GT20L24F6Y
Multi-Language Dot Matrix Font Chip

DATASHEET

- **ISO8859 (14 sets)：5x7, 8x16**
- **ASCII (11 sets)：5x7 ~ 24x24**
- **LCM Character Set (8 sets)：5x7, 5x10**
- **Unicode (173 coutries' language)：5x7 ~ 24x24**

  （Latin, Greek, Cyrillic, Arabic, Hebrew, Thai）
- **Data Arrangement: Vertical Byte,Horizontal String**
- **Bus Interface: SPI**
- **Package: SOT23-6**

VER 4.0

2011-4

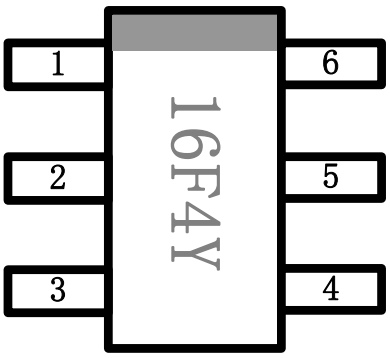
INDEX

1 General

GT20L24F6Y supports multi-country language character set,including 11 sets of ASCII, 8 sets of LCM characters, 14 sets of ISO8859, and support up to Unicode up to 173 countries' language. The data arrangement format is vertical byte, horizontal string. The user may obtain the address of certain character dot matrix with the calculation method given by this datasheet, which enables the user to access to more character data by continually reading from the address already obtained.

1.1 Chip Feature

- Bus Interface: SPI
- Data Arrangement: Vertical byte, horizontal string
- Frequency: 30MHz(max.) @3.3V
- Operating Voltage: 2.2V~3.6V
- Current:
  Operating: 8mA
  Standby: 8uA
- Package: SOT23-6
- Package Size: 2.9mmX1.6 mm x1.10mm
- Operating Temperature: -20℃~70℃



1.2 Pin Description

SOT23-6	Name	I/O	Description
1	SCLK	I	Serial clock input
2	GND		Ground
3	CS#	I	Chip enable input
4	VCC		+ 3.3V Power Supply
5	SO	O	Serial data output
6	SI	I	Serial data input

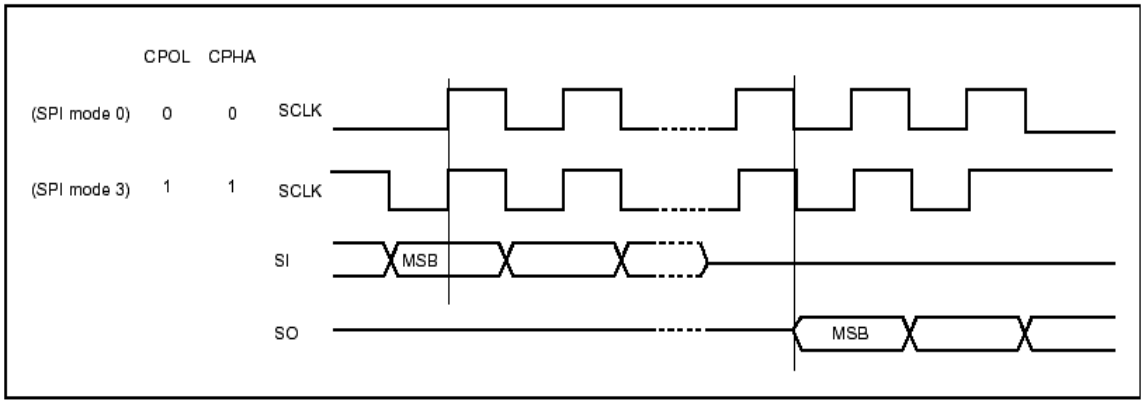
Serial Data Output (SO): Data shift-out on the falling edge of the serial clock.

Serial Data Input (SI): Data shift-in on the rising edge of the serial clock.

Serial Clock Input (SCLK): Data shift-out on the falling edge of the serial clock, shift-in on the rising

edge of the serial clock.

Chip Enable Input (CS#): The device is enabled by a high to low transition on CE#. CE# must remain low for the duration of any command sequence



1.3 Font Content

Character Set / Font Size / Number of Character			5X7 5X10	7X8	6X12	8X16	8X16 Bold	12X24	Proportional Adjusted: 12 Arial	12 Times	16 Arial	16	24
ASCII			96	96	96	96	96	96	96	96	96	96	96
LCM Character		Basic	256										
		Extended	256 X7										
UNI-CODE	Latin (130 Countries)	Basic	96			96		96	96		96		
		Supplement	96			96		96	96		96		
		Extended A	115			128		128	128		128		
		Extended B				80		80	80		80		
		Extended Additional				96		96	96		96		
	Greek (2 Countries)	Basic	86			96		96	96		96		
	Cyrillic (15 Countries)	Basic	94			208		208	208		208		
	Arabic (24 Countries)	Basic									256		256
		Form A									176		176
		Form B									144		144
	Hebrew (1 Country)	Basic	32			112							
	Thai (1 Country)	Basic	96			128							
ISO-8859	NO.1～NO.16 (NO.6 和 NO.12 无)		128X 14			128X 14							

Note: See 6.8 for 5X7 ISO8859 to UNICODE conversion table

1.4 Font Sample

1） ASCII

5x7 ASCII **7x8 ASCII** **6X12 ASCII** **8X16 ASCII**

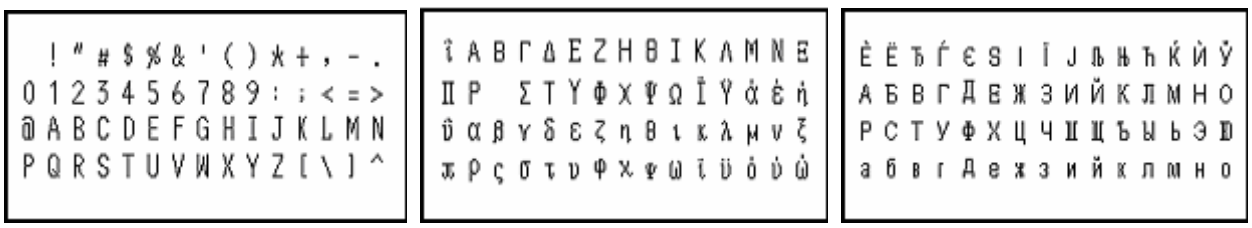
12 dots ASCII **16 dots ASCII** **24 dots ASCII**

2） LCM Character（5x7）

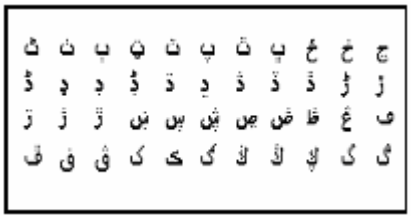
LCM-0 Common **LCM-1 English&Japanese** **LCM-2 English&Russian** **LCM-3 English&European**

3） Unicode（16 dot matrix）

Latin **Greek** **Cyrillic**

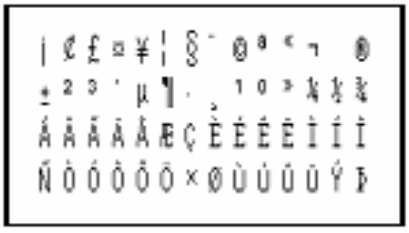


Hebrew **Thai** **Arabic**

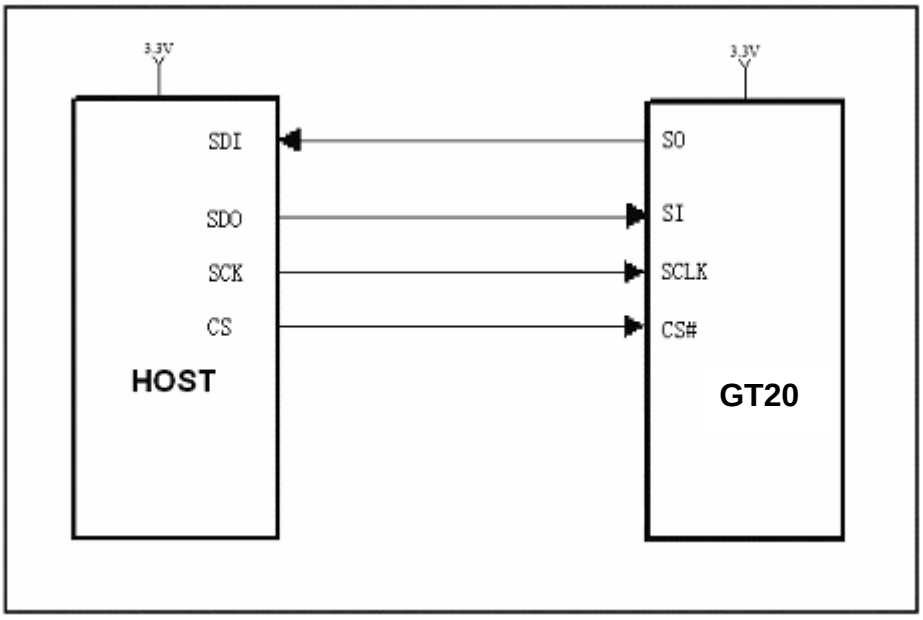


4） ISO8859（8x16）

ISO-8859-1 **ISO-8859-2** **ISO-8859-3**



1.5 SPI Connection Block Diagram



2 Operating Instruction

2.1 Instruction Parameter

Instruction Set

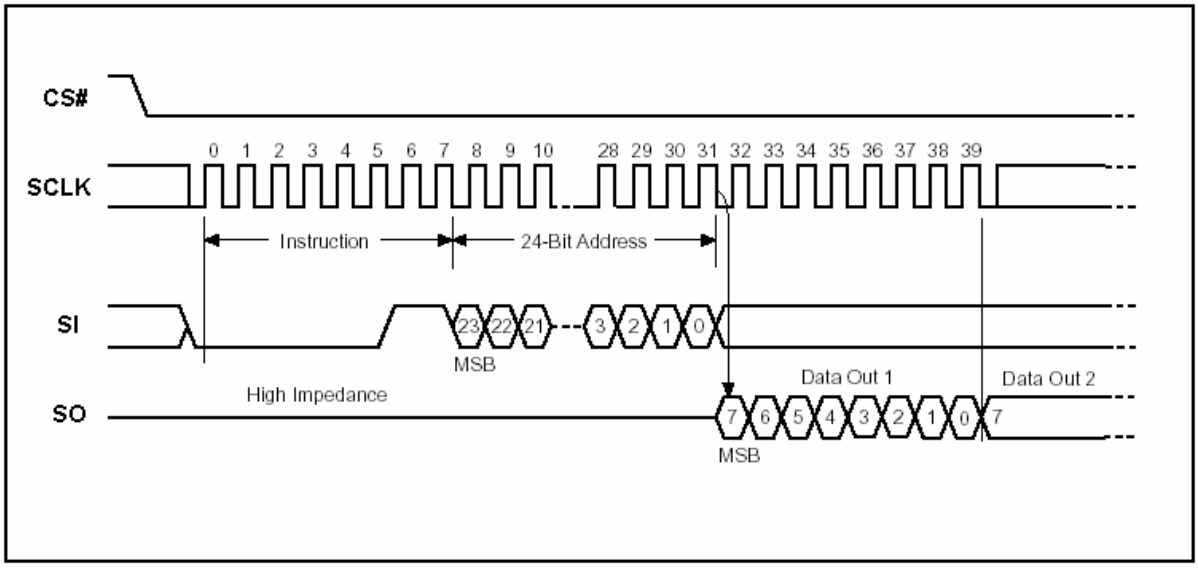
Instruction	Description	Instruction Code(One-Byte)		Address Bytes	Dummy Bytes	Data Bytes
READ	Read Data Bytes	0000 0011	03 h	3	—	1 to ∞
FAST_READ	Read Data Bytes at Higher Speed	0000 1011	0B h	3	1	1 to ∞

2.2 Read Data Bytes

The Read instruction supports up to 20 MHz, It outputs the data starting from the specified address location. The data output stream is continuous through all addresses until terminated by a low to high transition on CE#. The internal address pointer will automatically increment.

The Read instruction is initiated by executing an 8-bit command,03H, followed by address bits [A23-A0]. CE# must remain active low for the duration of the Read cycle.

Figure：Read Data Bytes (READ) Instruction Sequence and Data-out sequence:

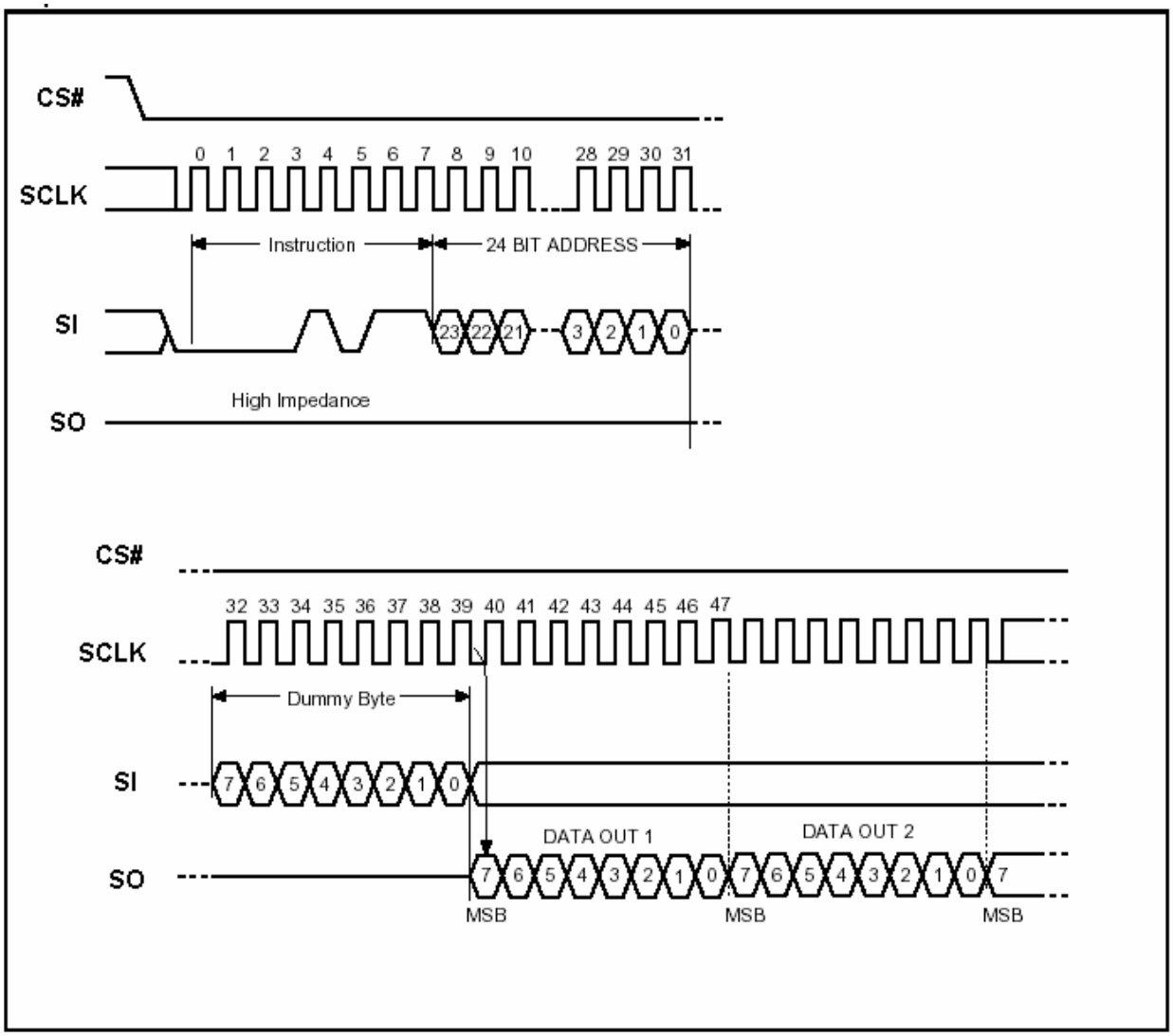


2.3 Read Data Bytes at Higher Speed

The High-Speed-Read instruction supporting up to 30 MHz is initiated by executing an 8-bit command, 0BH, followed by address bits [A23-A0] and a dummy byte. CE# must remain active low for the duration of the High-Speed-Read cycle.

Following a dummy byte (8 clocks input dummy cycle), the High-Speed-Read instruction outputs the data starting from the specified address location. The data output stream is continuous through all addresses until terminated by a low to high transition on CE#. The internal address pointer will automatically increment.

Read Data Bytes at Higher Speed (READ_FAST) Instruction Sequence and Data-out sequence:



3 Electrical Characteristic

3.1 Absolute Maximum Rating

Symbol	Parameter	Min.	Max.	Unit	Condition
T_{OP}	Operating Temperature	-20	70	℃	
T_{STG}	Storage Temperature	-65	150	℃	
VCC	Supply Voltage	-0.3	3.6	V	
V_{IN}	Input Voltage	-0.3	VCC+0.3	V	
GND	Power Ground	-0.3	0.3	V	

3.2 DC Characteristic

Condition：T_{OP} =-20℃ to 70℃，GND=0V

Symbol	Parameter	Min.	Max.	Unit	Condition
I_{DD}	VCC Supply Current(active)		8	mA	
I_{SB}	VCC Standby Current		8	uA	
V_{IL}	Input LOW Voltage	-0.3	0.3VCC	V	VCC=2.2~3.6V
V_{IH}	Input HIGH Voltage	0.7VCC	VCC+0.4	V	
V_{OL}	Output LOW Voltage		0.4 (I_{OL}=1.6mA)	V	
V_{OH}	Output HIGH Voltage	0.8VCC (I_{OH}=-100uA)		V	
I_{LI}	Input Leakage Current	0	2	uA	
I_{LO}	Output Leakage Current	0	2	uA	

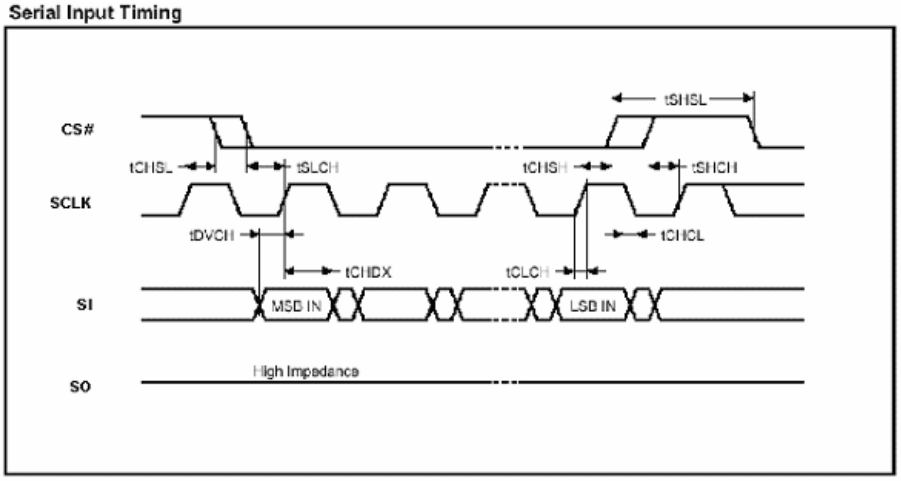
Note：I_{IL}：Input LOW Current，I_{IH}：Input HIGH Current，
I_{OL}：Output LOW Current，I_{OH}：Output HIGH Current，

3.3 AC Characteristic

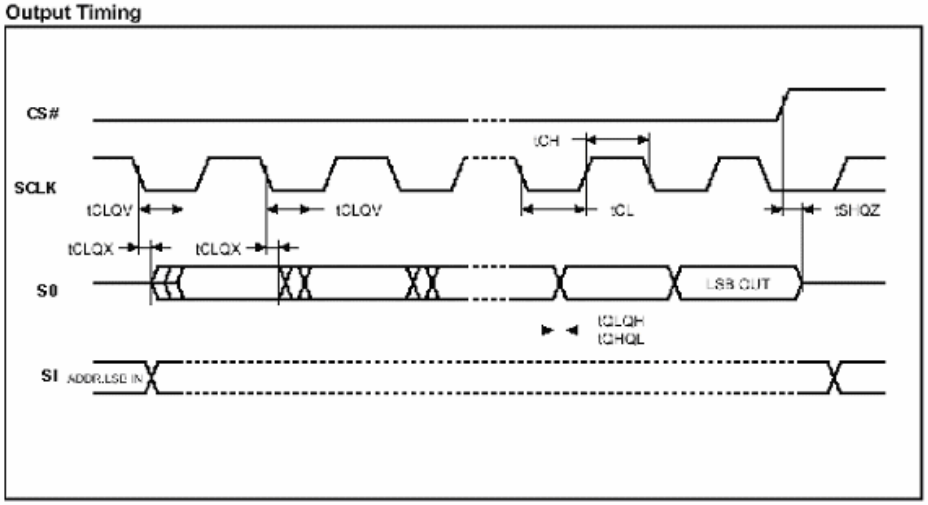
Symbol	Alt.	Parameter	Min.	Max.	Unit
Fc	Fc	Clock Frequency	D.C.	30	MHz
t_{CH}	t_{CLH}	Clock High Time	15		ns
t_{CL}	t_{CLL}	Clock Low Time	15		ns
t_{CLCH}		Clock Rise Time(peak to peak)	0.1		V/ns
t_{CHCL}		Clock Fall Time (peak to peak)	0.1		V/ns
t_{SLCH}	t_{CSS}	CS# Active Setup Time (relative to SCLK)	5		ns
t_{CHSL}		CS# Not Active Hold Time (relative to SCLK)	5		ns
t_{DVCH}	t_{DSU}	Data In Setup Time	2		ns
t_{CHDX}	t_{DH}	Data In Hold Time	5		ns
t_{CHSH}		CS# Active Hold Time (relative to SCLK)	5		ns
t_{SHCH}		CS# Not Active Setup Time (relative to SCLK)	5		ns
t_{SHSL}	t_{CSH}	CS# Deselect Time	100		ns
t_{SHQZ}	t_{DIS}	Output Disable Time		9	ns

t CLQV	tV	Clock Low to Output Valid		9	ns
t CLQX	tHO	Output Hold Time	0		ns

Serial Input Timing

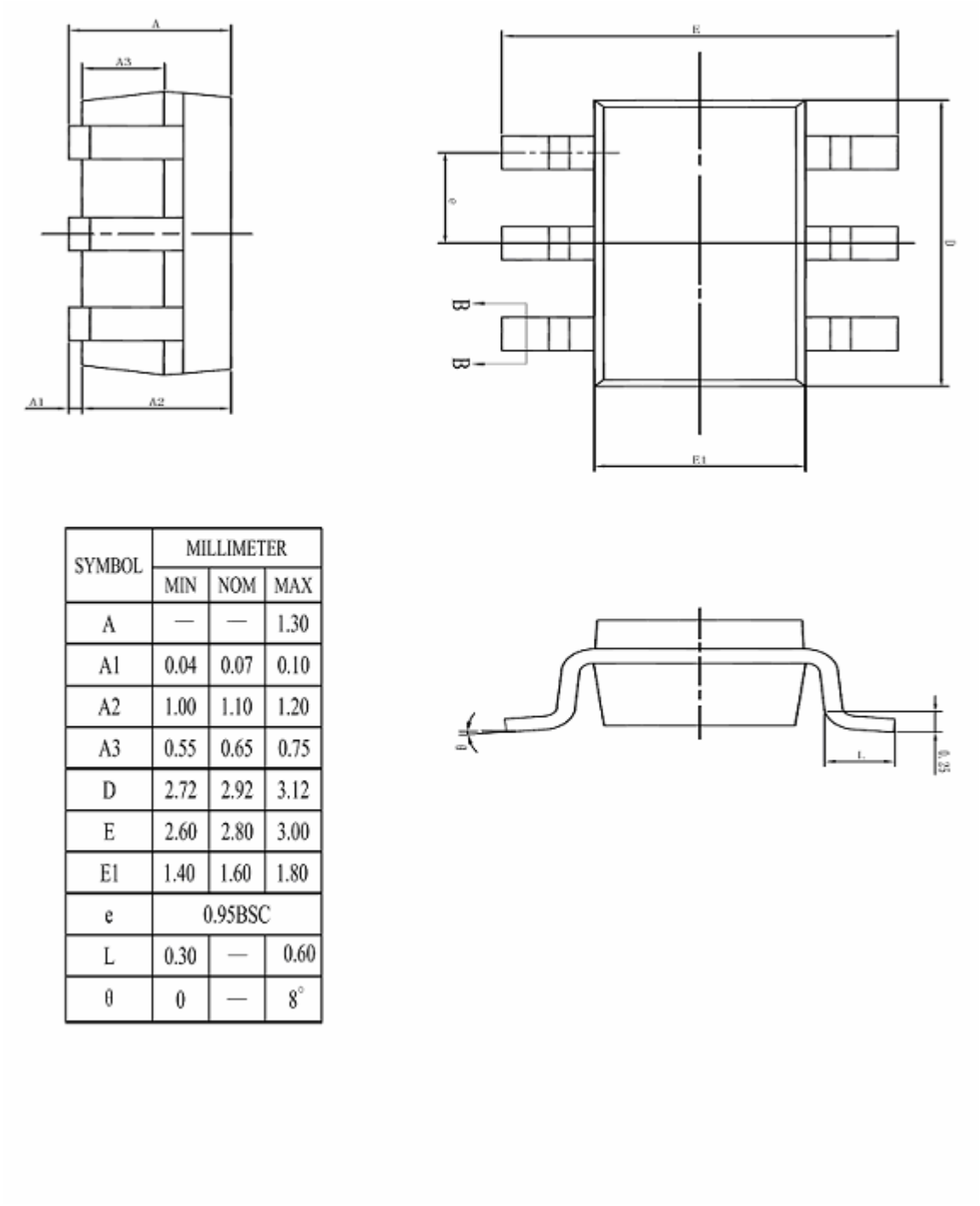


Output Timing



4 Package Size

SOT23-6 Package



SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	—	—	1.30
A1	0.04	0.07	0.10
A2	1.00	1.10	1.20
A3	0.55	0.65	0.75
D	2.72	2.92	3.12
E	2.60	2.80	3.00
E1	1.40	1.60	1.80
e	0.95BSC		
L	0.30	—	0.60
θ	0	—	8°

SOT23-6

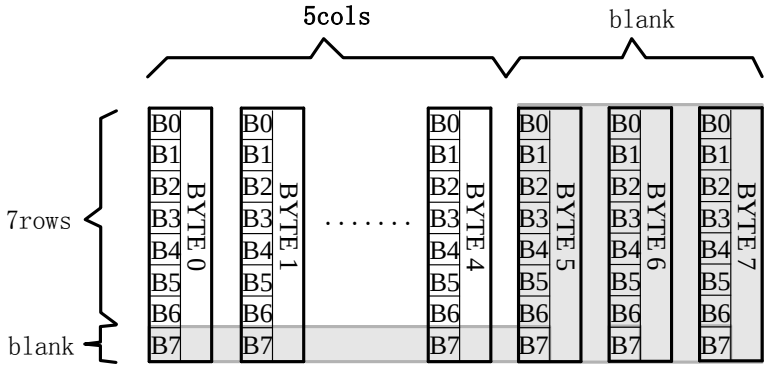
5 Font Read Method

5.1 Character Dot Matrix Arrangement(Data Arrangement Format)

Each character is stored in the Chinese dot matrix format, each dot is expressed by a binary bit. 1 represents for lightened dot, 0 represents for unlightened dot. The data arrangement format is byte vertical, string horizontal. The biggest bit of BYTE represents the most left point, the smallest bit of BYTE represents the most right point. Advances when horizontal row is booked. Chinese will display when using the above method.

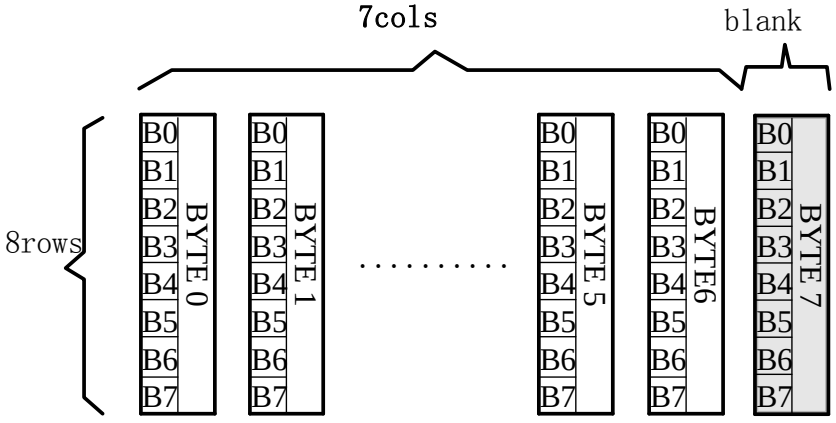
5.2 5x7

5X7 dots ASCII font requires 8 bytes(BYTE 0 – BYTE7) to display. Data arrangement format of this ASCII font is byte horizontal, string horizontal, the detailed arrangement structure is showed below:



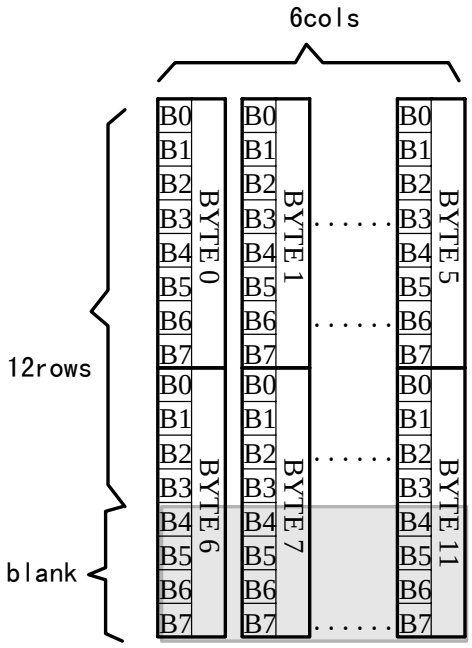
5.2.1 7X8

7X8 dots ASCII font requires 8 bytes (BYTE 0 – BYTE7) to display. Data arrangement format of this ASCII font is byte vertical, string horizontal, the detailed arrangement structure is showed below:



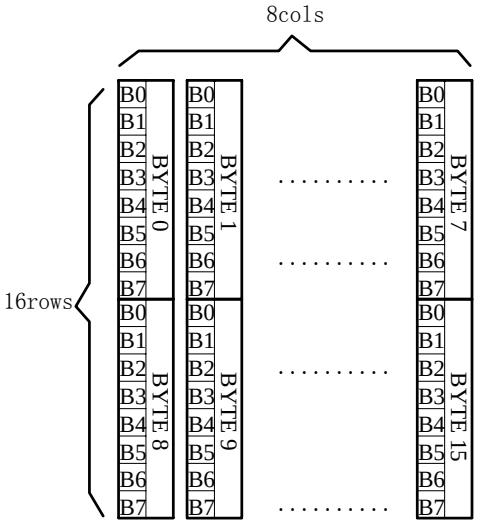
5.2.2 6X12

6X12 dots ASCII font requires 12 bytes (BYTE 0 – BYTE11) to display. Data arrangement format of this ASCII font is byte vertical, string horizontal, the detailed arrangement structure is showed below:



5.3 8X16

8X16 dots font requires 16 bytes（BYTE 0 – BYTE15）to display. Data arrangement format of this font is byte vertical, string horizontal, the detailed arrangement structure is showed below:



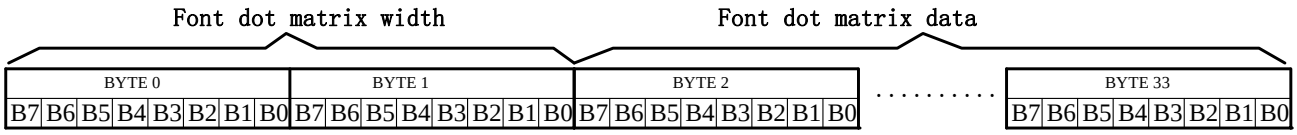
5.3.1 12 Dot Matrix Proportional Adjusted Font

The following font has the same dot matrix format
12 dot matrix proportional adjusted Arial ASCII
12 dot matrix proportional adjusted Times New Roman ASCII
12 dot matrix proportional adjusted Unicode (Latin, Greek, Cyrillic）

- Storage Format

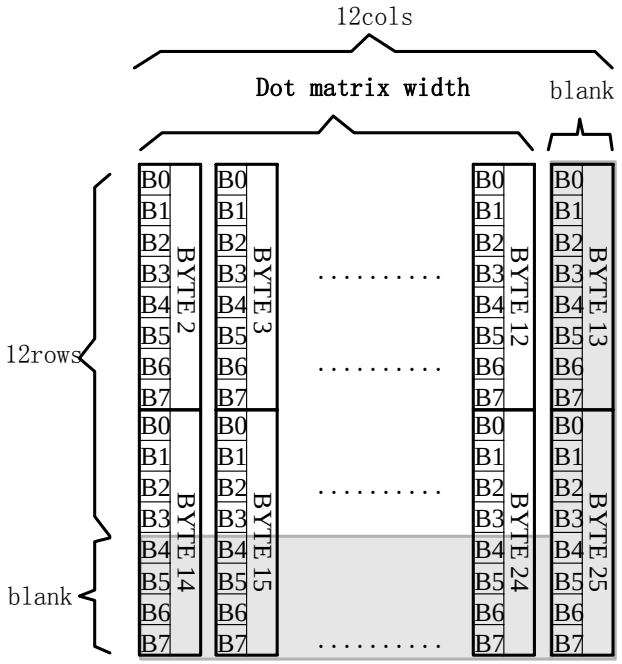
12 dots proportionally adjusted font requires 26 bytes（BYTE 0 – BYTE25）to display. For the font is proportionally adjusted, BYTE0~ BYTE1 are stored font width data,

BYTE2-25 are stored dots matrix data.



■ Storage Structure

The dots matrix storage width of proportionally adjusted font uses BYTE as its unit. Different font width will reveal corresponding blanks. With the font's actual width data stored in BYTE0~BYTE 1, it can be used as reference for the position of the next word.



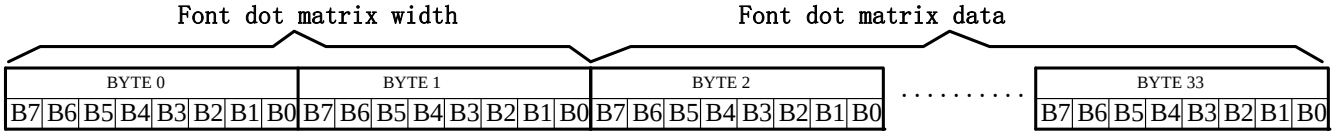
5.4 16 Dot Matrix Proportional Adjusted Font

The following font has the same dot matrix format
16 dot matrix proportional adjusted ASCII Arial
16 dot matrix proportional adjusted ASCII Times New Roman
16 dot matrix proportional adjusted Unicode (Latin, Greek, Cyrillic）
16 dot matrix proportional adjusted Arabic

■ Storage Format

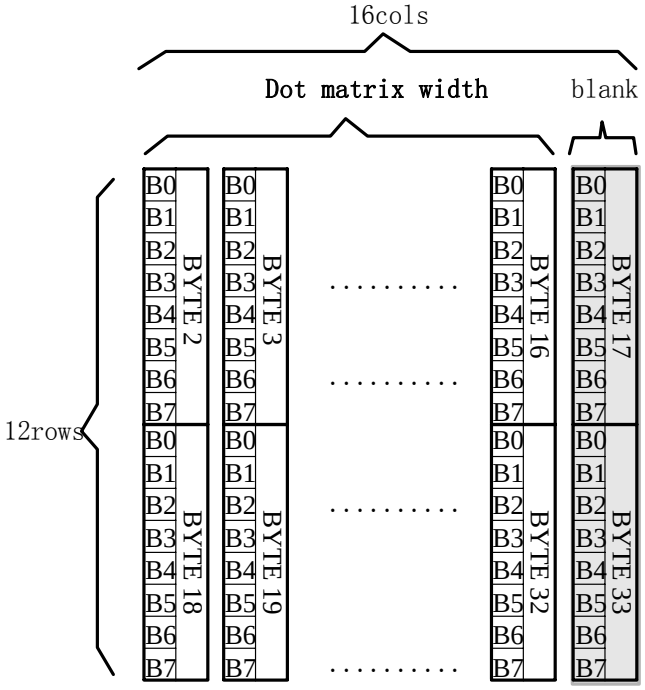
16 dots proportionally adjusted font requires 34 bytes（BYTE 0 – BYTE33）to display.

For the font is proportionally adjusted, BYTE0~ BYTE1 are stored font width data, BYTE2-33 are stored dots matrix data.



■ Storage Structure

The dots matrix storage width of proportionally adjusted font uses BYTE as its unit. Different font width will reveal corresponding blanks. With the font's actual width data stored in BYTE0~BYTE 1, it can be used as reference for the position of the next word.



For Example: ASCII Arial Font "B"

0-33 BYTE: 00 0C 00 00 00 00 00 00 7F 80 7F C0 60 C0 60 C0 60 C0 7F 80 7F C0 60 E0 60 60 60 60 7F C0 7F 80 00 00

In BYTE0~BYTE1: "00 0C" is width data, 12 bit width, 4 blank bits is reserved. The typeset of the next word may shift forward considering the blank bits.

In BYTE2~BYTE33: "00 00 00 00 00 00 7F 80 7F C0 60 C0 60 C0 60 C0 7F 80 7F C0 60 E0 60 60 60 60 7F C0 7F 80 00 00" is dot matrix data.

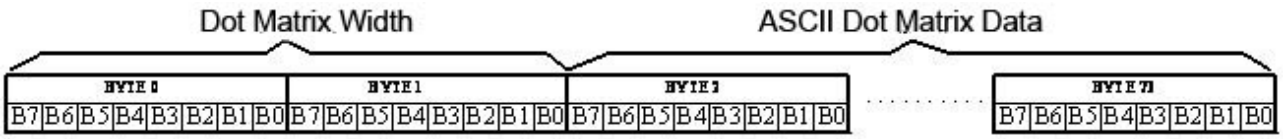
5.4.1 24 Dot Matrix Proportional Adjusted Font

The following font has the same dot matrix format
24 dot matrix proportional adjusted ASCII Arial
24 dot matrix proportional adjusted Arabic

■ Storage Format

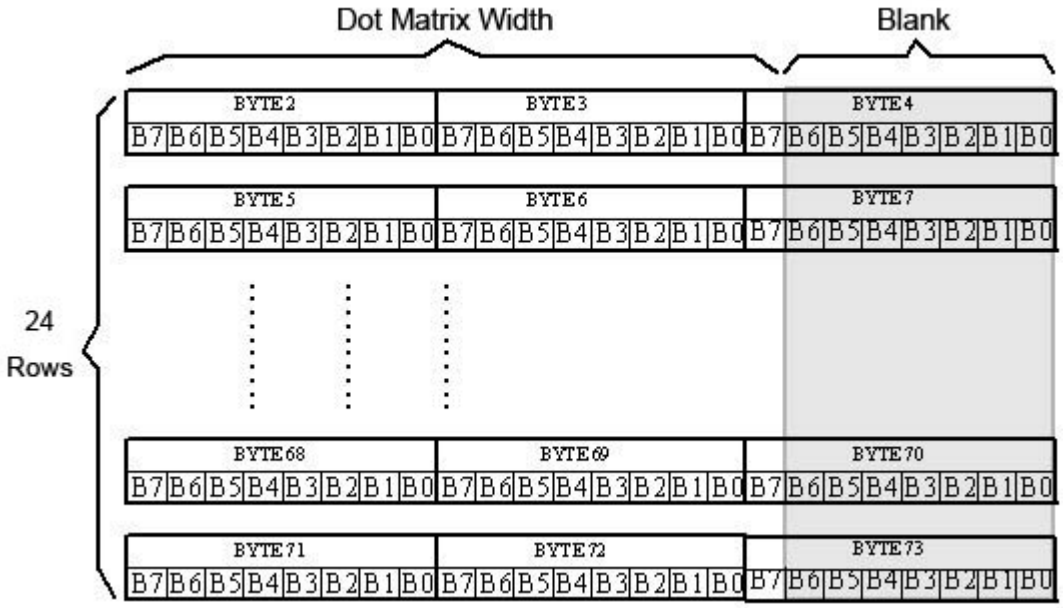
24 dots proportionally adjusted font requires 74 bytes（BYTE 0 – BYTE73）to display.

For the font is proportionally adjusted, BYTE0~ BYTE1 are stored font width data, BYTE2-73 are stored dots matrix data.



■ Storage Structure

The dots matrix storage width of proportionally adjusted font uses BYTE as its unit. Different font width will reveal corresponding blanks. With the font's actual width data stored in BYTE0~BYTE 1, it can be used as reference for the position of the next word.

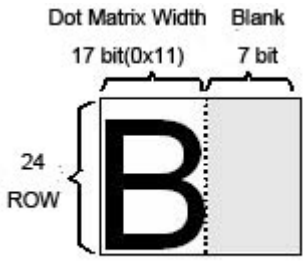


For Example: ASCII Arial Font "B"

0-73 BYTE: 00 11 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 3F F8 00 3F FC 00 3F FC 00 38 1E 00 38 0E 00 38 0E 00 38 1E 00 3F FC 00 3F FC 00 3F FE 00 38 0F 00 38 07 00 38 07 00 38 0F 00 3F FE 00 3F FE 00 3F F8 00 00 00 00

BYTE0~ BYTE1:

"00 11" is width data, 17 bit width, 7 blank bits is reserved. The typeset of the next word may shift forward considering the blank bits.



BYTE2-73: "00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 3F F8 00 3F FC 00 3F FC 00 38 1E 00 38 0E 00 38 0E 00 38 1E 00 3F FC 00 3F FC 00 3F FE 00 38 0F 00 38 07 00 38 07 00 38 0F 00 3F FE 00 3F FE 00 3F F8 00 00 00 00" is dot matrix data.。

5.5 Dot Matrix Font Address Table

No	Type	Ver.	Font	Char Set	Characters	Capacity	Base Address
1	ASCII		5X7 ASCII	ASCII	96	768	0
2			7X8 ASCII	ASCII	96	768	768
3			6X12 ASCII	ASCII	128	1,536	1,536
4			8X16 ASCII	ASCII	128	2,048	3,072
5			8X16 Bold ASCII	ASCII	96	1,536	5,120
6			12 dot proportional adjusted Arial	ASCII	96	2,496	6,656
7			12 dot proportional adjusted Times New Roman	ASCII	96	2,496	9,152
8			16 dot proportional adjusted Arial	ASCII	96	3,264	11,648
9			16 dot proportional adjusted Times New Roman	ASCII	96	3,264	14,912
10			24 dot proportional adjusted Arial	ASCII	96	7,104	18,176
11	UNICODE	5.0	8X16 Latin characters	Basic	496	7,936	25,280
12			8X16 Latin characters	Supplement			
13			8X16 Latin characters	Extended A			
14			8X16 Latin characters	Extended B			
15			8X16 Latin characters	Extended Additional			
16			8X16 Greek characters	Basic	96	1,536	33,216
17			8X16 Cyrillic characters	Basic	208	3,328	34,752
18			8X16 Hebrew characters	Basic	112	1,792	38,080
19			8X16 Thai characters	Basic	128	2,048	39,872
20		5.0	12X24 Latin characters	Basic	496	23,808	41,920
21			12X24 Latin characters	Supplement			
22			12X24 Latin characters	Extended A			
23			12X24 Latin characters	Extended B			
24			12X24 Latin characters	Extended Additional			
25			12X24 Greek characters	Basic	96	4,608	65,728
26			12X24 Hebrew characters	Basic	208	9,984	70,336
27		3.0	16 dot proportional adjusted Arabic	Basic+Supplement+Form B	576	19,584	80,320
28		5.0	16 dot proportional adjusted Latin	Basic	496	16,864	99,904
29			16 dot proportional adjusted Latin	Supplement			
30			16 dot proportional adjusted Latin	Extended A			
31			16 dot proportional adjusted Latin	Extended B			
32			16 dot proportional adjusted Latin	Extended Additional			
33			16 dot proportional adjusted Greek	Basic	96	3,264	116,768
34			16 dot proportional adjusted Cyrillic	Basic	208	7,072	120,032
35			12 dot proportional adjusted Latin	Basic	496	12,896	127,104
36			12 dot proportional adjusted Latin	Supplement			
37			12 dot proportional adjusted Latin	Extended A			

38			12 dot proportional adjusted Latin	Extended B			
39			12 dot proportional adjusted Latin	Extended Additional			
40			12 dot proportional adjusted Greek	Basic	96	2,496	140,000
41			12 dot proportional adjusted Cyrillic	Basic	208	5,408	142,496
42	**ISO8859**	**1998 -8X16**	ISO 8859-1 (8x16)	ISO8859	128	2,048	147,904
43			ISO 8859-2 (8x16)	ISO8859	128	2,048	149,952
44			ISO 8859-3 (8x16)	ISO8859	128	2,048	152,000
45			ISO 8859-4 (8x16)	ISO8859	128	2,048	154,048
46			ISO 8859-5 (8x16)	ISO8859	128	2,048	156,096
47			ISO 8859-7 (8x16)	ISO8859	128	2,048	158,144
48			ISO 8859-8 (8x16)	ISO8859	128	2,048	160,192
49			ISO 8859-9 (8x16)	ISO8859	128	2,048	162,240
50			ISO 8859-10 (8x16)	ISO8859	128	2,048	164,288
51			ISO 8859-11 (8x16)	ISO8859	128	2,048	166,336
52			ISO 8859-13 (8x16)	ISO8859	128	2,048	168,384
53			ISO 8859-14 (8x16)	ISO8859	128	2,048	170,432
54			ISO 8859-15 (8x16)	ISO8859	128	2,048	172,480
55			ISO 8859-16 (8x16)	ISO8859	128	2,048	174,528
56		**1998 -5X7**	ISO 8859-1 (5x7)	ISO8859	128	1,024	176,576
57			ISO 8859-2 (5x7)	ISO8859	128	1,024	177,600
58			ISO 8859-3 (5x7)	ISO8859	128	1,024	178,624
59			ISO 8859-4 (5x7)	ISO8859	128	1,024	179,648
60			ISO 8859-5 (5x7)	ISO8859	128	1,024	180,672
61			ISO 8859-7 (5x7)	ISO8859	128	1,024	181,696
62			ISO 8859-8 (5x7)	ISO8859	128	1,024	182,720
63			ISO 8859-9 (5x7)	ISO8859	128	1,024	183,744
64			ISO 8859-10 (5x7)	ISO8859	128	1,024	184,768
65			ISO 8859-11 (5x7)	ISO8859	128	1,024	185,792
66			ISO 8859-13 (5x7)	ISO8859	128	1,024	186,816
67			ISO 8859-14 (5x7)	ISO8859	128	1,024	187,840
68			ISO 8859-15 (5x7)	ISO8859	128	1,024	188,864
69			ISO 8859-16 (5x7)	ISO8859	128	1,024	189,888
70	**LCM Character**	**LCM**	LCM-1 (5x10)	SPLC780C-01 Compatible	256	2,560	190,912
71			LCM-2 (5x10)	SPLC780C-02 Compatible	256	2,560	193,472
72			LCM-3 (5x10)	SPLC780C-03 Compatible	256	2,560	196,032
73			LCM-8 (5x10)	SPLC780C-08 Compatible	256	2,560	198,592
74			LCM-11 (5x10)	SPLC780C-11 Compatible	256	2,560	201,152

75			LCM-12 (5x10)	SPLC780C-12 Compatible	256	2,560	203,712
76			LCM 13 (5x10)	SPLC780C-13 Compatible	256	2,560	206,272
77			LCM-0 (5x7)	7032 Compatible	256	2,048	208,832
78			LCM reserved area - 1 (5x10)		256	2,560	210,880
			LCM reserved area - 2 (5x10)		256	2,560	213,440
79	**UNICODE**	**3.0**	24 dot proportional adjusted Arabic	Basic+Supplement +Form B	576	42,624	216,000

5.6 Calculation of Character Address

With certain calculation method, the user may obtain certain character dots address using character code.

5.6.1 ASCII

5X7 ASCII

Parameter：

ASCIICode：ASCII character code（8bits）

Address：Address of ASCII character data in chip.

Calculation of character address：

if(ASCIICODE >=0x00 && ASCIICODE <=0xFF)

Address =ASCIICODE*8

7X8 ASCII

Parameter：

ASCIICode：ASCII character code（8bits）

Address：Address of ASCII character data in chip.

Calculation of character address：

if(ASCIICODE >=0x00 && ASCIICODE <=0xFF)

Address =ASCIICODE*8 + 768

6X12 ASCII

Parameter：

ASCIICode：ASCII character code（8bits）

Address：Address of ASCII character data in chip.

Calculation of character address：

if(ASCIICODE >=0x00 && ASCIICODE <=0xFF)

Address =(ASCIICODE-0x20) * 12 + 1536

8X16 ASCII

Parameter：

ASCIICode：ASCII character code（8bits）

Address：Address of ASCII character data in chip.

Calculation of character address：

if(ASCIICODE >=0x00 && ASCIICODE <=0xFF)

Address =(ASCIICODE-0x20) * 16 + 3072;

8X16 Bold ASCII

Parameter：

ASCIICode：ASCII character code（8bits）

Address：Address of ASCII character data in chip.

Calculation of character address：

if(ASCIICODE >=0x00 && ASCIICODE <=0xFF)

Address =(ASCIICODE-0x20) * 16 + 5120

12 dot proportional adjusted Arial ASCII

Parameter：

ASCIICode：ASCII character code（8bits）

Address：Address of ASCII character data in chip.

Calculation of character address：

if(ASCIICODE >=0x00 && ASCIICODE <=0xFF)

Address =(ASCIICODE-0x20) * 26 + 6656

12 dot proportional adjusted Times New Roman ASCII

Parameter：

ASCIICode：ASCII character code（8bits）

Address：Address of ASCII character data in chip.

Calculation of character address：

if(ASCIICODE >=0x00 && ASCIICODE <=0xFF)

Address =(ASCIICODE-0x20) * 26 + 9152

16 dot proportional adjusted Arial ASCII

Parameter：

ASCIICode：ASCII character code（8bits）

Address：Address of ASCII character data in chip.

Calculation of character address：

if(ASCIICODE >=0x00 && ASCIICODE <=0xFF)

Address =(ASCIICODE-0x20) * 34 + 11648

16 dot proportional adjusted Times New Roman ASCII

Parameter：

ASCIICode：ASCII character code（8bits）

Address：Address of ASCII character data in chip.

Calculation of character address：

if(ASCIICODE >=0x00 && ASCIICODE <=0xFF)

Address =(ASCIICODE-0x20) * 34 + 14912

24 dot proportional adjusted Arial ASCII

Parameter：

ASCIICode：ASCII character code（8bits）

Address：Address of ASCII character data in chip.

Calculation of character address：

if(ASCIICODE >=0x00 && ASCIICODE <=0xFF)

Address =(ASCIICODE-0x20) * 74 + 18176

5.6.2 Unicode

8X16 Latin characters

Parameters:

FontCode: Unicode code (16bits)

Address: Address of character data in chip

Calculation of character address:

```
if (FontCode>=0x0020 && FontCode<=0x007F)
      Address=(FontCode-0x0020) * 16 +25280;
Else if (FontCode>=0x00A0 && FontCode<=0x017F)
            Address=(FontCode-0x0040) * 16 +25280;
else if (FontCode>=0x01A0 && FontCode<=0x01CF)
            Address=(FontCode-0x01A0+320) * 16 +25280;
else if (FontCode>=0x01F0 && FontCode<=0x01FF)
            Address=(FontCode-0x01F0+368) * 16 +25280;
else if (FontCode>=0x0210 && FontCode<=0x021F)
            Address=(FontCode-0x0210+384) * 16 +25280;
else if (FontCode>=0x1EA0 && FontCode<=0x1EFF)
            Address=(FontCode-0x1EA0+400) * 16 +25280;
```

8X16 Greek characters

Parameters:

FontCode: Unicode code (16bits)

Address: Address of character data in chip

Calculation of character address:

```
if (FontCode>=0x0370 && FontCode<=0x03CF)
            Address=(FontCode-0x0370) * 16 +33216;
```

8X16 Cyrillic

Parameters:

FontCode: Unicode code (16bits)

Address: Address of character data in chip

Calculation of character address:

```
if (FontCode>=0x0400 && FontCode<=0x045F)
            Address=(FontCode-0x0400) * 16 +34752;
else if (FontCode>=0x0490 && FontCode<=0x04FF)
            Address=(FontCode-0x0490+96) * 16 +34752;
```

8X16 Hebrew characters

Parameters:

FontCode: Unicode code (16bits)

Address: Address of character data in chip

Calculation of character address:

```
if (FontCode>=0x0590 && FontCode<=0x05FF)
            Address=(FontCode-0x0590) * 16 +38080;
```

8X16 Thai characters

Parameters:

FontCode: Unicode code (16bits)

Address: Address of character data in chip

Calculation of character address:

if (FontCode>=0x0E00 && FontCode<=0x0E5F)

Address=(FontCode-0x0E00) * 16 +39872;

12X24 Latin characters

Parameters:

FontCode: Unicode code (16bits)

Address: Address of character data in chip

Calculation of character address:

if (FontCode>=0x0020 && FontCode<=0x007F)

Address=(FontCode-0x0020) * 48 +41920;

else if (FontCode>=0x00A0 && FontCode<=0x017F)

Address=(FontCode-0x00A0+96) * 48 +41920;

else if (FontCode>=0x01A0 && FontCode<=0x01CF)

Address=(FontCode-0x01A0+320) * 48 +41920;

else if (FontCode>=0x01F0 && FontCode<=0x01FF)

Address=(FontCode-0x01F0+368) * 48 +41920;

else if (FontCode>=0x0210 && FontCode<=0x021F)

Address=(FontCode-0x0210+384) * 48 +41920;

else if (FontCode>=0x1EA0 && FontCode<=0x1EFF)

Address=(FontCode-0x1EA0+400) * 48 +41920;

12X24 Greek characters

Parameters:

FontCode: Unicode code (16bits)

Address: Address of character data in chip

Calculation of character address:

if (FontCode>=0x0370 && FontCode<=0x03CF)

Address=(FontCode-0x0370) * 48 +65728;

12X24 Cyrillic

Parameters:

FontCode: Unicode code (16bits)

Address: Address of character data in chip

Calculation of character address:

if (FontCode>=0x0400 && FontCode<=0x045F)

Address=(FontCode-0x0400) * 48 +70336;

else if (FontCode>=0x0490 && FontCode<=0x04FF)

Address=(FontCode-0x0490+96) * 48 +70336;

16 proportional adjusted Arabic

Parameters:

FontCode: Unicode code (16bits)

Address: Address of character data in chip

Calculation of character address:

BaseAdd= 80320;

if(unicode_alb >= 0x0600 && unicode_alb <= 0x06FF)//alb 1

Address = 34*(unicode_alb-0x0600)+ BaseAdd;

else if(unicode_alb >= 0xfb50 && unicode_alb <= 0xfbff)//alb 2

Address = 34*(16*16+unicode_alb-0xfb50)+ BaseAdd;

```
else if( unicode_alb >= 0xfe70 && unicode_alb <= 0xfeff )//alb 3
    Address = 34*(16*11+16*16+unicode_alb-0xfe70)+ BaseAdd;
```

16 dot proportional adjusted Latin characters

Parameters:

FontCode: Unicode code (16bits)

Address: Address of character data in chip

Calculation of character address:

```
if (FontCode>=0x0020 && FontCode<=0x007F)
        Address=(FontCode-0x0020) * 34 +99904;
else if (FontCode>=0x00A0 && FontCode<=0x017F)
        Address=(FontCode-0x00A0+96) * 34 +99904;
else if (FontCode>=0x01A0 && FontCode<=0x01CF)
        Address=(FontCode-0x01A0+320) * 34 +99904;
else if (FontCode>=0x01F0 && FontCode<=0x01FF)
        Address=(FontCode-0x01F0+368) * 34 +99904;
else if (FontCode>=0x0210 && FontCode<=0x021F)
        Address=(FontCode-0x0210+384) * 34 +99904;
else if (FontCode>=0x1EA0 && FontCode<=0x1EFF)
        Address=(FontCode-0x1EA0+400) * 34 +99904;
```

16 dot propotional adjusted Greek

Parameters:

FontCode: Unicode code (16bits)

Address: Address of character data in chip

Calculation of character address:

```
if (FontCode>=0x0370 && FontCode<=0x03CF)
        Address=(FontCode-0x0370) * 34 +116768;
```

16 dot proportional adjusted Cyrillic

Parameters:

FontCode: Unicode code (16bits)

Address: Address of character data in chip

Calculation of character address:

```
if (FontCode>=0x0400 && FontCode<=0x045F)
        Address=(FontCode-0x0400) * 34 +120032;
else if (FontCode>=0x0490 && FontCode<=0x04FF)
        Address=(FontCode-0x0490+96) * 34 +120032;
```

12 proportional adjusted Latin characters

Parameters:

FontCode: Unicode code (16bits)

Address: Address of character data in chip

Calculation of character address:

```
if (FontCode>=0x0020 && FontCode<=0x007F)
        Address=(FontCode-0x0020) * 26 +127104;
else if (FontCode>=0x00A0 && FontCode<=0x017F)
        Address=(FontCode-0x00A0+96) * 26 +127104;
else if (FontCode>=0x01A0 && FontCode<=0x01CF)
        Address=(FontCode-0x01A0+320) * 26 +127104;
```

else if (FontCode>=0x01F0 && FontCode<=0x01FF)
Address=(FontCode-0x01F0+368) * 26 +127104;
else if (FontCode>=0x0210 && FontCode<=0x021F)
Address=(FontCode-0x0210+384) * 26 +127104;
else if (FontCode>=0x1EA0 && FontCode<=0x1EFF)
Address=(FontCode-0x1EA0+400) * 26 +127104;

12 dot proportional adjusted Greek

Parameters:

FontCode: Unicode code (16bits)

Address: Address of character data in chip

Calculation of character address:

if (FontCode>=0x0370 && FontCode<=0x03CF)
Address=(FontCode-0x0370) * 26 +140000;

12 dot proportional adjusted Cyrillic

Parameters:

FontCode: Unicode code (16bits)

Address: Address of character data in chip

Calculation of character address:

if (FontCode>=0x0400 && FontCode<=0x045F)
Address=(FontCode-0x0400) * 26 +142496;
else if (FontCode>=0x0490 && FontCode<=0x04FF)
Address=(FontCode-0x0490+96) * 26 +143496;

5.6.3 ISO8859

8*16 ISO8859-1

Parameters:

FontCode: Unicode code (16bits)

Address: Address of character data in chip

Calculation of character address:

if(FontCode >=0x0080 && FontCode <=0x00FF)
Address =(FONTCODE-0x80) * 16 + 147904;

8*16 ISO8859-2

Parameters:

FontCode: Unicode code (16bits)

Address: Address of character data in chip

Calculation of character address:

if(FONTCODE >=0x0080 && FONTCODE <=0x00FF)
Address =(FONTCODE-0x80) * 16 + 149952;

8*16 ISO8859-3

Parameters:

FontCode: Unicode code (16bits)

Address: Address of character data in chip

Calculation of character address:

if(FONTCODE >=0x0080 && FONTCODE <=0x00FF)

Address =(FONTCODE-0x80) * 16 + 152000;

8*16 ISO8859-4

Parameters:

FontCode: Unicode code (16bits)

Address: Address of character data in chip

Calculation of character address:

if(FONTCODE >=0x0080 && FONTCODE <=0x00FF)

Address =(FONTCODE-0x80) * 16 + 154048;

8*16 ISO8859-5

Parameters:

FontCode: Unicode code (16bits)

Address: Address of character data in chip

Calculation of character address:

if(FONTCODE >=0x0080 && FONTCODE <=0x00FF)

Address =(FONTCODE-0x80) * 16 + 156096;

8*16 ISO8859-7

Parameters:

FontCode: Unicode code (16bits)

Address: Address of character data in chip

Calculation of character address:

if(FONTCODE >=0x0080 && FONTCODE <=0x00FF)

Address =(FONTCODE-0x80) * 16 + 158144;

8*16 ISO8859-8

Parameters:

FontCode: Unicode code (16bits)

Address: Address of character data in chip

Calculation of character address:

if(FONTCODE >=0x0080 && FONTCODE <=0x00FF)

Address =(FONTCODE-0x80) * 16 + 160192;

8*16 ISO8859-9

Parameters:

FontCode: Unicode code (16bits)

Address: Address of character data in chip

Calculation of character address:

if(FONTCODE >=0x0080 && FONTCODE <=0x00FF)

Address =(FONTCODE-0x80) * 16 + 162240;

8*16 ISO8859-10

Parameters:

FontCode: Unicode code (16bits)

Address: Address of character data in chip

Calculation of character address:

if(FONTCODE >=0x0080 && FONTCODE <=0x00FF)

Address =(FONTCODE-0x80) * 16 + 164288;

8*16 ISO8859-11

Parameters:

FontCode: Unicode code (16bits)

Address: Address of character data in chip

Calculation of character address:

if(FONTCODE >=0x0080 && FONTCODE <=0x00FF)

Address =(FONTCODE-0x80) * 16 + 166336;

8*16 ISO8859-13

Parameters:

FontCode: Unicode code (16bits)

Address: Address of character data in chip

Calculation of character address:

if(FONTCODE >=0x0080 && FONTCODE <=0x00FF)

Address =(FONTCODE-0x80) * 16 + 168384;

8*16 ISO8859-14

Parameters:

FontCode: Unicode code (16bits)

Address: Address of character data in chip

Calculation of character address:

if(FONTCODE >=0x0080 && FONTCODE <=0x00FF)

Address =(FONTCODE-0x80) * 16 + 170432;

8*16 ISO8859-15

Parameters:

FontCode: Unicode code (16bits)

Address: Address of character data in chip

Calculation of character address:

if(FONTCODE >=0x0080 && FONTCODE <=0x00FF)

Address =(FONTCODE-0x80) * 16 + 172480;

8*16 ISO8859-16

Parameters:

FontCode: Unicode code (16bits)

Address: Address of character data in chip

Calculation of character address:

if(FONTCODE >=0x0080 && FONTCODE <=0x00FF)

Address =(FONTCODE-0x80) * 16 + 174528;

5*7 ISO8859-1

Parameters:

FontCode: Unicode code (16bits)

Address: Address of character data in chip

Calculation of character address:

if(FONTCODE >=0x0080 && FONTCODE <=0x00FF)

Address =(FONTCODE-0x80) * 8 + 176576;

5*7 ISO8859-2

Parameters:

FontCode: Unicode code (16bits)

Address: Address of character data in chip

Calculation of character address:

if(FONTCODE >=0x0080 && FONTCODE <=0x00FF)

Address =(FONTCODE-0x80) * 8 + 177600;

5*7 ISO8859-3

Parameters:

FontCode: Unicode code (16bits)

Address: Address of character data in chip

Calculation of character address:

if(FONTCODE >=0x0080 && FONTCODE <=0x00FF)

Address =(FONTCODE-0x80) * 8 + 178624;

5*7 ISO8859-4

Parameters:

FontCode: Unicode code (16bits)

Address: Address of character data in chip

Calculation of character address:

if(FONTCODE >=0x0080 && FONTCODE <=0x00FF)

Address =(FONTCODE-0x80) * 8 + 179648;

5*7 ISO8859-5

Parameters:

FontCode: Unicode code (16bits)

Address: Address of character data in chip

Calculation of character address:

if(FONTCODE >=0x0080 && FONTCODE <=0x00FF)

Address =(FONTCODE-0x80) * 8 + 180672;

5*7 ISO8859-7

Parameters:

FontCode: Unicode code (16bits)

Address: Address of character data in chip

Calculation of character address:

if(FONTCODE >=0x0080 && FONTCODE <=0x00FF)

Address =(FONTCODE-0x80) * 8 + 181696;

5*7 ISO8859-8

Parameters:

FontCode: Unicode code (16bits)

Address: Address of character data in chip

Calculation of character address:

if(FONTCODE >=0x0080 && FONTCODE <=0x00FF)

Address =(FONTCODE-0x80) * 8 + 182720;

5*7 ISO8859-9

Parameters:

FontCode: Unicode code (16bits)

Address: Address of character data in chip

Calculation of character address:

if(FONTCODE >=0x0080 && FONTCODE <=0x00FF)

Address =(FONTCODE-0x80) * 16 + 183744;

5*7 ISO8859-10

Parameters:

FontCode: Unicode code (16bits)

Address: Address of character data in chip

Calculation of character address:

if(FONTCODE >=0x0080 && FONTCODE <=0x00FF)

Address =(FONTCODE-0x80) * 8 + 184768;

5*7 ISO8859-11

Parameters:

FontCode: Unicode code (16bits)

Address: Address of character data in chip

Calculation of character address:

if(FONTCODE >=0x0080 && FONTCODE <=0x00FF)

Address =(FONTCODE-0x80) * 8 + 185792;

5*7 ISO8859-13

Parameters:

FontCode: Unicode code (16bits)

Address: Address of character data in chip

Calculation of character address:

if(FONTCODE >=0x0080 && FONTCODE <=0x00FF)

Address =(FONTCODE-0x80) * 8 + 186816;

5*7 ISO8859-14

Parameters:

FontCode: Unicode code (16bits)

Address: Address of character data in chip

Calculation of character address:

if(FONTCODE >=0x0080 && FONTCODE <=0x00FF)

Address =(FONTCODE-0x80) * 8 + 187840;

5*7 ISO8859-15

Parameters:

FontCode: Unicode code (16bits)

Address: Address of character data in chip

Calculation of character address:

if(FONTCODE >=0x0080 && FONTCODE <=0x00FF)

Address =(FONTCODE-0x80) * 8 + 188864;

5*7 ISO8859-16

Parameters:

FontCode: Unicode code (16bits)

Address: Address of character data in chip

Calculation of character address:

if(FONTCODE >=0x0080 && FONTCODE <=0x00FF)

Address =(FONTCODE-0x80) * 8 + 189888;

5*10 LCM-1

Parameters:

FontCode: Unicode code (16bits)

Address: Address of character data in chip

Calculation of character address:

if (FontCode>=0x0000 && FontCode<=0x00FF)

Address=FontCode * 10 +190912;

5*10 LCM-2

Parameters:

FontCode: Unicode code (16bits)

Address: Address of character data in chip

Calculation of character address:

if (FontCode>=0x0000 && FontCode<=0x00FF)

Address=FontCode * 10 +193472;

5*10 LCM-3

Parameters:

FontCode: Unicode code (16bits)

Address: Address of character data in chip

Calculation of character address:

if (FontCode>=0x0000 && FontCode<=0x00FF)

Address=FontCode * 10 +196032;

5*10 LCM-8

Parameters:

FontCode: Unicode code (16bits)

Address: Address of character data in chip

Calculation of character address:

if (FontCode>=0x0000 && FontCode<=0x00FF)

Address=FontCode * 10 +198592;

5*10 LCM-11

Parameters:

FontCode: Unicode code (16bits)

Address: Address of character data in chip

Calculation of character address:

if (FontCode>=0x0000 && FontCode<=0x00FF)

Address=FontCode * 10 +201152;

5*10 LCM-12

Parameters:

FontCode: Unicode code (16bits)

Address: Address of character data in chip

Calculation of character address:

if (FontCode>=0x0000 && FontCode<=0x00FF)

Address=FontCode * 10 +203712;

5*10 LCM-13

Parameters:

FontCode: Unicode code (16bits)

Address: Address of character data in chip

Calculation of character address:

if (FontCode>=0x0000 && FontCode<=0x00FF)

Address=FontCode * 10 +206272;

5*10 LCM-0

Parameters:

FontCode: Unicode code (16bits)

Address: Address of character data in chip

Calculation of character address:

```
if (FontCode>=0x0000 && FontCode<=0x00FF)
        Address=FontCode * 10 +208832;
```

24 dot proportional adjusted Arabic

Parameters:

FontCode: Unicode code (16bits)

Address: Address of character data in chip

Calculation of character address:

```
BaseAdd= 216000;
if( unicode_alb >= 0x0600 && unicode_alb <= 0x06FF )//alb 1
    Address = 74 *(unicode_alb-0x0600)+ BaseAdd;
else if( unicode_alb >= 0xfb50 && unicode_alb <= 0xfbff )//alb 2
    Address = 74 *(16*16+unicode_alb-0xfb50)+ BaseAdd;
else if( unicode_alb >= 0xfe70 && unicode_alb <= 0xfeff )//alb 3
    Address = 74 *(16*11+16*16+unicode_alb-0xfe70)+ BaseAdd;
```

6 Appendix

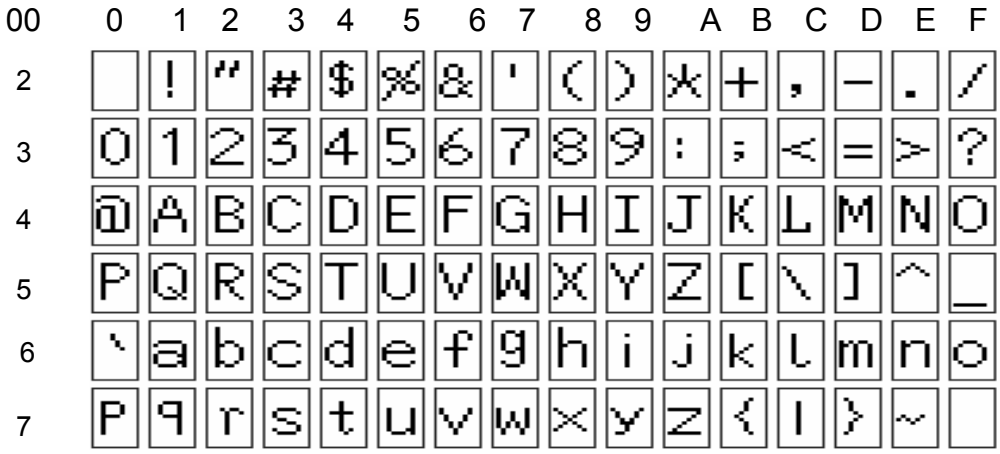
6.1 Unicode Section

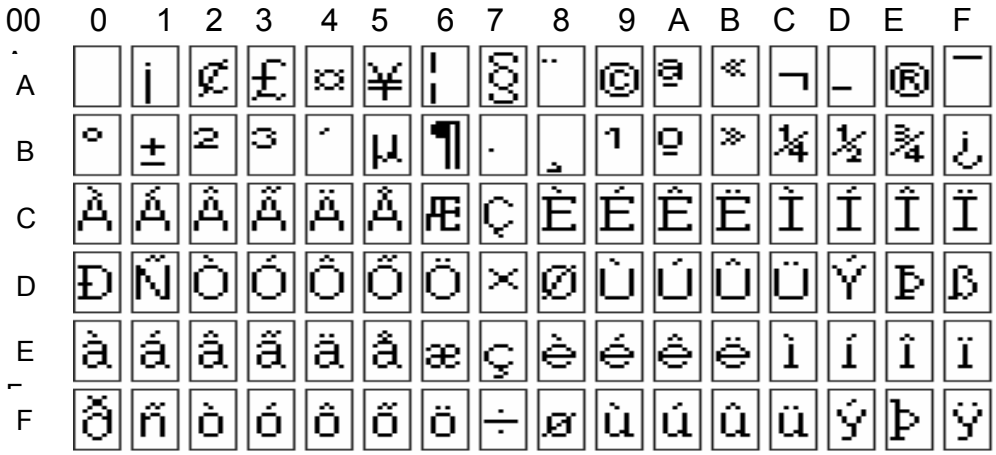
Unicode section include Latin, Greek, Cyrillic, Hebrew, Thai, and Arabic.

6.1.1 Latin (496 characters)

Scope of code: 0x20~0x70、0xA0~0xFF、0x0100~0x0170、0x01A0~0x01CF、0x01F0~0x01FF、0x0210~0x021F、0x1EA0~0x1EFF、0x1EA0~0x1EFF, total 496 characters.

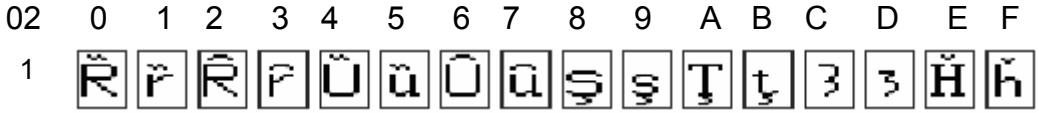
Note：The scope of code for “Latin Basic” equal to ASCII code, which is 0020~007E, therefore not listed in Unicode-Latin Section.






01 0 1 2 3 4 5 6 7 8 9 A B C D E F
0
1
2
3
4
5
6
7

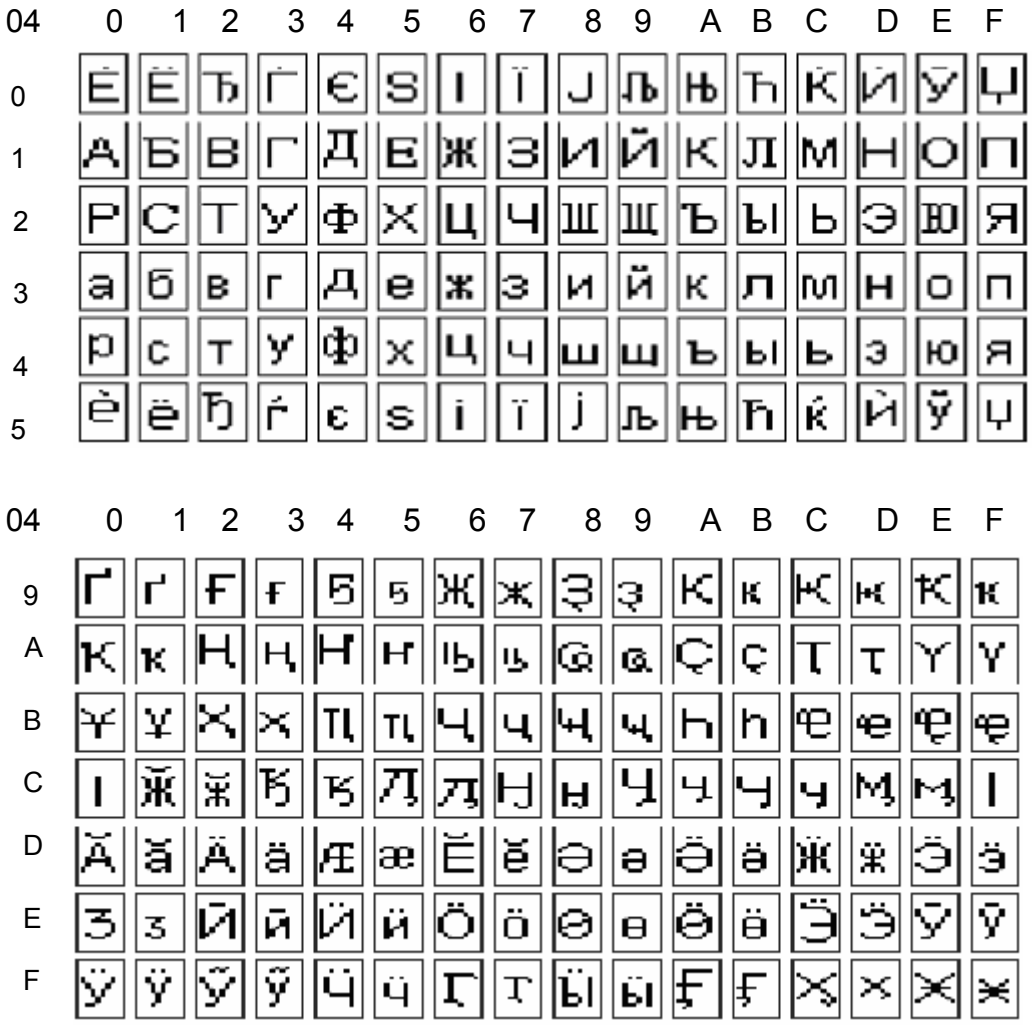

01 0 1 2 3 4 5 6 7 8 9 A B C D E F
A
B
C
F


02 0 1 2 3 4 5 6 7 8 9 A B C D E F
1


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

6.1.2 Cyrillic (208 characters)

Scope of code: 0x0400~0x045F、0x0490~0x04FF, total of 208 characters.



6.1.3 Greek (96 characters)

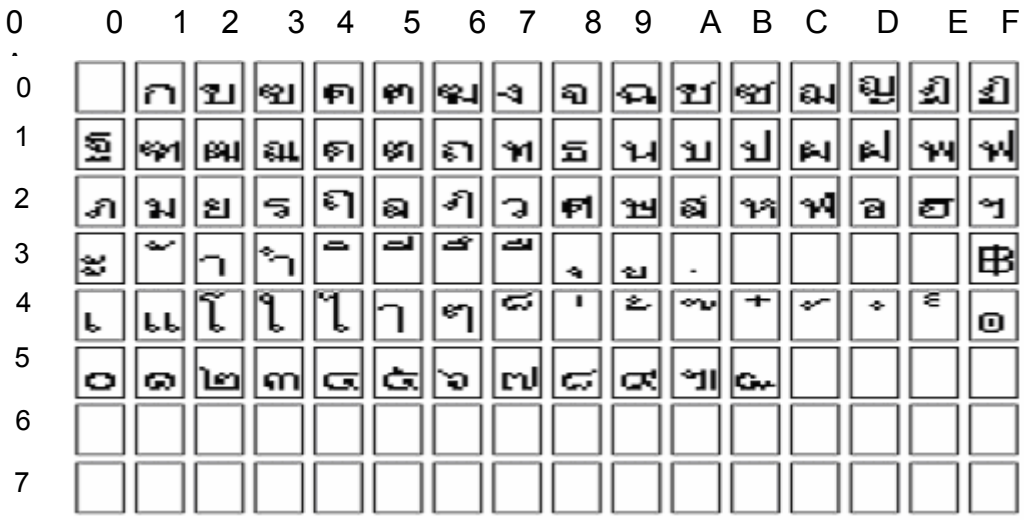
Scope of code: 0x0370~0x03CF, total of 96 characters.

6.1.4 Hebrew (112 characters)

Scope of code: 0x0590~0x05FF, total of 112 characters.

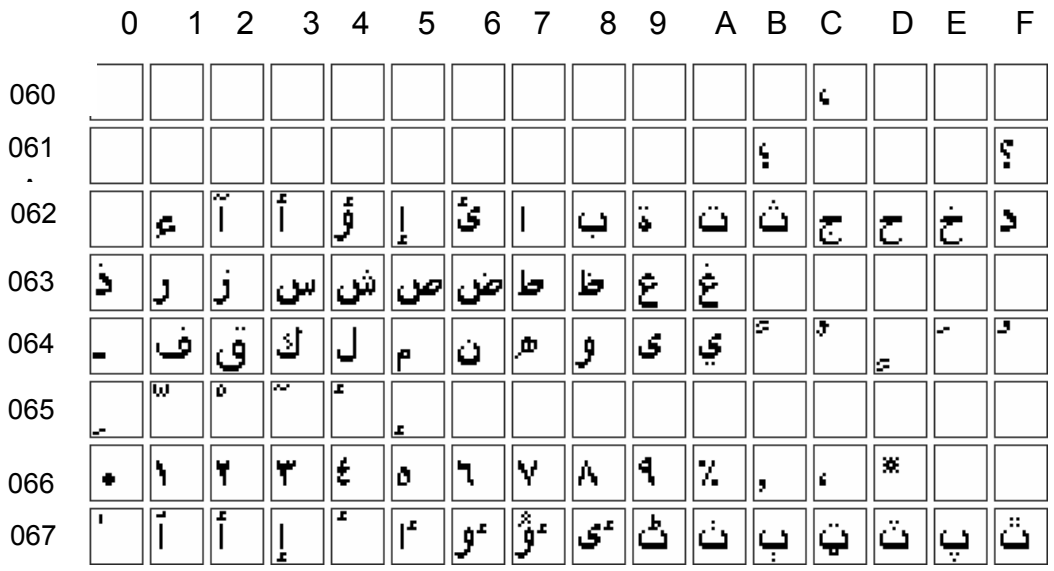
6.1.5 Thai (128 characters)

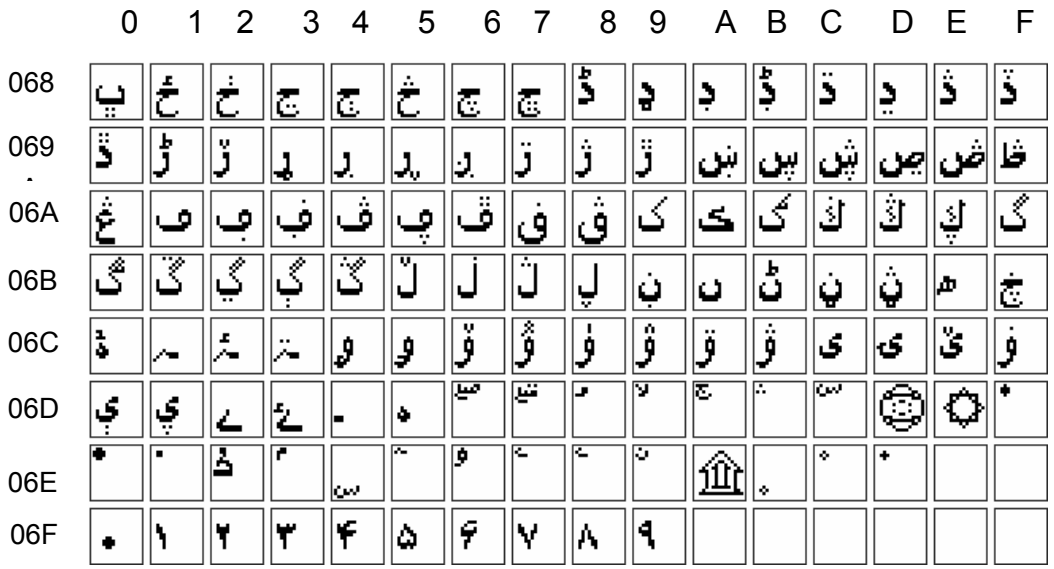
Code of Scope: 0x0E00~0x0E7F, total of 128 characters.

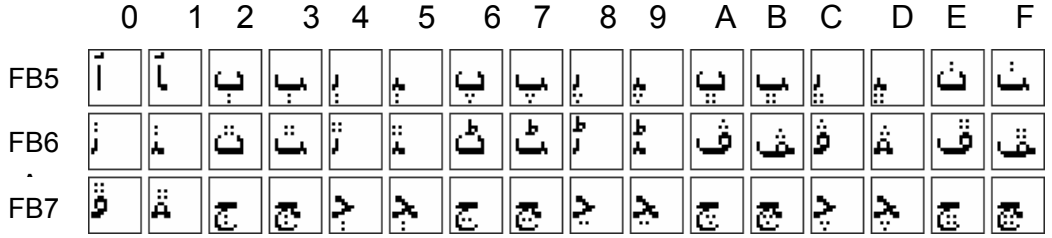


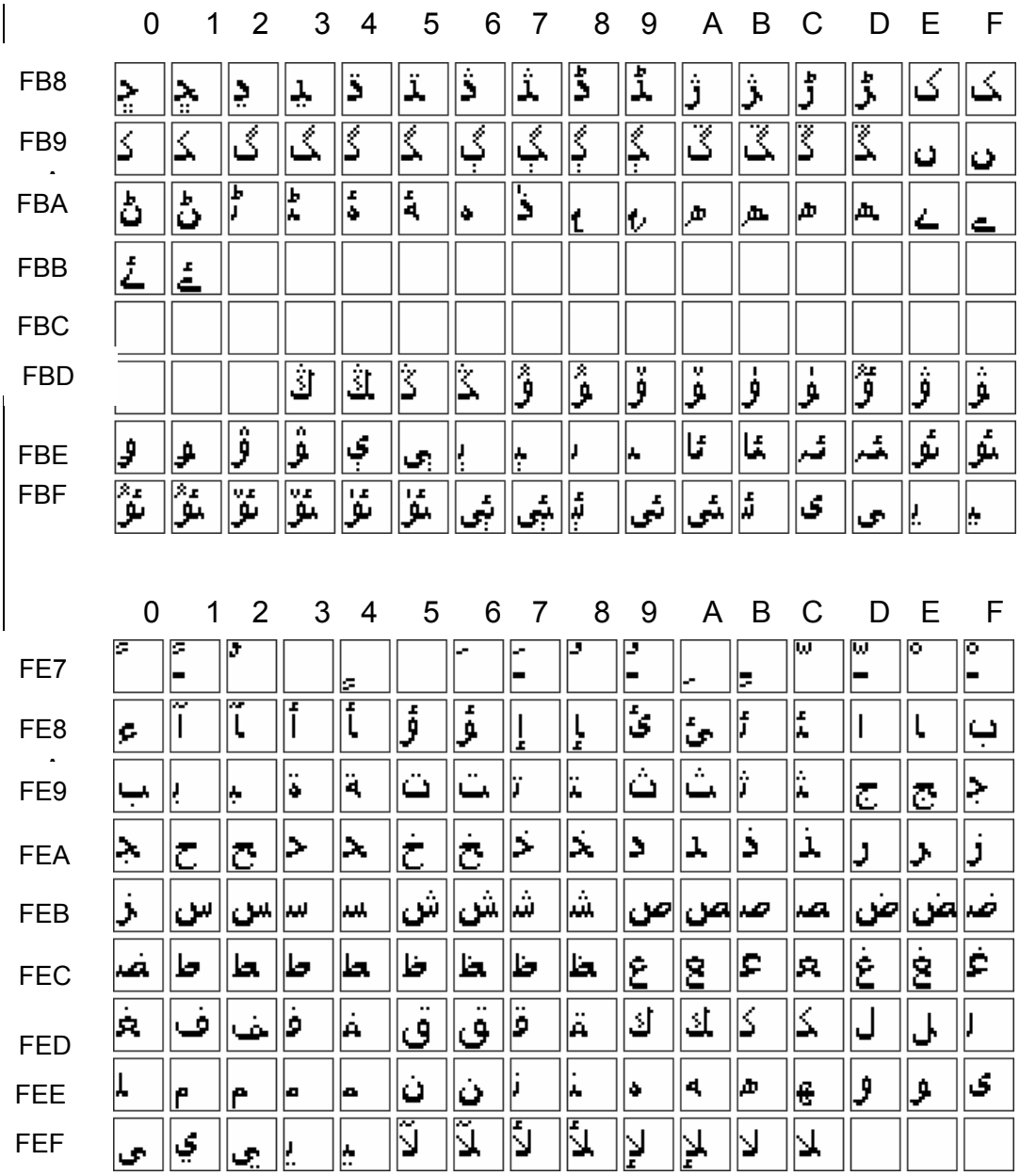
6.1.6 Arabic (576 characters)

Scope of code: 0x0600~0x06FF、0xFB50~0xFBFF、0xFE70~0xFEFF, total of 576 characters





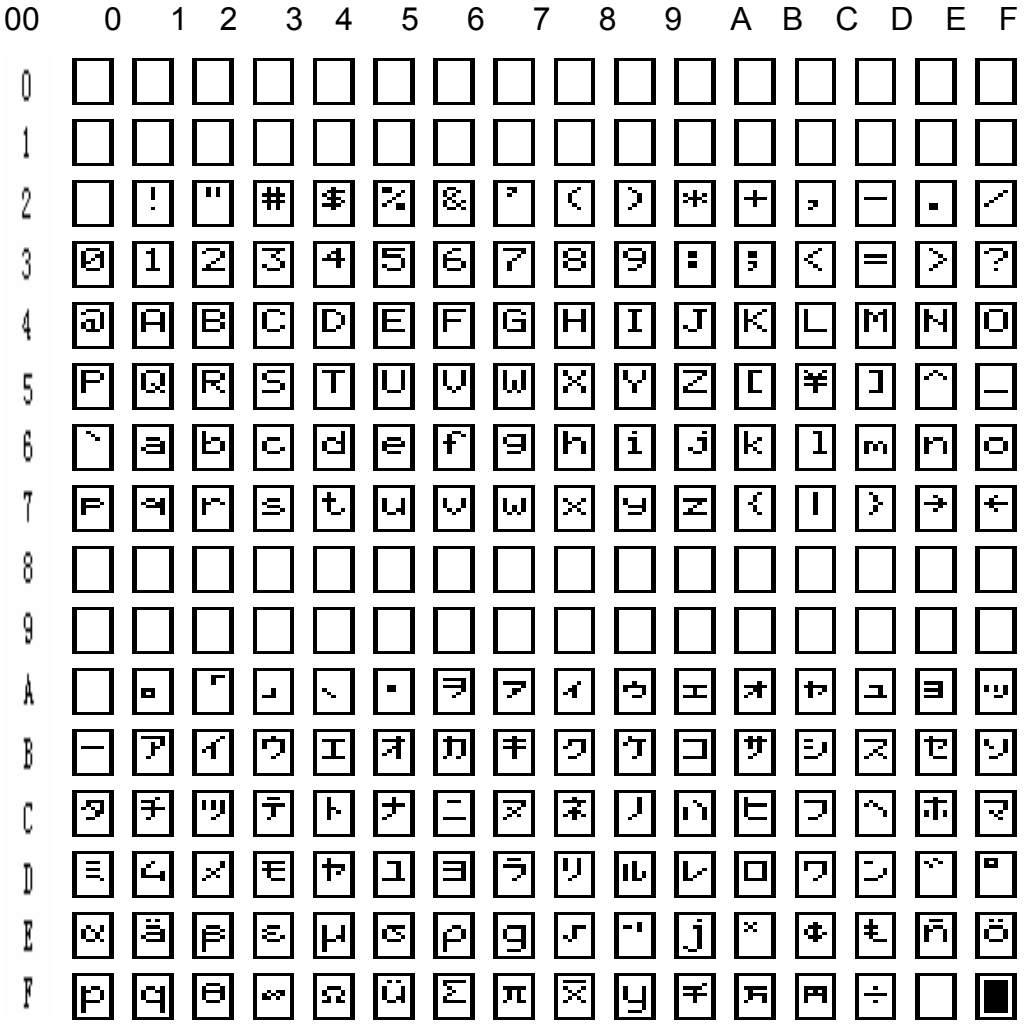



0 1 2 3 4 5 6 7 8 9 A B C D E F
FB8
FB9
FBA
FBB
FBC
FBD
FBE
FBF
0 1 2 3 4 5 6 7 8 9 A B C D E F
FE7
FE8
FE9
FEA
FEB
FEC
FED
FEE
FEF

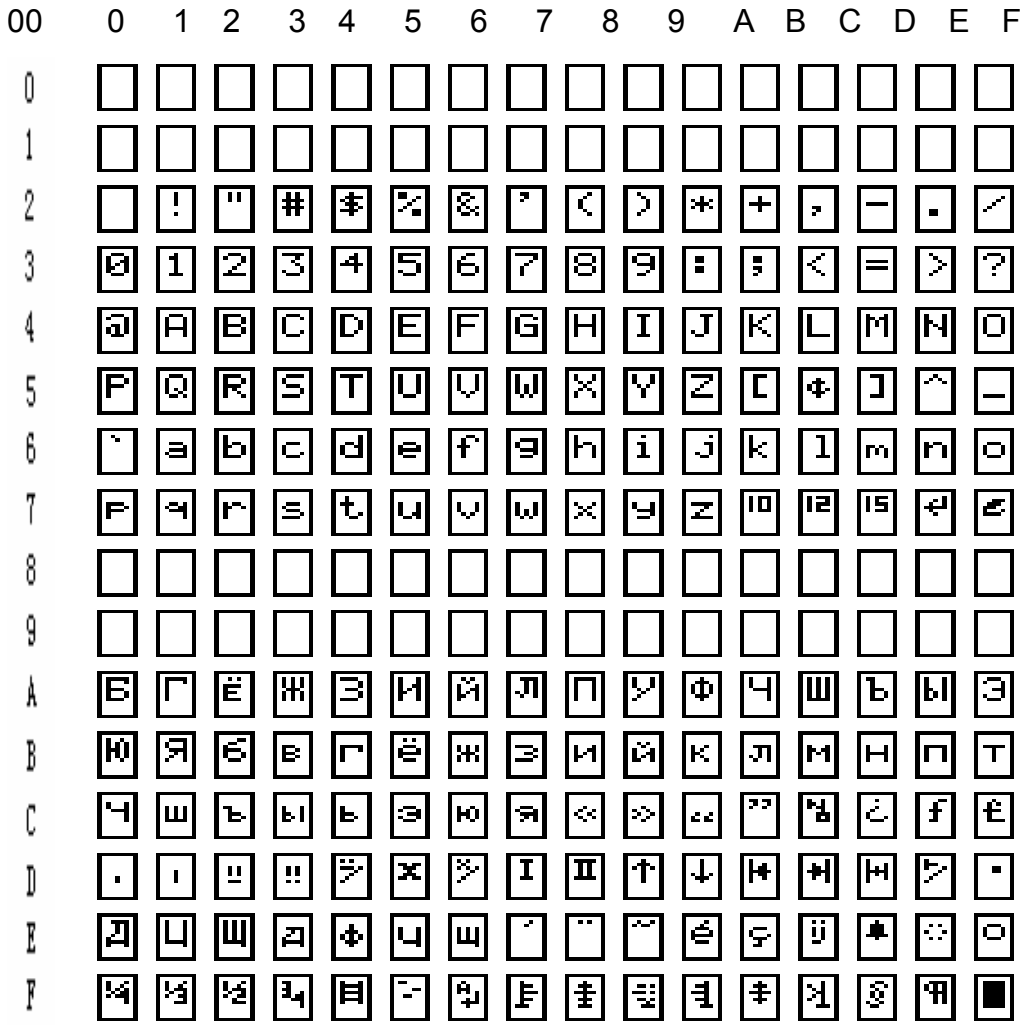
6.2 LCM Characters- Including LCM 5X10、LCM5X7

6.2.1 **LCM 5X10**

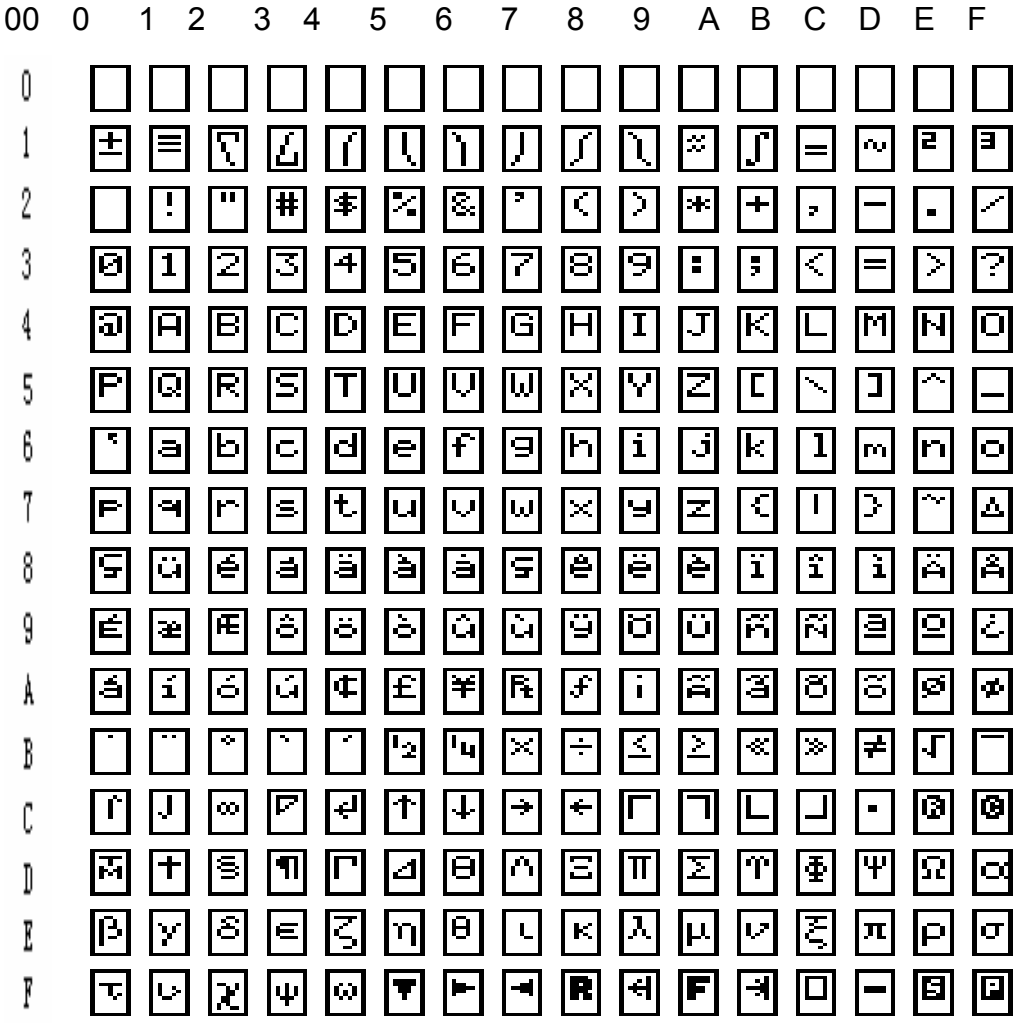
LCM5X10-1



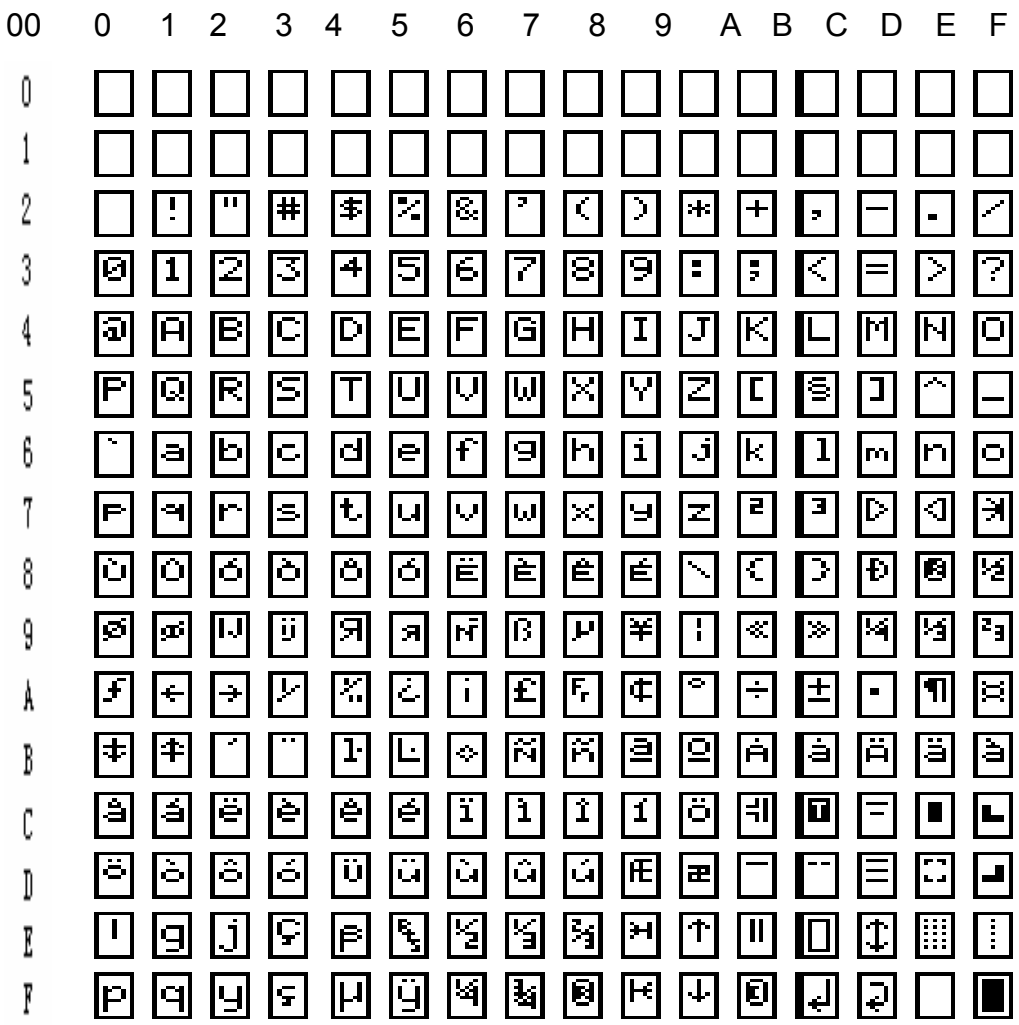
LCM5X10-2



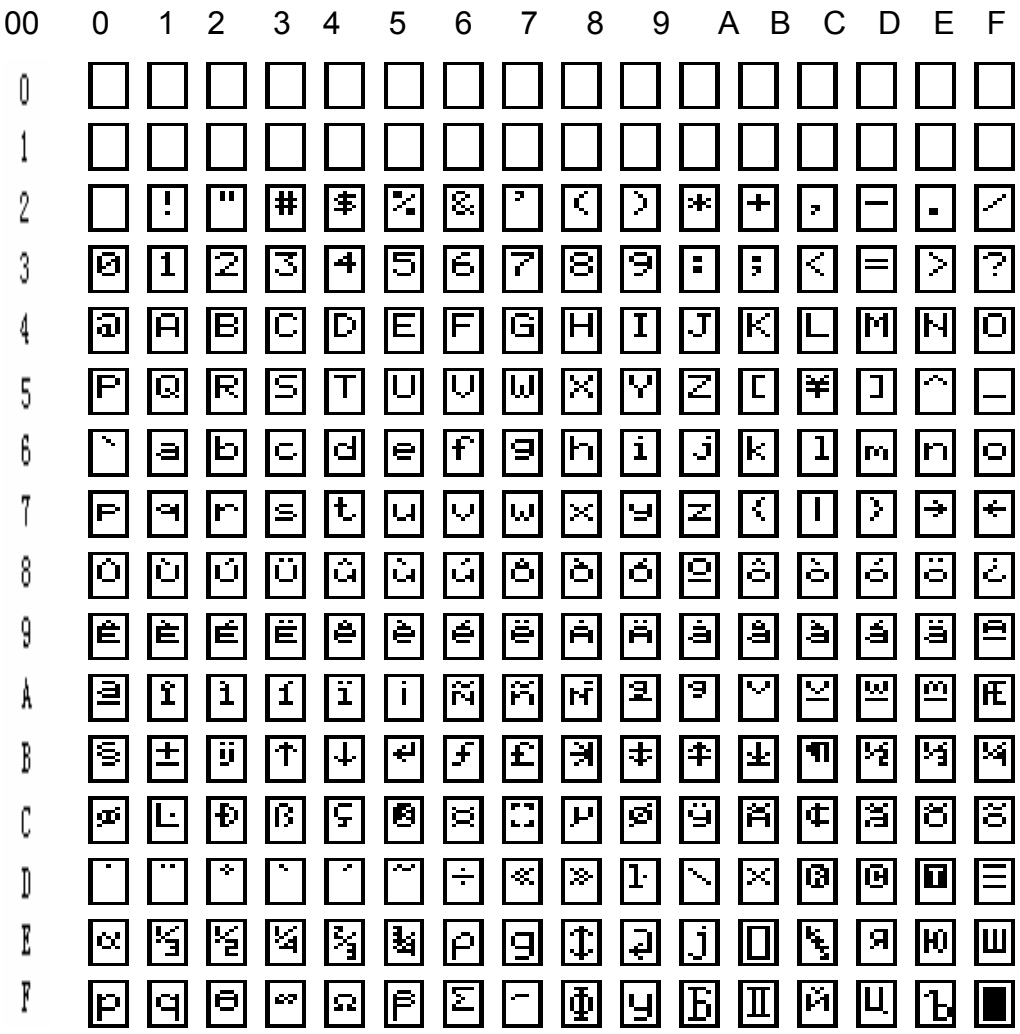
LCM5X10-3



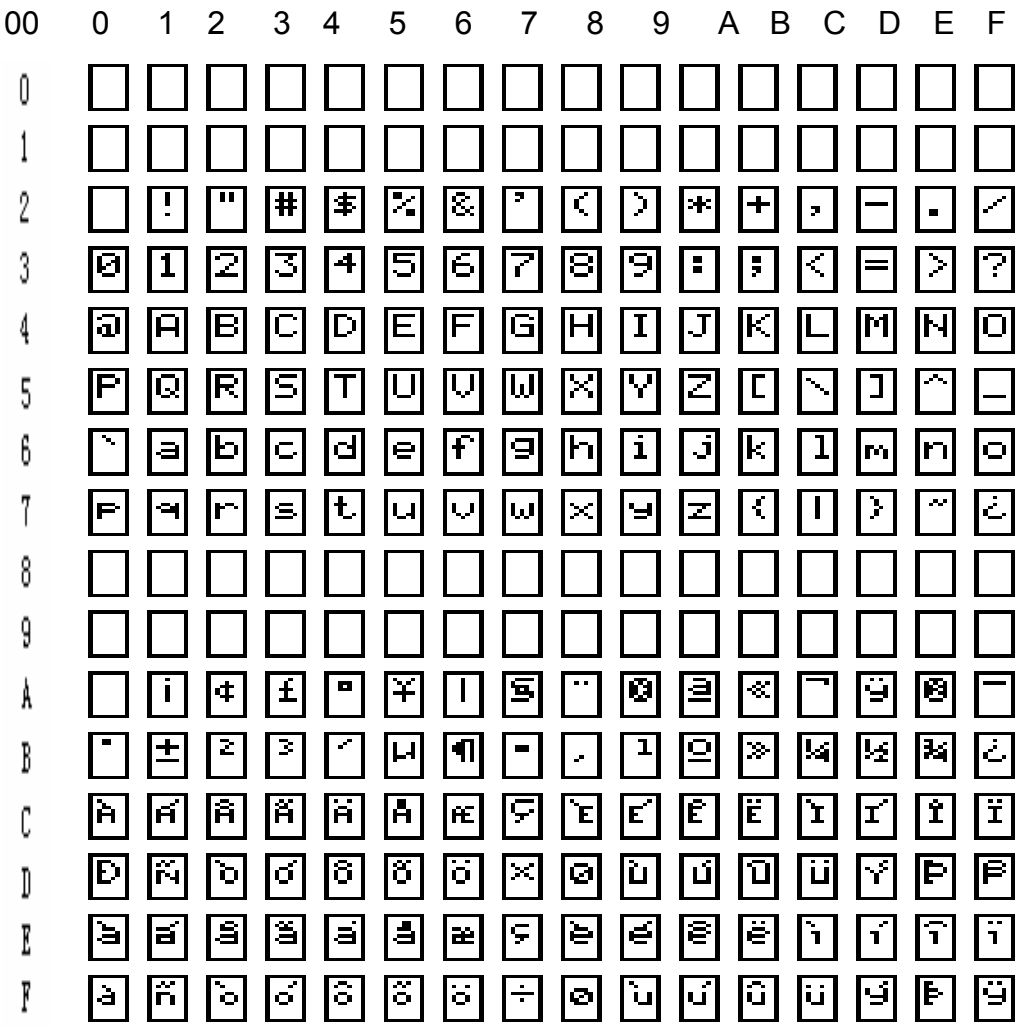
LCM5X10-8



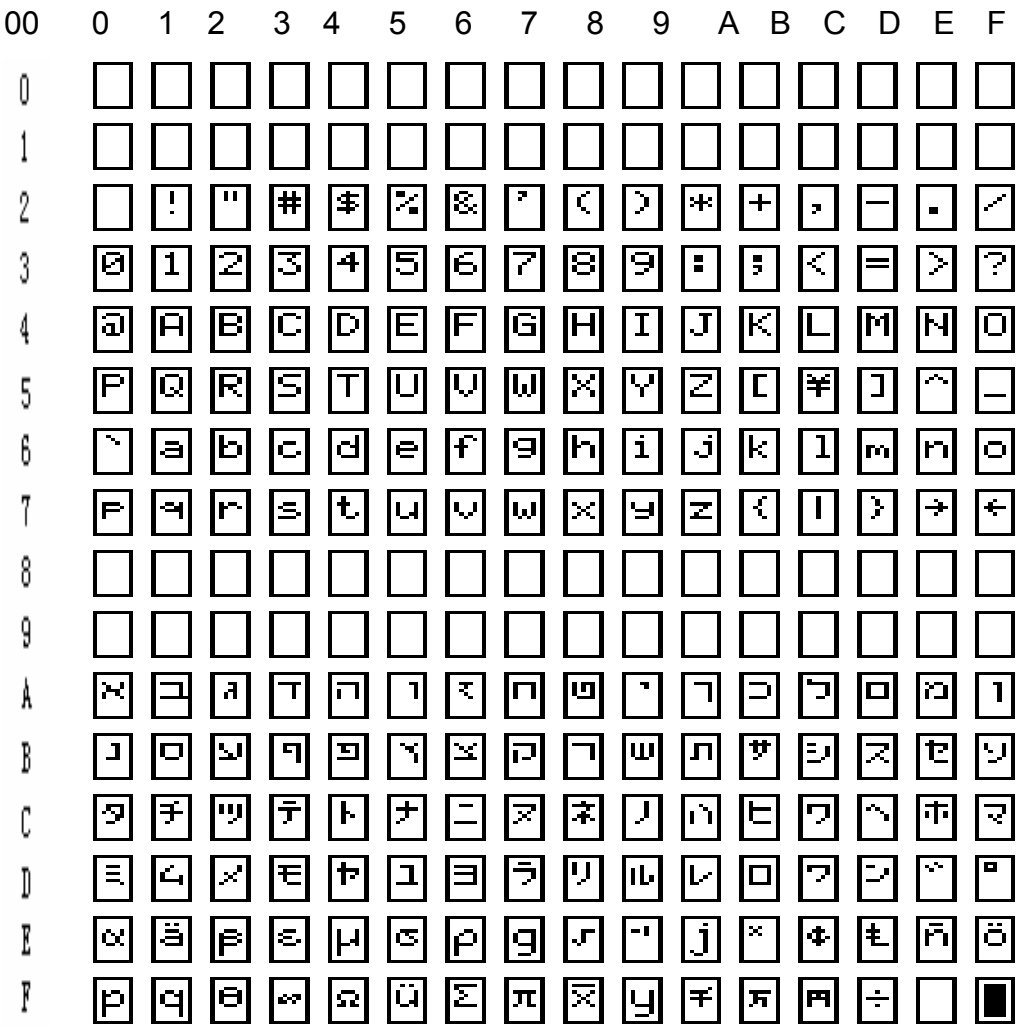
LCM5X10-11



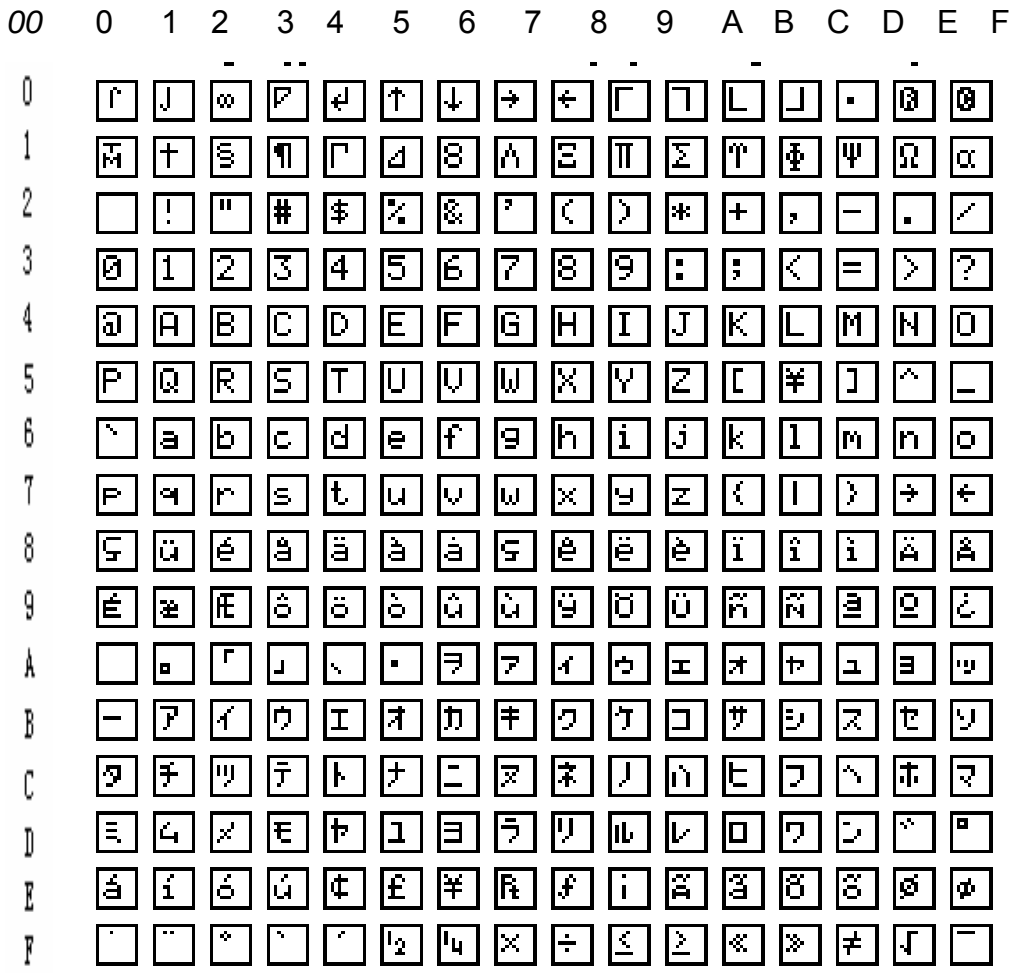
LCM5X10-12



LCM5X10-13



6.2.2 LCM 5X7（7032 Compatibel）Character Set



6.3 173 Countries Language Check List

173 Coutries Language Check List

FOR GT20L24F6Y UNICODE&ISO8859

Language Family	Area	No.	Country	Language	ISO-8859
Latin (English)	European	1	Britain or United Kingdom	English	ISO8859-1
		2	Ireland		
	North America	3	USA	English	ISO8859-1
		4	Canada	English、French	ISO8859-1
		5	Belize	English	ISO8859-1
		6	Jamaica		
		7	Trinidad and Tobago		
		8	Bahamas		
		9	Antigua and Barbuda		
		10	Dominica		
		11	St.Vincent		
		12	St.Lucia		
		13	Grenada		
		14	St.Kitts-Nevis		
	South Africa	15	Guyana	English	ISO8859-1
	Australia	16	Australia	English	ISO8859-1
		17	New Zealand		
		18	Tonga		
		19	Fiji		
		20	Palau		
		21	Solomon		
		22	Vanuatu		
		23	Kiribati		
		24	Nauru		
		25	Marshall Islands		
	Africa	26	South Africa	English、South Africa Dutch	ISO8859-1
		27	Zimbabwe	English	ISO8859-1
		28	Gambia		
		29	Sierra Leone		
		30	Liberia		
		31	Ghana		
		32	Nigeria		
		33	Uganda		
		34	Zambia		

		35	Malawi		
		36	Seychelles		
		37	Mauritius		
		38	Botswana		
		39	Namibia		
		40	Lesotho		
Latin（French）	Europe	41	France	French	ISO8859-15
		42	Belgium	French、Dutch	ISO8859-15
		43	Monaco	French、Italian	ISO8859-15
	North America	44	Haiti	French	ISO8859-15
	Africa	45	Senegal	French	ISO8859-15
		46	Mali		
		47	Burkina Faso		
		48	Guinea		
		49	cote dlvoire		
		50	Togo		
		51	Benin		
		52	Niger		
		53	Cameroon		
		54	Chad		
		55	Central African Republic		
Latin（French）	Africa	56	Djibouti	French	ISO8859-15
		57	Burundi		
		58	Republic of Democratic Congo		
		59	Congo		
		60	Gabon		
		61	Comoros		
		62	Madagascar		
Latin（Spanish）	Europe	63	Spain	Spanish、Catalan	ISO8859-1、-15
		64	Andorra	Spanish	ISO8859-1、-15
	North America	65	Mexico	Spanish	ISO8859-1 ISO8859-15
		66	Guatemala		
		67	Costa Rica		
		68	Panama		
		69	Dominican Republic		
		70	El Salvador		
		71	Honduras		
		72	Nicaragua		
		73	Puerto Rico		
		74	Cuba		
	South Africa	75	Venezuela	Spanish	ISO8859-1 ISO8859-15
		76	Colombia		

		77	Peru		
		78	Argentina		
		79	Ecuador		
		80	Chile		
		81	Uruguay		
		82	Paraguay		
		83	Bolivia		
	Africa	84	Equatorial New Guinea	Spanish	ISO8859-1 ISO8859-15
		85	Ceuta and Melilla		
Latin（Portuguese）	Europe	86	Portugal	Portuguese	ISO8859-1 ISO8859-15
	South Africa	87	Brazil		
	Africa	88	Cape Verde		
		89	Guinea-Bissau		
		90	Sao Tome and Principe		
		91	Angola		
		92	Mozambique		
Latin（German）	Europe	93	Germany	German	ISO8859-1、-15
		94	Switzerland	German、French	ISO8859-1、-15
		95	Austria	German	ISO8859-1、-15
		96	Luxembourg	German、French	ISO8859-1、-15
		97	Liechtenstein	German	ISO8859-1、-15
Latin（Dutch）	Europe	98	Holland	Dutch	ISO8859-1 ISO8859-15
	South Africa	99	Surinam		
Latin（Nordic Europe）	Europe	100	Denmark	Dannish	ISO8859-1、-10
		101	Norway	Norwegian	ISO8859-1、-10
		102	Sweden	Swedish	ISO8859-1、-10
		103	Faroes，The	Faroese	ISO8859-1、-10
		104	Greenland	Greelandic	ISO8859-1、-10
		105	Iceland	Icelandic	ISO8859-1、-10
		106	Finland	Finnish、Swedish	ISO8859-13、-15
		107	Estonia	Estonian	ISO8859-4、-13
		108	Latvia	Latvian	ISO8859-4、-13
		109	Lithuania	Lithuanian	ISO8859-4、-13
Latin（Central Europe）	Europe	110	Czech	Czech	ISO8859-2
		111	Slovakia	Slovak	ISO8859-2
		112	Poland	Polish	ISO8859-2、-16
		113	Hungary	Hungarian	ISO8859-2、-16
		114	Romania	Romannian	ISO8859-16
	Europe	115	Slovenia	Slovenian	ISO8859-2、-16
		116	Croatia	Crotian	ISO8859-2、-16

Latin（Southern Europe）	Europe	117	Italy	Italian	ISO8859-1 ISO8859-16
		118	San Marino		
		119	Vatican		
		120	Turkey	Turkish	ISO8859-9
		121	Malta	Maltese	ISO8859-3、-9
		122	Albania	Albanian	ISO8859-1、-16
Latin（Southeast Asia）	Asia	123	Vietnam	Vietnamese	ISO8859-1
		124	Malaysia	Malaysian	ISO8859-1
		125	Brunei		
		126	Indonesia	Indonesian	ISO8859-1
		127	East Timor		
		128	Philippines，The	English, Tagalog	ISO8859-1
Latin（Africa）	Africa	129	Kenya	Kiswahili	ISO8859-1
		130	Tanzania		
Cyrillic（Eastern Europe）	Europe	131	Russia	Rusian	ISO8859-5
		132	Byelorussia or Belarus		
		133	Ukraine	RussianUkrainian	ISO8859-5
		134	Bulgaria	Bulgarian	ISO8859-5
		135	Moldova	Russian	ISO8859-5
		136	F.R.Yugoslavia	Serbian	ISO8859-5
		137	Barbados	Serbian	ISO8859-5
		138	Macedonia	Macedonian	ISO8859-5
Cyrillic（Asia）	Asia	139	Azerbaijan	Azeri	ISO8859-5
		140	Kirghizstan	Kyrgyz	ISO8859-5
		141	Tajikistan	Tajik	ISO8859-5
		142	Turkmenistan	Turkmen	ISO8859-5
		143	Uzbekistan	Uzbek	ISO8859-5
		144	Kazakhstan	Kazakh	ISO8859-5
		145	Mongolia	Mongolian	ISO8859-5
Greek	Asia	146	Greece	Greek	ISO8859-7
		147	Cyprus		
Arabic（Africa）	Africa	148	Egypt	Arabic	ISO8859-6
		149	Tunisia		
		150	Libya		
		151	Morocco		
		152	Algeria		
		153	Sudan，The		
		154	Somalia		
		155	Djibouti		
		156	Mauritania		
Arabic	Asia	157	Syria	Arabic	ISO8859-6

（Asia）		158	United Arab Emirates，The		
		159	Lebanon		
		160	Yemen		
		161	Kuwait		
		162	Qatar		
		163	Bahrain		
		164	Oman		
		165	Jordan		
		166	Iraq		
		167	Saudi Arabia		
		168	Palestine		
		169	Iran	Farsi	
		170	Pakistan	Urdu、Arabic	
		171	Afghanistan	Pashto	
Hebrew	Asia	172	Israel	Hebrew	ISO8859-8
Thai	Asia	173	Thailand	Thai	ISO8859-11

6.4 173 Countries Checklist (Alpabet Order)

173 Countries Checklist (Alpabet Order)

	Country			Country			Country	
A	Afghanistan	171	G	Gambia	28	O	Oman	164
	Albania	122		Germany	93	P	Palau	20
	Algeria	152		Ghana	31		Pakistan	170
	Andorra	64		Greece	146		Palestine	168
	Angola	91		Greenland	104		Panama	68
	Antigua and Barbuda	9		Grenada	13		Paraguay	82
	Argentina	78		Guatemala	66		Peru	77
	Australia	16		Guinea	48		Philippines，The	128
	Austria	95		Guinea-Bissau	89		Poland	112
	Azerbaijan	139		Guyana	15		Portugal	86
B	Bahamas	8	H	Haiti	44		Puerto Rico	73
	Bahrain	163		Holland	98	Q	Qatar	162
	Barbados	137		Honduras	71	R	Republic of Democratic Congo	58
	Belgium	42		Hungary	113		Romania	114
	Belize	5	I	Iceland	105		Russia	131
	Benin	51		Indonesia	126	S	San Marino	118
	Bolivia	83		Iran	169		Sao Tome and Principe	90
	Botswana	38		Iraq	166		Saudi Arabia	167
	Brazil	87		Israel	172		Senegal	45
	Britain 或 United Kingdom	1		Italy	117		Seychelles	36
	Brunei	125	J	Jamaica	6		Sierra Leone	29
	Bulgaria	134		Jordan	165		Slovakia	111
	Burkina Faso	47	K	Kazakhstan	144		Slovenia	115
	Burundi	57		Kenya	129		Solomon	21
	Byelorussia or Belarus	132		Kirghizstan	140		Somalia	154
C	Cameroon	53		Kiribati	23		South Africa	26
	Canada	4		Kuwait	161		Spain	63
	Cape Verde	88	L	Latvia	108		St.Kitts-Nevis	14
	Central African Republic	55		Lebanon	159		St.Lucia	12
	Ceuta and Melilla	85		Lesotho	40		St.Vincent	11

	Chad	54		Liberia	30		Sudan，The	153
	Chile	80		Libya	150		Surinam	99
	Colombia	76		Liechtenstein	97		Sweden	102
	Comoros	61		Lithuania	109		Switzerland	94
	Congo	59		Ireland	2		Syria	157
	Costa Rica	67		Luxembourg	96	T	Tajikistan	141
	cote dlvoire	49	M	Macedonia	138		Tanzania	130
	Croatia	116		Madagascar	62		Thailand	173
	Cuba	74		Malawi	35		Togo	50
	Cyprus	147		Malaysia	124		Tonga	18
	Czech	110		Mali	46		Trinidad and Tobago	7
D	Denmark	100		Malta	121		Tunisia	149
	Djibouti	56		Marshall Islands	25		Turkey	120
	Djibouti	155		Mauritania	156		Turkmenistan	142
	Dominica	10		Mauritius	37	U	Uganda	33
	Dominican Republic	69		Mexico	65		Ukraine	133
E	East Timor	127		Moldova	135		United Arab Emirates，The	158
	Ecuador	79		Monaco	43		Uruguay	81
	Egypt	148		Mongolia	145		USA	3
	El Salvador	70		Morocco	151		Uzbekistan	143
	Equatorial New Guinea	84		Mozambique	92	V	Vanuatu	22
	Estonia	107	N	Namibia	39		Vatican	119
F	F.R.Yugoslavia	136		Nauru	24		Venezuela	75
	Faroes，The	103		New Zealand	17		Vietnam	123
	Fiji	19		Nicaragua	72	Y	Yemen	160
	Finland	106		Niger	52	Z	Zambia	34
	France	41		Nigeria	32		Zimbabwe	27
G	Gabon	60		Norway	101			

6.5 ISO/IEC 8859

6.5.1 ISO 8859-1

Latin-1 Western European

Perhaps the most widely used part of ISO/IEC 8859, covering most Western European languages: Danish (partial), Dutch (partial), English, Faeroese, Finnish (partial), French (partial), German, Icelandic, Irish, Italian, Norwegian, Portuguese, Rhaeto-Romanic, Scottish Gaelic, Spanish, and Swedish. Languages from other parts of the world are also covered, including: Eastern European Albanian, Southeast Asian Indonesian, as well as the African languages Afrikaans and Swahili. The missing euro sign and capital *Ÿ* are in the revised version ISO/IEC 8859-15.

ISO/IEC 8859-1																
	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x																
1x																
2x	SP	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6x	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	
8x																
9x																
Ax	NBSP	¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	SHY	®	¯
Bx	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
Cx	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
Dx	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
Ex	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
Fx	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

6.5.2 ISO 8859-2

Latin-2 Central European

Supports those Central and Eastern European languages that use the Latin alphabet, including Bosnian, Polish, Croatian, Czech, Slovak, Slovene, Serbian, and Hungarian. The missing euro sign can be found in version ISO/IEC 8859-16.

ISO/IEC 8859-2																
	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x																
1x																
2x	*SP*	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6x	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	
8x																
9x																
Ax	*NBSP*	Ą	˘	Ł	¤	Ľ	Ś	§	¨	Š	Ş	Ť	Ź	*SHY*	Ž	Ż
Bx	°	ą	˛	ł	´	ľ	ś	ˇ	¸	š	ş	ť	ź	˝	ž	ż
Cx	Ŕ	Á	Â	Ă	Ä	Ĺ	Ć	Ç	Č	É	Ę	Ë	Ě	Í	Î	Ď
Dx	Đ	Ń	Ň	Ó	Ô	Ő	Ö	×	Ř	Ů	Ú	Ű	Ü	Ý	Ţ	ß
Ex	ŕ	á	â	ă	ä	ĺ	ć	ç	č	é	ę	ë	ě	í	î	ď
Fx	đ	ń	ň	ó	ô	ő	ö	÷	ř	ů	ú	ű	ü	ý	ţ	˙

6.5.3 ISO 8859-3

Latin-3 South European

Turkish, Maltese, and Esperanto. Largely superseded by ISO/IEC 8859-9 for Turkish and Unicode for Esperanto.

ISO/IEC 8859-3															
	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x																
1x																
2x	*SP*	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6x	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	
8x																
9x																
Ax	*NBSP*	Ħ	˘	£	¤		Ĥ	§	¨	İ	Ş	Ğ	Ĵ	*SHY*		Ż
Bx	°	ħ	²	³	´	µ	ĥ	·	¸	ı	ş	ğ	ĵ	½		ż
Cx	À	Á	Â		Ä	Ċ	Ĉ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
Dx		Ñ	Ò	Ó	Ô	Ġ	Ö	×	Ĝ	Ù	Ú	Û	Ü	Ŭ	Ŝ	ß
Ex	à	á	â		ä	ċ	ĉ	ç	è	é	ê	ë	ì	í	î	ï
Fx		ñ	ò	ó	ô	ġ	ö	÷	ĝ	ù	ú	û	ü	ŭ	ŝ	˙

6.5.4 ISO 8859-4

Latin-3 North European

Estonian, Latvian, Lithuanian, Greenlandic, and Sami.

ISO/IEC 8859-4															
	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x																
1x																
2x	*SP*	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6x	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	
8x																
9x																
Ax	*NBSP*	Ą	ĸ	Ŗ	¤	Ĩ	Ļ	§	¨	Š	Ē	Ģ	Ŧ	*SHY*	Ž	¯
Bx	°	ą	˛	ŗ	´	ĩ	ļ	ˇ	¸	š	ē	ģ	ŧ	Ŋ	ž	ŋ
Cx	Ā	Á	Â	Ã	Ä	Å	Æ	Į	Č	É	Ę	Ë	Ė	Í	Î	Ī
Dx	Đ	Ņ	Ō	Ķ	Ô	Õ	Ö	×	Ø	Ų	Ú	Û	Ü	Ũ	Ū	ß
Ex	ā	á	â	ã	ä	å	æ	į	č	é	ę	ë	ė	í	î	ī
Fx	đ	ņ	ō	ķ	ô	õ	ö	÷	ø	ų	ú	û	ü	ũ	ū	˙

6.5.5 ISO 8859-5

Latin/Cyrillic

Covers mostly Slavic languages that use a Cyrillic alphabet, including Belarusian, Bulgarian, Macedonian, Russian, Serbian, and Ukrainian (partial)

ISO/IEC 8859-5															
	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x																
1x																
2x	*SP*	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6x	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	
8x																
9x																
Ax	*NBSP*	Ё	Ђ	Ґ	Є	S	І	Ї	J	Љ	Њ	Ћ	Ќ	*SHY*	Ў	Џ
Bx	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
Cx	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
Dx	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
Ex	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
Fx	№	ё	ђ	ґ	є	s	і	ї	j	љ	њ	ћ	ќ	§	ў	џ

6.5.6 ISO 8859-6

Latin/Arabic

Covers the most common Arabic language characters. Doesn't support other languages using the Arabic script. Needs to be BiDi and cursive joining processed for display.

ISO/IEC 8859-6															
	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x																
1x																
2x	SP	!	"	#	$	%	&	'	(	)	*	+	،	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	؟
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6x	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	
8x																
9x																
Ax	NBSP				¤								؛	SHY		
Bx												ء				؟
Cx		ء	آ	أ	ؤ	إ	ئ	ا	ب	ة	ت	ث	ج	ح	خ	د
Dx	ذ	ر	ز	س	ش	ص	ض	ط	ظ	ع	غ					
Ex	ـ	ف	ق	ك	ل	م	ن	ه	و	ى	ي	ً	ٌ	ٍ	َ	ُ
Fx	ِ	ّ	ْ													

6.5.7 ISO 8859-7

Latin/Greek

Covers the modern Greek language (monotonic orthography). Can also be used for Ancient Greek written without accents or in monotonic orthography, but lacks the diacritics for polytonic orthography. These were introduced with Unicode.

ISO/IEC 8859-7															
	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x																
1x																
2x	*SP*	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6x	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	
8x																
9x																
Ax	*NBSP*	‘	’	£	€		¦	§	¨	©		«	¬	*SHY*		—
Bx	°	±	²	³	΄	΅	Ά	·	Έ	Ή	Ί	»	Ό	½	Ύ	Ώ
Cx	ΐ	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο
Dx	Π	Ρ		Σ	Τ	Υ	Φ	Χ	Ψ	Ω	Ϊ	Ϋ	ά	έ	ή	ί
Ex	ΰ	α	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο
Fx	π	ρ	ς	σ	τ	υ	φ	χ	ψ	ω	ϊ	ϋ	ό	ύ	ώ	

6.5.8 ISO 8859-8

Latin/Hebrew

Covers the modern Hebrew alphabet as used in Israel. In practice two different encodings exist, logical order (needs to be BiDi processed for display) and visual (left-to-right) order (in effect, after bidi processing and line breaking).

ISO/IEC 8859-8															
	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x																
1x																
2x	*SP*	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6x	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	
8x																
9x																
Ax	*NBSP*		¢	£	¤	¥	¦	§	¨	©	×	«	¬	*SHY*	®	¯
Bx	°	±	²	³	´	µ	¶	·	¸	¹	÷	»	¼	½	¾	
Cx																
Dx																‗
Ex	א	ב	ג	ד	ה	ו	ז	ח	ט	י	ך	כ	ל	ם	מ	ן
Fx	נ	ס	ע	ף	פ	ץ	צ	ק	ר	ש	ת			*LRM*	*RLM*	

6.5.9 ISO 8859-9

Latin-5 Turkish

Largely the same as ISO/IEC 8859-1, replacing the rarely used Icelandic letters with Turkish ones. It is also used for Kurdish.

位置	0xD0	0xDD	0xDE	0xF0	0xFD	0xFE
8859-9	Ğ	İ	Ş	ğ	ı	ş
8859-1	Ð	Ý	Þ	ð	ý	þ

ISO/IEC 8859-9															
	x0	**x1**	**x2**	**x3**	**x4**	**x5**	**x6**	**x7**	**x8**	**x9**	**xA**	**xB**	**xC**	**xD**	**xE**	**xF**
0x																
1x																
2x	*SP*	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6x	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	
8x																
9x																
Ax	*NBSP*	¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	*SHY*	®	¯
Bx	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
Cx	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
Dx	Ğ	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	İ	Ş	ß
Ex	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
Fx	ğ	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ı	ş	ÿ

6.5.10 ISO 8859-10

Latin-6 Nordic

a rearrangement of Latin-4. Considered more useful for Nordic languages. Baltic languages use Latin-4 more.

ISO/IEC 8859-10															
	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x																
1x																
2x	*SP*	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6x	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	
8x																
9x																
Ax	*NBSP*	Ą	Ē	Ģ	Ī	Ĩ	Ķ	§	Ļ	Đ	Š	Ŧ	Ž	*SHY*	Ū	Ŋ
Bx	°	ą	ē	ģ	ī	ĩ	ķ	·	ļ	đ	š	ŧ	ž	―	ū	ŋ
Cx	Ā	Á	Â	Ã	Ä	Å	Æ	Į	Č	É	Ę	Ë	Ė	Í	Î	Ï
Dx	Ð	Ņ	Ō	Ó	Ô	Õ	Ö	Ũ	Ø	Ų	Ú	Û	Ü	Ý	Þ	ß
Ex	ā	á	â	ã	ä	å	æ	į	č	é	ę	ë	ė	í	î	ï
Fx	ð	ņ	ō	ó	ô	õ	ö	ũ	ø	ų	ú	û	ü	ý	þ	ĸ

6.5.11 ISO 8859-11

Latin/Thai

Contains characters needed for the Thai language. Virtually identical to TIS 620.

ISO/IEC 8859-11															
	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x																
1x																
2x	SP	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6x	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	
8x																
9x																
Ax	NBSP	ก	ข	ฃ	ค	ฅ	ฆ	ง	จ	ฉ	ช	ซ	ฌ	ญ	ฎ	ฏ
Bx	ฐ	ฑ	ฒ	ณ	ด	ต	ถ	ท	ธ	น	บ	ป	ผ	ฝ	พ	ฟ
Cx	ภ	ม	ย	ร	ฤ	ล	ฦ	ว	ศ	ษ	ส	ห	ฬ	อ	ฮ	ฯ
Dx	ะ	ั	า	ำ	ิ	ี	ึ	ื	ุ	ู	ฺ					฿
Ex	เ	แ	โ	ใ	ไ	ๅ	ๆ	็	่	้	๊	๋	์	ํ	๎	๏
Fx	๐	๑	๒	๓	๔	๕	๖	๗	๘	๙	๚	๛				

6.5.12 ISO 8859-13

Latin-7 Baltic Rim

Added some characters for Baltic languages which were missing from Latin-4 and Latin-6.

ISO/IEC 8859-13															
	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x																
1x																
2x	*SP*	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6x	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	
8x																
9x																
Ax	*NBSP*	”	¢	£	¤	„	¦	§	Ø	©	Ŗ	«	¬	*SHY*	®	Æ
Bx	°	±	²	³	“	µ	¶	·	ø	¹	ŗ	»	¼	½	¾	æ
Cx	Ą	Į	Ā	Ć	Ä	Å	Ę	Ē	Č	É	Ź	Ė	Ģ	Ķ	Ī	Ļ
Dx	Š	Ń	Ņ	Ó	Ō	Õ	Ö	×	Ų	Ł	Ś	Ū	Ü	Ż	Ž	ß
Ex	ą	į	ā	ć	ä	å	ę	ē	č	é	ź	ė	ģ	ķ	ī	ļ
Fx	š	ń	ņ	ó	ō	õ	ö	÷	ų	ł	ś	ū	ü	ż	ž	’

6.6 UTF-8 to UTF-16 Conversion Program

```
//--------------------------------------------------------------------
//Function name：UFT8toUTF16(unsigned char *pUTF8,unsigned short *pUTF16 )
//Function description:UTF8 to UTF16 code conversion fuction
//Parameter：pUTF8: UTF8 data pointer；pUTF16: UTF16 data pointer
//Back to instruction: The capacity this UTF8 character occupied.
//--------------------------------------------------------------------
unsigned char UFT8toUTF16(unsigned char *pUTF8,unsigned short *pUTF16 )
{
unsigned char bytes[3];
unsigned short unicode16;
bytes[0] = *pUTF8++;
if( bytes[0] < 0x80 )
{
*pUTF16 = bytes[0];
return(1);
}
bytes[1] = *pUTF8++;
if( bytes[0] >= 0xC0 && bytes[0] < 0xE0 )
{
unicode16 = 0x1f&bytes[0];
*pUTF16 = (unicode16<<6)+(0x3f&bytes[1]);
return(2);
}
bytes[2] = *pUTF8++;
if( bytes[0] >= 0xE0 && bytes[0] < 0xF0 )
{
unicode16 = 0x0f&bytes[0];
unicode16 = (unicode16<<6)+(0x3f&bytes[1]);
*pUTF16 = (unicode16<<6)+(0x3f&bytes[2]);
return(3);
}
else
return(0x00);
}
```

6.7 5X7 Character ISO8859 to UNICODE Conversion Program

No.	Patch Description	Patch Mode	Corresponding File	Version	Release Date
1	5X7 ISO8859 to Unicode Conversion Program	Program	<<ISO8859_Unicode.c>>	V1.00	2011-1-12

关于GT20X 焊接工艺说明

GENITOP 出品的GT20X 标准字库系列芯片，封装均为无铅电镀产品（Lead-FreePackage）。按照GENTIOP 外协封装厂企业标准（外协厂均为国内知名上市企业）回流焊接最高极限温度（Maximum Reflow Temperature）为240℃+0/-5℃。推荐回流焊接采用通常符合IPC/JEDEC J-STD-020 标准工艺，峰值温度控制在235℃+0/-5℃。

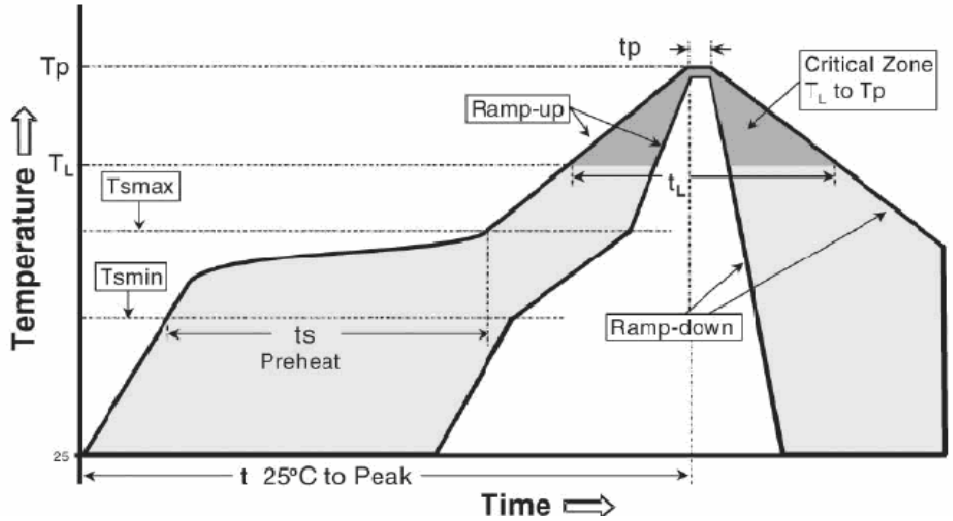
若回流焊接工艺需要采用温度 260℃+0/-5℃，需额外咨询我司工业级别芯片。

Package Thickness	Volume mm^3 < 350	Volume mm^3 350 - 2000	Volume mm^3 > 2000
< 1.6 mm	260 °C *	260 °C *	260 °C *
1.6 mm - 2.5 mm	260 °C *	250 °C *	245 °C *
> 2.5 mm	250 °C *	245 °C *	245 °C *
* Tolerance: The device manufacturer/supplier shall assure process compatibility up to and including the stated classification temperature at the rated MSL level			

焊接表-1 **Pb-free Process - Package Peak Reflow Temperatures**

Profile Feature	Sn-Pb Eutectic Assembly	Pb-Free Assembly
Average ramp-up rate (Tsmax to Tp)	3° C/second max.	3° C/second max.
Preheat - Temperature Min (Ts_{min}) - Temperature Max (Ts_{max}) - Time (Ts_{min} to Ts_{max}) (ts)	 100 °C 150 °C 60-120 seconds	 150 °C 200 °C 60-180 seconds
Time maintained above: - Temperature (T_L) - Time (t_L)	 183 °C 60-150 seconds	 217 °C 60-150 seconds
Peak Temperature (Tp)	See Table 4.1	See Table 4.2
Time within 5°C of actual Peak Temperature (tp)[2]	10-30 seconds	20-40 seconds
Ramp-down Rate	6 °C/second max.	6 °C/second max.
Time 25°C to Peak Temperature	6 minutes max.	8 minutes max.

焊接表-2 **Classification Reflow Profiles**



焊接图-1 **Classification Reflow Profile**