

Multi-Font Displays

Font Tables and Application Guide

Newhaven Display International, Inc.

2661 Galvin Ct.

Elgin IL, 60124

Ph: 847-844-8795

Fax: 847-844-8796

www.newhavendisplay.com

nhtech@newhavendisplay.com

nhsales@newhavendisplay.com

Table of Contents

1. Document Revision History
2. Font Tables
 - 2.1. ASCII
 - 2.2. Latin Basic
 - 2.3. Latin Supplement
 - 2.4. Latin Extended A
 - 2.5. Latin Extended B
 - 2.6. Latin Extended Additional
 - 2.7. Greek
 - 2.8. Cyrillic
 - 2.9. Hebrew
 - 2.10. Thai
 - 2.11. Arabic
 - 2.12. ISO8859
 - 2.13. LCM 5x10
3. Font Data Arrangement
 - 3.1. 5x7 font
 - 3.2. 7x8 font
 - 3.3. 8x16 font
 - 3.4. Width-Adjusted font
 - 3.5. CJK font
4. Calculation of Font Addresses
 - 4.1. 5x7 ASCII
 - 4.2. 7x8 ASCII
 - 4.3. 8x16 ASCII
 - 4.4. Width-Adjusted Arial ASCII
 - 4.5. 8x16 Latin
 - 4.6. 8x16 Greek
 - 4.7. 8x16 Cyrillic
 - 4.8. 8x16 Hebrew
 - 4.9. 8x16 Thai
 - 4.10. Width-Adjusted Latin
 - 4.11. Width-Adjusted Greek
 - 4.12. Width-Adjusted Cyrillic
 - 4.13. Width-Adjusted Arabic
 - 4.14. GB2312 (Simplified Chinese)
 - 4.15. KSC5605 (Korean)
 - 4.16. JIS0208 (Japanese)
 - 4.17. ISO8859-1
 - 4.18. ISO8859-2
 - 4.19. ISO8859-3
 - 4.20. ISO8859-4
 - 4.21. ISO8859-5
 - 4.22. ISO8859-7
 - 4.23. ISO8859-8
 - 4.24. ISO8859-9
 - 4.25. ISO8859-10
 - 4.26. ISO8859-11
 - 4.27. ISO8859-13
 - 4.28. ISO8859-14
 - 4.29. ISO8859-15

- 4.30. ISO8859-16
 - 4.31. LCM 5x10-1
 - 4.32. LCM 5x10-2
 - 4.33. LCM 5x10-3
 - 4.34. LCM 5x10-8
 - 4.35. LCM 5x10-11
 - 4.36. LCM 5x10-12
 - 4.37. LCM 5x10-13
5. How to use the Multi-Font displays.

1. Document Revision History

Revision	Date	Description	Changed by
0	10/15/2012	Preliminary Release	-
1	11/5/2012	Initial Product Release	

2. Font Tables

2.1. ASCII

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
ASCII	0	!	“	#	\$	%	&	‘	()	*	+	,	-	.	/
1	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
2	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
3	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
4	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
5	p	q	r	s	t	u	v	w	x	y	z	{		}	~	

2.2. Latin Basic

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Latin Basic	0	!	“	#	\$	%	&	‘	()	*	+	,	-	.	/
1	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
2	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
3	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
4	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
5	p	q	r	s	t	u	v	w	x	y	z	{		}	~	

2.3. Latin Supplement

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Latin Supplement	0	ı	ç	£	¤	¥	ı	§	¨	©	ª	«	¬	-	®	-
1	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
2	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
3	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
4	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
5	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

2.4. Latin Extended A

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Latin Extended A	Ā	ā	Ă	ă	Ą	ą	Ć	ć	Ĉ	ĉ	Č	č	Ď	ď	Đ	đ
1	Ð	đ	Ē	ē	Ĕ	ĕ	Ė	ė	Ę	ę	Ě	ě	Ĝ	ĝ	Ğ	ğ
2	Ġ	ġ	Ģ	ģ	Ĥ	ĥ	Ħ	ħ	Ĩ	ĩ	Ī	ī	Ĵ	ĵ	Ķ	ķ
3	Ĭ	ĭ	Ĵ	ĵ	Ķ	ķ	κ	Ł	ł	Ļ	ļ	Ľ	ľ	Ł	ł	Ł
4	ł	Ł	ł	Ń	ń	Ņ	ņ	Ň	ň	Ŋ	ŋ	Ō	ō	Ŏ	ö	
5	Ŏ	ö	Œ	œ	Ř	ř	Ŕ	ŕ	Ř	ř	Ś	ś	Ŝ	ŝ	Ş	ş
6	Š	š	Ţ	ţ	Ť	ť	Ŧ	ŧ	Ũ	ũ	Ū	ū	Ŭ	ŭ	Ů	ů
7	Ů	ů	Ū	ū	Ŭ	ŭ	Ÿ	ÿ	Ÿ	ÿ	Ž	ž	Ž	ž	Ž	ž

2.5. Latin Extended B

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Latin Extended B	Œ	œ	Ɔ	ɔ	Ɔ	ɔ	Ɔ	ɔ	Ɔ	ɔ	Ɔ	ɔ	Ɔ	ɔ	Ɔ	ɔ
1	ur	Ů	Ū	Y	ý	Z	z	Ɔ	ɔ	Ɔ	ɔ	Ɔ	ɔ	Ɔ	ɔ	Ɔ
2	ı	ı	ı	ı	DŽ	Dž	dž	LJ	Lj	lj	NJ	Nj	nj	Ă	ă	Ĭ
3	Ĵ	DZ	Dz	dz	Ĝ	ĝ	Hu	h	Ń	ń	Ă	ă	Æ	æ	Ø	ó
4	Ŕ	ř	Ŕ	ř	Ů	ů	Ū	ū	Ş	ş	Ţ	ţ	Ɔ	ɔ	Ť	ť

2.6. Latin Extended Additional

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
Latin Extended Additional	0	À	á	Â	â	Ã	ã	Ä	ä	Å	å	Ā	ā	Ă	ă	Ȧ	ȧ
	1	Ȧ	ȧ	Ȧ	ȧ	Ȧ	ȧ	Ȧ	ȧ	Ȧ	ȧ	Ȧ	ȧ	Ȧ	ȧ	Ȧ	ȧ
	2	Ë	è	Ê	ê	Ë	ë	Ě	ě	Ī	ī	Į	į	Ō	ō	Ȫ	ȫ
	3	Ō	ó	Ȫ	ȫ	Ō	ó	Ȫ	ȫ	Ō	ó	Ȫ	ȫ	Ō	ó	Ȫ	ȫ
	4	Ō	õ	Ȫ	ȫ	Ū	ū	Ū	ů	Ū	ú	Ū	ű	Ū	ů	Ū	ű
	5	Ū	ұ	Ÿ	ÿ	Ÿ	ÿ	Ÿ	ÿ	Ÿ	ÿ						

2.7. Greek

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
Greek	0				´	˘						Ϸ	ϸ	Ϲ	;		
	1				´	˘	Α	·	Ε	Η	Ι		Ο		Υ	Ω	
	2	ι	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο
	3	Π	Ρ		Σ	Τ	Υ	Φ	Χ	Ψ	Ω	Ϊ	Ϋ	ά	έ	ή	ί
	4	ϐ	α	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο
	5	π	ρ	ς	σ	τ	υ	φ	χ	ψ	ω	ϊ	ϋ	ό	ύ	ώ	

2.8. Cyrillic

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
Cyrillic	0	Ё	ё	Ђ	ђ	Є	є	І	і	Ј	љ	Њ	Ћ	Ќ	Ў	Ѳ	
	1	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
	2	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
	3	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
	4	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
	5	ё	ё	ђ	ђ	є	є	і	і	ј	љ	њ	ћ	ќ	ў	Ѳ	Ѳ
	6	Ѓ	ѓ	Є	є	І	і	Ј	ј	Љ	љ	Њ	њ	Ћ	ќ	Ў	Ѳ
	7	К	к	Ѓ	ѓ	Н	н	Ѓ	ѓ	Ѳ	Ѳ	Ѓ	ѓ	Ѓ	ѓ	У	у
	8	У	у	Х	х	Ц	ц	Ч	ч	Ч	ч	Н	н	Ѳ	Ѳ	Ѳ	Ѳ
	9	І	Ѓ	Ѓ	Ѓ	Л	л	Н	н	Ѓ	Ѓ	Ч	ч	М	м	І	
	A	Ă	ă	Ă	ă	Æ	æ	Ë	ë	Ə	ə	Ë	ë	Ж	ж	Ž	ž
	B	Ɔ	Ɔ	Й	й	Й	й	Ö	ö	Ө	ө	Ө	ө	Э	э	У	у
	C	Ÿ	ÿ	Ÿ	ÿ	Č	č	Г	г	Б	б	Г	г	Х	х	Ж	ж

2.9. Hebrew

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
Hebrew (note: some symbols are shown with 0 for reference)	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	0
	3		0	0													
	4	א	ב	ג	ד	ה	ו	ז	ח	ט	י	ך	כ	ל	ם	מ	ן
	5	נ	ס	ע	ף	פ	ץ	צ	ק	ר	ש	ת					
	6	ן	ן	ן	ן	ן											

2.10. Thai

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F		
Thai (note: some symbols are shown with character for reference)	0	ก	ข	ฃ	ค	ฅ	ฉ	ง	จ	ฉ	ช	ฌ	ฉ	ญ	ฎ	ฏ		
	1	ฐ	ฑ	ฒ	ณ	ด	ต	ถ	ธ	น	บ	ป	ผ	ฝ	พ	ฟ		
	2	ภ	ม	ย	ร	ฤ	ล	ฬ	ว	ศ	ษ	ห	ฬ	อ	ฮ	๑		
	3	๕	๖	๗	๘	๙	๑๐	๑๑	๑๒	๑๓	๑๔	๑๕						฿
	4	๑	๒	๓	๔	๕	๖	๗	๘	๙	๑๐	๑๑	๑๒	๑๓	๑๔	๑๕	๑๖	๑๗
	5	๐	๑	๒	๓	๔	๕	๖	๗	๘	๙	๑๐	๑๑					
	6																	
	7																	
	2	Ĝ ĝ	Ĝ ĝ	Ĝ ĝ	Ĝ ĝ	Ĥ ĥ	Ĥ ĥ	Ĥ ĥ	Ĥ ĥ	Ĭ ĭ	Ĭ ĭ	Ĭ ĭ	Ĭ ĭ	Ĭ ĭ	Ĭ ĭ	Ĭ ĭ	Ĭ ĭ	Ĭ ĭ
	3	Ĭ ĭ	Ĭ ĭ	Ĭ ĭ	Ĭ ĭ	Ĭ ĭ	Ĭ ĭ	Ĭ ĭ	Ĭ ĭ	Ĭ ĭ	Ĭ ĭ	Ĭ ĭ	Ĭ ĭ	Ĭ ĭ	Ĭ ĭ	Ĭ ĭ	Ĭ ĭ	Ĭ ĭ
	4	Ĭ ĭ	Ĭ ĭ	Ĭ ĭ	Ĭ ĭ	Ĭ ĭ	Ĭ ĭ	Ĭ ĭ	Ĭ ĭ	Ĭ ĭ	Ĭ ĭ	Ĭ ĭ	Ĭ ĭ	Ĭ ĭ	Ĭ ĭ	Ĭ ĭ	Ĭ ĭ	Ĭ ĭ
	5	Ŏ ŏ	Ŏ ŏ	Ŏ ŏ	Ŏ ŏ	Ŏ ŏ	Ŏ ŏ	Ŏ ŏ	Ŏ ŏ	Ŏ ŏ	Ŏ ŏ	Ŏ ŏ	Ŏ ŏ	Ŏ ŏ	Ŏ ŏ	Ŏ ŏ	Ŏ ŏ	Ŏ ŏ
	6	Š š	Š š	Š š	Š š	Š š	Š š	Š š	Š š	Š š	Š š	Š š	Š š	Š š	Š š	Š š	Š š	Š š
	7	Ů ů	Ů ů	Ů ů	Ů ů	Ů ů	Ů ů	Ů ů	Ů ů	Ů ů	Ů ů	Ů ů	Ů ů	Ů ů	Ů ů	Ů ů	Ů ů	Ů ů
	4	Ŏ ŏ	Ŏ ŏ	Ŏ ŏ	Ŏ ŏ	Ŏ ŏ	Ŏ ŏ	Ŏ ŏ	Ŏ ŏ	Ŏ ŏ	Ŏ ŏ	Ŏ ŏ	Ŏ ŏ	Ŏ ŏ	Ŏ ŏ	Ŏ ŏ	Ŏ ŏ	Ŏ ŏ
5	Ů ů	Ů ů	Ů ů	Ů ů	Ů ů	Ů ů	Ů ů	Ů ů	Ů ů	Ů ů	Ů ů	Ů ů	Ů ů	Ů ů	Ů ů	Ů ů	Ů ů	

2.11. Arabic

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
Width-Adjusted x 16 Arabic (note: some symbols are combined with character for reference)	0																
	1																؟
	2		ء	آ	أ	ؤ	إ	ئ	ا	ب	ة	ن	ث	ج	ح	خ	د
	3	ذ	ر	ز	س	ش	ص	ض	ط	ظ	ع	غ					
	4	-	ف	ق	ك	ل	م	ن	ه	و	ى	ي	ح	خ	ح	ح	ح
	5	ح	ح	ح	ح	ح											
	6	٠	١	٢	٣	٤	٥	٦	٧	٨	٩	٪	,	,	*		
	7	ح	أ	أ	إ	أ	و	و	ي	ث	ث	ب	ب	ب	ب	ب	ب
	8	ب	ب	ب	ب	ب	ب	ب	ب	ب	ب	ب	ب	ب	ب	ب	ب
	9	ب	ب	ب	ب	ب	ب	ب	ب	ب	ب	ب	ب	ب	ب	ب	ب
	A	ب	ب	ب	ب	ب	ب	ب	ب	ب	ب	ب	ب	ب	ب	ب	ب
	B	ب	ب	ب	ب	ب	ب	ب	ب	ب	ب	ب	ب	ب	ب	ب	ب
	C	ه	ه	ه	ه	ه	ه	ه	ه	ه	ه	ه	ه	ه	ه	ه	ه
D	ي	ي	ي	ي	ي	ي	ي	ي	ي	ي	ي	ي	ي	ي	ي	ي	
E	ح	ح	ح	ح	ح	ح	ح	ح	ح	ح	ح	ح	ح	ح	ح	ح	
F	٠	١	٢	٣	٤	٥	٦	٧	٨	٩							
10	أ	أ	ب	ب	ب	ب	ب	ب	ب	ب	ب	ب	ب	ب	ب	ب	
11	ث	ث	ث	ث	ث	ث	ث	ث	ث	ث	ث	ث	ث	ث	ث	ث	
12	ث	ث	ث	ث	ث	ث	ث	ث	ث	ث	ث	ث	ث	ث	ث	ث	

13	چ	چ	ب	ب	ت	ت	ث	ث	ط	ط	ظ	ظ	ك	ك
14	ك	ك	گ	گ	گ	گ	ج	ج	ج	ج	گ	گ	س	س
15	ط	ط	ظ	ظ	ه	ه	ه	ب	ب	ه	ه	ه	ع	ع
16	ع	ع												
17														
18				ث	ث	ح	ح	و	و	و	و	و	و	و
19	و	و	و	و	ي	ي	پ	پ	د	د	نا	نا	نه	نو
1A	نو	نو	نو	نو	نو	نو	ئي	ئي	ئي	ئي	ئي	ئي	ي	ي
1B	ء	ء					ء	ء	ء	ء	ء	ء	ء	ء
1C	ء	ا	ا	ا	ا	و	و	ا	ا	ئ	ئ	ئ	ا	ب
1D	ب	ب	ب	ة	ة	ت	ت	ت	ت	ث	ث	ث	ج	ج
1E	ج	ح	ح	ح	خ	خ	خ	خ	د	د	ذ	ذ	ر	ز
1F	ز	س	س	س	س	ش	ش	ش	ص	ص	ص	ص	ض	ض
20	ظ	ط	ط	ط	ظ	ظ	ظ	ظ	ع	ع	ع	ع	غ	غ
21	غ	ف	ف	ف	ق	ق	ق	ق	ك	ك	ك	ك	ل	ل
22	ل	م	م	م	ن	ن	ن	ن	ه	ه	ه	ه	و	ي
23	ي	ي	ي	ي	لا	لا	لا	لا	لا	لا	لا	لا		

2.12. ISO8859

2.12.1. All ISO8859 tables - characters 0x20 ~ 0x7F are equal to ASCII:

20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F		
	!	"	#	\$	%	&	'	()	*	+	,	-	.	/		
30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F		
	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?	
40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F		
	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	
50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F		
	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_	
60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F		
	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	
70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F		
	p	q	r	s	t	u	v	w	x	y	z	{		}	~		

2.12.2. ISO8859-1 (characters 0xA0 ~ 0xFF)

A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF		
	ı	ϕ	£	κ	¥	ı	§	·	©	≡	«	¬	-	®	-		
B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF		
	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿		
C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF		
	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï	
D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF		
	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß	
E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF		
	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï	
F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF		
	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ	

2.12.3. ISO8859-2 (characters 0xA0 ~ 0xFF)

A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF		
	Ā	ı	Ł	κ	Ĺ	Š	§	·	Š	Š	Ť	Ž	-	Ž	Ž		
B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF		
	ą	ˆ	ł	ˆ	ĺ	š	ˆ	ˆ	š	š	ť	ž	ˆ	ž	ž		
C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF		
	Ř	Ā	Ā	Ā	Ā	Ĺ	Č	Ç	Č	Ě	Ě	Ě	Ī	Ī	Ī	Ī	
D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF		
	Ð	Ñ	Ñ	Ò	Ó	Ö	×	Ř	Û	Ú	Û	Ü	Ý	Ť	ß		
E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF		
	ř	á	â	ã	ä	ı	č	ç	ě	ě	ě	ë	í	î	ï	ď	
F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF		
	đ	ñ	ň	ó	ô	ö	÷	ř	û	ú	ű	ü	ý	ť	·		

2.12.4. ISO8859-3 (characters 0xA0 ~ 0xFF)

A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF		
	Ĥ	ı	£	κ		Ĥ	§	·	İ	Š	Ğ	Ĵ	-		Ž		
B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF		
	ĥ	²	³	´	µ	ĥ	·	¸	ı	š	ğ	ĵ	¼		ž		
C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF		
	Ā	Ā	Ā	Ā	Č	Č	Ç	È	É	Ê	Ë	Ī	Ī	Ī	Ī		
D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF		
	Ñ	Ò	Ó	Ô	Ğ	Ö	×	Ğ	Û	Ú	Û	Ü	Ü	Ŝ	ß		
E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF		
	à	á	â	ã	ä	č	ç	è	é	ê	ë	ì	í	î	ï		
F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF		
	ñ	ò	ó	ô	ğ	ö	÷	ğ	û	ú	ű	ü	ű	ș	·		

2.12.5. ISO8859-4 (characters 0xA0 ~ 0xFF)

A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF
À	Á	Â	Ã	Ä	Å	Ĳ	Š	Š	Š	Ě	Ĝ	Ŧ	-	Ž	-
à	á	â	ã	ä	å	ĳ	š	š	š	ě	ĝ	ŧ	ð	ž	ŋ
Ā	Ā	Ā	Ā	Ā	Ā	Æ	Ī	Ī	Ī	Ē	Ē	Ē	Ē	Ī	Ī
ā	ā	ā	ā	ā	ā	æ	ī	ī	ī	ē	ē	ē	ē	ī	ī
Ð	Ŋ	Ō	Ɔ	Ô	Õ	Ö	×	Ø	Ū	Ū	Ū	Ü	Ŭ	Ū	Ɔ
ð	ŋ	ō	Ɔ	ô	õ	ö	×	ø	ū	ū	ū	ü	ŭ	ū	Ɔ

2.12.6. ISO8859-5 (characters 0xA0 ~ 0xFF)

A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF
Ě	Ђ	Ѓ	Ѕ	Є	Ѕ	І	Ї	Ј	Љ	Њ	Ћ	Ќ	-	Ў	Ў
ě	ђ	ѓ	ѕ	є	ѕ	і	ї	ј	љ	њ	ќ	ќ	–	ў	ў
Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
№	ë	ђ	ѓ	ѕ	є	і	ї	ј	љ	њ	ќ	ќ	–	ў	ў

2.12.7. ISO8859-7 (characters 0xA0 ~ 0xFF)

A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF
ı	Š	Š	Š	Š	Š	Š	Š	Š	Š	Š	Š	Š	Š	Š	Š
ı	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
ı	À	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο
ı	Π	Ρ	Σ	Τ	Υ	Φ	Χ	Ψ	Ω	İ	ÿ	α	ε	η	ι
ı	Ü	α	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ
ı	π	ρ	σ	τ	υ	φ	χ	ψ	ω	ı	ÿ	ö	ü	ı	ı

2.12.8. ISO8859-8 (characters 0xA0 ~ 0xFF)

A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF
ı	ı	ϕ	£	¥	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı
ı	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı
ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı
ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı
ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı
ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı
ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı
ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı

2.12.9. ISO8859-9 (characters 0xA0 ~ 0xFF)

A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF
B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF
C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF
D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF
E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF
F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF

2.12.10. ISO8859-10 (characters 0xA0 ~ 0xFF)

A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF
B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF
C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF
D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF
E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF
F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF

2.12.11. ISO8859-11 (characters 0xA0 ~ 0xFF)

A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF
B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF
C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF
D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF
E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF
F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF

2.12.12. ISO8859-13 (characters 0xA0 ~ 0xFF)

A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF
B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF
C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF
D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF
E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF
F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF

2.12.13. ISO8859-14 (characters 0xA0 ~ 0xFF)

A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF
	Ā	ā	£	Č	č	Ď	š	Ŧ	©	Ű	đ	ÿ	-	®	ÿ
B0	Ā	ā	Ġ	ġ	Ĭ	ĭ	Ŧ	Ĳ	ŵ	ŷ	š	ÿ	Ŵ	ŵ	š
C0	Ā	Ā	Ā	Ā	Ā	Ā	Æ	Ç	È	É	Ê	Ë	Ī	Ī	Ī
D0	Ŵ	Ŷ	Ÿ	Ź	Ż	Ź	Ź	Ź	Ź	Ź	Ź	Ź	Ź	Ź	Ź
E0	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î
F0	û	ü	ý	ÿ											

2.12.14. ISO8859-15 (characters 0xA0 ~ 0xFF)

A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF
	ı	ϕ	£	€	¥	Š	š	Š	©	≡	«	¬	-	®	-
B0	°	±	²	³	ž	μ	Ŧ	•	ž	ı	»	œ	œ	ÿ	ı
C0	Ā	Ā	Ā	Ā	Ā	Ā	Æ	Ç	È	É	Ê	Ë	Ī	Ī	Ī
D0	Đ	Ŷ	Ÿ	Ź	Ż	Ź	×	Ø	Ű	Ű	Ű	Ű	Ÿ	Ɔ	Ɔ
E0	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î
F0	ð	ñ	õ	ö	÷	ø	ù	ú	û	ü	ý	ÿ	Ɔ	ÿ	ÿ

2.12.15. ISO8859-16 (characters 0xA0 ~ 0xFF)

A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF
	À	á	£	€	”	Š	š	Š	©	Š	«	Ž	-	ž	ž
B0	°	±	Č	č	Ž	”	Ŧ	•	ž	č	š	»	œ	œ	ÿ
C0	Ā	Ā	Ā	Ā	Ā	Ā	Æ	Ç	È	É	Ê	Ë	Ī	Ī	Ī
D0	Đ	Ŷ	Ÿ	Ź	Ż	Ź	Š	Ű	Ű	Ű	Ű	Ű	Ÿ	Ŧ	Ɔ
E0	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î
F0	đ	ñ	õ	ö	ó	ö	š	ű	ű	ű	ű	ű	ÿ	Ŧ	ÿ

2.13. LCM 5x10

2.13.1. LCM 5x10-1

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
1	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
2	□	!	"	#	\$	%	&	'	<	>	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	a	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[¥]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	P	q	r	s	t	u	v	w	x	y	z	<		>	→	←
8	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
9	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
A	□	。	「	」	、	・	ヲ	ア	イ	ウ	エ	オ	カ	ユ	ヨ	ツ
B	一	ア	イ	ウ	エ	オ	カ	キ	ク	ケ	コ	サ	シ	ス	セ	ソ
C	タ	チ	ツ	テ	ト	ナ	ニ	ヌ	ネ	ノ	ハ	ヒ	フ	ヘ	ホ	マ
D	ミ	ム	メ	モ	ヤ	ユ	ヨ	ラ	リ	ル	レ	ロ	ワ	ヅ	ヅ	□
E	α	ä	ß	ε	μ	σ	ρ	g	∫	"	j	*	φ	も	ñ	ö
F	p	q	θ	ω	Ω	Ü	Σ	π	∞	γ	¥	₤	₹	÷	□	■

2.13.2. LCM 5x10-2

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
1	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
2	□	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	a	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[]	^	_	
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	P	q	r	s	t	u	v	w	x	y	z	10	12	15	e	e
8	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
9	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
A	Б	Г	Ё	Ж	З	И	Й	Л	П	У	Ф	Ч	Ш	Ъ	Ы	Э
B	Ю	Я	б	в	г	ё	ж	з	и	й	к	л	м	н	п	т
C	ч	ш	ъ	ы	ь	э	ю	я	«	»	„	”	№	¿	f	e
D	.	ı	ıı	ııı	ıııı	ııııı	ıııııı	ııııııı	ıııııııı	ııııııııı	ıııııııııı	ııııııııııı	ıııııııııııı	ııııııııııııı	ıııııııııııııı	ııııııııııııııı
E	Д	Ц	Щ	д	ф	ц	щ	’	”	~	ё	ф	ü	•	ˆ	o
F	¼	½	¾	¼	½	¾	¼	½	¾	¼	½	¾	¼	½	¾	¼

2.13.3. LCM 5x10-3

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
1	±	≡	∇	△	∫	∫	∫	∫	∫	∫	∫	∫	∫	∫	∫	∫
2	□	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	a	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6	°	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	Δ
8	ā	ā	ā	ā	ā	ā	ā	ā	ā	ā	ā	ā	ā	ā	ā	ā
9	ē	æ	Æ	ø	ö	ò	á	á	á	ü	ü	ÿ	ÿ	ÿ	ÿ	ÿ
A	š	í	ó	ú	č	ř	ř	ř	ř	ř	ř	ř	ř	ř	ř	ř
B	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ
C	Γ	J	∞	∇	∫	∫	∫	∫	∫	∫	∫	∫	∫	∫	∫	∫
D	∫	∫	∫	∫	∫	∫	∫	∫	∫	∫	∫	∫	∫	∫	∫	∫
E	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	π	ρ	σ
F	τ	υ	χ	ψ	ω	φ	φ	φ	φ	φ	φ	φ	φ	φ	φ	φ

2.13.4. LCM 5x10-8

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
1	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
2	□	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	a	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[]	^	_	
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	}	~		
8	ü	ö	ó	ò	ô	õ	ë	é	ê	ë	÷	ç	ç	ø	ø	ø
9	ø	ø	ü	ü	я	я	н	в	у	¥	!	«	»	»	»	»
A	f	←	→	√	%	¿	¡	£	£	¢	°	÷	±	·	¶	§
B	†	‡	´	¨	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı
C	á	á	ë	è	é	é	ı	ı	ı	ı	ö	ı	ı	ı	ı	ı
D	ö	ö	ö	ö	ü	ü	ü	ü	ü	ı	ı	ı	ı	ı	ı	ı
E	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı
F	p	q	y	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı

2.13.5. LCM 5x10-11

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
1	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
2	□	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	a	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[¥]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	P	q	r	s	t	u	v	w	x	y	z	{		}	+	←
8	o	ó	ú	ü	á	â	ã	ä	å	ö	ø	ù	û	ü	ö	ç
9	é	ê	ë	ë	é	è	é	ë	ä	ä	ä	ä	ä	ä	ä	ä
A	ë	ï	í	í	ï	í	ñ	ñ	ñ	ñ	ñ	ñ	ñ	ñ	ñ	ñ
B	ë	±	ü	↑	↓	←	↵	↵	↵	↵	↵	↵	↵	↵	↵	↵
C	ø	L	Ø	B	G	Ø	Ø	U	Ø	U	Ø	Ø	Ø	Ø	Ø	Ø
D	·	¨	°	´	˘	÷	×	×	1	√	×	Ø	Ø	Ø	Ø	Ø
E	α	¼	½	¾	¾	¾	ρ	g	†	‡	j	□	κ	Я	Ю	Ш
F	P	q	Ø	∞	Ω	Ρ	Σ	∩	Φ	Υ	Β	Π	Ψ	Ц	Ъ	■

2.13.6. LCM 5x10-12

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
1	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
2	□	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	a	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	P	q	r	s	t	u	v	w	x	y	z	{		}	~	
8	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
9	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
A	□	i														
B																
C																
D																
E																
F																

2.13.7. LCM 5x10-13

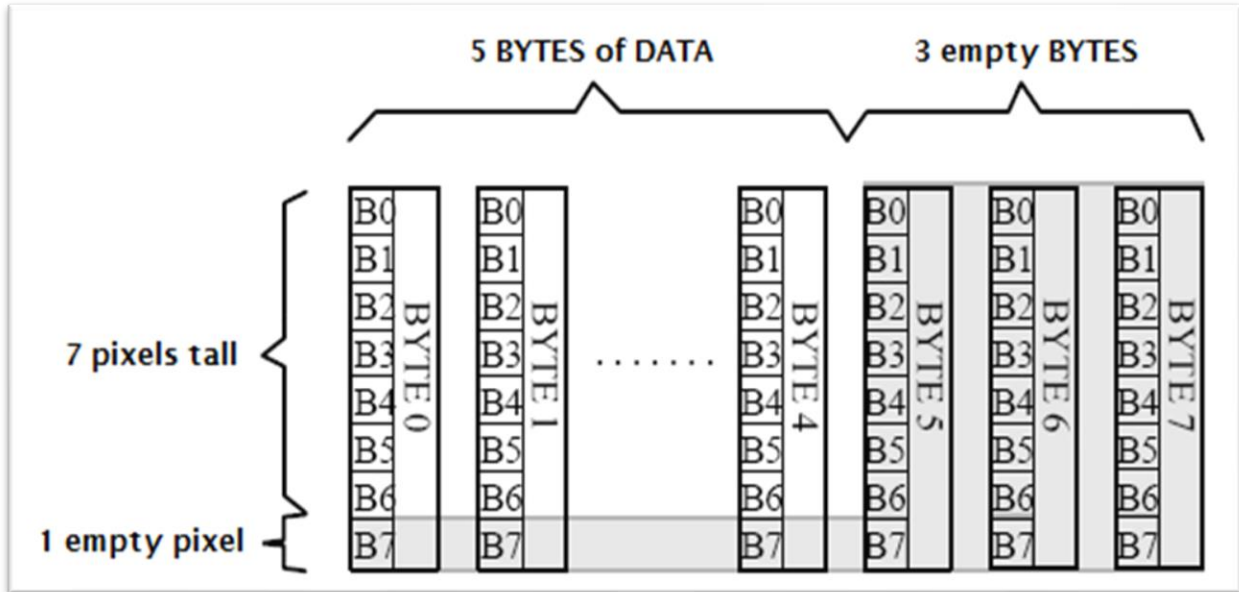
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
1	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
2	□	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	a	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[¥]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	P	q	r	s	t	u	v	w	x	y	z	{		}	→	←
8	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
9	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
A	ア	イ	ウ	エ	オ	カ	キ	ク	ケ	コ	サ	シ	ス	セ	ソ	タ
B	チ	リ	フ	ル	レ	ロ	ワ	ヰ	ヱ	ヰ	ヱ	ヰ	ヱ	ヰ	ヱ	ヰ
C	夕	チ	ツ	テ	ト	ナ	ニ	ヌ	ネ	ノ	ハ	ヒ	フ	ヘ	ホ	マ
D	ミ	ム	メ	モ	ヤ	ユ	ヨ	ラ	リ	ル	レ	ロ	ワ	ヰ	ヱ	ヰ
E	α	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο	π
F	ρ	σ	τ	υ	φ	χ	ψ	ω	⊗	⊙	⊚	⊛	⊜	⊝	⊞	■

3. Font Data Arrangement Format

Each font character is stored in dot matrix format. Each dot is expressed by a binary bit; 1 represents an 'ON' pixel, 0 represents an 'OFF' pixel. The data arrangement format is byte-vertical, string-horizontal.

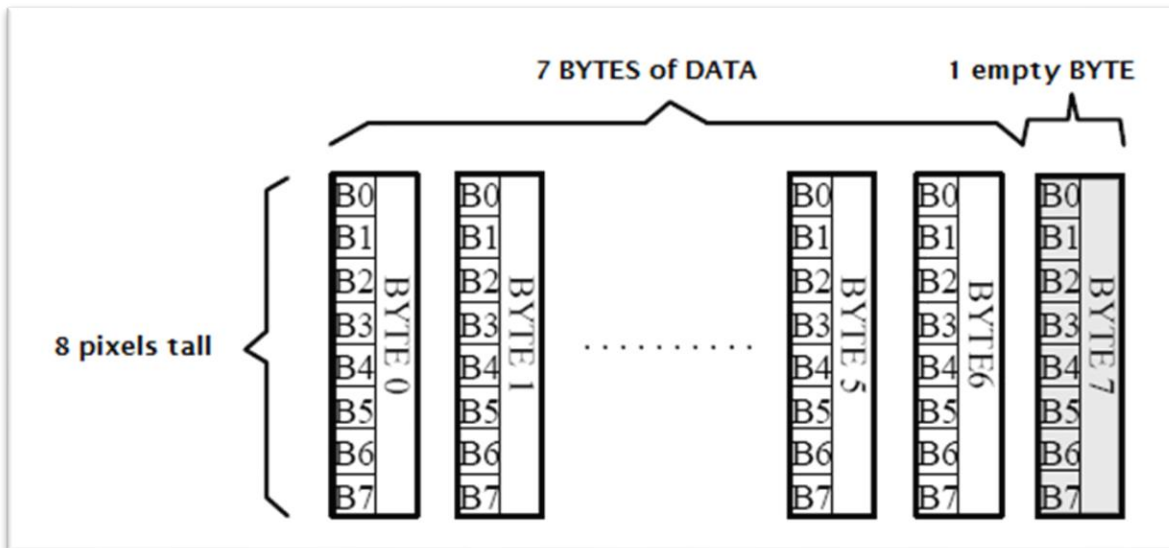
3.1. 5x7 Font Data Arrangement

5x7 dot fonts require 8 bytes (BYTE 0 – BYTE 7) to display. BYTE 0 represents the left most column of the font. BYTE 5 through BYTE 7 are empty data (0x00).



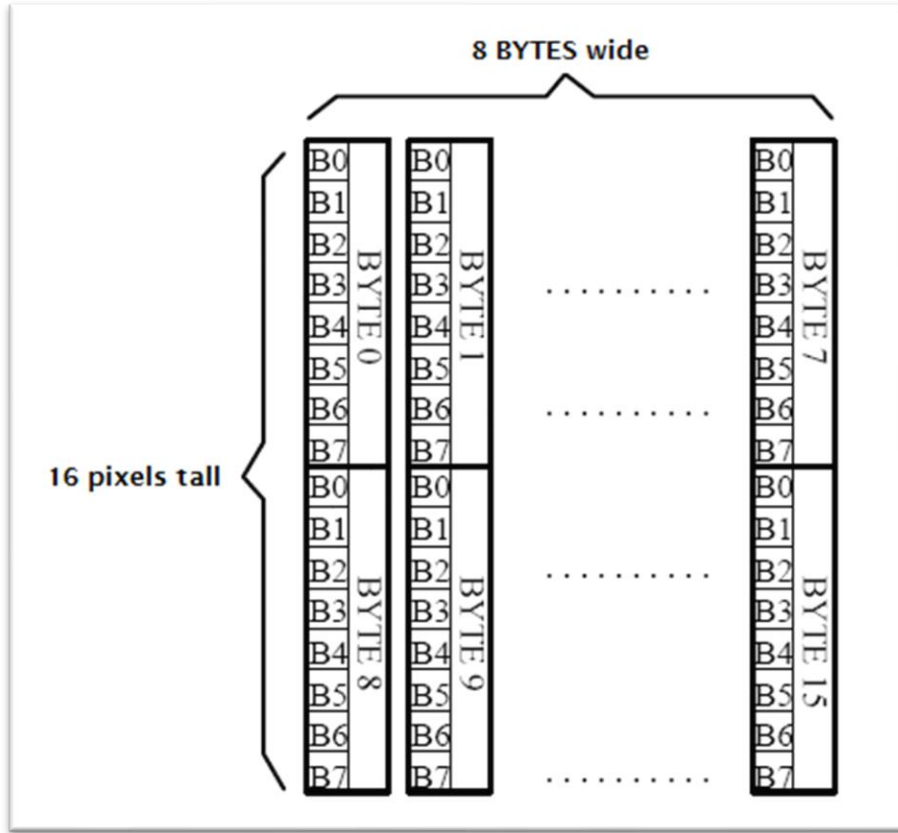
3.2. 7x8 Font Data Arrangement

7x8 dot fonts require 8 bytes (BYTE 0 – BYTE 7) to display. BYTE 0 represents the left most column of the font. BYTE 7 is empty data (0x00).



3.3. 8x16 Font Data Arrangement

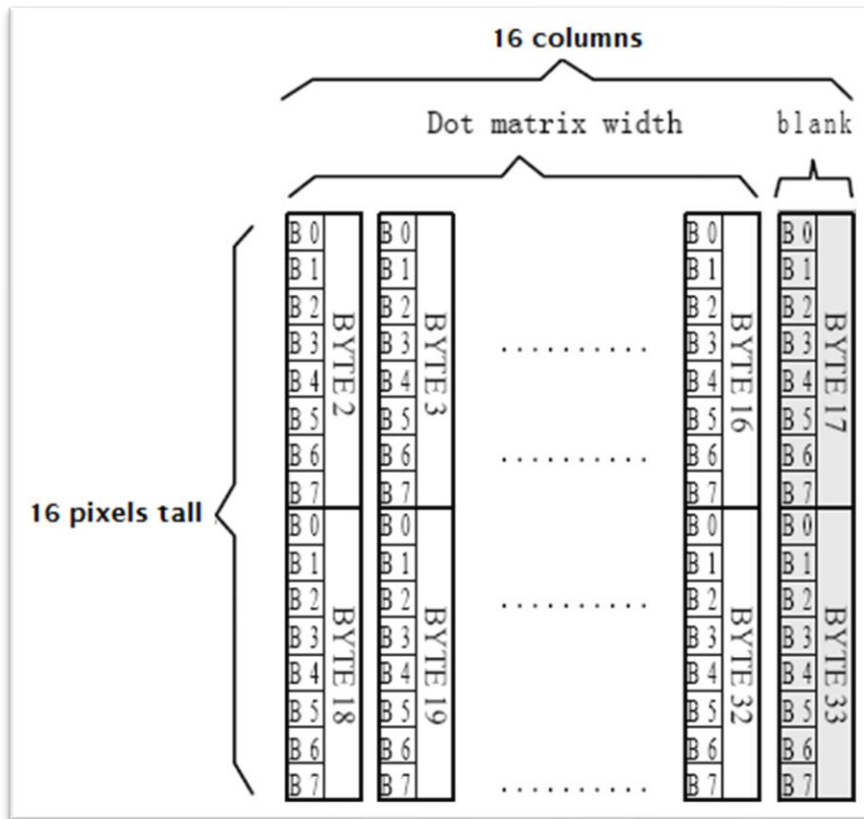
8x16 dot fonts require 16 bytes (BYTE 0 – BYTE 15) to display. BYTE 0 represents the top-left most column of the font. BYTE 0 through BYTE 7 represent the top half of the font. BYTE 8 represents the bottom-left most column of the font. BYTE 8 through BYTE 15 represent the bottom half of the font.



3.4. Width Adjusted Font Data Arrangement

Width-Adjusted dot fonts require 34 bytes (BYTE 0 – BYTE 33) to display. Because each font is proportionally adjusted, BYTE 0 through BYTE 1 represents the width of the font. BYTE 2 - 33 represent the dot matrix font data. BYTE 2 through BYTE 16 have the lower 3 bits empty, BYTE 18 through 32 have the highest bit empty, giving each character a maximum height of 12 pixels. BYTE 17 and BYTE 33 are always empty to allow one pixel space between characters (0x00).





The font width in BYTE 0~BYTE 1 can be used as reference for the position of the next displayed character. For example: ASCII Arial Font “B” reads BYTES 0 ~33:

(hex)

00 0C

00 F8 F8 18 18 18 18 18 F8 F0 00 00 00 00 00 00

00 7F 7F 63 63 63 63 63 67 3E 1C 00 00 00 00 00

BYTE0~BYTE1 = 0000C. This means the character “B” is 12 pixels wide, with 4 empty columns at the end.

BYTE2~BYTE33 is the dot matrix data:

	12 bytes wide												4 blank bytes						
BYTES 2~17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
BYTES 18~33	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	
	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	
	0	1	1	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	
	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	
	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	
	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

3.5. CJK Font Data Arrangement

Chinese, Japanese, and Korean fonts are 15x16 dots, each font require 32 bytes (BYTE 0 – BYTE 31) to display.

BYTE 0 represents the top-left most column of the font.

BYTE 0 through BYTE 7 represent the top-left quarter of the font.

BYTE 8 through BYTE 15 represent the top-right quarter of the font.

BYTE 16 through BYTE 23 represent the bottom-left quarter of the font.

BYTE 24 through BYTE 31 represent the bottom-right quarter of the font.

For example: Chinese Font “美” (Unicode U+7F8E , or GB2312 font C3C0) reads BYTES 0 ~31:

(hex)

00 04 24 24 25 26 24 FC 24 26 25 24 24 04 00 00

81 89 89 49 49 29 19 0F 19 29 49 49 89 89 81 00

BYTES 0~15	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0
	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0
	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
BYTES 16~31	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0
	0	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0
	1	1	1	0	0	0	0	0	0	0	0	0	1	1	1	1	0

4. Calculation of Font Addresses

4.1. 5x7 ASCII

Parameters:

ASCII CODE: 8bit ASCII character code

Address: Address of character data

```
if(ASCII CODE >= 0x20 && ASCII CODE <= 0xFF)
```

```
    Address = (ASCII CODE - 0x20) * 8;
```

4.2. 7x8 ASCII

Parameters:

ASCII CODE: 8bit ASCII character code

Address: Address of character data

```
if(ASCII CODE >= 0x20 && ASCII CODE <= 0xFF)
```

```
    Address = ((ASCII CODE - 0x20) * 8) + 768;
```

4.3. 8x16 ASCII

Parameters:

ASCII CODE: 8bit ASCII character code

Address: Address of character data

```
if(ASCII CODE >= 0x20 && ASCII CODE <= 0xFF)
```

```
    Address = (ASCII CODE - 0x20) * 16 + 1,536;
```

4.4. Width-Adjusted Arial ASCII

Parameters:

ASCII CODE: 8bit ASCII character code

Address: Address of character data

```
if(ASCII CODE >= 0x20 && ASCII CODE <= 0xFF)
```

```
    Address = (ASCII CODE - 0x20) * 34 + 3,072;
```

4.5. 8x16 Latin

Parameters:

UNICODE: 16-bit Unicode

Address: Address of character data

```
if(UNICODE >= 0x0020 && UNICODE <= 0x007F)
```

```
    Address = ((UNICODE - 0x0020) * 16) + 6,336;
```

```
else if(UNICODE >= 0x00A0 && UNICODE <= 0x017F)
```

```
    Address = ((UNICODE - 0x00A0 + 96) * 16) + 6,336;
```

```
else if(UNICODE >= 0x01A0 && UNICODE <= 0x01CF)
```

```
    Address = ((UNICODE - 0x01A0 + 320) * 16) + 6,336;
```

```
else if(UNICODE >= 0x01F0 && UNICODE <= 0x01FF)
```

```
    Address = ((UNICODE - 0x01F0 + 368) * 16) + 6,336;
```

```
else if(UNICODE >= 0x0210 && UNICODE <= 0x021F)
```

```
    Address = ((UNICODE - 0x0210 + 384) * 16) + 6,336;
```

```
else if(UNICODE >= 0x1EA0 && UNICODE <= 0x1EFF)
```

```
    Address = ((UNICODE - 0x1EA0 + 400) * 16) + 6,336;
```

4.6. 8x16 Greek

Parameters:

UNICODE: 16-bit Unicode

Address: Address of character data

```
if(UNICODE >= 0x0370 && UNICODE <= 0x03CF)
    Address = ((UNICODE - 0x0370)*16) + 14,272;
```

4.7. 8x16 Cyrillic

Parameters:

UNICODE: 16-bit Unicode

Address: Address of character data

```
if(UNICODE >= 0x0400 && UNICODE <= 0x045F)
    Address = ((UNICODE - 0x0400)*16) + 15,808;
else if(UNICODE >= 0x0490 && UNICODE <= 0x04FF)
    Address = ((UNICODE - 0x0490 + 96)*16) + 15,808;
```

4.8. 8x16 Hebrew

Parameters:

UNICODE: 16-bit Unicode

Address: Address of character data

```
if(UNICODE >= 0x0590 && UNICODE <= 0x05FF)
    Address = ((UNICODE - 0x0590)*16) + 19,136;
```

4.9. 8x16 Thai

Parameters:

UNICODE: 16-bit Unicode

Address: Address of character data

```
if(UNICODE >= 0x0E00 && UNICODE <= 0x0E5F)
    Address = ((UNICODE - 0x0E00)*16) + 20,928;
```

4.10. Width-Adjusted Latin

Parameters:

UNICODE: 16-bit Unicode

Address: Address of character data

```
if(UNICODE >= 0x0020 && UNICODE <= 0x007F)
    Address = ((UNICODE - 0x0020)*34) + 22,976;
else if(UNICODE >= 0x00A0 && UNICODE <= 0x017F)
    Address = ((UNICODE - 0x00A0 + 96)*34) + 22,976;
else if(UNICODE >= 0x01A0 && UNICODE <= 0x01CF)
    Address = ((UNICODE - 0x01A0 + 320)*34) + 22,976;
else if(UNICODE >= 0x01F0 && UNICODE <= 0x01FF)
    Address = ((UNICODE - 0x01F0 + 368)*34) + 22,976;
else if(UNICODE >= 0x0210 && UNICODE <= 0x021F)
    Address = ((UNICODE - 0x0210 + 384)*34) + 22,976;
else if(UNICODE >= 0x1EA0 && UNICODE <= 0x1EFF)
    Address = ((UNICODE - 0x1EA0 + 400)*34) + 22,976;
```


4.11. Width-Adjusted Greek

Parameters:

UNICODE: 16-bit Unicode

Address: Address of character data

```
if(UNICODE >= 0x0370 && UNICODE <= 0x03CF)
    Address = ((UNICODE - 0x0370)*34) + 39,840;
```

4.12. Width-Adjusted Cyrillic

Parameters:

UNICODE: 16-bit Unicode

Address: Address of character data

```
if(UNICODE >= 0x0400 && UNICODE <= 0x045F)
    Address = ((UNICODE - 0x0400)*34) + 43,104;
else if(UNICODE >= 0x0490 && UNICODE <= 0x04FF)
    Address = ((UNICODE - 0x0490 + 96)*34) + 43,104;
```

4.13. Width-Adjusted Arabic

Parameters:

UNICODE: 16-bit Unicode

Address: Address of character data

```
if(UNICODE >= 0x0600 && UNICODE <= 0x06FF)
    Address = ((UNICODE - 0x0600)*34) + 50,176;
else if(UNICODE >= 0xFB50 && UNICODE <= 0xFBFF)
    Address = ((UNICODE - 0xFB50 + 256) *34) + 50,176;
else if(UNICODE >= 0xFE70 && UNICODE <= 0xFEFF)
    Address = ((UNICODE - 0xFE70 + 432) *34) + 50,176;
```

4.14. GB2312 Simplified Chinese

Parameters:

GBCode: 16-bit GB2312 character code

MSB: Higher byte of GB code

LSB: Lower byte of GB code

Address: Address of character data

```
if(MSB >= 0xA1 && MSB <= 0xA9 && LSB >=0xA1)
    Address = (((MSB - 0xA1) * 94) + (LSB - 0xA1)) * 32 + 69,760;
else if(MSB >= 0xB0 && MSB <= 0xF7 && LSB >= 0xA1)
    Address = (((MSB - 0xB0) * 94) + (LSB - 0xA1) + 846) * 32 + 69,760;
```

4.15. KSC5601 Korean

Parameters:

GBCode: 16-bit GB2312 character code

MSB: Higher byte of GB code

LSB: Lower byte of GB code

Address: Address of character data

```
if(MSB >= 0xA1 && MSB <= 0xB0 && LSB >=0xA1)
    Address = (((MSB - 0xA1) * 94) + (LSB - 0xA1)) * 32 + 379,744;
else if(MSB >= 0xB0 && MSB <= 0xC8 && LSB >= 0xA1)
    Address = (((MSB - 0xB0) * 94) + (LSB - 0xA1)) * 32 + 379,744 + 35,680;
```

4.16. JIS0208 Japanese

Parameters:

GBCode: 16-bit GB2312 character code

MSB: Higher byte of GB code

LSB: Lower byte of GB code

Address: Address of character data

```
if((MSB >= 1 && MSB <= 94) && (LSB >= 1 && LSB <= 94 )  
    Address = (((MSB - 1) * 94) + (LSB - 1)) * 32 + 490,624;
```

4.17. ISO8859-1 (for characters 0x20~0x7F, reference 5x7 ASCII calculation)

Parameters:

UNICODE: 16-bit Unicode

Address: Address of character data

```
if(UNICODE >= 0x0080 && UNICODE <= 0x00FF)  
    Address =( (UNICODE - 0x80) *8) + 946,992;
```

4.18. ISO8859-2 (for characters 0x20~0x7F, reference 5x7 ASCII calculation)

Parameters:

UNICODE: 16-bit Unicode

Address: Address of character data

```
if(UNICODE >= 0x0080 && UNICODE <= 0x00FF)  
    Address =( (UNICODE - 0x80) *8) + 946,992 + 1,024;
```

4.19. ISO8859-3 (for characters 0x20~0x7F, reference 5x7 ASCII calculation)

Parameters:

UNICODE: 16-bit Unicode

Address: Address of character data

```
if(UNICODE >= 0x0080 && UNICODE <= 0x00FF)  
    Address =( (UNICODE - 0x80) *8) + 946,992 + (1,024*2);
```

4.20. ISO8859-4 (for characters 0x20~0x7F, reference 5x7 ASCII calculation)

Parameters:

UNICODE: 16-bit Unicode

Address: Address of character data

```
if(UNICODE >= 0x0080 && UNICODE <= 0x00FF)  
    Address =( (UNICODE - 0x80) *8) + 946,992 + (1,024*3);
```

4.21. ISO8859-5 (for characters 0x20~0x7F, reference 5x7 ASCII calculation)

Parameters:

1.1. UNICODE: 16-bit Unicode

Address: Address of character data

```
if(UNICODE >= 0x0080 && UNICODE <= 0x00FF)  
    Address =( (UNICODE - 0x80) *8) + 946,992 + (1,024*4);
```

4.22. ISO8859-7 (for characters 0x20~0x7F, reference 5x7 ASCII calculation)

Parameters:

UNICODE: 16-bit Unicode

Address: Address of character data

```
if(UNICODE >= 0x0080 && UNICODE <= 0x00FF)
    Address = ( (UNICODE - 0x80) *8) + 946,992 + (1,024*5);
```

4.23. ISO8859-8 (for characters 0x20~0x7F, reference 5x7 ASCII calculation)

Parameters:

UNICODE: 16-bit Unicode

Address: Address of character data

```
if(UNICODE >= 0x0080 && UNICODE <= 0x00FF)
    Address = ( (UNICODE - 0x80) *8) + 946,992 + (1,024*6);
```

4.24. ISO8859-9 (for characters 0x20~0x7F, reference 5x7 ASCII calculation)

Parameters:

UNICODE: 16-bit Unicode

Address: Address of character data

```
if(UNICODE >= 0x0080 && UNICODE <= 0x00FF)
    Address = ( (UNICODE - 0x80) *8) + 946,992 + (1,024*7);
```

4.25. ISO8859-10 (for characters 0x20~0x7F, reference 5x7 ASCII calculation)

Parameters:

UNICODE: 16-bit Unicode

Address: Address of character data

```
if(UNICODE >= 0x0080 && UNICODE <= 0x00FF)
    Address = ( (UNICODE - 0x80) *8) + 946,992 + (1,024*8);
```

4.26. ISO8859-11 (for characters 0x20~0x7F, reference 5x7 ASCII calculation)

Parameters:

UNICODE: 16-bit Unicode

Address: Address of character data

```
if(UNICODE >= 0x0080 && UNICODE <= 0x00FF)
    Address = ( (UNICODE - 0x80) *8) + 946,992 + (1,024*9);
```

4.27. ISO8859-13 (for characters 0x20~0x7F, reference 5x7 ASCII calculation)

Parameters:

UNICODE: 16-bit Unicode

Address: Address of character data

```
if(UNICODE >= 0x0080 && UNICODE <= 0x00FF)
    Address = ( (UNICODE - 0x80) *8) + 946,992 + (1,024*10);
```

4.28. ISO8859-14 (for characters 0x20~0x7F, reference 5x7 ASCII calculation)

Parameters:

UNICODE: 16-bit Unicode

Address: Address of character data

```
if(UNICODE >= 0x0080 && UNICODE <= 0x00FF)
    Address = ( (UNICODE - 0x80) *8) + 946,992 + (1,024*11);
```

4.29. ISO8859-15 (for characters 0x20~0x7F, reference 5x7 ASCII calculation)

Parameters:

UNICODE: 16-bit Unicode

Address: Address of character data

```
if(UNICODE >= 0x0080 && UNICODE <= 0x00FF)
    Address = ( (UNICODE - 0x80) *8) + 946,992 + (1,024*12);
```

4.30. ISO8859-16 (for characters 0x20~0x7F, reference 5x7 ASCII calculation)

Parameters:

UNICODE: 16-bit Unicode

Address: Address of character data

```
if(UNICODE >= 0x0080 && UNICODE <= 0x00FF)
    Address = ( (UNICODE - 0x80) *8) + 946,992 + (1,024*13);
```

4.31. 5x10 LCM - 1

Parameters:

UNICODE: 16-bit Unicode

Address: Address of character data

```
if(UNICODE >= 0x0000 && UNICODE <= 0x00FF)
    Address = (UNICODE *10) + 961,328;
```

4.32. 5x10 LCM - 2

Parameters:

UNICODE: 16-bit Unicode

Address: Address of character data

```
if(UNICODE >= 0x0000 && UNICODE <= 0x00FF)
    Address = (UNICODE *10) + 961,328 + 2,560;
```

4.33. 5x10 LCM - 3

Parameters:

UNICODE: 16-bit Unicode

Address: Address of character data

```
if(UNICODE >= 0x0000 && UNICODE <= 0x00FF)
    Address = (UNICODE *10) + 961,328 + (2,560*2);
```

4.34. 5x10 LCM - 8

Parameters:

UNICODE: 16-bit Unicode

Address: Address of character data

```
if(UNICODE >= 0x0000 && UNICODE <= 0x00FF)
    Address = (UNICODE *10) + 961,328 + (2,560*3);
```

4.35. 5x10 LCM - 11

Parameters:

UNICODE: 16-bit Unicode

Address: Address of character data

```
if(UNICODE >= 0x0000 && UNICODE <= 0x00FF)
    Address = (UNICODE *10) + 961,328 + (2,560*4);
```

4.36. 5x10 LCM - 12

Parameters:

UNICODE: 16-bit Unicode

Address: Address of character data

```
if(UNICODE >= 0x0000 && UNICODE <= 0x00FF)
    Address = (UNICODE *10) + 961,328 + (2,560*5);
```

4.37. 5x10 LCM - 13

Parameters:

UNICODE: 16-bit Unicode

Address: Address of character data

```
if(UNICODE >= 0x0000 && UNICODE <= 0x00FF)
    Address = (UNICODE *10) + 961,328 + (2,560*6);
```

5. How to use the Multi-Font Displays

The Multi-font displays have a built-in serial interface memory IC with preloaded font tables. These font tables contain the dot-matrix (pixel) data that makes up each individual character in the supported languages.

The multi-font IC communicates with the main host MPU by receiving the specified font address and then sending the dot-matrix data back to the main host MPU. When the data is received by the main host MPU, it can then be written to the Display Data RAM.

The following steps with examples are used to get and display the multi-font data:

1) Determine what characters or strings should be shown on the display.

example: **“Newhaven Display”**

2) Determine what languages or fonts this string should be shown in.

example: Chinese = “**纽黑文显示**”

3) Go to Section 4 and determine what character code is required for the address calculation.

example: Chinese requires GB code

4) Save the strings into program code.

example:

```
unsigned int NameChinese[] = {0xC5A6, 0xBADA, 0xCEC4, 0xCFD4, 0xCABE};        //GB code for 纽黑文显示
```

5) Send string data to address calculation routine in Section 4.

example: get the multi-font address for 0xC5A6 Chinese character:

MSB = 0xC5

LSB = 0xA6

since (MSB >= 0xB0 && MSB <= 0xF7 && LSB >= 0xA1) then:

Address = ((0xC5 – 0xB0)*94 + (0xA6 – 0xA1) + 846)*32 + 69,760

Address = ((0x15)*94 + 0x05 + 846)*32 + 69,760

Address = (1,974 + 5 + 846)*32 + 69,760

Address = 160,160

Address = 0x0271A0

6) Send READ command and Address to Multi-Font IC.

example:

```
SPI_OUT(0x0B);           //READ command
SPI_OUT(0x02);           //Address Byte1 (MSB)
SPI_OUT(0x71);           //Address Byte2
SPI_OUT(0xA0);           //Address Byte3 (LSB)
SPI_OUT(0xFF);           //Dummy Byte
```

7) Read dot-matrix font data and store in buffer or send to display.

example: Chinese fonts are 32 bytes of data

```
for(i = 1 ; i <= 32 ; i++){
    *readByte = SPI_IN();           //read one byte
    FontBuffer[1][i] = readByte;    //save byte in buffer
}

OLED_12864_SetAddress(column , row); //set address location for font to be displayed

for(i = 1 ; i <= 16 ; i++){        //for the first 16 bytes of font data
    OLED_12864_Output(FontBuffer[1][i]); //write byte to display
    column++;                        //next byte will be for the next column
    OLED_12864_SetAddress(column , row); //set address location for next column
}

column = column - 16;              //go back to first column of the font
row = row + 16;                    //move down to show the bottom half of font
OLED_12864_SetAddress(column , row); //set address for bottom half of font

for(i = 17 ; i <= 32 ; i++){      //for the second 16 bytes of font data
    OLED_12864_Output(FontBuffer[1][i]); //write byte to display
    column++;                        //next byte will be for the next column
    OLED_12864_SetAddress(column , row); //set address location for next column
}
```