Product Specification

Multi-Language OLED Display
with built in language tables

Part Number: **FDS128x64(75.52.7)TBP**

PREPARED BY	CHECKED BY	APPROVED BY

Focus Display Solutions, Inc.

Notes:

1. Please contact Focus Display Solutions, Inc. before assigning your product based on this module specification

2. The information contained herein is presented merely to indicate the characteristics and performance of our products. No responsibility is assumed by Focus Display Solutions, Inc. For any intellectual property claims or other problems that may result from application based on the module described herein

1 Basic Specification

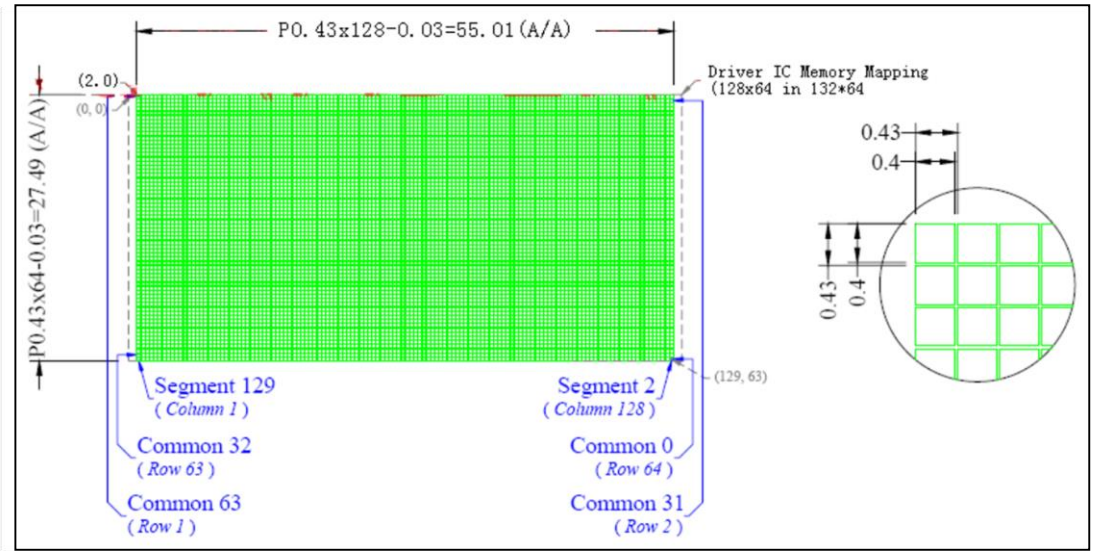
1.1 Display Specifications

Item	Specs
Display Mode	Passive Matrix—OLED: Yellow &Green
Interface	Serial Interface
Drive Duty	1/64
Driver IC	SSD1305
Shell	0.5T
Other	

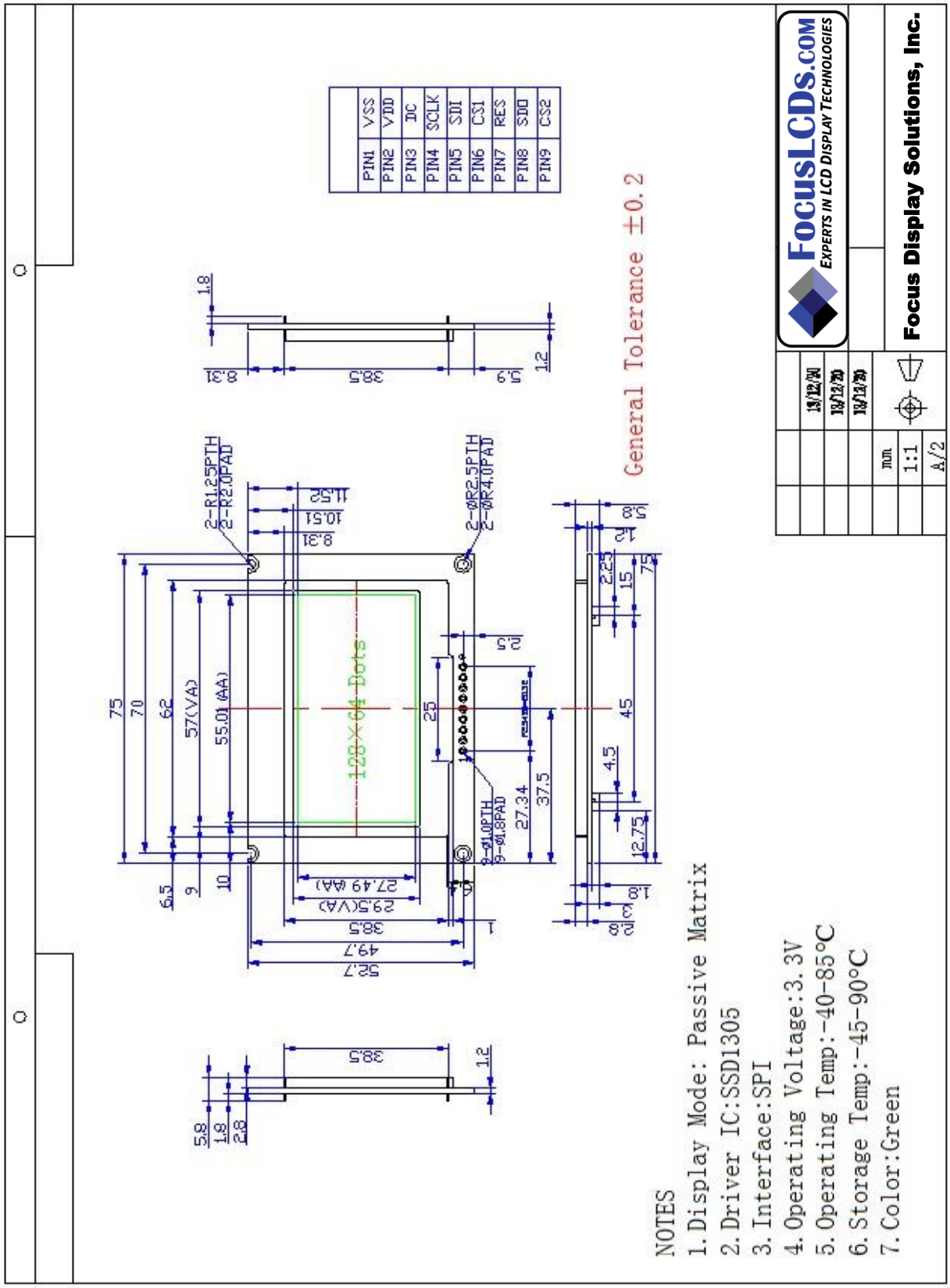
1.2 Mechanical Specifications

Item	Specs	Unit	Remark
Outline Drawing	75(W) x52.7 (H) x5.8Max (T)	mm	
View Area	57(W) x29.5 (H)	mm	
Active Area	55(W) x27.5 (H)	mm	
Lattice	128dots x 64dots	--	Pixel
Pitch	0.43(W) x0.43 (H)	mm	
Pixel Size	0.40(W)x0.40(H)	mm	

1.3 Active Area & Pixel Construction



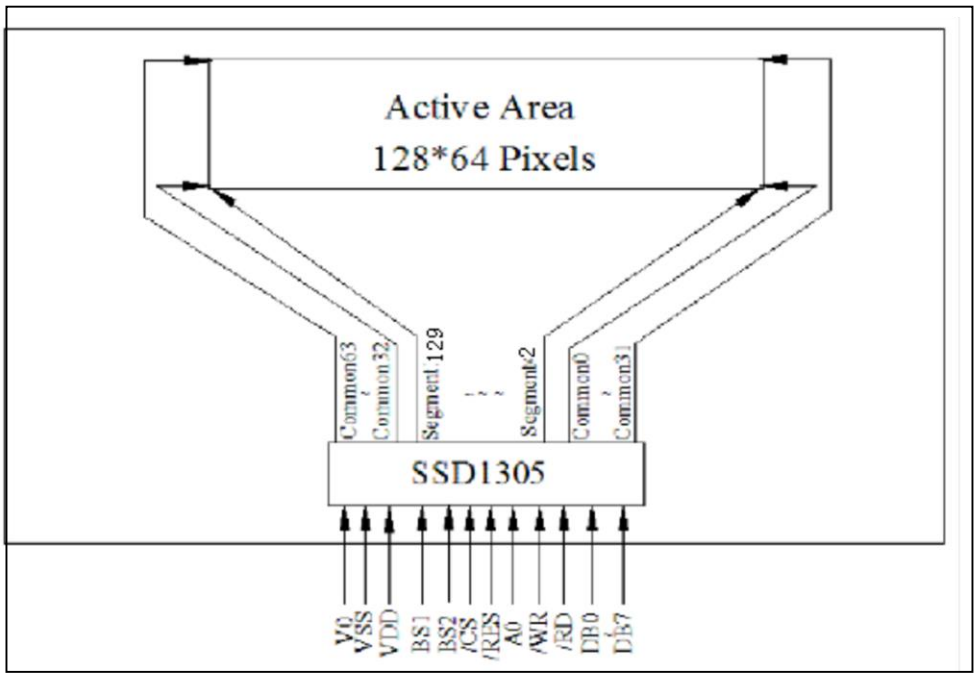
1.4 Mechanical Drawing



1.5 Pin Definition

Pin No.	Symbol	I/O	Function
1	VSS	P	Ground of Logic Circuit This is a ground pin. It also acts as a reference for the Logic pins. It must be connected to external ground.
2	VDD		Power Supply for Display Module Circuit This is a voltage supply pin. It connected to external Source.
3	DC	I	This is Data/Command control pin. When it is pulled HIGH, the data at D[7:0] is treated as data. When it is pulled LOW, the data at D[7:0] will be transferred to the command register.
4	SCLK	I	Series clock for SSD1305and multi-font IC
5	SDI	I	Series data input for SSD1305and multi-font IC
6	CS1	I	SSD1305 This pin is the chip select input. (active LOW)
7	RES		This pin is reset signal input. When the pin is LOW, initialization of the chip is executed. Keep this pin HIGH during normal operation.
8	SDO	O	Output the series data of font IC when clock pause is going down.
9	CS2		For multi font IC this pin is the chip select input. All series data transmission occurs on the trailing edge of the CS# pulse, and CS# must be low level when data are been transmitting, and will be high level between 2 instruments.

1.6 Elements Block Diagram



2. Absolute Maximum Ratings

Parameter	Symbol	Min	Max	Unit	Notes
Supply Voltage	VBAT	3.0	5	V	-
Driver Supply Voltage	Vcc	0	15	V	1,2
Vcc Supply Current	Icc	-	28	mA	1,2
Operating Temperature	Top	-40	+85	℃	-
Storage Temperature	Tstg	-45	+90	℃	-

Note 1: All the above voltages are on the basis of "GND=0V".

Note 2: When this module is used beyond the above absolute maximum Ratings, permanent breakage of the module may occur. Also, for normal operations, it is desirable to use this module under the conditions according to Section 3. "Electrical Characteristics". If this module is used beyond these conditions, malfunctioning of the module can occur and the reliability of the module may deteriorate.

3. Electrical Characteristics

3.1 DC Characteristics

Characteristics	Symbol	Conditions	Min	Typ	Max	Unit
Supply Voltage	VBAT		2.8	3.0	5	V
High Level Input	VIH	Iout =100Ua 3.3MHz	0.8xVdd	-	Vdd	V
Low Level Input	VIL	Iout =100Ua 3.3MHz	0	-	0.2xVdd	V
Supply Current	IVBAT	Note	-	90	--	mA
Display voltage	VCC	Ta = 25℃	12.0	12.5	13	V

Note: VDD=3.0V, VCC=12.5V(VDD and VCC Supply by the module internal generated) 100% Display Area Turn on.

3.2 Optics & Electrical Characteristics

Characteristics	Symbol	Conditions	Min	Typ	Max	Unit
Brightness	Lbr	With Polarizer	100	120	-	Cd/m2
C.I.E.(Green)	(X) (Y)	C.I.E 1931	0.27 0.58	0.31 0.62	0.35 0.66	
Dark Room Contrast	CR		-	>2000:1	-	
View Angle			>160	-	-	degree

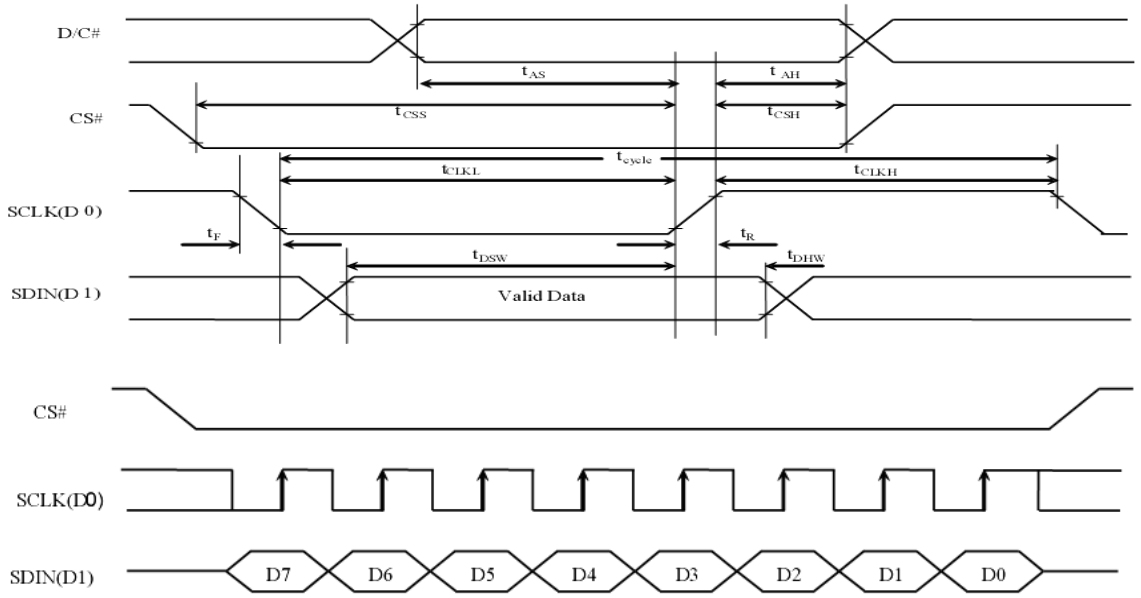
- Optical measurement taken at VDD =2.8V.VCC=12.5V, and software configuration follows Sec4.8 “Software Initial Setting”

3.3 Serial Interface Timing Characteristics

(1) SSD1305 Serial Interface Timing

(V_{DD} - V_{SS} = 2.4V to 3.5V , V_{DDIO} = V_{DD} ,T_A = 25°C)

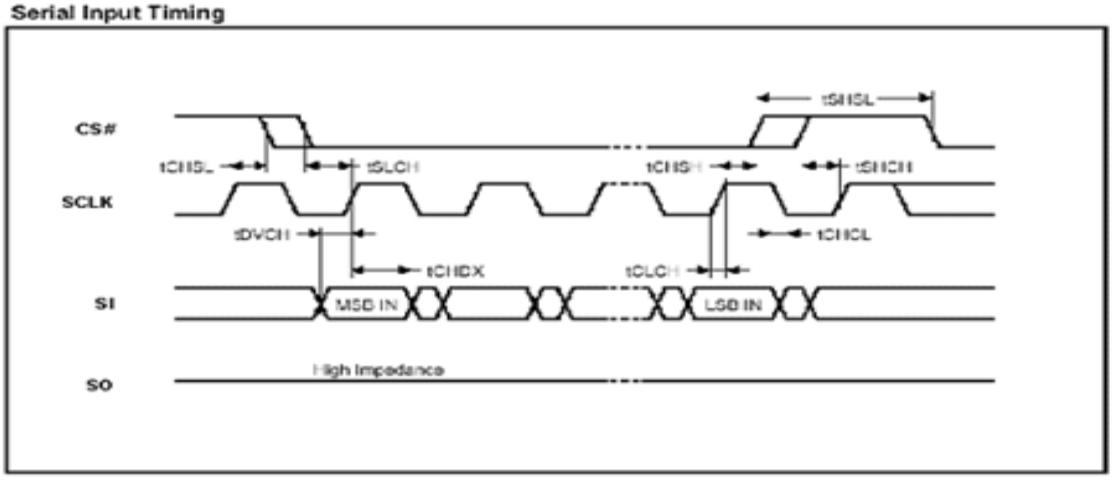
Symbol	Parameter	Min	Typ	Max	Unit
t_{cycle}	Clock Cycle Time	250	-	-	ns
t_{AS}	Address Setup Time	150	-	-	ns
t_{AH}	Address Hold Time	150	-	-	ns
t_{CSS}	Chip Select Setup Time	120	-	-	ns
t_{CSH}	Chip Select Hold Time	60	-	-	ns
t_{DSW}	Write Data Setup Time	50	-	-	ns
t_{DHW}	Write Data Hold Time	15	-	-	ns
t_{CLKL}	Clock Low Time	100	-	-	ns
t_{CLKH}	Clock High Time	100	-	-	ns
t_R	Rise Time	-	-	40	ns
t_F	Fall Time	-	-	40	ns



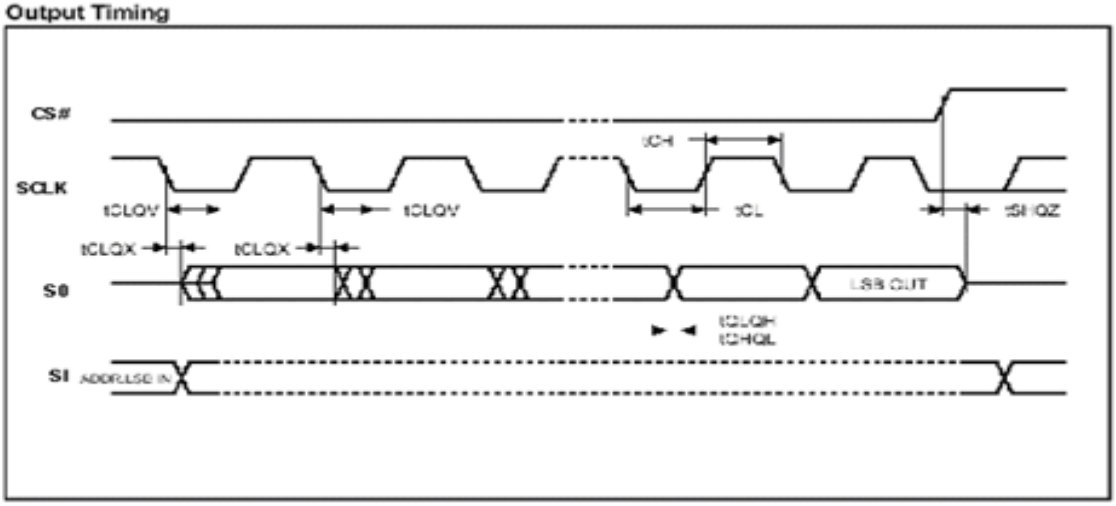
(2) Multi-font IC Serial Interface Timing

Symbol	Alt.	Parameter	Min.	Max.	Unit
Fc	Fc	Clock Frequency	D.C.	30	MHz
tCH	tCLH	Clock High Time	15		ns
tCL	tCLL	Clock Low Time	15		ns
tCLCH		Clock Rise Time(peak to peak)	0.1		V/ns
tCHCL		Clock Fall Time (peak to peak)	0.1		V/ns
tSLCH	tCSS	CS# Active Setup Time (relative to SCLK)	5		ns
tCHSL		CS# Not Active Hold Time (relative to SCLK)	5		ns
tDVCH	tDSU	Data In Setup Time	2		ns
tCHDX	tDH	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	tCSH	CS# Deselect Time	100		ns
t SHQZ	tDIS	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 Functional Specifications

MCU Interface selection:

MCU Interface assignment under different bus interface mode:

Pin Name / Bus Interface	Data/Command Interface								Control Signal				
	D7	D6	D5	D4	D3	D2	D1	D0	E	R/W#	CS#	D/C#	RES#
8-bit 8080	D[7:0]								RD#	WR#	CS#	D/C#	RES#
8-bit 6800	D[7:0]								E	R/W#	CS#	D/C#	RES#
SPI	Tie LOW					NC	SDIN	SCLK	Tie LOW		CS#	D/C#	RES#
I^2C	Tie LOW					SDA_{OUT}	SDA_{IN}	SCL	Tie LOW			SA0	RES#

4.1 MCU Serial Interface

The serial interface consists of serial clock SCLK, serial data SDIN, D/C#, CS#. In SPI mode, D0 acts as SCLK, D1 acts as SDIN. For the unused data pins, D2 should be left open. The pins from D3 to D7, E and R/W# can be connected to an external ground.

Control pins of Serial interface

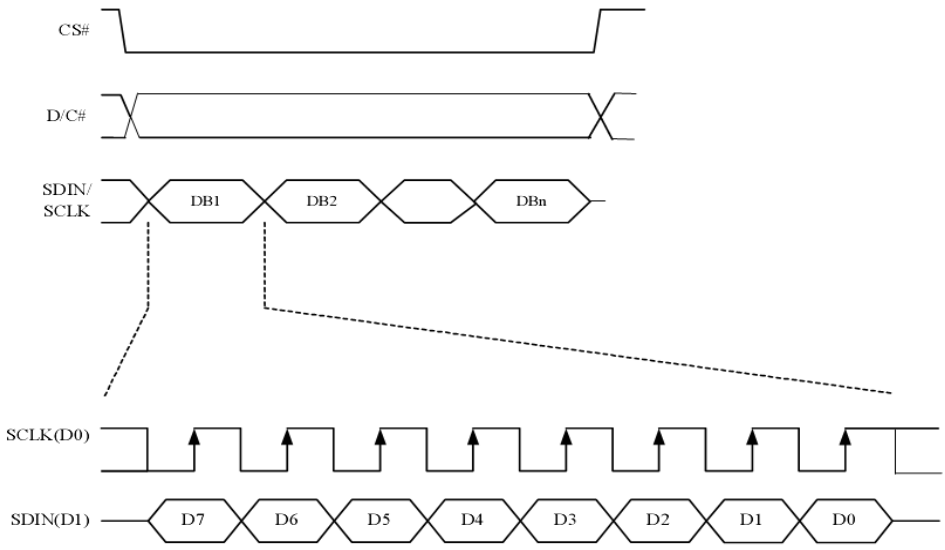
Function	E(RD#)	R/W#(WR#)	CS#	D/C#	D0
Write command	Tie LOW	Tie LOW	L	L	↑
Write data	Tie LOW	Tie LOW	L	H	↑

Note :

(1) ↑ stands for rising edge of signal.
(2) H stands for HIGH in signal.
(3) L stands for LOW in signal.

SDIN is shifted into an 8-bit shift register on every rising edge of SCLK in the order of D7, D6, ... D0. D/C# is sampled on every eighth clock and the data byte in the shift register is written to the Graphic Display Data RAM (GDDRAM) or command register in the same clock. Under serial mode, only write operations are allowed.

Write procedure in SPI mode



5. Multi-font Operation instruments

5.1 Command 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 ∞

There are only 2 operations to multi-font IC, they are Read Data Bytes (normal READ) and Read Data Bytes at Higher Speed (FAST_READ).

5.2 Read Data Bytes

Read Data Bytes will be operated when each operation. The timing of READ Instrument is as below:

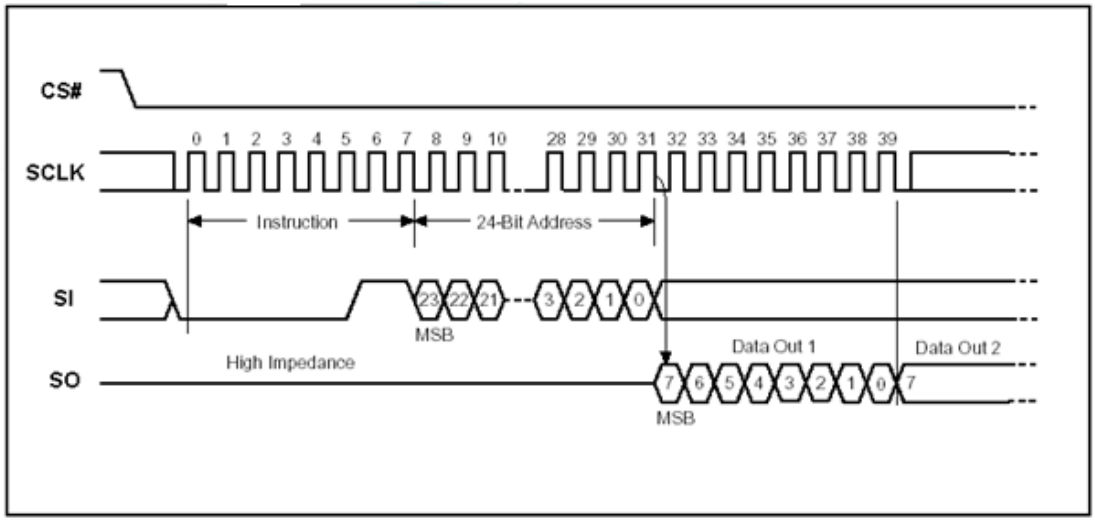
At first CS# goes down, and then one-byte command (03h) and 3-byteaddress shifted in through series data input terminal (SI), each is locked on the rising edge.

Then the address data is shifted out through the series data output terminal (SO), each is moved out on series clock trailing edge.

After reading operation, then CS# signal becomes as high level and ends the operation.

If CS# keeps as low level, then next address byte data will be shifted out through series data output terminal (SO).

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



5.3 Read Data Bytes at Higher Speed

Read Data Bytes at Higher Speed need to be operated by instruments. READ_FAST timing is as below：

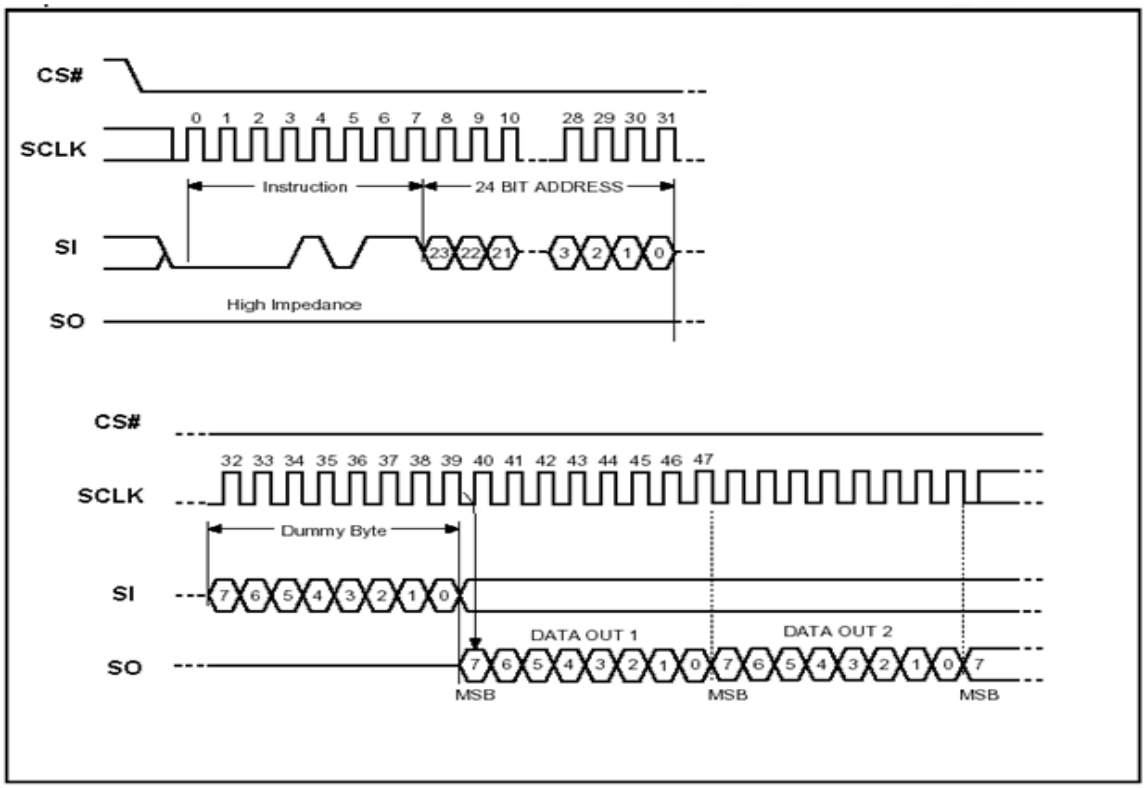
At first CS# goes down, and then one-byte command (0Bh) and 3-byteaddress, & one-byte Dummy_byte will be shifted in through series data input terminal (SI), each is locked on the rising edge of series clock (SCLK).

Then the address data is shifted out through the series data output terminal (SO), each is moved out on series clock (SCLK) trailing edge.

If CS# keeps as low level, then next address byte data will continuously be shifted out through series data output terminal (SO). For instance, a reading of 15x16 dots Chinese font need a 32 bytes, so after reading of 32 bytes, an operation of reading of one Chinese font ends.

If don't need to continue reading data, then change CS# signal as high level, the operation ends.

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



5.4 Character Patterns Map

No.	Description of font	Code #	Font Qty	Start address	reference
1	16x16 GB2312 (simple) font data	GB2312	7614	0x11080	5.5.1
2	16x16 BIG5(complex) font data	BIG5	5401	0x11080	5.5.2
3	16x16 KSC5601 font	KSC5601	3465	0x5CB60	5.5.3
4	16x16 JIS0208 Japanese font data	JIS0208	8366	0x77C80	
5	16x16 KSC5601chinese	KSC5601	4888	0xED920	
6	8x16 shift-JIS Japanese half kana	Shift –JIS	63	0x124D32	
7	5x7 ASCII standard font	ASCII	96	0x0	5.6.1
8	7x8 ASCII standard font	ASCII	96	0x300	5.6.2
9	8x16 ASCII standard font	ASCII	96	0x40C	5.6.3
10	16dots ASCII Arial (square) font	ASCII	96	0xC00	5.6.4
11	ISO8859 (14 sets)	ISO8859	1792	0xE7331	
12	ISO8859 Unicode	ISO8859	509	0xEF130	
13	LCM (3 sets)	LCM	256	0xEAB31	
14	8x16 Latin series fonts	Unicode	96	0x18C0	5.7.1
15	8x16 Greek series fonts	Unicode	96	0x37C0	5.7.2
16	8x16 Cyrillic series font	Unicode	208	0x3DC0	5.7.3
17	8x16 wide Hebrew font	Unicode	112	0x4AC0	
18	8x16 wide Thai font	Unicode	128	0x51C0	5.7.4
19	16 dot Latin series font	Unicode	96	0x59C0	5.7.5
20	16 dot Greek font	Unicode	96	0x9BA0	5.7.6
21	16 dot Cyrillic font	Unicode	208	0xA860	5.7.7
22	16 dot Latin font	Unicode	576	0x139C0	5.7.8
23	UNICODE-> GB2312 transcoding			0xCBA9E	
24	UNICODE-> BIG5 transcoding			0x113C20	
25	UNICODE-> JIS020 transcoding			0xB9240	
26	UNICODE-> KSC560 transcoding			0xD7B3E	
27	Shift -JIS-> JIS0208 transcoding			0x11FA68	

5.5 Method for address of character pattern data

As long as the user knows the font code # , they can acquire the address (start address) of the chip, user can read the data of font from the address for display.

5.5.0 Method for Chinese font data

5.5.1 16x16 GB2312 font data

Parameters :

GBCode means code #.
MSB means high 8 bits of GBCode.
LSB means low 8 bits of GBCode.
Address means the byte address of Chinese or ASCII font.
BaseAddr means the start address of font data in the chip.

Method :

```
BaseAddr=0x11080 ;
if(MSB >=0xA1 && MSB <= 0Xa9 && LSB >=0xA1)
        Address =( (MSB - 0xA1) * 94 + (LSB - 0xA1))*32+ BaseAddr;
 else if(MSB >=0xB0 && MSB <= 0xF7 && LSB >=0xA1)
        Address = ((MSB - 0xB0) * 94 + (LSB - 0xA1)+ 846)*32+ BaseAddr;
```

5.5.2 16x16 BIG5 font data

BYTE readbyte(DWORD address):
function : reading one-byte data from any address from font chip.
Parameter: the address where the font data is stored .
Return value : content of font data address .

Method for BIG5 standard font data address.
MSB:BIG5 - high byte of BIG5 code
LSB:BIG5 - low byte of BIG5 code
Address: means the font data address of BIG5 font.

```
BaseAddr=0x11080;
Big5Table=936190;
if(MSB >=0xA1 && MSB <= 0xA3)       // font  area
{
   if(LSB >=0x40 && LSB <= 0X7E)
          Address =((MSB - 0xA4) * 157 + (LSB - 0x40))*32+0x125140;
   else if(LSB >=0XA1 && LSB <= 0XFE)
          Address =((MSB - 0xA4) * 157 + 63 + (LSB - 0xA1))*32+ 0x1251
 }
    else if(MSB >=0xA4 && MSB <= 0XC6)   //  text area
 {
    if(LSB >=0x40 && LSB <= 0X7E)
          Big5Index =(MSB - 0xA4) * 157 + (LSB - 0x40);
    else if(LSB >=0XA1 && LSB <= 0XFE)
          Big5Index =(MSB - 0xA4) * 157 + 63 + (LSB - 0xA1));
    Index = readbyte(Big5Table+Big5Index*2 )* 256 + readbyte(Big5Table+Big5Index*2+1);
   Address =Index * 32 + BaseAddr;
 }
```

5.5.3 16x16 KSC5601 font data

Parameters：

MSB：means high 8 bits of Korean Code.

LSB：means low 8 bits of Korean Code.

Address means byte address of font in the chip.

BaseAddr：start address of font data in the chip.

Method：

```
BaseAddr=0x5CB60
if(MSB>=0xA1&&MSB<0xB0&&LSB>=0xA1)
        Address =((MSB-0xA1)*94+(LSB-0xA1))*32+ BaseAddr;
else if(MSB>=0xB0 && MSB<=0XC8 && LSB>=0xA1)
        Address = ((MSB-0xA1)*94+(LSB-0xA1))*32+35680+BaseAddr;
```

5.6 Method for ASCII characters data address

5.6.1 5x7 pattern ASCII standard characters

Parameters：

ASCIICode：means ASCII code（8bits）

Address：byte address of ASCII standard font in the chip.

Method：

```
if(ASCIICODE >=0x20 && ASCIICODE <=0x7F)
   addr =(ASCIICODE-0x20)*8
```

5.6.2 7x8 pattern ASCII standard characters

Parameters：

ASCIICode：means ASCII code （8bits）

Address：byte address of ASCII characters in the chip.

Method：

```
if(ASCIICODE >=0x20 && ASCIICODE <=0x7F)
      Address =(ASCIICODE-0x20)*8 +0x300
```

5.6.3 8x16 pattern ASCII characters

Parameters：

ASCIICode：means ASCII code（8bits）

Address：byte address of ASCII characters in the chip.

Method：

```
if((ASCIICode >=0x20)&&( ASCIICode <=0x7F))
        addr = (ASCIICode -0x20)*16 +0x40C
```

5.6.4 16 dot ASCII Arial characters

Parameters：

ASCIICode：means ASCII code （8bits）.

Address：byte address of ASCII Arial characters in the chip.

Method：

```
if((ASCIICode >=0x20)&&( ASCIICode <=0x7F))
        addr = (ASCIICode -0x20)*34 +0xC00;
```

5.7 Method for addressing of multi-fonts data

5.7.1 8x16 pattern of Latin font set

Description:

FontCode：means unicode code #（16bits）

Address：byte address of Latin font in the chip.

Method :

```
if (FontCode>=0x0020 && FontCode<=0x007F)
        Address=(FontCode-0x0020) * 16 +0x18C0;
else if (FontCode>=0x00A0 && FontCode<=0x017F)
        Address=(FontCode-0x0040) * 16 +0x18C0;
else if (FontCode>=0x01A0 && FontCode<=0x01CF)
        Address=(FontCode-0x01A0+320) * 16 +0x18C0;
else if (FontCode>=0x01F0 && FontCode<=0x01FF)
        Address=(FontCode-0x01F0+368) * 16 +0x18C0;
else if (FontCode>=0x0210 && FontCode<=0x021F)
        Address=(FontCode-0x0210+384) * 16 +0x18C0;
else if (FontCode>=0x1EA0 && FontCode<=0x1EFF)
       Address=(FontCode-0x1EA0+400) * 16 +0x18C0
```

5.7.2 8x16 pattern of Greek font set

Description:

FontCode：means unicode code #（16bits）

Address：byte address of Greek font in the chip.

Method :

```
if (FontCode>=0x0370 && FontCode<=0x03CF)
      Address=(FontCode-0x0370) * 16 +0x37C0
```

5.7.3 8x16 pattern Cyrillic font set

Description:

FontCode：means unicode code #（16bits）

Address：byte address of Cyrillic font in the chip.

Method :

```
if (FontCode>=0x0400 && FontCode<=0x045F)
        Address=(FontCode-0x0400) * 16 +0x3DC0
else if (FontCode>=0x0490 && FontCode<=0x04FF)
        Address=(FontCode-0x0490+96) * 16 +0x3DC0
```

5.7.4 8x16 pattern Thai font set

Description:

FontCode：means unicode code #（16bits）

Address：byte address of Thai font in the chip.

Method :

```
if (FontCode>=0x0E00 && FontCode<=0x0E5F)
        Address=(FontCode-0x0E00) * 16 +0x51C0
```

5.7.5 16 dots Latin font set

Description:

FontCode：means unicode code #（16bits）

Address：byte address of Latin font in the chip.

Method :

```
if (FontCode>=0x0020 && FontCode<=0x007F)
        Address=(FontCode-0x0020) * 34 +0x59C0;
else if (FontCode>=0x00A0 && FontCode<=0x017F)
        Address=(FontCode-0x00A0+96) * 34 +0x59C0;
else if (FontCode>=0x01A0 && FontCode<=0x01CF)
        Address=(FontCode-0x01A0+320) * 34 +0x59C0;
else if (FontCode>=0x01F0 && FontCode<=0x01FF)
        Address=(FontCode-0x01F0+368) * 34 +0x59C0;
else if (FontCode>=0x0210 && FontCode<=0x021F)
        Address=(FontCode-0x0210+384) * 34 +0x59C0;
else if (FontCode>=0x1EA0 && FontCode<=0x1EFF)
       Address=(FontCode-0x1EA0+400) * 34 +0x59C0;
```

5.7.6 16 dots Greek font set

Description:

FontCode：means unicode code #（16bits）

Address：byte address of Greek font in the chip. 。

Method :

```
if (FontCode>=0x0370 && FontCode<=0x03CF)
        Address=(FontCode-0x0370) * 34 +0x9BA0
```

5.7.7 16 dots Cyrillic font set

Description:

FontCode：means unicode code #（16bits）

Address：byte address of Cyrillic font in the chip.

Method :

```
if (FontCode>=0x0400 && FontCode<=0x045F)
        Address=(FontCode-0x0400) * 34 +0xA860;
else if (FontCode>=0x0490 && FontCode<=0x04FF)
        Address=(FontCode-0x0490+96) * 34 +0xA860;
```

5.7.8 16 dots Latin font set

Description:

unicode_alb：means unicode code #（16bits）

Address：byte address of Latin font in the chip.

Method :

```
BaseAdd=  0x139C0;
if( unicode_alb >= 0x0600 && unicode_alb <= 0x06FF )
        Address = 34*(unicode_alb-0x0600)+ 50176;
else if( unicode_alb >= 0xfb50 && unicode_alb <= 0xfbff )
        Address = 34*(16*16+unicode_alb-0xfb50)+ 50176;
else if( unicode_alb >= 0xfe70 && unicode_alb <= 0xfeff )
   Address = 34*(16*11+16*16+unicode_alb-0xfe70)+ 50176
```

6. Reliability

6.1Contents of Reliability Tests

Item	Conditions		Criteria
High Temperature Operation	85℃	TBD	
Low Temperature Operation	-40℃	TBD	
High Temperature Storage	90℃	TBD	
Low Temperature Storage	-45℃	TBD	
High Temperature/Humidity Operation	60	TBD	
Thermal Shock	-40℃to90℃	TBD	

* The samples used for the above tests do not include polarizer.

* No moisture condensation is observed during tests.

6.2 Lifetime

End of lifetime is specified as 50% of initial brightness.

Parameter	Min	Max	Unit	Condition	Notes
Operating Life Time	50,000	-	Hrs	80 cd/m2,50%checkerboard	6
Storage Life Time	100,000	-	Hrs	Ta=25 ,50%RH	-

Note 6: The average operating lifetime at room temperature is estimated by the accelerated operation at high temperature conditions.

6.3 Failure Check Standard

After the completion of the described reliability test, the samples were left at room temperature for 2 hrs prior to conducting the failure test at 23+/-5℃; 55+/-15%RH.

7. Appendixes

7.1 Display-module Software Initial Setting

```
Write_Command(0xae); /* set    display off */
Write_Command(0x02); /* set    lower column start address */
Write_Command(0x10); /* set    higher column start address */
Write_Command(0x40); /* set    display start line */
Write_Command(0x2E);
Write_Command(0x81); /* set    contrast control */
Write_Command(0x32);
Write_Command(0x82);
Write_Command(0x80);
Write_Command(0xa1); /* set    segment remap    */
Write_Command(0xa6); /* set    normal display */
Write_Command(0xa8); /* set    multiplex ratio */
Write_Command(0x3f); /* 1/64 */
Write_Command(0xad); /* master configuration */
Write_Command(0x8e); /* external vcc supply */
Write_Command(0xc8); /* set    com scan direction */
Write_Command(0xd3); /* set    display offset    */
Write_Command(0x40);
Write_Command(0xd5); /* set    display clock divide/oscillator frequency */
Write_Command(0xf0);
Write_Command(0xD8);        /*set area color mode off */
Write_Command(0x05);
Write_Command(0xD9);
Write_Command(0xF1);
Write_Command(0xda); /* set    com pin configuration */
Write_Command(0x12);
Write_Command(0x91);
Write_Command(0x3F);
Write_Command(0x3F);
Write_Command(0x3F);
 Write_Command(0x3F);
Write_Command(0xaf); /* set    display on */
```

8. Appendix

8.1 Unicode font set

Unicode font set collects totally LATIN, GREEK, THAI and ARABIC fonts.

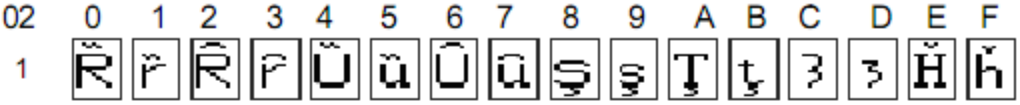
8.2 Arabic font set（496 fonts）

Unicode collects total 496 fonts of Arabic character.

Code # range: 0x20~0x70, 0xA0~0xFF, 0x0100~0x0170, 0x01A0~0x01CF, 0x01F0~0x01FF, 0x0210~0x021F, 0x1EA0~0x1EFF, 0x1EA0~0x1EFF.

Notes：the code # of Basic Latin (0020~007E) shares **with basic ASCII code area.**

Unicode字符区-拉丁文系

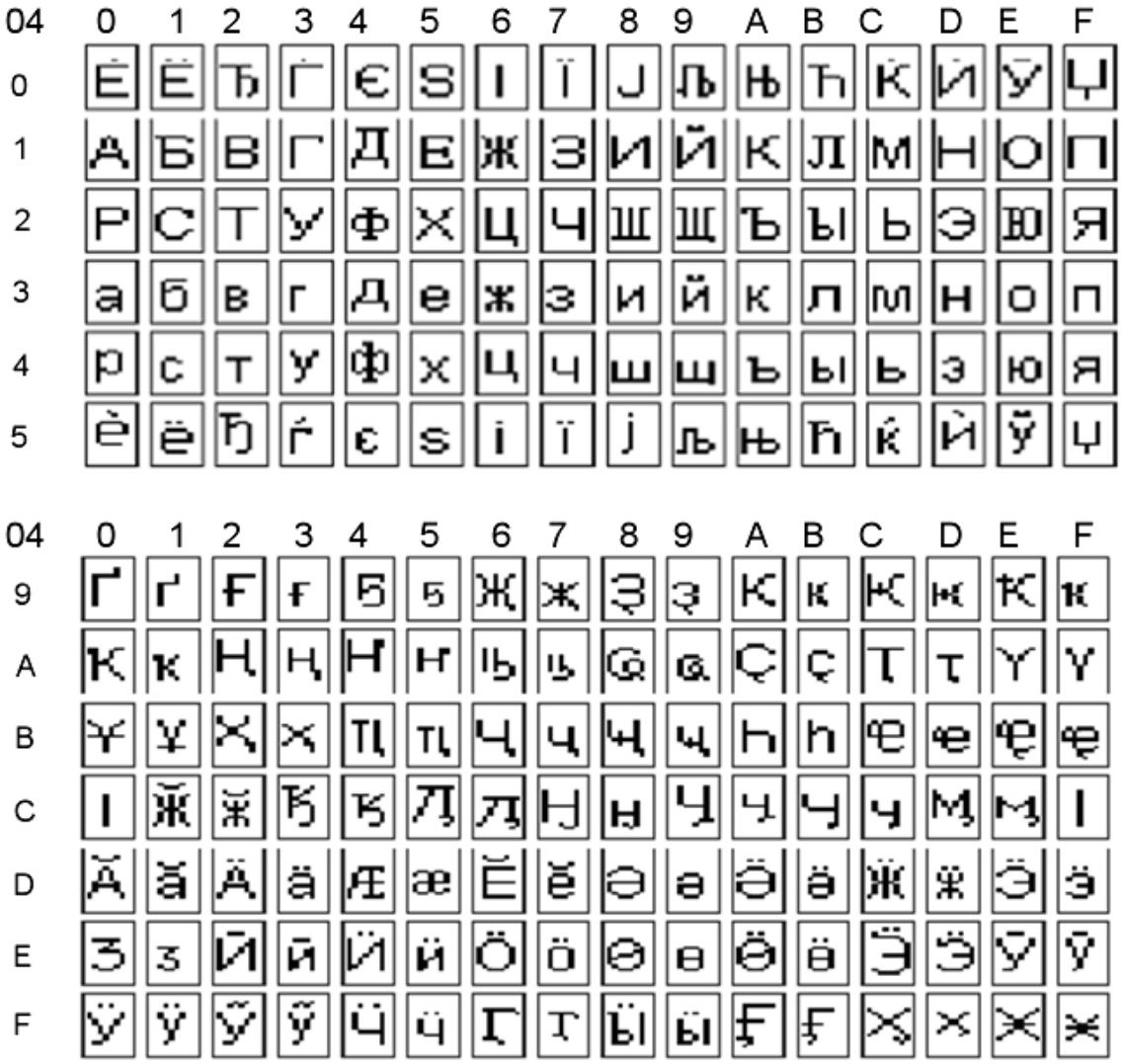


8.3 Cyrillic font set（208fonts）

Unicode area collects total 208 fonts of Cyrillic character.

Code # range: 0x0400~0x045F、0x0490~0x04FF。

Unicode area - Cyrillic font set



8.4 Greek font set（96 fonts）

Unicode area collects total 96 fonts of Greek character.

Code # range: 0x0370~0x03CF。

Unicode area - Greek font set

8.5 Hebrew font set（112 fonts）

Unicode area collects total 112 fonts of Hebrew character.

Code # range: 0x0590~0x05FF。

8.6 Thai font set（128 font）

Unicode area collects total 128 fonts of Thai character.

Code # range: 0x0E00~0x0E7F。

8.7 Latin font set (576 fonts)

Unicode area collects total 576 fonts of Latin character.

Code # range: 0x0600~0x06FF、0xFB50~0xFBFF、0xFE70~0xFEFF。

Unicode area - Latin font code set


0 1 2 3 4 5 6 7 8 9 A B C D E F
068
06F


FB5
FBB
FBD
FBF


FE7
0 1 2 3 4 5 6 7 8 9 A B C D E F
FEF