3

[Range] **(*p_L* + *p_H* × 256) = 4** (***p_L* = 4, *p_H* = 0**)
***fn* = 2**
***d1* = 79 (ASCII O)**
***d2* = 85 (ASCII U)**
***d3* = 84 (ASCII T)**

[Description] Ends user setting mode and performs a soft reset.

- [Notes]
- This function is performed only when the display is in user setting mode.
 - The settings you specified while in the user setting mode are activated by sending a Function 2 command.
 - Function 2 performs a software reset that returns the printer to the state it is in at power on.
 - This function clears the buffers and resets all settings, including user-defined characters, downloaded bit images, macros, and the display mode.



US (E *p_L* *p_H* *fn* [*a*₁ *b*_{1₈} *b*_{1₁}] ... [*a*_{*k*} *b*_{*k*₈} *b*_{*k*₁}] <Function 3>

[Format]	ASCII	US	(E	<i>p_L</i>	<i>p_H</i>	<i>fn</i>	[<i>a</i> ₁ <i>b</i> _{1₈} <i>b</i> _{1₁}] ... [<i>a</i> _{<i>k</i>} <i>b</i> _{<i>k</i>₈} <i>b</i> _{<i>k</i>₁}]
	Hex	1F	28	45	<i>p_L</i>	<i>p_H</i>	03	[<i>a</i> ₁ <i>b</i> _{1₈} <i>b</i> _{1₁}] ... [<i>a</i> _{<i>k</i>} <i>b</i> _{<i>k</i>₈} <i>b</i> _{<i>k</i>₁}]
	Decimal	31	40	69	<i>p_L</i>	<i>p_H</i>	3	[<i>a</i> ₁ <i>b</i> _{1₈} <i>b</i> _{1₁}] ... [<i>a</i> _{<i>k</i>} <i>b</i> _{<i>k</i>₈} <i>b</i> _{<i>k</i>₁}]

[Range] $10 \leq (p_L + p_H \times 256) \leq 65530$ where $(p_L + p_H \times 256) = 9 \times k + 1$; $0 \leq p_L \leq 255$, $0 \leq p_H \leq 255$
fn = 3

DM-D110 and **DM-D210**: *a* = 10 through 15

DM-D500: *a* = 1, 5

b = 48, 49, 50

$1 \leq k \leq 7281$

[Description] Changes the memory switch specified by *a* to the values specified by *b*.

- When *b* = 48, the applicable bit is turned off.
- When *b* = 49, the applicable bit is turned on.
- When *b* = 50, the applicable bit is not changed.

DM-D110 and **DM-D210**:

The specific value of the memory switch corresponds to the switch number defined by *a* (10 through 15), and the value is the sum of the bits from bit 8 (MSB) to bit 1 (LSB).

Function	Item to be set	Memory switch	Default	Setting range
Character code table	Page #	<i>a</i> = 10 (MSW10)	Page 0	0 – 15, 16 – 19, 254, 255 (See ESC t)
International character set	Country	<i>a</i> = 11 (MSW11)	U.S.A.	0 – 13 (See ESC R)
Brightness adjustment	Brightness	<i>a</i> = 12 (MSW12)	4 (brightest)	1 – 4 (See US X)
Peripheral device selection	Display/printer	<i>a</i> = 13 (MSW13)	2 (display)	1 – 3 (See ESC =)
Cursor display	Cursor	<i>a</i> = 14 (MSW14)	Selected	0, 1, 48, 49 (See ESC W)
Display number	Number of display	<i>a</i> = 15 (MSW15)	0	0 – 255 (See US (A))



- When the specified parameter is out of range, it is ignored.

[Model-dependent variations] **DM-D500**

DM-D500

The specific value of the memory switch corresponds to the switch number defined by **a** (1 or 5), and the value is the sum of the bits from bit 8 (MSB) to bit 1 (LSB).

- When **a** = 1, the memory switch is defined as follows:

Bit	Function	Value		Default
		0 (OFF)	1 (ON)	
1	Transmission of power information	Does not transmit	Transmits	0
2 – 8	Reserved	—		—

- When **a** = 5, the memory switch is defined as follows:

Bit	Function	Value		Default
		0 (OFF)	1 (ON)	
1	Reserved	—	Fixed to ON	1
2	Pass-through connection (256 × 64 mode)	Enabled	Disabled	1
3	Reserved	—		—
4	Emulation mode	DM-D2xx	DM-D1xx	0
5 – 8	Reserved	—		—



US (E *p_L* *p_H* *fn* *a* <Function 4>

[Format]	ASCII	US	(E	<i>p_L</i>	<i>p_H</i>	<i>fn</i>	<i>a</i>
	Hex	1F	28	45	03	00	04	<i>a</i>
	Decimal	31	40	69	3	0	4	<i>a</i>

[Range] **(*p_L* + *p_H* × 256) = 2** (***p_L* = 2, *p_H* = 0**)
***fn* = 4**

DM-D110 and **DM-D210**: ***a*** = 10 through 15

DM-D500: ***a*** = 1, 5

[Description] Transmits the value of the memory switch setting back to the host.

- The memory switch is specified by ***a***.
- The display transmits the following data back to the host:

Transmitted data	Hex	Decimal	Number of bytes
Header	57H	87	1 byte
Identifier	24H	36	1 byte
Display number	30 – 39H	48 – 57	0 – 2 bytes
Separate code	1FH	31	1 byte
Data	30H or 31H	48 or 49	8 bytes
NUL	00	0	1 byte

[Example for **DM-D110** and **DM-D210**]

Switch	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
11 (<i>a</i> = 11)	OFF	OFF	OFF	OFF	ON	ON	OFF	ON

- The display transmits 0000 1101 (30, 30, 30, 30, 31, 31, 30, 31 Hex), which is equivalent to 13, which is the Korean International Character Set.



CONFIDENTIAL

[Example for **DM-D500**]

Switch	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
5 (a = 5)	OFF	OFF	OFF	OFF	ON	OFF	ON	ON

- The display transmits 0000 1011 (30, 30, 30, 30, 31, 30, 31, 31 Hex). This shows that the pass-through connection is enabled and the emulation mode is DM-D2xx.



US (E *pl* *ph* *fn* *a* *d1* *d2* <Function 7>

[Format]	ASCII	US	(E	<i>pl</i>	<i>ph</i>	<i>fn</i>	<i>a</i>	<i>d1</i>	<i>d2</i>
	Hex	1F	28	45	04	00	07	<i>a</i>	<i>d1</i>	<i>d2</i>
	Decimal	31	40	69	4	0	7	<i>a</i>	<i>d1</i>	<i>d2</i>

[Range] **(*pl* + *ph* × 256) = 4** (***pl* = 4, *ph* = 0**)
***fn* = 7**
1 ≤ *a* ≤ 255
***d1* = 30, 31**
***d2* = 30, 31 (where *d1* ≠ *d2*)**

[Displays featuring this command] **DM-D500**.

[Displays not featuring this command] **DM-D110** and **DM-D210**.

[Description] Copies the data that is stored in the user-defined code page.

<i>d1</i>	<i>d2</i>	Function
31	30	Loads the character code page data specified by the font number indicated by parameter <i>a</i> from the memory area into the work area
30	31	Saves the character code page data specified by the font number indicated by parameter <i>a</i> from the work area into the memory area



CONFIDENTIAL

Fonts specified by parameter **a**:

a	Font	Dot matrix		Character code page	
		Horizontal dots	Vertical dots	Page 254	Page 255
1	Font B	6	8	<input type="checkbox"/>	
2	Font B	6	8		<input type="checkbox"/>
3	Font A	8	16	<input type="checkbox"/>	
4	Font A	8	16		<input type="checkbox"/>

- If **Function 2** is executed, the power is turned off, or the customer display is reset without copying the data in the work area to the memory area, the data in the work area is deleted.



CONFIDENTIAL

US (E *p_L* *p_H* *fn* *y* *c1* *c2* [*x* *d1* ... *d* (*y* × *x*)] *k* <Function 8>

[Format]	ASCII	US	(E	<i>p_L</i>	<i>p_H</i>	<i>fn</i>	<i>y</i>	<i>c1</i>	<i>c2</i>	[<i>x</i>	<i>d1</i> ... <i>d</i>(<i>y</i> × <i>x</i>)]<i>k</i>
	Hex	1F	28	45	<i>p_L</i>	<i>p_H</i>	08	<i>y</i>	<i>c1</i>	<i>c2</i>	[<i>x</i>	<i>d1</i> ... <i>d</i>(<i>y</i> × <i>x</i>)]<i>k</i>
	Decimal	31	40	69	<i>p_L</i>	<i>p_H</i>	8	<i>y</i>	<i>c1</i>	<i>c2</i>	[<i>x</i>	<i>d1</i> ... <i>d</i>(<i>y</i> × <i>x</i>)]<i>k</i>

[Range] $5 \leq (\mathbf{pL} + \mathbf{pH} \times 256) \leq 65535$ ($0 \leq \mathbf{pL} \leq 255$, $0 \leq \mathbf{pH} \leq 255$)
***fn* = 8**
 $1 \leq \mathbf{y} \leq 2$ (font A)
 $1 \leq \mathbf{y} \leq 1$ (font B)
 $128 \leq \mathbf{c1} \leq \mathbf{c2} \leq 255$
 $0 \leq \mathbf{x} \leq 8$ (font A)
 $0 \leq \mathbf{x} \leq 6$ (font B)
 $0 \leq \mathbf{d} \leq 255$
***k* = *c2* - *c1* + 1**

[Displays featuring this command] **DM-D500**.

[Displays not featuring this command] **DM-D110** and **DM-D210**.

[Description] Defines each character of the character data stored in the character code page in the work area.

- ***y*** specifies the number of bytes in the character pattern in the vertical direction as ***y*** bytes.
- ***c1*** specifies the character code that starts the pattern definition.
- ***c2*** specifies the character code that ends the pattern definition.
- ***x*** specifies the dots from the left end in the horizontal direction for the defined data.
- ***d*** specifies the defined data. The defined data for the character pattern is processed as column type.



CONFIDENTIAL

US (E *p_L* *p_H* *fn* *x* *c1* *c2* [*x* *d1* ... *d* (*y* × *x*)] *k* <Function 9>

[Format]	ASCII	US	(E	<i>p_L</i>	<i>p_H</i>	<i>fn</i>	<i>x</i>	<i>c1</i>	<i>c2</i>	[<i>x</i>	<i>d1</i> ... <i>d</i> (<i>y</i> × <i>x</i>)] <i>k</i>
	Hex	1F	28	45	<i>p_L</i>	<i>p_H</i>	09	<i>x</i>	<i>c1</i>	<i>c2</i>	[<i>x</i>	<i>d1</i> ... <i>d</i> (<i>y</i> × <i>x</i>)] <i>k</i>
	Decimal	31	40	69	<i>p_L</i>	<i>p_H</i>	9	<i>x</i>	<i>c1</i>	<i>c2</i>	[<i>x</i>	<i>d1</i> ... <i>d</i> (<i>y</i> × <i>x</i>)] <i>k</i>

[Range] $5 \leq (p_L + p_H \times 256) \leq 65535$ ($0 \leq p_L \leq 255$, $0 \leq p_H \leq 255$)
fn = 9
x = 1
 $128 \leq c1 \leq 255$
 $128 \leq c2 \leq 255$ (where $c1 \leq c2$)
 $0 \leq y \leq 16$ (font A)
 $0 \leq y \leq 8$ (font B)
 $0 \leq d \leq 255$
k = **c2** - **c1** + 1

[Displays featuring this command] **DM-D500**.

[Displays not featuring this command] **DM-D110** and **DM-D210**.

[Description] Defines each character of the character data stored in the character code page in the work area.

- **x** specifies the number of bytes in the character pattern in the horizontal direction as **x** bytes.
- **c1** specifies the character code that starts the pattern definition.
- **c2** specifies the character code that ends the pattern definition.
- **y** specifies the dots from the top end in the vertical direction for the defined data.
- **d** specifies the defined data. The defined data for the character pattern is processed as raster type.



CONFIDENTIAL

US (E *pL pH fn c1 c2* <Function 10>

[Format]	ASCII	US	(E	<i>pL</i>	<i>pH</i>	<i>fn</i>	<i>c1</i>	<i>c2</i>
	Hex	1F	28	45	03	00	0A	<i>c1</i>	<i>c2</i>
	Decimal	31	40	69	3	0	10	<i>c1</i>	<i>c2</i>

[Range] $(pL + pH \times 256) = 3$ ($pL = 3, pH = 0$)
 $fn = 10$
 $128 \leq c1 \leq 255$
 $128 \leq c2 \leq 255$ (where $c1 \leq c2$)

[Displays featuring this command] **DM-D500**.

[Displays not featuring this command] **DM-D110** and **DM-D210**.

[Description] Deletes the character data individually for the character code page in the work area.

- ***c1*** specifies the first character code pattern to be deleted.
- ***c2*** specifies the last character code pattern to be deleted.



US (E *p_L* *p_H* *fn* *a* [*x_L* *x_H* *y_L* *y_H* *d1...dk*]*1* ... [*x_L* *x_H* *y_L* *y_H* *d1...dk*]*k* <Function 11>

[Format]	ASCII	US (E	<i>p_L</i>	<i>p_H</i>	<i>fn</i>	<i>a</i>	<i>[x_L x_H y_L y_H d1...dk]1 ... [x_L x_H y_L y_H d1...dk]k</i>
	Hex	1F 28	45	<i>p_L</i>	<i>p_H</i>	<i>OB</i>	<i>a</i>	<i>[x_L x_H y_L y_H d1...dk]1 ... [x_L x_H y_L y_H d1...dk]k</i>
	Decimal	31 40	69	<i>p_L</i>	<i>p_H</i>	<i>11</i>	<i>a</i>	<i>[x_L x_H y_L y_H d1...dk]1 ... [x_L x_H y_L y_H d1...dk]k</i>

[Range] $7 \leq (p_L + p_H \times 256) \leq 65535$ ($0 \leq p_L \leq 255$, $0 \leq p_H \leq 255$)
fn = 11
 $7 \leq a \leq 255$
 $1 \leq (x_L + x_H \times 256) \leq 65534$ ($0 \leq x_L \leq 255$, $0 \leq x_H \leq 255$)
 $1 \leq (y_L + y_H \times 256) \leq 65534$ ($0 \leq y_L \leq 255$, $0 \leq y_H \leq 255$)
 $0 \leq d \leq 255$
 Total amount of defined area = 32 KB

[Displays featuring this command] **DM-D500.**

[Displays not featuring this command] **DM-D110** and **DM-D210.**

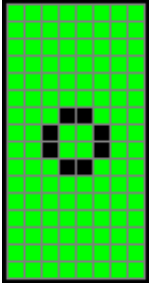
- [Description] Defines the NV bit image to be specified.
- ***a*** specifies the number of the NV bit image.
 - ***x_L*, *x_H*** specifies (***x_L* + *x_H* × 256**) dots in the horizontal direction for the NV bit image.
 - ***y_L*, *y_H*** specifies (***y_L* + *y_H* × 256**) dots in the vertical direction for the NV bit image.
 - ***d*** specifies the bit-image data.



[Examples]

Action	Command/Example
<p>Function 1 Change into user-setting mode</p>	<pre>PRINT #1, CHR\$(31); "(E"; PRINT #1, CHR\$(3); CHR\$(0); 'pL pH PRINT #1, CHR\$(1); 'fn PRINT #1, "IN"; 'd1 d2</pre>
<p>Function 4 Transmit the settings of the memory switch to the host. The example on the right shows settings for memory switch 1</p>	<pre>PRINT #1, CHR\$(31); "(E"; PRINT #1, CHR\$(2); CHR\$(0); 'pL pH PRINT #1, CHR\$(4); 'fn PRINT #1, CHR\$(1); 'a</pre> <div style="text-align: center;"> <p>ON OFF</p> </div>
<p>Function 3 Set value for the memory switch</p>	<pre>PRINT #1, CHR\$(31); "(E"; PRINT #1, CHR\$(10); CHR\$(0); 'pL pH PRINT #1, CHR\$(3); 'fn PRINT #1, CHR\$(1); 'a PRINT #1, "22222221"; '</pre> <div style="text-align: center;"> <p>ON OFF</p> </div>
<p>Function 7 Copy the data stored in the user-defined code page (Specify font A pattern.)</p>	<pre>PRINT #1, CHR\$(31); "(E"; PRINT #1, CHR\$(4); CHR\$(0); 'pL pH PRINT #1, CHR\$(7); 'fn PRINT #1, CHR\$(4); 'a PRINT #1, CHR\$(31); CHR\$(30); 'd1 d2</pre>



<p>Function 8 Define the character pattern to the character code page</p>	<pre>PRINT #1, CHR\$(31); "(E"; PRINT #1, CHR\$(21); CHR\$(0); 'pL pH PRINT #1, CHR\$(8); 'fn PRINT #1, CHR\$(2); 'y PRINT #1, CHR\$(128); CHR\$(128); 'c1 c2 PRINT #1, CHR\$(8); 'x PRINT #1, CHR\$(&HFF); CHR\$(&HFF); PRINT #1, CHR\$(&HFF); CHR\$(&HFF); PRINT #1, CHR\$(&HFE); CHR\$(&H7F); PRINT #1, CHR\$(&HFD); CHR\$(&HBF); PRINT #1, CHR\$(&HFD); CHR\$(&HBF); PRINT #1, CHR\$(&HFE); CHR\$(&H7F); PRINT #1, CHR\$(&HFF); CHR\$(&HFF); PRINT #1, CHR\$(&HFF); CHR\$(&HFF);</pre>
<p>Function 7 Copy the data stored in the user-defined code page</p>	<div style="text-align: center;">  </div> <pre>PRINT #1, CHR\$(31); "(E"; PRINT #1, CHR\$(4); CHR\$(0); 'pL pH PRINT #1, CHR\$(7); 'fn PRINT #1, CHR\$(4); 'a PRINT #1, CHR\$(30); CHR\$(31); 'd1 d2</pre>



Function 11

Define the NV bit image

```

PRINT #1, CHR$(31); "(E";
PRINT #1, CHR$(134); CHR$(0); 'pL pH
PRINT #1, CHR$(11); 'fn
PRINT #1, CHR$(1); 'a
PRINT #1, CHR$(32); CHR$(0); 'xL xH
PRINT #1, CHR$(32); CHR$(0); 'yL yH
PRINT #1, CHR$(0); CHR$(0); CHR$(0); CHR$(0);
PRINT #1, CHR$(0); CHR$(4); CHR$(32); CHR$(0);
PRINT #1, CHR$(0); CHR$(12); CHR$(96); CHR$(0);
PRINT #1, CHR$(0); CHR$(12); CHR$(96); CHR$(0);
PRINT #1, CHR$(0); CHR$(12); CHR$(96); CHR$(0);
PRINT #1, CHR$(0); CHR$(31); CHR$(240); CHR$(0);
PRINT #1, CHR$(0); CHR$(127); CHR$(252); CHR$(0);
PRINT #1, CHR$(1); CHR$(255); CHR$(255); CHR$(0);
PRINT #1, CHR$(3); CHR$(204); CHR$(127); CHR$(128);
PRINT #1, CHR$(7); CHR$(140); CHR$(103); CHR$(192);
PRINT #1, CHR$(14); CHR$(12); CHR$(97); CHR$(224);
PRINT #1, CHR$(28); CHR$(12); CHR$(96); CHR$(224);
PRINT #1, CHR$(28); CHR$(12); CHR$(96); CHR$(240);
PRINT #1, CHR$(56); CHR$(12); CHR$(96); CHR$(112);
PRINT #1, CHR$(56); CHR$(12); CHR$(96); CHR$(56);
PRINT #1, CHR$(48); CHR$(12); CHR$(96); CHR$(56);
PRINT #1, CHR$(112); CHR$(12); CHR$(96); CHR$(28);
PRINT #1, CHR$(112); CHR$(12); CHR$(96); CHR$(28);
PRINT #1, CHR$(112); CHR$(12); CHR$(96); CHR$(28);
PRINT #1, CHR$(112); CHR$(12); CHR$(96); CHR$(28);
PRINT #1, CHR$(112); CHR$(12); CHR$(96); CHR$(28);

```



<p>Function 11 Define the NV bit image (continued)</p>	<pre>PRINT #1, CHR\$(112); CHR\$(12); CHR\$(96); CHR\$(28); PRINT #1, CHR\$(48); CHR\$(12); CHR\$(96); CHR\$(56); PRINT #1, CHR\$(56); CHR\$(12); CHR\$(96); CHR\$(56); PRINT #1, CHR\$(56); CHR\$(12); CHR\$(64); CHR\$(112); PRINT #1, CHR\$(28); CHR\$(12); CHR\$(0); CHR\$(112); PRINT #1, CHR\$(28); CHR\$(0); CHR\$(0); CHR\$(224); PRINT #1, CHR\$(14); CHR\$(0); CHR\$(1); CHR\$(224); PRINT #1, CHR\$(14); CHR\$(0); CHR\$(0); CHR\$(0); PRINT #1, CHR\$(0); CHR\$(0); CHR\$(0); CHR\$(0); PRINT #1, CHR\$(0); CHR\$(0); CHR\$(0); CHR\$(0); PRINT #1, CHR\$(0); CHR\$(0); CHR\$(0); CHR\$(0);</pre>
<p>Function 2 End user-setting mode (The display performs a software reset.)</p>	<div data-bbox="1318 686 1780 1146" data-label="Image"> <p>The image shows a 24x24 grid of pixels. The letter 'E' is rendered in a bright green color against a black background. The 'E' is composed of several horizontal bars and a vertical stem, with a slight curve at the top and bottom. The grid is approximately 24 columns wide and 24 rows high.</p> </div> <pre>PRINT #1, CHR\$(31); "(E"; PRINT #1, CHR\$(4); CHR\$(0); 'pL pH PRINT #1, CHR\$(2); 'fn PRINT #1, "OUT"; 'd1 d2 d3</pre>



<p>Function 4 Transmit the settings of the memory switch 5 to the host</p>	<pre>PRINT #1, CHR\$(31); "(E"; PRINT #1, CHR\$(2); CHR\$(0); 'pL pH PRINT #1, CHR\$(4); 'fn PRINT #1, CHR\$(5); 'a</pre>
<p>Display character code 128</p>	<pre>PRINT #1, CHR\$(128);</pre>
<p>Select character code page 255</p>	<pre>PRINT #1, CHR\$(27); "t"; PRINT #1, CHR\$(255); 'n</pre>
<p>Display character code 128</p>	<pre>PRINT #1, CHR\$(128);</pre>
<p>US (F command, Function 1 Display NV bit image</p>	<pre>PRINT #1, CHR\$(31); "(F"; PRINT #1, CHR\$(12); CHR\$(0); 'pL pH PRINT #1, CHR\$(1); 'fn PRINT #1, CHR\$(1); 'a PRINT #1, CHR\$(1); CHR\$(1); 'x y PRINT #1, CHR\$(0); CHR\$(0); 'xnL xnH PRINT #1, CHR\$(0); CHR\$(0); 'ynL ynH PRINT #1, CHR\$(0); CHR\$(0); 'dxL dxH PRINT #1, CHR\$(0); CHR\$(0); 'dyL dyH</pre> <div data-bbox="1276 1000 1822 1138" style="text-align: center;"> </div>

[References]

US (B, US (F, Function 1



US (F

[Name] Display bit image

[Format] ASCII US (F ***pl ph fn [parameters]***
 Hex 1F 28 46 ***pl ph fn [parameters]***
 Decimal 31 40 70 ***pl ph fn [parameters]***

[Displays featuring this command] **DM-D500**.

[Displays not featuring this command] **DM-D110** and **DM-D210**.

[Description] Displays the bit image data.

- The value of ***fn*** specifies the function.

<i>fn</i>	Function	Description
1	Function 1	Displays the NV bit image
2	Function 2	Defines a downloaded bit image
3	Function 3	Displays a downloaded bit image
4	Function 4	Displays a bit image (raster type)
5	Function 5	Displays a bit image (column type)

- pl*** and ***ph*** specify the bytes following parameter ***ph*** (***fn*** and ***[parameters]***) as **$(pl + (ph \times 256))$** .



US (F *p_L* *p_H* *fn* *a* *x* *y* *x_{NL}* *x_{NH}* *y_{NL}* *y_{NH}* *dx_L* *dx_H* *dy_L* *dy_H* <Function 1>

[Format]	ASCII	US	(F	<i>p_L</i>	<i>p_H</i>	<i>fn</i>	<i>a</i>	<i>x</i>	<i>y</i>	<i>x_{NL}</i>	<i>x_{NH}</i>	<i>y_{NL}</i>	<i>y_{NH}</i>	<i>dx_L</i>	<i>dx_H</i>	<i>dy_L</i>	<i>dy_H</i>
	Hex	1F	28	46	0C	00	01	<i>a</i>	<i>x</i>	<i>y</i>	<i>x_{NL}</i>	<i>x_{NH}</i>	<i>y_{NL}</i>	<i>y_{NH}</i>	<i>dx_L</i>	<i>dx_H</i>	<i>dy_L</i>	<i>dy_H</i>
	Decimal	31	40	70	12	0	1	<i>a</i>	<i>x</i>	<i>y</i>	<i>x_{NL}</i>	<i>x_{NH}</i>	<i>y_{NL}</i>	<i>y_{NH}</i>	<i>dx_L</i>	<i>dx_H</i>	<i>dy_L</i>	<i>dy_H</i>

[Range] (*p_L* + *p_H* × 256) = 12 (*p_L* = 12, *p_H* = 0)

fn = 1

1 ≤ *a* ≤ 255

1 ≤ *x* ≤ 8

1 ≤ *y* ≤ 8

0 ≤ (*x_{NL}* + *x_{NH}* × 256) ≤ 65534 (0 ≤ *x_{NL}* ≤ 255, 0 ≤ *x_{NH}* ≤ 255)

0 ≤ (*y_{NL}* + *y_{NH}* × 256) ≤ 65534 (0 ≤ *y_{NL}* ≤ 255, 0 ≤ *y_{NH}* ≤ 255)

1 ≤ (*dx_L* + *dx_H* × 256) ≤ 65535 (0 ≤ *dx_L* ≤ 255, 0 ≤ *dx_H* ≤ 255)

1 ≤ (*dy_L* + *dy_H* × 256) ≤ 65535 (0 ≤ *dy_L* ≤ 255, 0 ≤ *dy_H* ≤ 255)

[Description] Displays the NV bit image defined by **US (E, Function 11** and identified by parameter *a* and enlarges it by *x* times in the horizontal and *y* times in the vertical direction.

If you want to display only a portion of the bit image, the parameters *x_{NL}*, *x_{NH}*, *y_{NL}*, *y_{NH}* define a starting point within the image, and the parameters *dx_L*, *dx_H*, *dy_L*, *dy_H* define the size of the portion of the bit image to be displayed.

- In the *x* direction, the display shows the dots after the dot number (*x_{NL}* + *x_{NH}* × 256).
- In the *y* direction, the display shows the dots after the dot number (*y_{NL}* + *y_{NH}* × 256).
- When (*dx_L* + *dx_H* × 256) = 0, all bit-image data in the *x* direction is displayed.
- When (*dx_L* + *dx_H* × 256) ≠ 0, only the specified bit-image data in the *x* direction is displayed.
- When (*dy_L* + *dy_H* × 256) = 0, all bit-image data in the *y* direction is displayed.
- When (*dy_L* + *dy_H* × 256) ≠ 0, only the specified bit-image data in the *y* direction is displayed.



CONFIDENTIAL

US (F *pL pH fn a [xL xH yL yH d1...dk]1 ... [xL xH yL yH d1...dk]k* <Function 2>

[Format]	ASCII	US	(F	<i>pL pH fn a [xL xH yL yH d1...dk]1 ... [xL xH yL yH d1...dk]k</i>
	Hex	1F	28	46	<i>pL pH 02 a [xL xH yL yH d1...dk]1 ... [xL xH yL yH d1...dk]k</i>
	Decimal	31	40	70	<i>pL pH 2 a [xL xH yL yH d1...dk]1 ... [xL xH yL yH d1...dk]k</i>

[Range] $7 \leq (pL + pH \times 256) \leq 65535$ ($0 \leq pL \leq 255$, $0 \leq pH \leq 255$)

fn = 2

$1 \leq a \leq 255$

$1 \leq (xL + xH \times 256) \leq 65534$ ($0 \leq xL \leq 255$, $0 \leq xH \leq 255$)

$1 \leq (yL + yH \times 256) \leq 65534$ ($0 \leq yL \leq 255$, $0 \leq yH \leq 255$)

$0 \leq d \leq 255$

Total amount of defined area = 4 KB

[Description] Defines a bit image to be downloaded.

- ***a*** specifies the number of the downloaded bit image you are defining.
- ***xL, xH*** specifies ($xL + xH \times 256$) dots in the horizontal direction for the downloaded image.
- ***yL, yH*** specifies ($yL + yH \times 256$) dots in the vertical direction for the downloaded image.
- ***d*** specifies the bit-image data.



CONFIDENTIAL

US (F *p_L* *p_H* *fn* *a* *x* *y* *x_{NL}* *x_{NH}* *y_{NL}* *y_{NH}* *dx_L* *dx_H* *dy_L* *dy_H* <Function 3>

[Format]	ASCII	US	(F	<i>p_L</i>	<i>p_H</i>	<i>fn</i>	<i>a</i>	<i>x</i>	<i>y</i>	<i>x_{NL}</i>	<i>x_{NH}</i>	<i>y_{NL}</i>	<i>y_{NH}</i>	<i>dx_L</i>	<i>dx_H</i>	<i>dy_L</i>	<i>dy_H</i>
	Hex	1F	28	46	0C	00	03	<i>a</i>	<i>x</i>	<i>y</i>	<i>x_{NL}</i>	<i>x_{NH}</i>	<i>y_{NL}</i>	<i>y_{NH}</i>	<i>dx_L</i>	<i>dx_H</i>	<i>dy_L</i>	<i>dy_H</i>
	Decimal	31	40	70	12	0	3	<i>a</i>	<i>x</i>	<i>y</i>	<i>x_{NL}</i>	<i>x_{NH}</i>	<i>y_{NL}</i>	<i>y_{NH}</i>	<i>dx_L</i>	<i>dx_H</i>	<i>dy_L</i>	<i>dy_H</i>

[Range] (*p_L* + *p_H* × 256) = 12 (*p_L* = 12, *p_H* = 0)

fn = 3

1 ≤ *a* ≤ 255

1 ≤ *x* ≤ 8

1 ≤ *y* ≤ 8

0 ≤ (*x_{NL}* + *x_{NH}* × 256) ≤ 65534 (0 ≤ *x_{NL}* ≤ 255, 0 ≤ *x_{NH}* ≤ 255)

0 ≤ (*y_{NL}* + *y_{NH}* × 256) ≤ 65534 (0 ≤ *y_{NL}* ≤ 255, 0 ≤ *y_{NH}* ≤ 255)

1 ≤ (*dx_L* + *dx_H* × 256) ≤ 65535 (0 ≤ *dx_L* ≤ 255, 0 ≤ *dx_H* ≤ 255)

1 ≤ (*dy_L* + *dy_H* × 256) ≤ 65535 (0 ≤ *dy_L* ≤ 255, 0 ≤ *dy_H* ≤ 255)

[Description] Displays the downloaded bit image identified by parameter *a* and enlarges it *x* times horizontally by *y* times vertically.

If you want to display only a portion of the bit image, the parameters *x_{NL}*, *x_{NH}*, *y_{NL}*, *y_{NH}* define a starting point within the image, and the parameters *dx_L*, *dx_H*, *dy_L*, *dy_H* define the size of the portion of the bit image to be displayed.

- In the *x* direction, the display shows the dots after the dot number (*x_{NL}* + *x_{NH}* × 256).
- In the *y* direction, the display shows the dots after the dot number (*y_{NL}* + *y_{NH}* × 256).
- When (*dx_L* + *dx_H* × 256) = 0, all bit-image data in the *x* direction is displayed.
- When (*dx_L* + *dx_H* × 256) ≠ 0, only the specified bit-image data in the *x* direction is displayed.
- When (*dy_L* + *dy_H* × 256) = 0, all bit-image data in the *y* direction is displayed.
- When (*dy_L* + *dy_H* × 256) ≠ 0, only the specified bit-image data in the *y* direction is displayed.



CONFIDENTIAL

US (F *p_L* *p_H* *fn* *a* *x* *y* *x_{NL}* *x_{NH}* *y_{NL}* *y_{NH}* *d1* ... *dk* <Function 4>

[Format]	ASCII	US	(F	<i>p_L</i>	<i>p_H</i>	<i>fn</i>	<i>a</i>	<i>x</i>	<i>y</i>	<i>x_{NL}</i>	<i>x_{NH}</i>	<i>y_{NL}</i>	<i>y_{NH}</i>	<i>d1</i>	...	<i>dk</i>
	Hex	1F	28	46	0C	00	04	<i>a</i>	<i>x</i>	<i>y</i>	<i>x_{NL}</i>	<i>x_{NH}</i>	<i>y_{NL}</i>	<i>y_{NH}</i>	<i>d1</i>	...	<i>dk</i>
	Decimal	31	40	70	12	0	4	<i>a</i>	<i>x</i>	<i>y</i>	<i>x_{NL}</i>	<i>x_{NH}</i>	<i>y_{NL}</i>	<i>y_{NH}</i>	<i>d1</i>	...	<i>dk</i>

[Range]

$8 \leq (p_L + p_H \times 256) \leq 65535$ ($0 \leq p_L \leq 255$, $0 \leq p_H \leq 255$)

fn = 4

a = 48

$1 \leq x \leq 8$

$1 \leq y \leq 8$

$1 \leq (x_{NL} + x_{NH} \times 256) \leq 65535$ ($0 \leq x_{NL} \leq 255$, $0 \leq x_{NH} \leq 255$)

$1 \leq (y_{NL} + y_{NH} \times 256) \leq 65535$ ($0 \leq y_{NL} \leq 255$, $0 \leq y_{NH} \leq 255$)

$0 \leq d \leq 255$

[Description]

Displays the raster-type bit image you define and enlarges it **x** times horizontally by **y** times vertically.

- **x_{NL}**, **x_{NH}** specifies (**x_{NL}** + **x_{NH}** × 256) dots in the horizontal direction for the image.
- **y_{NL}**, **y_{NH}** specifies (**y_{NL}** + **y_{NH}** × 256) dots in the vertical direction for the image.
- **d** specifies the bit-image data.



US (F p_L p_H fn a x y x_{NL} x_{NH} y_{NL} y_{NH} d1 ... dk <Function 5>

[Format]	ASCII	US	(F	p_L	p_H	fn	a	x	y	x_{NL}	x_{NH}	y_{NL}	y_{NH}	d1	...	dk
	Hex	1F	28	46	0C	00	05	a	x	y	x_{NL}	x_{NH}	y_{NL}	y_{NH}	d1	...	dk
	Decimal	31	40	70	12	0	5	a	x	y	x_{NL}	x_{NH}	y_{NL}	y_{NH}	d1	...	dk

[Range]

$8 \leq (p_L + p_H \times 256) \leq 65535$ ($0 \leq p_L \leq 255$, $0 \leq p_H \leq 255$)
fn = 5
a = 48
 $1 \leq x \leq 8$
 $1 \leq y \leq 8$
 $1 \leq (x_{NL} + x_{NH} \times 256) \leq 65535$ ($0 \leq x_{NL} \leq 255$, $0 \leq x_{NH} \leq 255$)
 $1 \leq (y_{NL} + y_{NH} \times 256) \leq 65535$ ($0 \leq y_{NL} \leq 255$, $0 \leq y_{NH} \leq 255$)
 $0 \leq d \leq 255$

[Description] Displays the column-type bit image you define and enlarges it **x** times horizontally by **y** times vertically.

- **x_{NL}**, **x_{NH}** specifies (**x_{NL}** + **x_{NH}** × 256) dots in the horizontal direction for the image.
- **y_{NL}**, **y_{NH}** specifies (**y_{NL}** + **y_{NH}** × 256) dots in the vertical direction for the image.
- **d** specifies the bit-image data.

[Examples]

Action	Command/Example
<p>Function 2 Define a downloaded bit image</p>	<pre>PRINT #1, CHR\$(31); "(F"; PRINT #1, CHR\$(134); CHR\$(0); 'pL pH PRINT #1, CHR\$(2); 'fn PRINT #1, CHR\$(1); 'a PRINT #1, CHR\$(32); CHR\$(0); 'xL xH PRINT #1, CHR\$(32); CHR\$(0); 'yL yH PRINT #1, CHR\$(0); CHR\$(0); CHR\$(0); CHR\$(0); PRINT #1, CHR\$(0); CHR\$(4); CHR\$(32); CHR\$(0); PRINT #1, CHR\$(0); CHR\$(12); CHR\$(96); CHR\$(0);</pre>



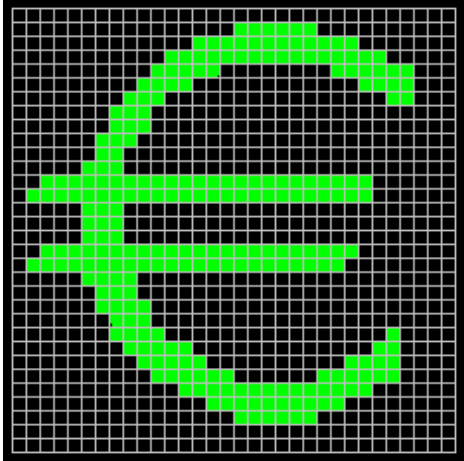

Function 2

Define a downloaded bit image (continued)

```

PRINT #1, CHR$(0); CHR$(12); CHR$(96); CHR$(0);
PRINT #1, CHR$(0); CHR$(12); CHR$(96); CHR$(0);
PRINT #1, CHR$(0); CHR$(31); CHR$(240); CHR$(0);
PRINT #1, CHR$(0); CHR$(127); CHR$(252); CHR$(0);
PRINT #1, CHR$(1); CHR$(255); CHR$(255); CHR$(0);
PRINT #1, CHR$(3); CHR$(204); CHR$(127); CHR$(128);
PRINT #1, CHR$(7); CHR$(140); CHR$(103); CHR$(192);
PRINT #1, CHR$(14); CHR$(12); CHR$(97); CHR$(224);
PRINT #1, CHR$(28); CHR$(12); CHR$(96); CHR$(224);
PRINT #1, CHR$(28); CHR$(12); CHR$(96); CHR$(240);
PRINT #1, CHR$(56); CHR$(12); CHR$(96); CHR$(112);
PRINT #1, CHR$(56); CHR$(12); CHR$(96); CHR$(56);
PRINT #1, CHR$(48); CHR$(12); CHR$(96); CHR$(56);
PRINT #1, CHR$(112); CHR$(12); CHR$(96); CHR$(28);
PRINT #1, CHR$(112); CHR$(12); CHR$(96); CHR$(28);
PRINT #1, CHR$(112); CHR$(12); CHR$(96); CHR$(28);
PRINT #1, CHR$(112); CHR$(12); CHR$(96); CHR$(28);
PRINT #1, CHR$(112); CHR$(12); CHR$(96); CHR$(28);
PRINT #1, CHR$(112); CHR$(12); CHR$(96); CHR$(28);
PRINT #1, CHR$(48); CHR$(12); CHR$(96); CHR$(56);
PRINT #1, CHR$(56); CHR$(12); CHR$(96); CHR$(56);
PRINT #1, CHR$(56); CHR$(12); CHR$(64); CHR$(112);
PRINT #1, CHR$(28); CHR$(12); CHR$(0); CHR$(112);
PRINT #1, CHR$(28); CHR$(0); CHR$(0); CHR$(224);
PRINT #1, CHR$(14); CHR$(0); CHR$(1); CHR$(224);
PRINT #1, CHR$(14); CHR$(0); CHR$(0); CHR$(0);
PRINT #1, CHR$(0); CHR$(0); CHR$(0); CHR$(0);
    
```



<p>Function 2 Define a downloaded bit image (continued)</p>	<pre>PRINT #1, CHR\$(0); CHR\$(0); CHR\$(0); CHR\$(0); PRINT #1, CHR\$(0); CHR\$(0); CHR\$(0); CHR\$(0);</pre> 
<p>Function 3 Display the downloaded bit image</p>	<pre>PRINT #1, CHR\$(31); "(F"; PRINT #1, CHR\$(12); CHR\$(0); 'pL pH PRINT #1, CHR\$(3); 'fn PRINT #1, CHR\$(1); 'a PRINT #1, CHR\$(1); CHR\$(1); 'x y PRINT #1, CHR\$(0); CHR\$(0); 'xnL xnH PRINT #1, CHR\$(0); CHR\$(0); 'ynL ynH PRINT #1, CHR\$(0); CHR\$(0); 'dxL dxH PRINT #1, CHR\$(0); CHR\$(0); 'dyL dyH</pre> 



Function 4

Display the defined raster-type bit image

```

PRINT #1, CHR$(31); "(F";
PRINT #1, CHR$(136); CHR$(0); 'pL pH
PRINT #1, CHR$(4); 'n
PRINT #1, CHR$(48); 'a
PRINT #1, CHR$(1); CHR$(2); 'x y
PRINT #1, CHR$(32); CHR$(0); 'xnL xnH
PRINT #1, CHR$(32); CHR$(0); 'ynL ynH
PRINT #1, CHR$(0); CHR$(0); CHR$(0); CHR$(0);
PRINT #1, CHR$(0); CHR$(4); CHR$(32); CHR$(0);
PRINT #1, CHR$(0); CHR$(12); CHR$(96); CHR$(0);
PRINT #1, CHR$(0); CHR$(12); CHR$(96); CHR$(0);
PRINT #1, CHR$(0); CHR$(12); CHR$(96); CHR$(0);
PRINT #1, CHR$(0); CHR$(31); CHR$(240); CHR$(0);
PRINT #1, CHR$(0); CHR$(127); CHR$(252); CHR$(0);
PRINT #1, CHR$(1); CHR$(255); CHR$(255); CHR$(0);
PRINT #1, CHR$(3); CHR$(204); CHR$(127); CHR$(128);
PRINT #1, CHR$(7); CHR$(140); CHR$(103); CHR$(192);
PRINT #1, CHR$(14); CHR$(12); CHR$(97); CHR$(224);
PRINT #1, CHR$(28); CHR$(12); CHR$(96); CHR$(224);
PRINT #1, CHR$(28); CHR$(12); CHR$(96); CHR$(240);
PRINT #1, CHR$(56); CHR$(12); CHR$(96); CHR$(112);
PRINT #1, CHR$(56); CHR$(12); CHR$(96); CHR$(56);
PRINT #1, CHR$(48); CHR$(12); CHR$(96); CHR$(56);
PRINT #1, CHR$(112); CHR$(12); CHR$(96); CHR$(28);
PRINT #1, CHR$(112); CHR$(12); CHR$(96); CHR$(28);
PRINT #1, CHR$(112); CHR$(12); CHR$(96); CHR$(28);
    
```



Function 4

Display the defined raster-type bit image

```
PRINT #1, CHR$(112); CHR$(12); CHR$(96); CHR$(28);
PRINT #1, CHR$(112); CHR$(12); CHR$(96); CHR$(28);
PRINT #1, CHR$(112); CHR$(12); CHR$(96); CHR$(28);
PRINT #1, CHR$(48); CHR$(12); CHR$(96); CHR$(56);
PRINT #1, CHR$(56); CHR$(12); CHR$(96); CHR$(56);
PRINT #1, CHR$(56); CHR$(12); CHR$(64); CHR$(112);
PRINT #1, CHR$(28); CHR$(12); CHR$(0); CHR$(112);
PRINT #1, CHR$(28); CHR$(0); CHR$(0); CHR$(224);
PRINT #1, CHR$(14); CHR$(0); CHR$(1); CHR$(224);
PRINT #1, CHR$(14); CHR$(0); CHR$(0); CHR$(0);
PRINT #1, CHR$(0); CHR$(0); CHR$(0); CHR$(0);
PRINT #1, CHR$(0); CHR$(0); CHR$(0); CHR$(0);
PRINT #1, CHR$(0); CHR$(0); CHR$(0); CHR$(0);
```



Function 5

Display the defined column-type bit image

```

PRINT #1, CHR$(31); "(F";
PRINT #1, CHR$(136); CHR$(0); 'pL pH
PRINT #1, CHR$(5); 'n
PRINT #1, CHR$(48); 'a
PRINT #1, CHR$(2); CHR$(2); 'x y
PRINT #1, CHR$(32); CHR$(0); 'xnL xnH
PRINT #1, CHR$(32); CHR$(0); 'ynL ynH
PRINT #1, CHR$(0); CHR$(0); CHR$(0); CHR$(0);
PRINT #1, CHR$(0); CHR$(4); CHR$(32); CHR$(0);
PRINT #1, CHR$(0); CHR$(12); CHR$(96); CHR$(0);
PRINT #1, CHR$(0); CHR$(12); CHR$(96); CHR$(0);
PRINT #1, CHR$(0); CHR$(12); CHR$(96); CHR$(0);
PRINT #1, CHR$(0); CHR$(31); CHR$(240); CHR$(0);
PRINT #1, CHR$(0); CHR$(127); CHR$(252); CHR$(0);
PRINT #1, CHR$(1); CHR$(255); CHR$(255); CHR$(0);
PRINT #1, CHR$(3); CHR$(204); CHR$(127); CHR$(128);
PRINT #1, CHR$(7); CHR$(140); CHR$(103); CHR$(192);
PRINT #1, CHR$(14); CHR$(12); CHR$(97); CHR$(224);
PRINT #1, CHR$(28); CHR$(12); CHR$(96); CHR$(224);
PRINT #1, CHR$(28); CHR$(12); CHR$(96); CHR$(240);
PRINT #1, CHR$(56); CHR$(12); CHR$(96); CHR$(112);
PRINT #1, CHR$(56); CHR$(12); CHR$(96); CHR$(56);
PRINT #1, CHR$(48); CHR$(12); CHR$(96); CHR$(56);
PRINT #1, CHR$(112); CHR$(12); CHR$(96); CHR$(28);
PRINT #1, CHR$(112); CHR$(12); CHR$(96); CHR$(28);
    
```



Function 5

Display the defined column-type bit image
(continued)

```

PRINT #1, CHR$(112); CHR$(12); CHR$(96); CHR$(28);
PRINT #1, CHR$(112); CHR$(12); CHR$(96); CHR$(28);
PRINT #1, CHR$(112); CHR$(12); CHR$(96); CHR$(28);
PRINT #1, CHR$(112); CHR$(12); CHR$(96); CHR$(28);
PRINT #1, CHR$(48); CHR$(12); CHR$(96); CHR$(56);
PRINT #1, CHR$(56); CHR$(12); CHR$(96); CHR$(56);
PRINT #1, CHR$(56); CHR$(12); CHR$(64); CHR$(112);
PRINT #1, CHR$(28); CHR$(12); CHR$(0); CHR$(112);
PRINT #1, CHR$(28); CHR$(0); CHR$(0); CHR$(224);
PRINT #1, CHR$(14); CHR$(0); CHR$(1); CHR$(224);
PRINT #1, CHR$(14); CHR$(0); CHR$(0); CHR$(0);
PRINT #1, CHR$(0); CHR$(0); CHR$(0); CHR$(0);
PRINT #1, CHR$(0); CHR$(0); CHR$(0); CHR$(0);
PRINT #1, CHR$(0); CHR$(0); CHR$(0); CHR$(0);
    
```



[References]

US (D Functions 1, 3, 6, **US (E** Function 11, **US (H** Function A1



US (G

[Name]	Select character style							
[Format]	ASCII	US	(G	<i>PL</i>	<i>PH</i>	<i>fn</i>	<i>[parameters]</i>
	Hex	1F	28	47	<i>PL</i>	<i>PH</i>	<i>fn</i>	<i>[parameters]</i>
	Decimal	31	40	71	<i>PL</i>	<i>PH</i>	<i>fn</i>	<i>[parameters]</i>

[Displays featuring this command] **DM-D500**.

[Displays not featuring this command] **DM-D110** and **DM-D210**.

[Description] Selects the character style. Some styles affect all character types, some affect one-byte alphanumeric character codes, and some affect two-byte Kanji characters.

- *fn* specifies the function number as shown in the tables that follow.
- The table below shows styles that affect all character types:

<i>fn</i>	Function Number	Description
32	Function A1	Specifies character size
33	Function A2	Specifies or cancels character highlighting
34	Function A3	Specifies or cancels reverse display of characters

- The table below shows styles that affect only one-byte, alphanumeric character types:

<i>fn</i>	Function Number	Description
64	Function B1	Specifies character font

- The table below shows styles that affect two-byte Kanji and external character types:

<i>fn</i>	Function Number	Description
96	Function C1	Specifies or cancels Kanji mode
97	Function C2	Selects the Kanji code system



fn	Function Number	Description
98	Function C3	Defines the user-defined characters as column type
99	Function C4	Defines the user-defined characters as raster type
100	Function C5	Deletes the data for the user-defined characters

- **pl** and **ph** specify the bytes following parameter **ph** (**fn** and [**parameters**]) as $(pl + (ph \times 256))$.
 - The other parameters are explained under each of the functions.
- [Note] ■ The command function is defined by the value of **fn**. The operation differs, depending on the function.



CONFIDENTIAL

US (G *p_L* *p_H* *fn* *x* *y* <Function A1>

[Format]	ASCII	US	(G	<i>p_L</i>	<i>p_H</i>	<i>fn</i>	<i>x</i>	<i>y</i>
	Hex	1F	28	47	03	00	20	x	y
	Decimal	31	40	71	3	0	32	x	y

[Range] **(*p_L* + *p_H* × 256) = 3** (***p_L* = 3, *p_H* = 0**)
***fn* = 32**
1 ≤ *x* ≤ 8
1 ≤ *y* ≤ 8

[Default] ***x* = 1, *y* = 1**

[Description] Specifies the amount the character is to be enlarged.

- ***x*** represents the enlargement in the horizontal direction.
- ***y*** specifies the enlargement in the vertical direction.
- The enlargement values for characters are defined within each window individually.



CONFIDENTIAL

US (G *p_L* *p_H* *fn* *m* <Function A2>

[Format]	ASCII	US	(G	<i>p_L</i>	<i>p_H</i>	<i>fn</i>	<i>m</i>
	Hex	1F	28	47	02	00	21	<i>m</i>
	Decimal	31	40	71	2	0	33	<i>m</i>

[Range] (***p_L*** + ***p_H*** × 256) = 2 (***p_L*** = 2, ***p_H*** = 0)
fn = 33
m = 0, 1, 48, 49

[Default] ***m*** = 0

[Description] Specifies character highlighting.

- When ***m*** = 1 or 49, highlighting is specified.
- When ***m*** = 0 or 48, highlighting is canceled.
- The highlighting values for characters are defined within each window individually.



CONFIDENTIAL

US (G *pL* *pH* *fn* *m* <Function A3>

[Format]	ASCII	US	(G	<i>pL</i>	<i>pH</i>	<i>fn</i>	<i>m</i>
	Hex	1F	28	47	02	00	22	<i>m</i>
	Decimal	31	40	71	2	0	34	<i>m</i>
[Range]	(<i>pL</i> + <i>pH</i> × 256) = 2 (<i>pL</i> = 2, <i>pH</i> = 0) <i>fn</i> = 34 <i>m</i> = 0, 1, 48, 49							
[Default]	<i>m</i> = 0							
[Description]	Specifies reverse display (dark characters on light background instead of light characters on dark background). <ul style="list-style-type: none">• When <i>m</i> = 1 or 49, reverse display is specified.• When <i>m</i> = 0 or 48, reverse display is canceled.■ The reverse display values for characters are defined within each window individually.							



CONFIDENTIAL

US (G *p_L* *p_H* *fn* *m* <Function B1>

[Format]	ASCII	US	(G	<i>p_L</i>	<i>p_H</i>	<i>fn</i>	<i>m</i>
	Hex	1F	28	47	02	00	40	<i>m</i>
	Decimal	31	40	71	2	0	64	<i>mm</i>

[Range] **(*p_L* + *p_H* × 256) = 2** (***p_L* = 2, *p_H* = 0**)
***fn* = 64**
***m* = 0, 1, 48, 49**

[Default] ***m* = 0**

[Description] Selects one-byte character codes (alphanumeric or user-defined).

- This command is defined within each window individually.
- This command is effective for windows only in graphics mode.

<i>m</i>	Function
0, 48	Selects font A (8 × 16)
1, 49	Selects font B (5 × 7)



CONFIDENTIAL

US (G *p_L* *p_H* *fn* *m* <Function C1>

[Format]	ASCII	US	(G	<i>p_L</i>	<i>p_H</i>	<i>fn</i>	<i>m</i>
	Hex	1F	28	47	02	00	60	<i>m</i>
	Decimal	31	40	71	2	0	96	<i>m</i>

[Range] $(p_L + p_H \times 256) = 2$ ($p_L = 2, p_H = 0$)
fn = 96
m = 0, 1, 48, 49

[Default] ***m*** = 0

[Description] Selects or cancels two-byte character codes.

- This command is defined within each window individually.

<i>m</i>	Function
0, 48	Kanji mode is canceled
1, 49	Kanji mode is selected



CONFIDENTIAL

US (G *p_L* *p_H* *fn* *m* <Function C2>

[Format]	ASCII	US	(G	<i>p_L</i>	<i>p_H</i>	<i>fn</i>	<i>m</i>
	Hex	1F	28	47	02	00	61	<i>m</i>
	Decimal	31	40	71	2	0	97	<i>m</i>

[Range] $(\mathbf{pL} + \mathbf{pH} \times 256) = 2$ ($\mathbf{pL} = 2, \mathbf{pH} = 0$)
fn = 97
m = 0, 1, 48, 49

[Default] ***m*** = 0

[Description] Selects the Kanji character code system for the Japanese Kanji model display.

<i>m</i>	Function
0, 48	JIS code system is specified
1, 49	SHIFT JIS code system is specified



CONFIDENTIAL

US (G *p_L* *p_H* *fn* *c1* *c2* *d1* ... *dk* <Function C3>

[Format]	ASCII	US	(G	<i>p_L</i>	<i>p_H</i>	<i>fn</i>	<i>c1</i>	<i>c2</i>	<i>d1</i> ... <i>dk</i>
	Hex	1F	28	47	<i>p_L</i>	<i>p_H</i>	62	<i>c1</i>	<i>c2</i>	<i>d1</i> ... <i>dk</i>
	Decimal	31	40	71	<i>p_L</i>	<i>p_H</i>	98	<i>c1</i>	<i>c2</i>	<i>d1</i> ... <i>dk</i>

[Range] (*p_L* + *p_H* × 256) = 35 (*p_L* = 35, *p_H* = 0)

fn = 98

c1, *c2* differ depending on the model and Kanji code system. See the table below.

0 ≤ *d* ≤ 255

k = 32

Model	Code system	<i>c1</i>	<i>c2</i>
Japanese Kanji	JIS X0208 (JIS)	77H	21H ≤ <i>c2</i> ≤ 7EH
	JIS X0208 (SHIFT JIS)	ECH	40H ≤ <i>c2</i> ≤ 7EH, 80H ≤ <i>c2</i> ≤ 9EH
Chinese Kanji	GB 2312	FEH	A1H ≤ <i>c2</i> ≤ FEH
Taiwanese Kanji	BIG-5	FEH	A1H ≤ <i>c2</i> ≤ FEH

[Description] Defines the user-defined Kanji character pattern for the character code specified by *c1* and *c2*.

- The character pattern data *d* used to define the character is processed as column type.
- Character pattern data used for definition is stored in sequence from the left end of the character pattern.



CONFIDENTIAL

US (G *p_L* *p_H* *fn* *c1* *c2* *d1* ... *dk* <Function C4>

[Format]	ASCII	US	(G	<i>p_L</i>	<i>p_H</i>	<i>fn</i>	<i>c1</i>	<i>c2</i>	<i>d1</i> ... <i>dk</i>
	Hex	1F	28	47	<i>p_L</i>	<i>p_H</i>	63	<i>c1</i>	<i>c2</i>	<i>d1</i> ... <i>dk</i>
	Decimal	31	40	71	<i>p_L</i>	<i>p_H</i>	99	<i>c1</i>	<i>c2</i>	<i>d1</i> ... <i>dk</i>

[Range] (*p_L* + *p_H* × 256) = 35 (*p_L* = 35, *p_H* = 0)

fn = 99

c1, *c2* differ depending on the model and Kanji code system. See the table below.

0 ≤ *d* ≤ 255

k = 32

Model	Code system	<i>c1</i>	<i>c2</i>
Japanese Kanji	JIS X0208 (JIS)	77H	21H ≤ <i>c2</i> ≤ 7EH
	JIS X0208 (SHIFT JIS)	ECH	40H ≤ <i>c2</i> ≤ 7EH, 80H ≤ <i>c2</i> ≤ 9EH
Chinese Kanji	GB 2312	FEH	A1H ≤ <i>c2</i> ≤ FEH
Taiwanese Kanji	BIG-5	FEH	A1H ≤ <i>c2</i> ≤ FEH

[Description] Defines the user-defined Kanji character pattern for the character code specified by *c1* and *c2*.

- The character pattern data *d* used to define the character is processed as raster type.
- Character pattern data used for definition is stored in sequence from the top of the character pattern.



US (G *p_L* *p_H* *fn* *c1* *c2* <Function C5>

[Format]	ASCII	US	(G	<i>p_L</i>	<i>p_H</i>	<i>fn</i>	<i>c1</i>	<i>c2</i>
	Hex	1F	28	47	<i>p_L</i>	<i>p_H</i>	64	<i>c1</i>	<i>c2</i>
	Decimal	31	40	71	<i>p_L</i>	<i>p_H</i>	100	<i>c1</i>	<i>c2</i>

[Range] (*p_L* + *p_H* × 256) = 3 (*p_L* = 3, *p_H* = 0)
fn = 100



c1, *c2* differ depending on the model and Kanji code system. See the table below.

Model	Code system	<i>c1</i>	<i>c2</i>
Japanese Kanji	JIS X0208 (JIS)	77H	21H ≤ <i>c2</i> ≤ 7EH
	JIS X0208 (SHIFT JIS)	ECH	40H ≤ <i>c2</i> ≤ 7EH, 80H ≤ <i>c2</i> ≤ 9EH
Chinese Kanji	GB 2312	FEH	A1H ≤ <i>c2</i> ≤ FEH
Taiwanese Kanji	BIG-5	FEH	A1H ≤ <i>c2</i> ≤ FEH

[Description] Deletes the user-defined character pattern for the character code specified by *c1* and *c2*.



[Examples]

Action	Command/Example
Display regular text	<pre>PRINT #1, CHR\$(10); PRINT #1, "EPSON ";</pre> 
<p>Function A1 Specify character size</p> <p>Function A2 Specify highlighting</p>	<pre>PRINT #1, CHR\$(31); "(G"; PRINT #1, CHR\$(3); CHR\$(0); 'pL pH PRINT #1, CHR\$(32); 'fn PRINT #1, CHR\$(2); CHR\$(2); 'x y PRINT #1, CHR\$(31); "(G"; PRINT #1, CHR\$(2); CHR\$(0); 'pL pH PRINT #1, "!1"; 'fn m PRINT #1, "DM-D500 ";</pre> 



<p>Function A1 Specify character size</p> <p>Function A2 Highlight character display</p> <p>Function A3 Specify reverse character display</p>	<pre>PRINT #1, CHR\$(31); "(G"; PRINT #1, CHR\$(3); CHR\$(0); 'pL pH PRINT #1, CHR\$(32); 'fn PRINT #1, CHR\$(1); CHR\$(1); 'x y PRINT #1, CHR\$(31); "(G"; PRINT #1, CHR\$(2); CHR\$(0); 'pL pH PRINT #1, "!0"; 'fn m PRINT #1, CHR\$(31); "(G"; PRINT #1, CHR\$(2); CHR\$(0); 'pL pH PRINT #1, CHR\$(34); "1"; 'fn m PRINT #1, "SERIES"</pre>
<p>Function A3 Cancel reverse display</p> <p>Function B1 Specify font B</p>	<div data-bbox="1276 743 1822 883" data-label="Image"> </div> <pre>PRINT #1, CHR\$(31); "(G"; PRINT #1, CHR\$(2); CHR\$(0); 'pL pH PRINT #1, CHR\$(34); "0"; 'fn m</pre>



CONFIDENTIAL

<p>Function A3 Cancel reverse display</p> <p>Function B1 Specify font B (continued)</p>	<pre>PRINT #1, CHR\$(31); "(G"; PRINT #1, CHR\$(2); CHR\$(0); 'pL pH PRINT #1, "@1"; 'fn m PRINT #1, "5 x 7 Font";</pre>
<p>Function B1 Specify font A</p> <p>Function C1 Specify the Kanji mode and display Kanji</p>	<div data-bbox="1276 386 1822 526" data-label="Image"> <p>A screenshot of an EPSON printer's LCD display showing the text "EPSON DM-D500 SERIES" in large green characters, with "5 x 7 Font" in smaller green characters below it.</p> </div> <pre>PRINT #1, CHR\$(31); "(G"; PRINT #1, CHR\$(2); CHR\$(0); 'pL pH PRINT #1, "@0"; 'n m PRINT #1, CHR\$(31); "(G"; PRINT #1, CHR\$(2); CHR\$(0); 'pL pH PRINT #1, "`1"; 'n m PRINT #1, "KANJI"; PRINT #1, CHR\$(&HA3); CHR\$(&HCB); PRINT #1, CHR\$(&HA3); CHR\$(&HC1); PRINT #1, CHR\$(&HA3); CHR\$(&HCE); PRINT #1, CHR\$(&HA3); CHR\$(&HCA); PRINT #1, CHR\$(&HA3); CHR\$(&HC9);</pre> <div data-bbox="1276 1162 1822 1302" data-label="Image"> <p>A screenshot of an EPSON printer's LCD display showing the text "EPSON DM-D500 SERIES" in large green characters, with "5 x 7 Font" in smaller green characters below it. The word "KANJI" is displayed in large green characters across the bottom of the screen.</p> </div>

[References]

US C, US X, US r, US ., US ,, US ;, US #, US (H Function C1



US (H

[Name]	Set display layout					
[Format]	ASCII	US	(H	fn	[parameters]
	Hex	1F	28	48	fn	[parameters]
	Decimal	31	40	72	fn	[parameters]

[Displays featuring this command] **DM-D500**.

[Displays not featuring this command] **DM-D110** and **DM-D210**.

[Description] Sets the display layout, as described in the table below.

- **fn** specifies the function number.

fn	Function number	Description
32	Function A1	Specifies the line spacing
33	Function A2	Moves the display data in the current window
64	Function B1	Specifies the character spacing for one-byte character codes
96	Function C1	Specifies the character spacing for two-byte character codes

- **pl** and **ph** specify the bytes following parameter **ph** (**fn** and **[parameters]**) as $(pl + (ph \times 256))$.
- The other parameters are explained under each of the functions.

[Note] ■ The command function is defined by the value of **fn**. The operation differs, depending on the function.



CONFIDENTIAL

US (H *pL* *pH* *fn* *n* <Function A1>

[Format]	ASCII	US	(H	<i>pL</i>	<i>pH</i>	<i>fn</i>	<i>n</i>
	Hex	1F	28	48	02	00	20	<i>n</i>
	Decimal	31	40	72	32	0	32	<i>n</i>

[Range] $(\mathbf{pL} + \mathbf{pH} \times 256) = 2$ ($\mathbf{pL} = 2, \mathbf{pH} = 0$)
 $\mathbf{fn} = 32$
 $0 \leq \mathbf{n} \leq 255$

[Default] $\mathbf{n} = 16$

[Description] Specifies the line spacing as ***n*** dots.

- The spacing values for lines are defined within each window individually.
- This command is effective for windows in graphics mode only.



US (H *p_L* *p_H* *fn* *x1_L* *x1_H* *y1_L* *y1_H* *dx_L* *dx_H* *dy_L* *dy_H* *x2_L* *x2_H* *y2_L* *y2_H* <Function A2>

[Format]	ASCII	US	(H	<i>p_L</i>	<i>p_H</i>	<i>fn</i>	<i>x1_L</i>	<i>x1_H</i>	<i>y1_L</i>	<i>y1_H</i>	<i>dx_L</i>	<i>dx_H</i>	<i>dy_L</i>	<i>dy_H</i>	<i>x2_L</i>	<i>x2_H</i>	<i>y2_L</i>	<i>y2_H</i>
	Hex	1F	28	48	0D	00	21	<i>x1_L</i>	<i>x1_H</i>	<i>y1_L</i>	<i>y1_H</i>	<i>dx_L</i>	<i>dx_H</i>	<i>dy_L</i>	<i>dy_H</i>	<i>x2_L</i>	<i>x2_H</i>	<i>y2_L</i>	<i>y2_H</i>
	Decimal	31	40	72	13	0	33	<i>x1_L</i>	<i>x1_H</i>	<i>y1_L</i>	<i>y1_H</i>	<i>dx_L</i>	<i>dx_H</i>	<i>dy_L</i>	<i>dy_H</i>	<i>x2_L</i>	<i>x2_H</i>	<i>y2_L</i>	<i>y2_H</i>

[Range] (***p_L*** + ***p_H*** × 256) = 13 (***p_L*** = 13, ***p_H*** = 0)
fn = 33

0 ≤ (***x1_L*** + ***x1_H*** × 256) ≤ 65534 (0 ≤ ***x1_L*** ≤ 255, 0 ≤ ***x1_H*** ≤ 255)

0 ≤ (***y1_L*** + ***y1_H*** × 256) ≤ 65534 (0 ≤ ***y1_L*** ≤ 255, 0 ≤ ***y1_H*** ≤ 255)

1 ≤ (***dx_L*** + ***dx_H*** × 256) ≤ 65535 (0 ≤ ***dx_L*** ≤ 255, 0 ≤ ***dx_H*** ≤ 255)

1 ≤ (***dy_L*** + ***dy_H*** × 256) ≤ 65535 (0 ≤ ***dy_L*** ≤ 255, 0 ≤ ***dy_H*** ≤ 255)

0 ≤ (***x2_L*** + ***x2_H*** × 256) ≤ 65534 (0 ≤ ***x2_L*** ≤ 255, 0 ≤ ***x2_H*** ≤ 255)

0 ≤ (***y2_L*** + ***y2_H*** × 256) ≤ 65534 (0 ≤ ***y2_L*** ≤ 255, 0 ≤ ***y2_H*** ≤ 255)

[Description] Moves the display data in the current window.

- ***x1_L***, ***x1_H***, ***y1_L***, ***y1_H*** specify the ***x*** and ***y*** coordinates of the origin point of the area to be moved. (Four bytes are required to define this point, because a high byte and low byte are necessary to define the maximum point, 65534.)
 - In graphics mode, the origin of the area to be moved is specified as (***x1_L*** + ***x1_H*** × 256) dots over horizontally and (***y1_L*** + ***y1_H*** × 256) dots down vertically.
 - In character mode, the origin of the area to be moved is specified as (***x1_L*** + ***x1_H*** × 256) columns over and (***y1_L*** + ***y1_H*** × 256) lines down.
- ***dx_L***, ***dx_H***, ***dy_L***, ***dy_H*** specify the size of the area to be moved. (Four bytes are required to define this point, because a high byte and low byte are needed to define the maximum, 65535.)
 - In graphics mode, the size of the area to be moved is specified as a point diagonal to the first point, which is (***dx_L*** + ***dx_H*** × 256) dots over horizontally and (***dy_L*** + ***dy_H*** × 256) dots down vertically.
 - In character mode, the size of the area to be moved is specified as a point diagonal to the first point, which is (***dx_L*** + ***dx_H*** × 256) columns over and (***dy_L*** + ***dy_H*** × 256) lines down.



CONFIDENTIAL

- **x2L, x2H, y2L, y2H** specify the **x** and **y** coordinates of the destination point of the area to be moved. (Four bytes are required to define this point, because a high byte and low byte are necessary to define the maximum point, 65534.)
 - In graphics mode, the destination of the area to be moved is specified as (**x2L** + **x2H** × 256) dots over horizontally and (**y2L** + **y2H** × 256) dots down vertically.
 - In character mode, the destination of the area to be moved is specified as (**x2L** + **x2H** × 256) columns over and (**y2L** + **y2H** × 256) lines down.



CONFIDENTIAL

US (H *pL* *pH* *fn* *n1* *n2* <Function B1>

[Format]	ASCII	US	(G	<i>pL</i>	<i>pH</i>	<i>fn</i>	<i>n1</i>	<i>n2</i>
	Hex	1F	28	47	03	00	40	n1	n2
	Decimal	31	40	71	3	0	64	n1	n2

[Range] $(pL + pH \times 256) = 3$ ($pL = 3, pH = 0$)
 $fn = 32$
 $0 \leq n1 \leq 255$
 $0 \leq n2 \leq 255$

[Default] $n1 = 0, n2 = 0$

- [Description]
- Specifies the amount of space to be added on the left or right side of a one-byte character.
n1 specifies the amount of space on the left side of a character as ***n1*** dots.
n2 specifies the amount of space on the right side of a character as ***n2*** dots.
 - The spacing values for characters are defined within each window individually.
 - This command is effective only for windows in graphics mode.



US (H *pL* *pH* *fn* *n1* *n2* <Function C1>

[Format]	ASCII	US	(G	<i>pL</i>	<i>pH</i>	<i>fn</i>	<i>n1</i>	<i>n2</i>
	Hex	1F	28	47	03	00	60	n1	n2
	Decimal	31	40	71	3	0	96	n1	n2


[Range] **(*pL* + *pH* × 256) = 3** **(*pL* = 3, *pH* = 0)**
***fn* = 96**
 0 ≤ ***n1*** ≤ 255
 0 ≤ ***n2*** ≤ 255

[Default] ***n1* = 0, *n2* = 0**

- [Description]
- Specifies the amount of space to be added on the left or right side of a two-byte character.
n1 specifies the amount of space on the left side of a character as ***n1*** dots.
n2 specifies the amount of space on the right side of a character as ***n2*** dots.
 - The spacing values for characters are defined within each window individually.
 - This command is effective only for windows in graphics mode.



[Examples]

Action	Command/Example
<p>US (G Function C1 Specify Kanji mode and send data</p>	<pre>PRINT #1, CHR\$(31); "(G"; PRINT #1, CHR\$(2); CHR\$(0); 'pL pH PRINT #1, CHR\$(96); 'fn C1 PRINT #1, CHR\$(49); 'n PRINT #1, "EPSON"; PRINT #1, CHR\$(&HA3); CHR\$(&HC5); PRINT #1, CHR\$(&HA3); CHR\$(&HD0); PRINT #1, CHR\$(&HA3); CHR\$(&HD3); PRINT #1, CHR\$(&HA3); CHR\$(&HCF); PRINT #1, CHR\$(&HA3); CHR\$(&HCE)</pre> 
<p>Function A1 Specify the line spacing Function B1 and Function C1 Specify the character spacing</p>	<pre>PRINT #1, CHR\$(31); "(H"; PRINT #1, CHR\$(2); CHR\$(0); 'pL pH PRINT #1, CHR\$(32); 'fn A1 PRINT #1, CHR\$(4); 'n PRINT #1, CHR\$(31); "(H"; PRINT #1, CHR\$(3); CHR\$(0); 'pL pH PRINT #1, CHR\$(64); 'fn B1 PRINT #1, CHR\$(3); CHR\$(3); 'n1 n2</pre>



<p>Function A1 Specify the line spacing</p> <p>Function B1 and Function C1 Specify the character spacing (continued)</p>	<pre>PRINT #1, CHR\$(31); "(H"; PRINT #1, CHR\$(3); CHR\$(0); 'pL pH PRINT #1, CHR\$(96); 'fn C1 PRINT #1, CHR\$(8); CHR\$(8); 'n1 n2</pre>
<p>US \$ Move cursor and send data</p>	<pre>PRINT #1, CHR\$(31); "\$"; PRINT #1, CHR\$(1); CHR\$(8); 'n m PRINT #1, "E"; PRINT #1, CHR\$(10); PRINT #1, "P"; PRINT #1, CHR\$(10); PRINT #1, "S"; PRINT #1, CHR\$(10); PRINT #1, "O"; PRINT #1, CHR\$(10); PRINT #1, "N"; PRINT #1, CHR\$(&HA3); CHR\$(&HC5); PRINT #1, CHR\$(31); CHR\$(10); PRINT #1, CHR\$(&HA3); CHR\$(&HD0); PRINT #1, CHR\$(31); CHR\$(10); PRINT #1, CHR\$(&HA3); CHR\$(&HD3); PRINT #1, CHR\$(31); CHR\$(10);</pre>



CONFIDENTIAL

<p>US \$ Move cursor and send data (continued)</p>	<pre>PRINT #1, CHR\$(&HA3); CHR\$(&HCF); PRINT #1, CHR\$(31); CHR\$(10); PRINT #1, CHR\$(&HA3); CHR\$(&HCE)</pre>
<p>Function A2 Move the display data in the current window</p>	<div data-bbox="1276 331 1822 467" style="background-color: black; color: green; padding: 5px; margin-bottom: 10px;"> <pre>EPSONEPSON EPSONEPSON</pre> </div> <pre>PRINT #1, CHR\$(31); "(H"; PRINT #1, CHR\$(13); CHR\$(0); 'pL pH PRINT #1, CHR\$(33); 'fn A2 PRINT #1, CHR\$(0); CHR\$(0); 'x1L x1H PRINT #1, CHR\$(0); CHR\$(0); 'y1L y1H PRINT #1, CHR\$(40); CHR\$(0); 'dxL dxH PRINT #1, CHR\$(16); CHR\$(0); 'dyL dyH PRINT #1, CHR\$(216); CHR\$(0); 'x2L x2H PRINT #1, CHR\$(48); CHR\$(0); 'y2L y2H</pre> <div data-bbox="1276 935 1822 1071" style="background-color: black; color: green; padding: 5px;"> <pre>EPSON EPSONEPSON EPSON</pre> </div>

[References]

BS, HT, LF, US LF, US \$, US MD1, US MD2, US (D Functions 1, 3, 6, **US (F** Functions 1, 3, 4, 5, **US (G** Functions A1, B1, C1

