Multi-Font Displays

Font Tables and Application Guide

Newhaven Display International, Inc.
2511 Technology Drive, Suite 101
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

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		¡	¢	£	¤	¥	¦	§	¨	©	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	0	Ā	ā	Ă	ă	Ą	ą	Ć	ć	Ĉ	ĉ	Ċ	ċ	Č	č	Ď	ď
	1	Đ	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	Ĭ	ĭ	Į	į
	3	İ	ı	Ĳ	ĳ	Ĵ	ĵ	Ķ	ķ	ĸ	Ĺ	ĺ	Ļ	ļ	Ľ	ľ	Ŀ
	4	ŀ	Ł	ł	Ń	ń	Ņ	ņ	Ň	ň	ŉ	Ŋ	ŋ	Ō	ō	Ŏ	ŏ
	5	Ő	ő	Œ	œ	Ŕ	ŕ	Ŗ	ŗ	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	Ű	ű	Ų	ų	Ŵ	ŵ	Ŷ	ŷ	Ÿ	Ź	ź	Ż	ż	Ž	ž	ſ

2.5. Latin Extended B

		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Latin Extended B	0	Ơ	ơ	Ƣ	ƣ	Ƥ	ƥ	Ʀ	Ƨ	ƨ	Ʃ	ƪ	ƫ	Ƭ	ƭ	Ʈ	Ư
	1	ư	Ʊ	Ʋ	Ƴ	ƴ	Ƶ	ƶ	Ʒ	Ƹ	ƹ	ƺ	ƻ	Ƽ	ƽ	ƾ	ƿ
	2	ǀ	ǁ	ǂ	ǃ	Ǆ	ǅ	ǆ	Ǉ	ǈ	ǉ	Ǌ	ǋ	ǌ	Ǎ	ǎ	Ǐ
	3	ǰ	Ǳ	ǲ	ǳ	Ǵ	ǵ	Ƕ	Ƿ	Ǹ	ǹ	Ǻ	ǻ	Ǽ	ǽ	Ǿ	ǿ
	4	Ȑ	ȑ	Ȓ	ȓ	Ȕ	ȕ	Ȗ	ȗ	Ș	ș	Ț	ț	Ȝ	ȝ	Ȟ	ȟ

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**	Ạ	ạ	Ả	ả	Ấ	ấ	Ầ	ầ	Ẩ	ẩ	Ẫ	ẫ	Ậ	ậ	Ắ	ắ
	1	Ằ	ằ	Ẳ	ẳ	Ẵ	ẵ	Ặ	ặ	Ẹ	ẹ	Ẻ	ẻ	Ẽ	ẽ	Ế	ế
	2	Ề	ề	Ể	ể	Ễ	ễ	Ệ	ệ	Ỉ	ỉ	Ị	ị	Ọ	ọ	Ỏ	ỏ
	3	Ố	ố	Ồ	ồ	Ổ	ổ	Ỗ	ỗ	Ộ	ộ	Ớ	ớ	Ờ	ờ	Ở	ở
	4	Ỡ	ỡ	Ợ	ợ	Ụ	ụ	Ủ	ủ	Ứ	ứ	Ừ	ừ	Ử	ử	Ữ	ữ
	5	Ự	ự	Ỳ	ỳ	Ỵ	ỵ	Ỷ	ỷ	Ỹ	ỹ						

2.7. Greek

		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Greek	**0**					ʹ	͵					ͺ	ͻ	ͼ	ͽ	;	
	1					΄	΅	Ά	·	Έ	Ή	Ί		Ό		Ύ	Ώ
	2	ΐ	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	ΰ	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**	Ѐ	Ё	Ђ	Ѓ	Є	Ѕ	І	Ї	Ј	Љ	Њ	Ћ	Ќ	Ѝ	Ў	Џ
	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	ѐ	ё	ђ	ѓ	є	ѕ	і	ї	ј	љ	њ	ћ	ќ	ѝ	ў	џ
	6	Ґ	ґ	Ғ	ғ	Ҕ	ҕ	Җ	җ	Ҙ	ҙ	Қ	қ	Ҝ	ҝ	Ҟ	ҟ
	7	Ҡ	ҡ	Ң	ң	Ҥ	ҥ	Ҧ	ҧ	Ҩ	ҩ	Ҫ	ҫ	Ҭ	ҭ	Ү	ү
	8	Ұ	ұ	Ҳ	ҳ	Ҵ	ҵ	Ҷ	ҷ	Ҹ	ҹ	Һ	һ	Ҽ	ҽ	Ҿ	ҿ
	9	Ӏ	Ӂ	ӂ	Ӄ	ӄ	Ӆ	ӆ	Ӈ	ӈ	Ӊ	ӊ	Ӌ	ӌ	Ӎ	ӎ	ӏ
	A	Ӑ	ӑ	Ӓ	ӓ	Ӕ	ӕ	Ӗ	ӗ	Ә	ә	Ӛ	ӛ	Ӝ	ӝ	Ӟ	ӟ
	B	Ӡ	ӡ	Ӣ	ӣ	Ӥ	ӥ	Ӧ	ӧ	Ө	ө	Ӫ	ӫ	Ӭ	ӭ	Ӯ	ӯ
	C	Ӱ	ӱ	Ӳ	ӳ	Ӵ	ӵ	Ӷ	ӷ	Ӹ	ӹ	Ӻ	ӻ	Ӽ	ӽ	Ӿ	ӿ

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 ס for reference)	**0**		ס֑	ס֒	ס֓	ס֔	ס֕	ס֖	ס֗	ס֘	ס֙	ס֚	ס֛	ס֜	ס֝	ס֞	ס֟
	1	ס֠	ס֡	ס֢	ס֣	ס֤	ס֥	ס֦	ס֧	ס֨	ס֩	ס֪	ס֫	ס֬	ס֭	ס֮	ס֯
	2	סְ	סֱ	סֲ	סֳ	סִ	סֵ	סֶ	סַ	סָ	סֹ	וֺ	סֻ	סּ	סֽ	־	סֿ
	3	׀	◌ׁס	◌ׂס	׃	סׄ	סׅ	׆	סׇ								
	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	װ	ױ	ײ	׳	״											

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	ฆ	ง	จ	ฉ	ช	ซ	ฌ	ญ	ฎ	ฏ
	1	ฐ	ฑ	ฒ	ณ	ด	ต	ถ	ท	ธ	น	บ	ป	ผ	ฝ	พ	ฟ
	2	ภ	ม	ย	ร	ฤ	ล	ฦ	ว	ศ	ษ	ส	ห	ฬ	อ	ฮ	ฯ
	3	ะ	ก◌ั	า	◌ำ	กิ	กี	กึ	กื	กุ	กู	กฺ					฿
	4	เ	แ	โ	ใ	ไ	ๅ	ๆ	ก็	ก่	ก้	ก๊	ก๋	ก์	กํ	ก๎	๏
	5	๐	๑	๒	๓	๔	๕	๖	๗	๘	๙	๚	๛				
	6																
	7																
	2	Ġ	ġ	Ģ	ģ	Ĥ	ĥ	Ħ	ħ	Ĩ	ĩ	Ī	ī	Ĭ	ĭ	Į	į
	3	İ	ı	Ĳ	ĳ	Ĵ	ĵ	Ķ	ķ	ĸ	Ĺ	ĺ	Ļ	ļ	Ľ	ľ	Ŀ
	4	ŀ	Ł	ł	Ń	ń	Ņ	ņ	Ň	ň	ŉ	Ŋ	ŋ	Ō	ō	Ŏ	ŏ
	5	Ő	ő	Œ	œ	Ŕ	ŕ	Ŗ	ŗ	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	Ű	ű	Ų	ų	Ŵ	ŵ	Ŷ	ŷ	Ÿ	Ź	ź	Ż	ż	Ž	ž	ſ
	4	Ỡ	ỡ	Ợ	ợ	Ụ	ụ	Ủ	ủ	Ứ	ứ	Ừ	ừ	Ử	ử	Ữ	ữ
	5	Ự	ự	Ỳ	ỳ	Ỵ	ỵ	Ỷ	ỷ	Ỹ	ỹ						

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	ت	ث	ج	ح	خ	د
	3	ذ	ر	ز	س	ش	ص	ض	ط	ظ	ع	غ					
	4	ـ	ف	ق	ك	ل	م	ن	ه	و	ى	ي	حً	حٌ	حٍ	حَ	حُ
	5	حِ	حّ	حْ	ح◌ٓ	ح ◌ٔ											
	6	٠	١	٢	٣	٤	٥	٦	٧	٨	٩	٪	٫	٬	٭		
	7	حٰ	ٱ	ٲ	ٳ	ٴ	ٵ	ٶ	ٷ	ٸ	ٹ	ٺ	ٻ	ټ	ٽ	پ	ٿ
	8	ڀ	ځ	ڂ	ڃ	ڄ	څ	چ	ڇ	ڈ	ډ	ڊ	ڋ	ڌ	ڍ	ڎ	ڏ
	9	ڐ	ڑ	ڒ	ړ	ڔ	ڕ	ږ	ڗ	ژ	ڙ	ښ	ڛ	ڜ	ڝ	ڞ	ڟ
	A	ڠ	ڡ	ڢ	ڣ	ڤ	ڥ	ڦ	ڧ	ڨ	ک	ڪ	ګ	ڬ	ڭ	ڮ	گ
	B	ڰ	ڱ	ڲ	ڳ	ڴ	ڵ	ڶ	ڷ	ڸ	ڹ	ں	ڻ	ڼ	ڽ	ھ	ڿ
	C	ۀ	ہ	ۂ	ۃ	ۄ	ۅ	ۆ	ۇ	ۈ	ۉ	ۊ	ۋ	ی	ۍ	ێ	ۏ
	D	ې	ۑ	ے	ۓ	۔	ە	حۖ	حۗ	حۘ	حۙ	حۚ	حۛ	حۜ	۝	۞	ح۟
	E	ح۠	حۡ	حۢ	حۣ	حۤ	ۥ	ۦ	حۧ	حۨ	۩	ح۪	ح۫	ح۬	حۭ		
	F	۰	۱	۲	۳	۴	۵	۶	۷	۸	۹						
	10	ﭐ	ﭑ	ﭒ	ﭓ	ﭔ	ﭕ	ﭖ	ﭗ	ﭘ	ﭙ	ﭚ	ﭛ	ﭜ	ﭝ	ﭞ	ﭟ
	11	ﭠ	ﭡ	ﭢ	ﭣ	ﭤ	ﭥ	ﭦ	ﭧ	ﭨ	ﭩ	ﭪ	ﭫ	ﭬ	ﭭ	ﭮ	ﭯ
	12	ﭰ	ﭱ	ﭲ	ﭳ	ﭴ	ﭵ	ﭶ	ﭷ	ﭸ	ﭹ	ﭺ	ﭻ	ﭼ	ﭽ	ﭾ	ﭿ

	13	ﮀ	ﮁ	ﮂ	ﮃ	ﮄ	ﮅ	ﮆ	ﮇ	ﮈ	ﮉ	ﮊ	ﮋ	ﮌ	ﮍ	ﮎ	ﮏ
	14	ﮐ	ﮑ	ﮒ	ﮓ	ﮔ	ﮕ	ﮖ	ﮗ	ﮘ	ﮙ	ﮚ	ﮛ	ﮜ	ﮝ	ﮞ	ﮟ
	15	ﮠ	ﮡ	ﮢ	ﮣ	ﮤ	ﮥ	ﮦ	ﮧ	ﮨ	ﮩ	ﮪ	ﮫ	ﮬ	ﮭ	ﮮ	ﮯ
	16	ﮰ	ﮱ														
	17																
	18				ﯓ	ﯔ	ﯕ	ﯖ	ﯗ	ﯘ	ﯙ	ﯚ	ﯛ	ﯜ	ﯝ	ﯞ	ﯟ
	19	ﯠ	ﯡ	ﯢ	ﯣ	ﯤ	ﯥ	ﯦ	ﯧ	ﯨ	ﯩ	ﯪ	ﯫ	ﯬ	ﯭ	ﯮ	ﯯ
	1A	ﯰ	ﯱ	ﯲ	ﯳ	ﯴ	ﯵ	ﯶ	ﯷ	ﯸ	ﯹ	ﯺ	ﯻ	ﯼ	ﯽ	ﯾ	ﯿ
	1B	ﹰ	ﹱ			ﹴ			ﹷ	ﹸ	ﹹ	ﹺ	ﹻ	ﹼ	ﹽ	ﹾ	ﹿ
	1C	ﺀ	ﺁ	ﺂ	ﺃ	ﺄ	ﺅ	ﺆ	ﺇ	ﺈ	ﺉ	ﺊ	ﺋ	ﺌ	ﺍ	ﺎ	ﺏ
	1D	ﺐ	ﺑ	ﺒ	ﺓ	ﺔ	ﺕ	ﺖ	ﺗ	ﺘ	ﺙ	ﺚ	ﺛ	ﺜ	ﺝ	ﺞ	ﺟ
	1E	ﺠ	ﺡ	ﺢ	ﺣ	ﺤ	ﺥ	ﺦ	ﺧ	ﺨ	ﺩ	ﺪ	ﺫ	ﺬ	ﺭ	ﺮ	ﺯ
	1F	ﺰ	ﺱ	ﺲ	ﺳ	ﺴ	ﺵ	ﺶ	ﺷ	ﺸ	ﺹ	ﺺ	ﺻ	ﺼ	ﺽ	ﺾ	ﺿ
	20	ﻀ	ﻁ	ﻂ	ﻃ	ﻄ	ﻅ	ﻆ	ﻇ	ﻈ	ﻉ	ﻊ	ﻋ	ﻌ	ﻍ	ﻎ	ﻏ
	21	ﻐ	ﻑ	ﻒ	ﻓ	ﻔ	ﻕ	ﻖ	ﻗ	ﻘ	ﻙ	ﻚ	ﻛ	ﻜ	ﻝ	ﻞ	ﻟ
	22	ﻠ	ﻡ	ﻢ	ﻣ	ﻤ	ﻥ	ﻦ	ﻧ	ﻨ	ﻩ	ﻪ	ﻫ	ﻬ	ﻭ	ﻮ	ﻯ
	23	ﻰ	ﻱ	ﻲ	ﻳ	ﻴ	ﻵ	ﻶ	ﻷ	ﻸ	ﻹ	ﻺ	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 0	31 1	32 2	33 3	34 4	35 5	36 6	37 7	38 8	39 9	3A :	3B ;	3C <	3D =	3E >	3F ?
40 @	41 A	42 B	43 C	44 D	45 E	46 F	47 G	48 H	49 I	4A J	4B K	4C L	4D M	4E N	4F O
50 P	51 Q	52 R	53 S	54 T	55 U	56 V	57 W	58 X	59 Y	5A Z	5B [	5C \	5D]	5E ^	5F _
60 `	61 a	62 b	63 c	64 d	65 e	66 f	67 g	68 h	69 i	6A j	6B k	6C l	6D m	6E n	6F o
70 p	71 q	72 r	73 s	74 t	75 u	76 v	77 w	78 x	79 y	7A z	7B {	7C |	7D }	7E ~	

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

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

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

A0	A1 Ą	A2 ˘	A3 Ł	A4 ¤	A5 Ľ	A6 Ś	A7 §	A8 ¨	A9 Š	AA Ş	AB Ť	AC Ź	AD -	AE Ž	AF Ż
B0 °	B1 ą	B2 ˛	B3 ł	B4 ´	B5 ľ	B6 ś	B7 ˇ	B8 ¸	B9 š	BA ş	BB ť	BC ź	BD ˝	BE ž	BF ż
C0 Ŕ	C1 Á	C2 Â	C3 Ă	C4 Ä	C5 Ĺ	C6 Ć	C7 Ç	C8 Č	C9 É	CA Ę	CB Ë	CC Ě	CD Í	CE Î	CF Ď
D0 Đ	D1 Ń	D2 Ň	D3 Ó	D4 Ô	D5 Ő	D6 Ö	D7 ×	D8 Ř	D9 Ů	DA Ú	DB Ű	DC Ü	DD Ý	DE Ţ	DF ß
E0 ŕ	E1 á	E2 â	E3 ă	E4 ä	E5 ĺ	E6 ć	E7 ç	E8 č	E9 é	EA ę	EB ë	EC ě	ED í	EE î	EF ď
F0 đ	F1 ń	F2 ň	F3 ó	F4 ô	F5 ő	F6 ö	F7 ÷	F8 ř	F9 ů	FA ú	FB ű	FC ü	FD ý	FE ţ	FF ˙

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

A0	A1 Ħ	A2 ˘	A3 £	A4 ¤		A6 Ĥ	A7 §	A8 ¨	A9 İ	AA Ş	AB Ğ	AC Ĵ	AD -		AF Ż
B0 °	B1 ħ	B2 ²	B3 ³	B4 ´	B5 µ	B6 ĥ	B7 ·	B8 ¸	B9 ı	BA ş	BB ğ	BC ĵ	BD ½		BF ż
C0 À	C1 Á	C2 Â		C4 Ä	C5 Ċ	C6 Ĉ	C7 Ç	C8 È	C9 É	CA Ê	CB Ë	CC Ì	CD Í	CE Î	CF Ï
	D1 Ñ	D2 Ò	D3 Ó	D4 Ô	D5 Ġ	D6 Ö	D7 ×	D8 Ĝ	D9 Ù	DA Ú	DB Û	DC Ü	DD Ŭ	DE Ŝ	DF ß
E0 à	E1 á	E2 â		E4 ä	E5 ċ	E6 ĉ	E7 ç	E8 è	E9 é	EA ê	EB ë	EC ì	ED í	EE î	EF ï
	F1 ñ	F2 ò	F3 ó	F4 ô	F5 ġ	F6 ö	F7 ÷	F8 ĝ	F9 ù	FA ú	FB û	FC ü	FD ŭ	FE ŝ	FF ˙

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

A0	A1 Ą	A2 ĸ	A3 Ŗ	A4 ¤	A5 Ĩ	A6 Ļ	A7 §	A8 ¨	A9 Š	AA Ē	AB Ģ	AC Ŧ	AD -	AE Ž	AF ¯
B0 °	B1 ą	B2 ˛	B3 ŗ	B4 ´	B5 ĩ	B6 ļ	B7 ˇ	B8 ¸	B9 š	BA ē	BB ģ	BC ŧ	BD Ŋ	BE ž	BF ŋ
C0 Ā	C1 Á	C2 Â	C3 Ã	C4 Ä	C5 Å	C6 Æ	C7 Į	C8 Č	C9 É	CA Ę	CB Ë	CC Ė	CD Í	CE Î	CF Ī
D0 Đ	D1 Ņ	D2 Ō	D3 Ķ	D4 Ô	D5 Õ	D6 Ö	D7 ×	D8 Ø	D9 Ų	DA Ú	DB Û	DC Ü	DD Ũ	DE Ū	DF ß
E0 ā	E1 á	E2 â	E3 ã	E4 ä	E5 å	E6 æ	E7 į	E8 č	E9 é	EA ę	EB ë	EC ė	ED í	EE î	EF ī
F0 đ	F1 ņ	F2 ō	F3 ķ	F4 ô	F5 õ	F6 ö	F7 ÷	F8 ø	F9 ų	FA ú	FB û	FC ü	FD ũ	FE ū	FF ˙

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

A0	A1 Ё	A2 Ђ	A3 Ѓ	A4 Є	A5 Ѕ	A6 І	A7 Ї	A8 Ј	A9 Љ	AA Њ	AB Ћ	AC Ќ	AD -	AE Ў	AF Џ
B0 А	B1 Б	B2 В	B3 Г	B4 Д	B5 Е	B6 Ж	B7 З	B8 И	B9 Й	BA К	BB Л	BC М	BD Н	BE О	BF П
C0 Р	C1 С	C2 Т	C3 У	C4 Ф	C5 Х	C6 Ц	C7 Ч	C8 Ш	C9 Щ	CA Ъ	CB Ы	CC Ь	CD Э	CE Ю	CF Я
D0 а	D1 б	D2 в	D3 г	D4 д	D5 е	D6 ж	D7 з	D8 и	D9 й	DA к	DB л	DC м	DD н	DE о	DF п
E0 р	E1 с	E2 т	E3 у	E4 ф	E5 х	E6 ц	E7 ч	E8 ш	E9 щ	EA ъ	EB ы	EC ь	ED э	EE ю	EF я
F0 №	F1 ё	F2 ђ	F3 ѓ	F4 є	F5 ѕ	F6 і	F7 ї	F8 ј	F9 љ	FA њ	FB ћ	FC ќ	FD §	FE ў	FF џ

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

A0	A1 ‘	A2 ’	A3 £			A6 ¦	A7 §	A8 ¨	A9 ©		AB «	AC ¬	AD -		AF ―
B0 °	B1 ±	B2 ²	B3 ³	B4 ΄	B5 ΅	B6 Ά	B7 ·	B8 Έ	B9 Ή	BA Ί	BB »	BC Ό	BD ½	BE Ύ	BF Ώ
C0 ΐ	C1 Α	C2 Β	C3 Γ	C4 Δ	C5 Ε	C6 Ζ	C7 Η	C8 Θ	C9 Ι	CA Κ	CB Λ	CC Μ	CD Ν	CE Ξ	CF Ο
D0 Π	D1 Ρ		D3 Σ	D4 Τ	D5 Υ	D6 Φ	D7 Χ	D8 Ψ	D9 Ω	DA Ϊ	DB Ϋ	DC ά	DD έ	DE ή	DF ί
E0 ΰ	E1 α	E2 β	E3 γ	E4 δ	E5 ε	E6 ζ	E7 η	E8 θ	E9 ι	EA κ	EB λ	EC μ	ED ν	EE ξ	EF ο
F0 π	F1 ρ	F2 ς	F3 σ	F4 τ	F5 υ	F6 φ	F7 χ	F8 ψ	F9 ω	FA ϊ	FB ϋ	FC ό	FD ύ	FE ώ	

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

A0		A2 ¢	A3 £	A4 ¤	A5 ¥	A6 ¦	A7 §	A8 ¨	A9 ©	AA ×	AB «	AC ¬	AD -	AE ®	AF ¯
B0 °	B1 ±	B2 ²	B3 ³	B4 ´	B5 µ	B6 ¶	B7 ·	B8 ¸	B9 ¹	BA ÷	BB »	BC ¼	BD ½	BE ¾	
															DF ‗
E0 א	E1 ב	E2 ג	E3 ד	E4 ה	E5 ו	E6 ז	E7 ח	E8 ט	E9 י	EA ך	EB כ	EC ל	ED ם	EE מ	EF ן
F0 נ	F1 ס	F2 ע	F3 ף	F4 פ	F5 ץ	F6 צ	F7 ק	F8 ר	F9 ש	FA ת					

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

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

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

A0	A1 Ą	A2 Ē	A3 Ģ	A4 Ī	A5 Ĩ	A6 Ķ	A7 §	A8 Ļ	A9 Đ	AA Š	AB Ŧ	AC Ž	AD -	AE Ū	AF Ŋ
B0 °	B1 ą	B2 ē	B3 ģ	B4 ī	B5 ĩ	B6 ķ	B7 ·	B8 ļ	B9 đ	BA š	BB ŧ	BC ž	BD ―	BE ū	BF ŋ
C0 Ā	C1 Á	C2 Â	C3 Ã	C4 Ä	C5 Å	C6 Æ	C7 Į	C8 Č	C9 É	CA Ę	CB Ë	CC Ė	CD Í	CE Î	CF Ï
D0 Ð	D1 Ņ	D2 Ō	D3 Ó	D4 Ô	D5 Õ	D6 Ö	D7 Ũ	D8 Ø	D9 Ų	DA Ú	DB Û	DC Ü	DD Ý	DE Þ	DF ß
E0 ā	E1 á	E2 â	E3 ã	E4 ä	E5 å	E6 æ	E7 į	E8 č	E9 é	EA ę	EB ë	EC ė	ED í	EE î	EF ï
F0 ð	F1 ņ	F2 ō	F3 ó	F4 ô	F5 õ	F6 ö	F7 ũ	F8 ø	F9 ų	FA ú	FB û	FC ü	FD ý	FE þ	FF ĸ

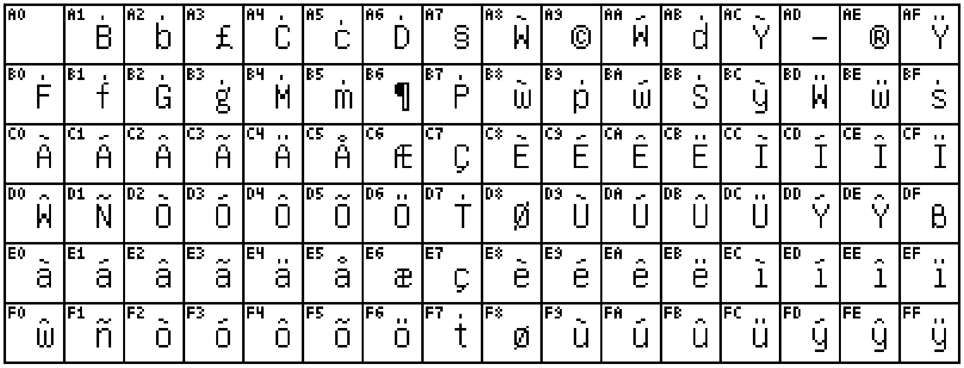
2.12.11. ISO8859-11 (characters 0xA0 ~ 0xFF)

	A1 ก	A2 ข	A3 ฃ	A4 ค	A5 ฅ	A6 ฆ	A7 ง	A8 จ	A9 ฉ	AA ช	AB ซ	AC ฌ	AD ญ	AE ฎ	AF ฏ
B0 ฐ	B1 ฑ	B2 ฒ	B3 ณ	B4 ด	B5 ต	B6 ถ	B7 ท	B8 ธ	B9 น	BA บ	BB ป	BC ผ	BD ฝ	BE พ	BF ฟ
C0 ภ	C1 ม	C2 ย	C3 ร	C4 ฤ	C5 ล	C6 ฦ	C7 ว	C8 ศ	C9 ษ	CA ส	CB ห	CC ฬ	CD อ	CE ฮ	CF ฯ
D0 ะ	D1 ั	D2 า	D3 ำ	D4 ิ	D5 ี	D6 ึ	D7 ื	D8 ุ	D9 ู	DA ฺ					DF ฿
E0 เ	E1 แ	E2 โ	E3 ใ	E4 ไ	E5 ๅ	E6 ๆ	E7 ็	E8 ่	E9 ้	EA ๊	EB ๋	EC ์	ED ํ	EE ๎	EF ๏
F0 ๐	F1 ๑	F2 ๒	F3 ๓	F4 ๔	F5 ๕	F6 ๖	F7 ๗	F8 ๘	F9 ๙	FA ๚	FB ๛				

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

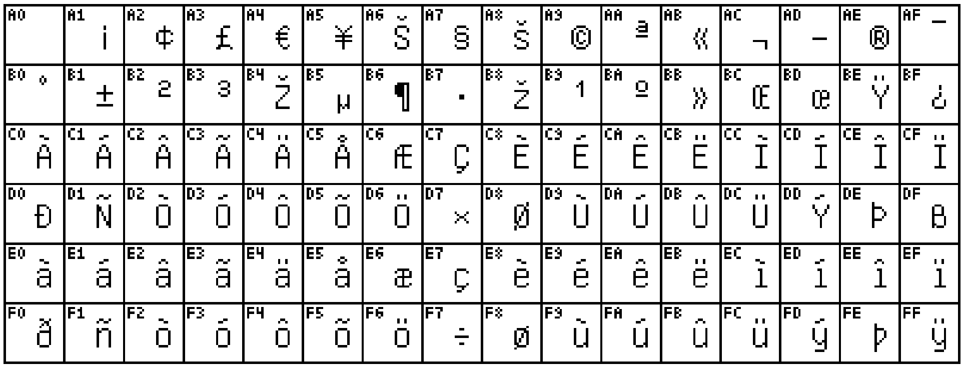
A0	A1 ”	A2 ¢	A3 £	A4 ¤	A5 „	A6 ¦	A7 §	A8 Ø	A9 ©	AA Ŗ	AB «	AC ¬	AD -	AE ®	AF Æ
B0 °	B1 ±	B2 ²	B3 ³	B4 “	B5 µ	B6 ¶	B7 ·	B8 ø	B9 ¹	BA ŗ	BB »	BC ¼	BD ½	BE ¾	BF æ
C0 Ą	C1 Į	C2 Ā	C3 Ć	C4 Ä	C5 Å	C6 Ę	C7 Ē	C8 Č	C9 É	CA Ź	CB Ė	CC Ģ	CD Ķ	CE Ī	CF Ļ
D0 Š	D1 Ń	D2 Ņ	D3 Ó	D4 Ō	D5 Õ	D6 Ö	D7 ×	D8 Ų	D9 Ł	DA Ś	DB Ū	DC Ü	DD Ż	DE Ž	DF ß
E0 ą	E1 į	E2 ā	E3 ć	E4 ä	E5 å	E6 ę	E7 ē	E8 č	E9 é	EA ź	EB ė	EC ģ	ED ķ	EE ī	EF ļ
F0 š	F1 ń	F2 ņ	F3 ó	F4 ō	F5 õ	F6 ö	F7 ÷	F8 ų	F9 ł	FA ś	FB ū	FC ü	FD ż	FE ž	FF ’

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



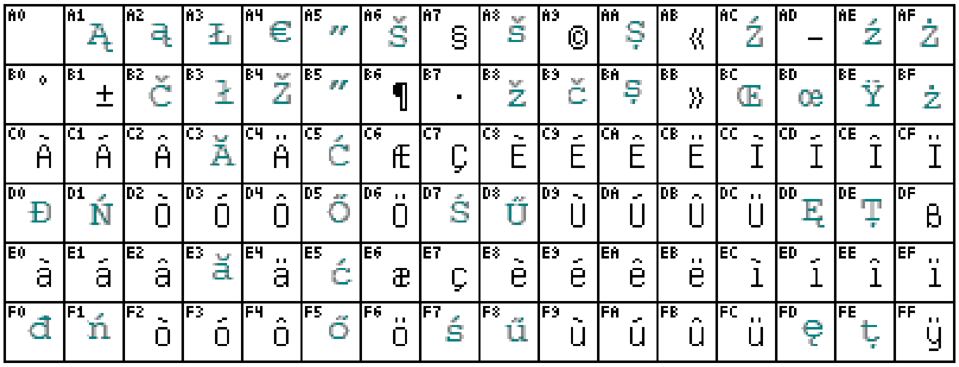
A0	A1 Ḃ	A2 ḃ	A3 £	A4 Ċ	A5 ċ	A6 Ḋ	A7 §	A8 Ẁ	A9 ©	AA Ẃ	AB ḋ	AC Ỳ	AD -	AE ®	AF Ÿ
B0 Ḟ	B1 ḟ	B2 Ġ	B3 ġ	B4 Ṁ	B5 ṁ	B6 ¶	B7 Ṗ	B8 ẁ	B9 ṗ	BA ẃ	BB Ṡ	BC ỳ	BD Ẅ	BE ẅ	BF ṡ
C0 À	C1 Á	C2 Â	C3 Ã	C4 Ä	C5 Å	C6 Æ	C7 Ç	C8 È	C9 É	CA Ê	CB Ë	CC Ì	CD Í	CE Î	CF Ï
D0 Ŵ	D1 Ñ	D2 Ò	D3 Ó	D4 Ô	D5 Õ	D6 Ö	D7 Ṫ	D8 Ø	D9 Ù	DA Ú	DB Û	DC Ü	DD Ý	DE Ŷ	DF ß
E0 à	E1 á	E2 â	E3 ã	E4 ä	E5 å	E6 æ	E7 ç	E8 è	E9 é	EA ê	EB ë	EC ì	ED í	EE î	EF ï
F0 ŵ	F1 ñ	F2 ò	F3 ó	F4 ô	F5 õ	F6 ö	F7 ṫ	F8 ø	F9 ù	FA ú	FB û	FC ü	FD ý	FE ŷ	FF ÿ

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



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

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



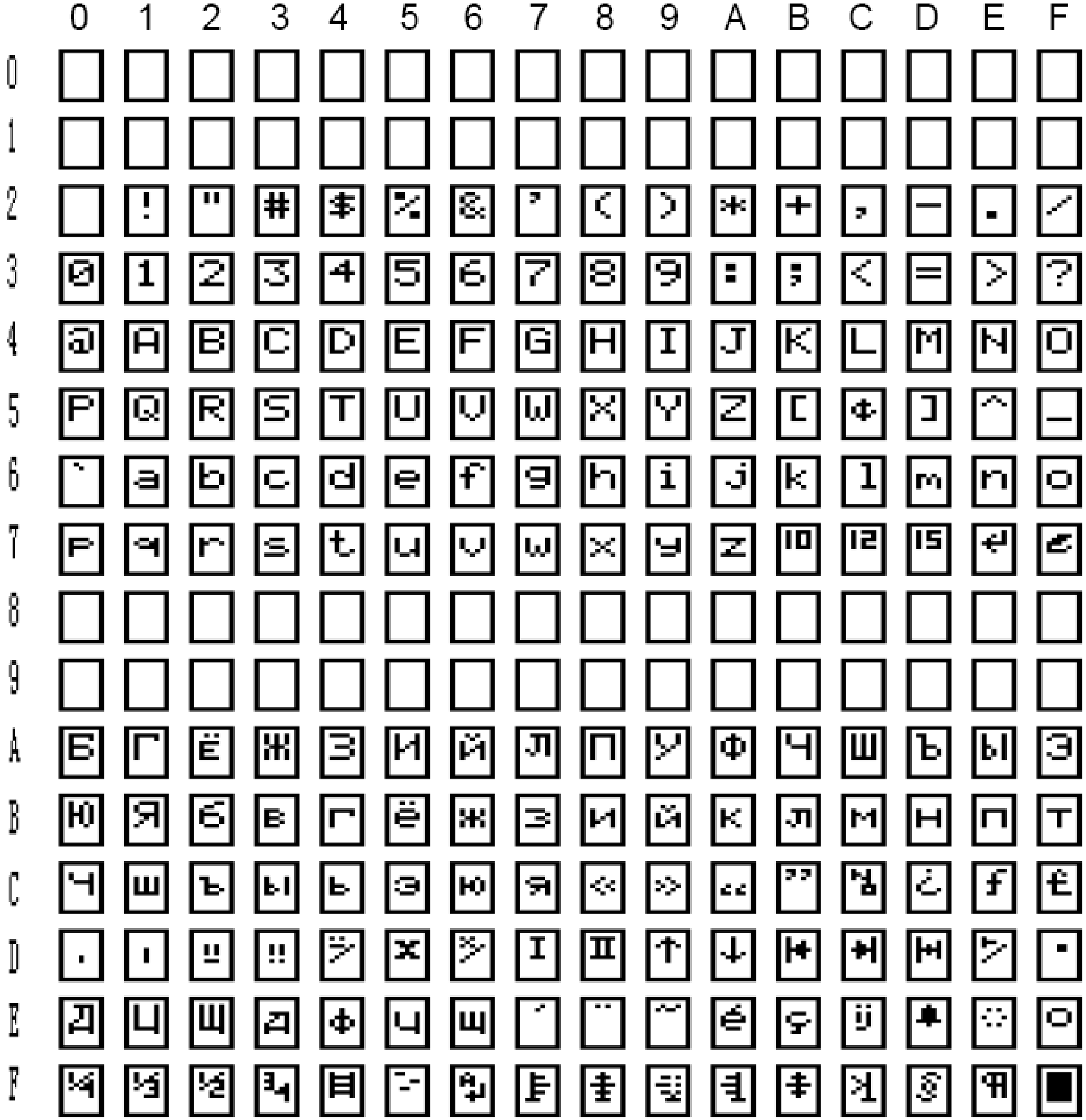
A0	A1 Ą	A2 ą	A3 Ł	A4 €	A5 „	A6 Š	A7 §	A8 š	A9 ©	AA Ș	AB «	AC Ź	AD -	AE ź	AF Ż
B0 °	B1 ±	B2 Č	B3 ł	B4 Ž	B5 ”	B6 ¶	B7 ·	B8 ž	B9 č	BA ș	BB »	BC Œ	BD œ	BE Ÿ	BF ż
C0 À	C1 Á	C2 Â	C3 Ă	C4 Ä	C5 Ć	C6 Æ	C7 Ç	C8 È	C9 É	CA Ê	CB Ë	CC Ì	CD Í	CE Î	CF Ï
D0 Đ	D1 Ń	D2 Ò	D3 Ó	D4 Ô	D5 Ő	D6 Ö	D7 Ś	D8 Ű	D9 Ù	DA Ú	DB Û	DC Ü	DD Ę	DE Ț	DF ß
E0 à	E1 á	E2 â	E3 ă	E4 ä	E5 ć	E6 æ	E7 ç	E8 è	E9 é	EA ê	EB ë	EC ì	ED í	EE î	EF ï
F0 đ	F1 ń	F2 ò	F3 ó	F4 ô	F5 ő	F6 ö	F7 ś	F8 ű	F9 ù	FA ú	FB û	FC ü	FD ę	FE ț	FF ÿ

2.13. LCM 5x10

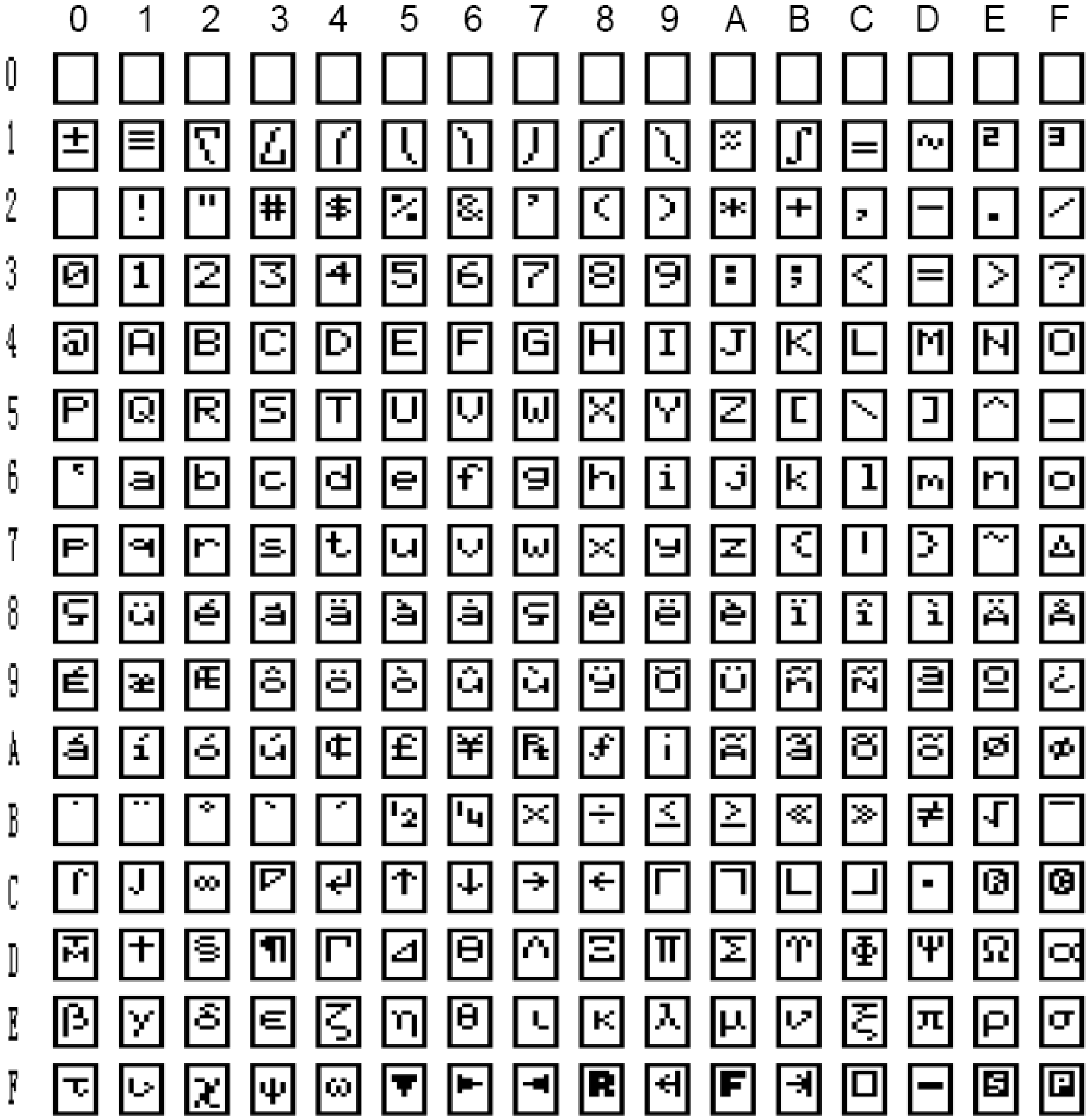
2.13.1. LCM 5x10-1

	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	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	ァ	ィ	ゥ	ェ	ォ	ャ	ュ	ョ	ッ
B	ー	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	ン	゛	゜
E	α	ä	β	ε	μ	σ	ρ	g	√	⁻¹	j	ˣ	¢	£	ñ	ö
F	p	q	θ	∞	Ω	ü	Σ	π	x̄	y	千	万	円	÷		█

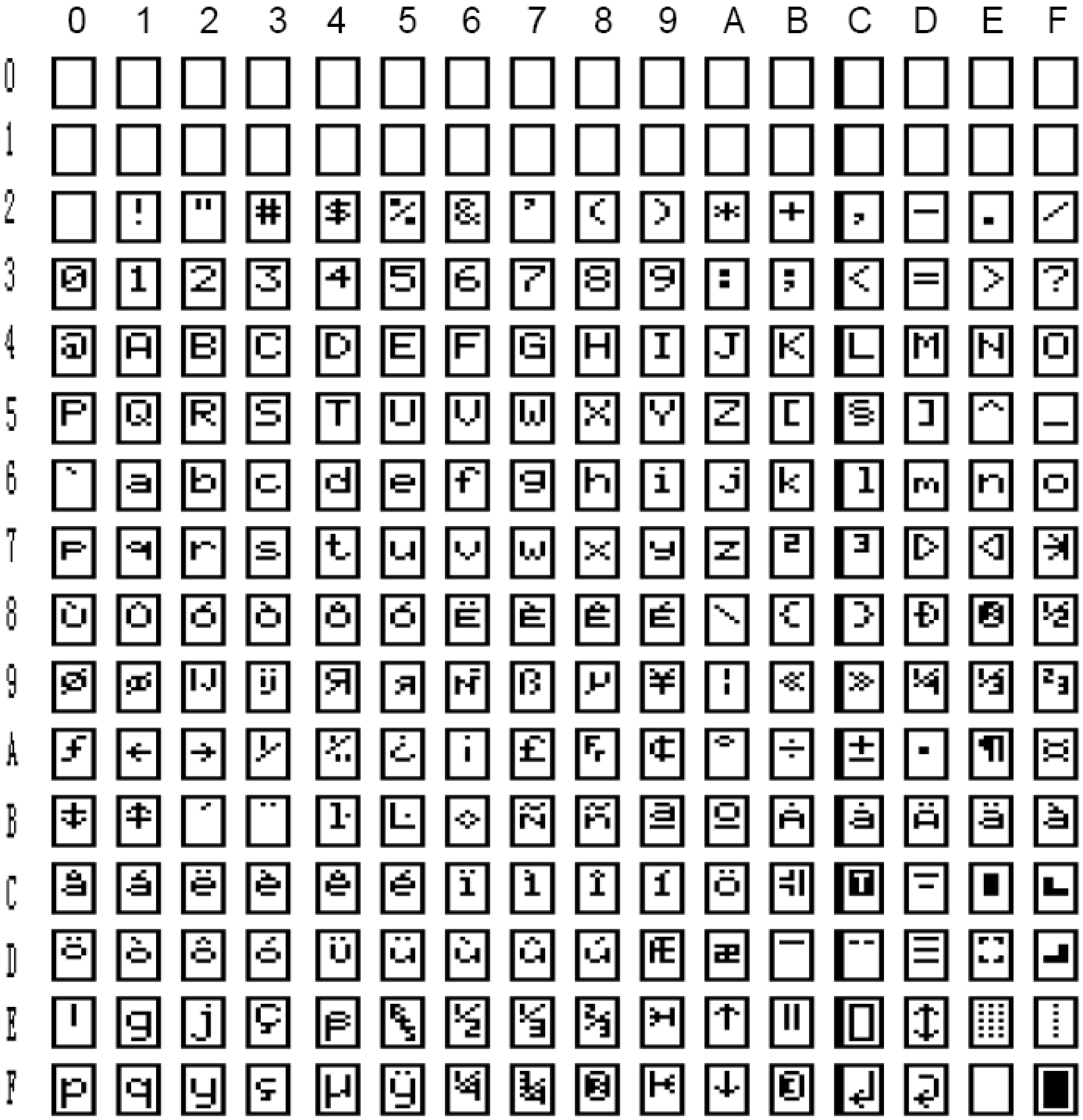
2.13.2. LCM 5x10-2



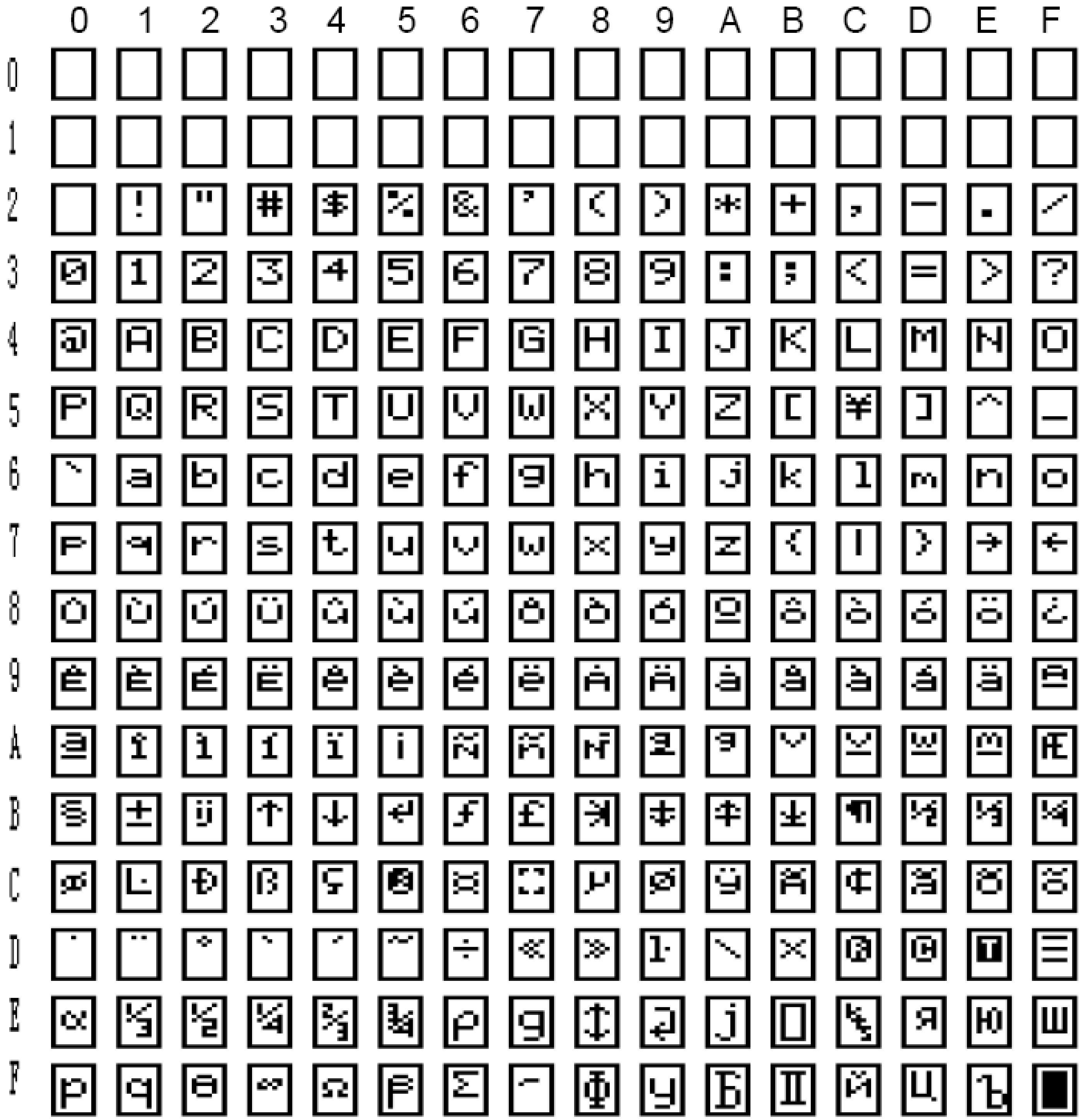
2.13.3. LCM 5x10-3


0
1
2
3
4
5
6
7
8
9
A
B
C
D
E
F

2.13.4. LCM 5x10-8



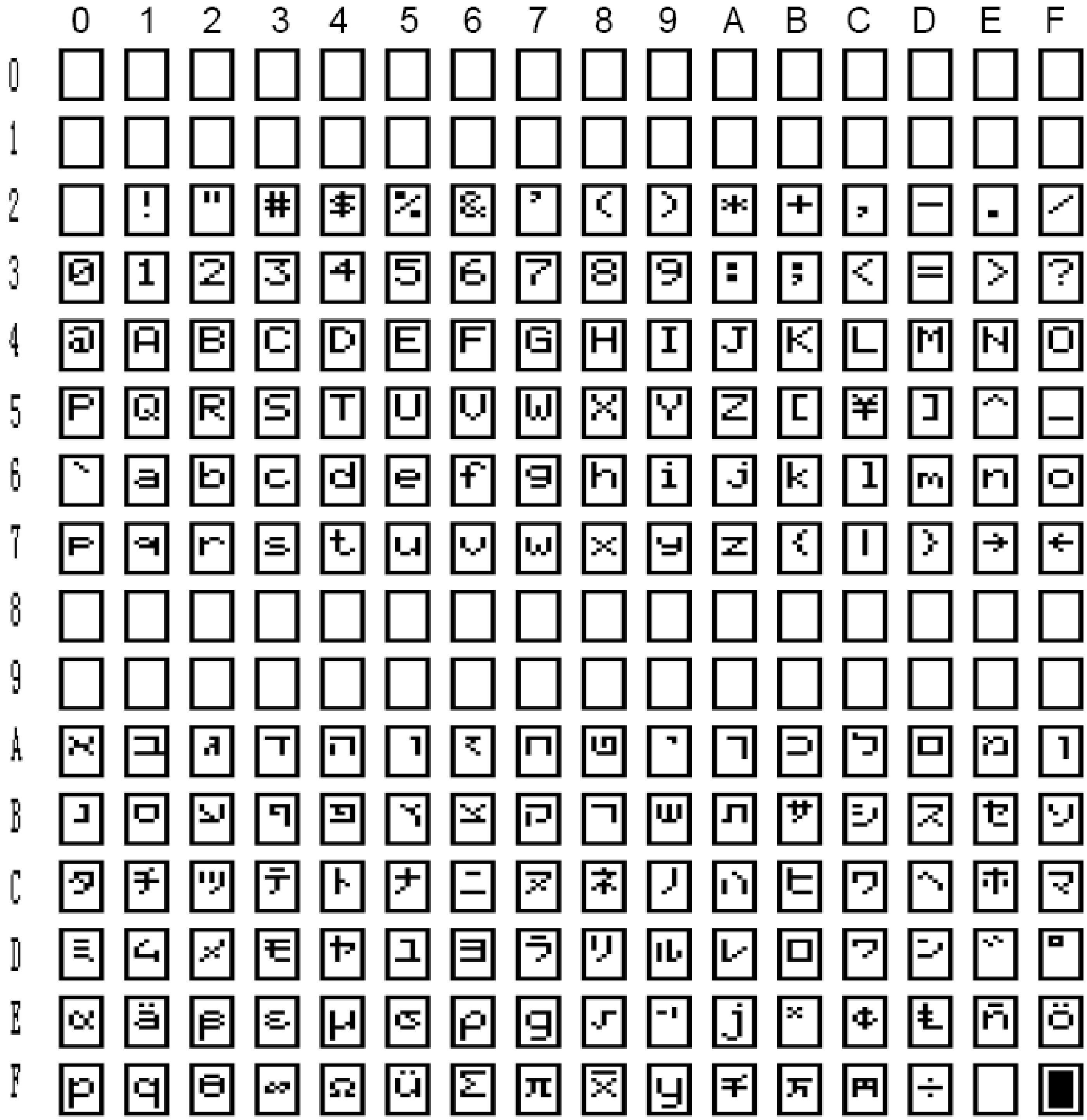
2.13.5. LCM 5x10-11


0
1
2
3
4
5
6
7
8
9
A
B
C
D
E
F

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

2.13.7. LCM 5x10-13

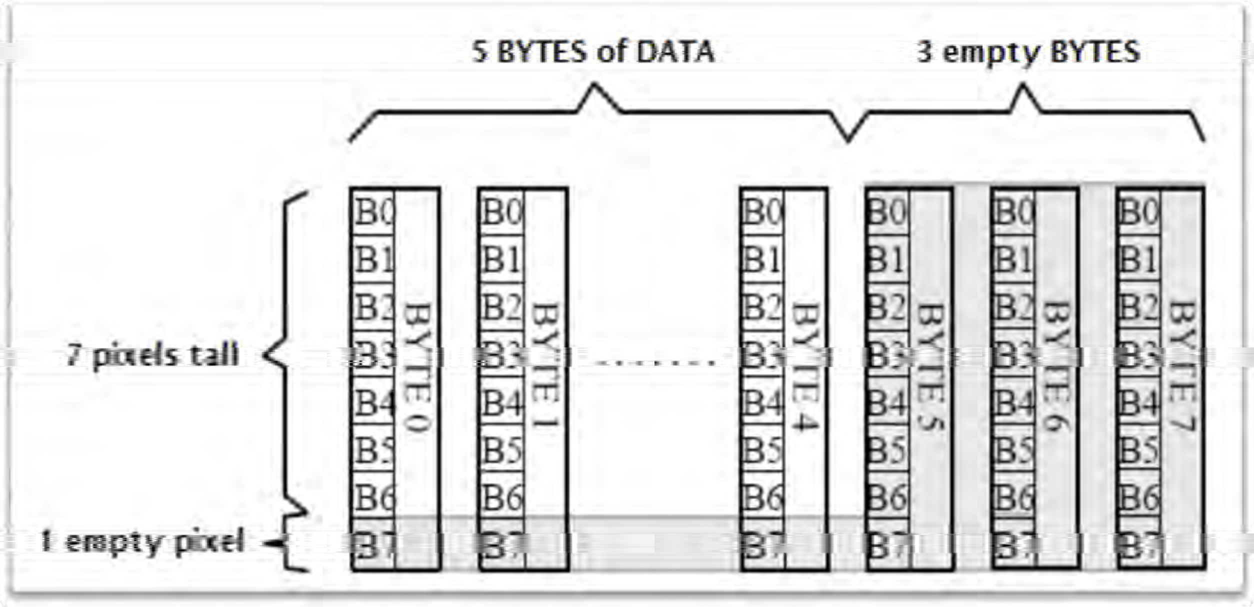

0
1
2
3
4
5
6
7
8
9
A
B
C
D
E
F

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.

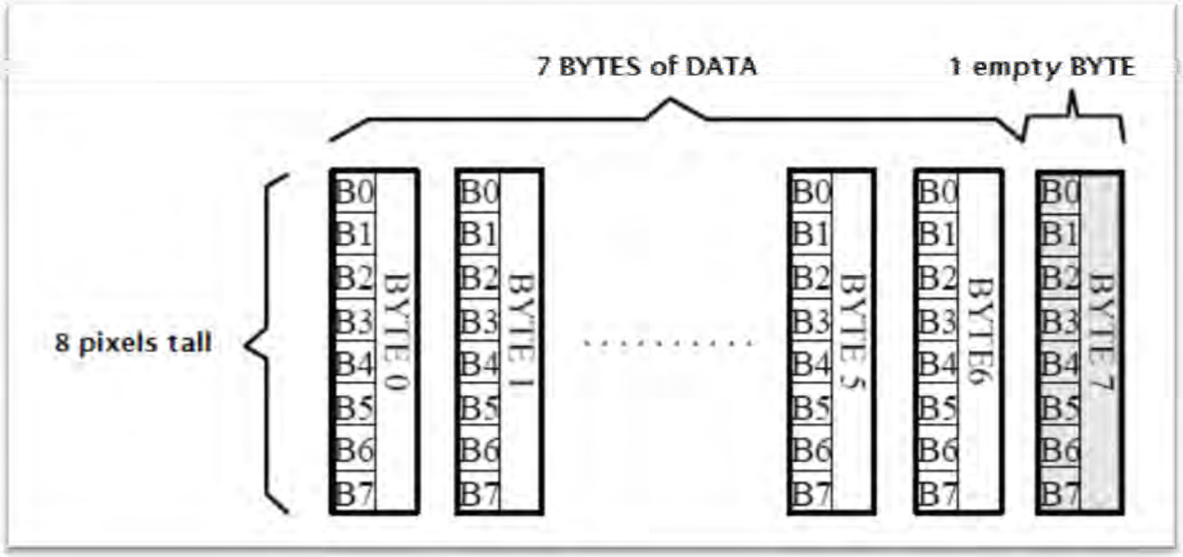
3.1. 5x7 Font Data Arrangement

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).



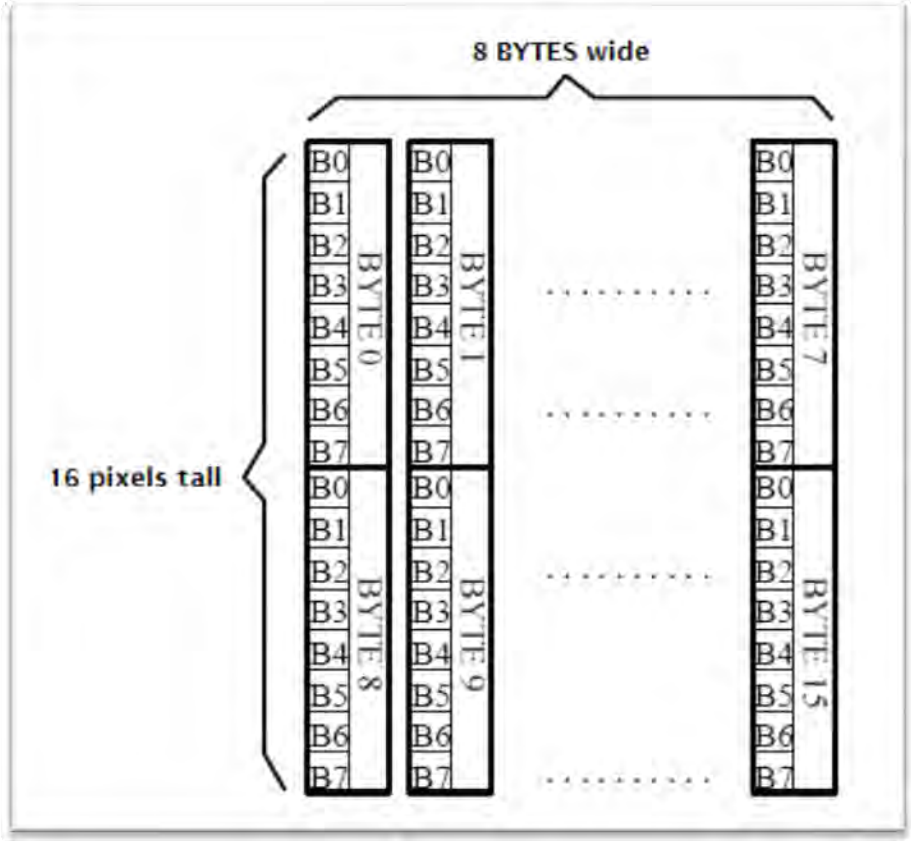
3.2. 7x8 Font Data Arrangement

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).



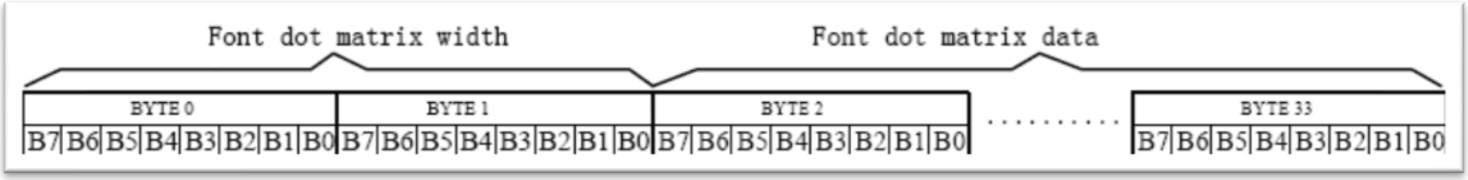
3.3. 8x16 Font Data Arrangement

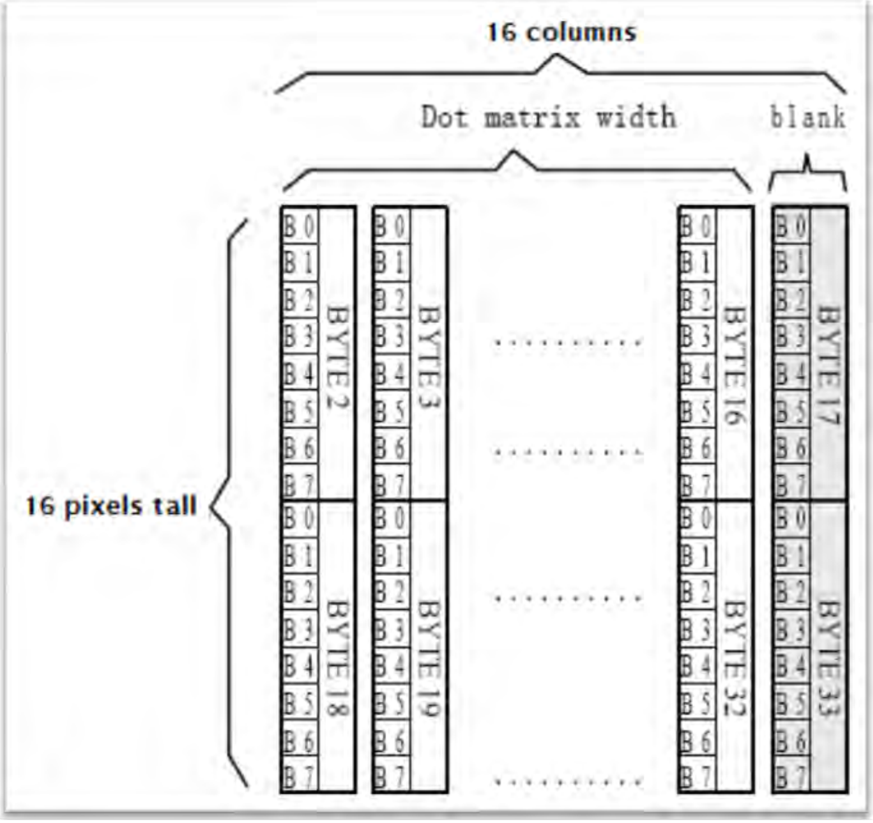
8x16 dot fonts require 16 bytes (BYTE 0 – BYTE 15) to display. BYTE 0 represents the top-left most column of the font. BYTE 0 through BYTE 7 represent the top half of the font. BYTE 8 represents the bottom-left most column of the font. BYTE 8 through BYTE 15 represent the bottom half of the font.



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	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0	0
	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0	0
	0	1	1	0	0	0	0	0	1	1	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	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
	0	1	1	0	0	0	0	0	1	1	1	0	0	0	0	0
	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0
	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0
	0	1	1	1	1	1	1	1	1	1	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

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	1	0	0	0	0	0
	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0
	0	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	1	0	0	0	0	0	0	0	0
	0	0	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	1	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	0
	0	0	0	0	0	0	0	1	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	1	1	1	1	1	1	1	1	1	1	1	1	1	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	1	1	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	0

4. Calculation of Font Addresses

4.1. 5x7 ASCII

Parameters:
ASCIICODE: 8bit ASCII character code
Address: Address of character data

```
if(ASCIICODE >= 0x20 && ASCIICODE <= 0xFF)
        Address = (ASCIICODE – 0x20) *8;
```

4.2. 7x8 ASCII

Parameters:
ASCIICODE: 8bit ASCII character code
Address: Address of character data

```
if(ASCIICODE >= 0x20 && ASCIICODE <= 0xFF)
        Address = ((ASCIICODE – 0x20) *8) + 768;
```

4.3. 8x16 ASCII

Parameters:
ASCIICODE: 8bit ASCII character code
Address: Address of character data

```
if(ASCIICODE >= 0x20 && ASCIICODE <= 0xFF)
        Address =( (ASCIICODE – 0x20) *16) + 1,536;
```

4.4. Width-Adjusted Arial ASCII

Parameters:
ASCIICODE: 8bit ASCII character code
Address: Address of character data

```
if(ASCIICODE >= 0x20 && ASCIICODE <= 0xFF)
        Address =( (ASCIICODE – 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
        }
```