

GT21L16T1W

Standard Chinese Font Chip

DATASHEET

- Font size:15X16 dots
- Traditional Chinese character set:
GB12345, BIG5 basic set
- Japanese character set:
JIS0208, Compatible with Unicode
- Multi languages:
Languages of 150 countries including Latin, Cyril, Arabian
- Data arrangement: Horizontal byte, horizontal string
- Bus interface: SPI serial bus
- Chip package: SO8, COG

VER 3.0

2008-Q2

INDEX

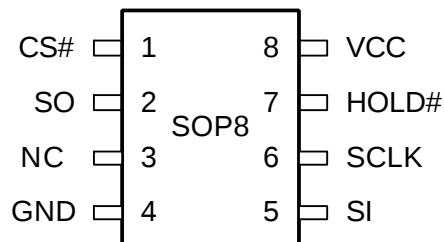
1 GENERAL	3
1.1 CHIP FEATURE	3
1.2 CHIP CONTENT.....	3
2 PIN DESCRIPTION AND INTERFACE CONNECTION.....	5
2.1 PIN CONFIGURATION.....	5
2.2 PIN DESCRIPTION.....	5
2.3 SPI CONNECTION BLOCK DIAGRAM	6
3 OPERATING INSTRUCTION	8
3.1 INSTRUCTION PARAMETER.....	8
3.2 READ DATA BYTES	8
3.3 READ DATA BYTES AT HIGHER SPEED	8
4 ELECTRIC CHARACTERISTIC	10
4.1 ABSOLUTE MAXIMUM RATING.....	10
4.2 DC CHARACTERISTIC	10
4.3 AC CHARACTERISTIC	10
5 PACKAGE SIZE	12
6 FONT READ METHOD	13
6.1 CHARACTER DOTS ARRANGEMENT.....	13
6.2 DOTS FONT ADDRESS TABLE	18
6.3 CALCULATION OF CHARACTER ADDRESS	18
7 APPENDIX	27
7.1 CHARACTER OF GB12345 (846 NON-CHINESE CHARACTERS)	27
7.2 CHARACTER OF UNICODE(NON- CHINESE CHARACTERS)	30
7.3 CHARACTER CHECKLIST (150 COUNTRIES)	39

1 General

GT21L16T1W is a 15X16 dots fonts chip, it supports Chinese standard GB12345 traditional Chinese character set (with official license from NIT), BIG5 traditional Chinese character basic set, Japanese standard JIS0208 Japanese character set(compatible with Unicode), and a total of 150 countries' character. The data is arrayed byte horizontally and string horizontally. The user may obtain the address of certain character dots with the calculation method given by this datasheet, which enables the user to get access of more character data by continually reading from the address already obtained.

1.1 Chip Feature

- Bus interface: SPI serial bus
- Data arrangement: byte horizontal, string horizontal
- Frequency: 30MHz(max.)@ 3.3V
- Supply voltage: 3.3V±10%
- Current:
 - Operating: 8mA
 - Standby: 8uA
- Package: SOP8, COG
- Package Size(SOP8): 4.90mmX3.90mm(193milX154mil)



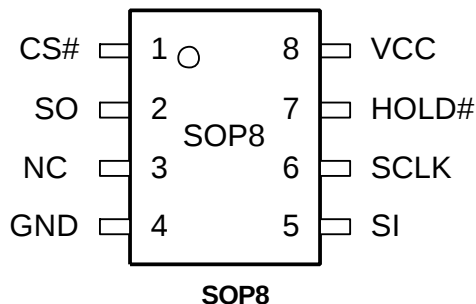
1.2 Chip Content

Type	Content	Character set	Characters
Chinese Font	15X16 dots GB12345 font	GB12345	6866+846
	15X16 dots BIG5 basic font	BIG5	5401+408
Japanese	15X16 dots JIS0208 basic font	JIS0208	6398+1644
ASCII font	5X7 dots ASCII font	ASCII	96
	7X8 dots ASCII font	ASCII	96
	6X12 dots ASCII font	ASCII	96
	8X16 dots ASCII font	ASCII	96
	8X16 dots bold ASCII font	ASCII	96
	12 dots Arial font	ASCII	96
	16 dots Arial font	ASCII	96
Unicode font	8X16 dots Latin font	Unicode	376
	8X16 dots Greek font	Unicode	96
	8X16 dots Cyril font	Unicode	250
	12 dots Unicode font (Latin, Greek, Cyril)	Unicode	555
	16 dots Unicode font (Latin, Greek, Cyril)	Unicode	555
	12 dots Arabia font	Unicode	250
	12 dots Arabia extendable font	Customize	498
	16 dots Arabia font	Unicode	250
	16 dots Arabia extendable font	Customize	498
Character set index table	BIG5 index table	BIG5	
	JIS0208 index table	JIS0208	
	Unicode index table	Unicode	

Language serial	language	country	Latin countries	countries
Latin	English	UK,USA etc.	39	112
	French	France, Niger etc.	22	
	Spanish	Mexico, Spain etc.	22	
	Portuguese	Portugal, Brazil etc.	7	
	German	Germany, Austria etc.	5	
	Italian	Italy, San Marino etc.	3	
	Malay	Malaysia, Brunei etc.	2	
	Swahili	Tanzania, Kenya etc.	2	
	other	Netherlands, Sweden etc.	10	
Arabian	Arabian	Egypt, Jordan etc.		21
Cyril	12 languages	Russia, Kazakhstan etc.		15
Greek	Greek	Greece, Cyprus etc.		2
				Sum 150

2 Pin description and interface connection

2.1 Pin configuration



2.2 Pin description

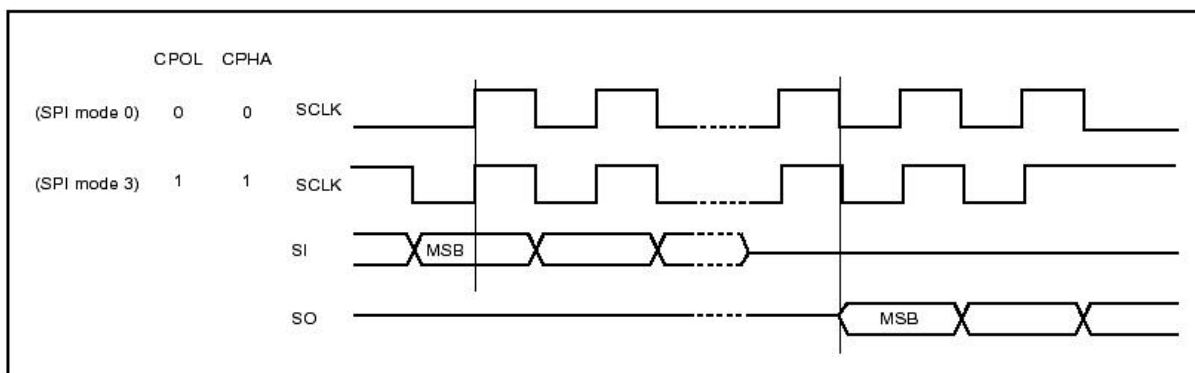
SOP8	name	I/O	description
1	CS#	I	Chip enable input
2	SO	O	Serial data output
3	NC		No Connected
4	GND		Ground
5	SI	I	Serial data input
6	SCLK	I	Serial clock input
7	HOLD#	I	Hold, to pause the device without
8	VCC		+ 3.3V Power Supply

Serial data output(SO): Data is shifted out on the falling edge of the serial clock.

Serial data input(SI) : Inputs are latched on the rising edge of the serial clock.

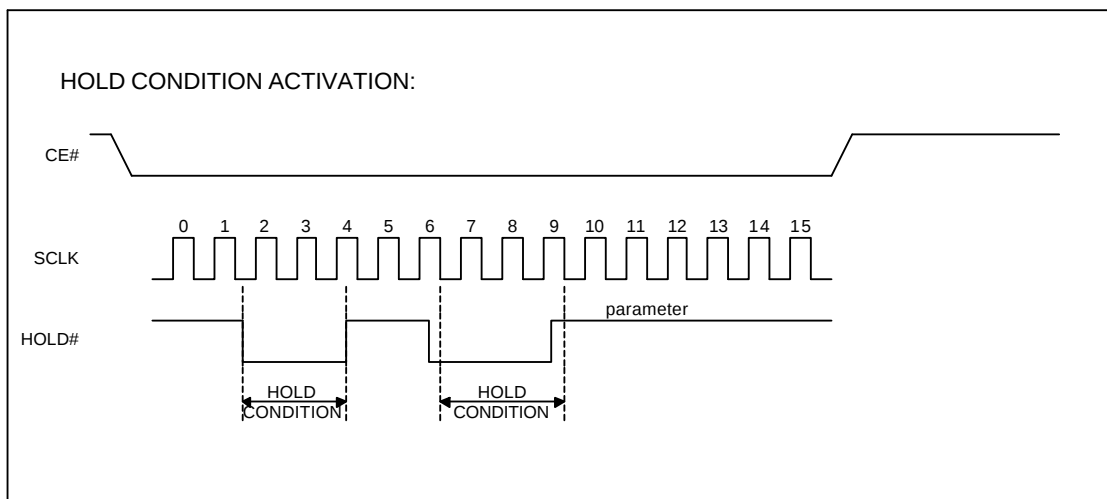
Serial clock input(SCLK): To provide the timing of the serial interface.

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.

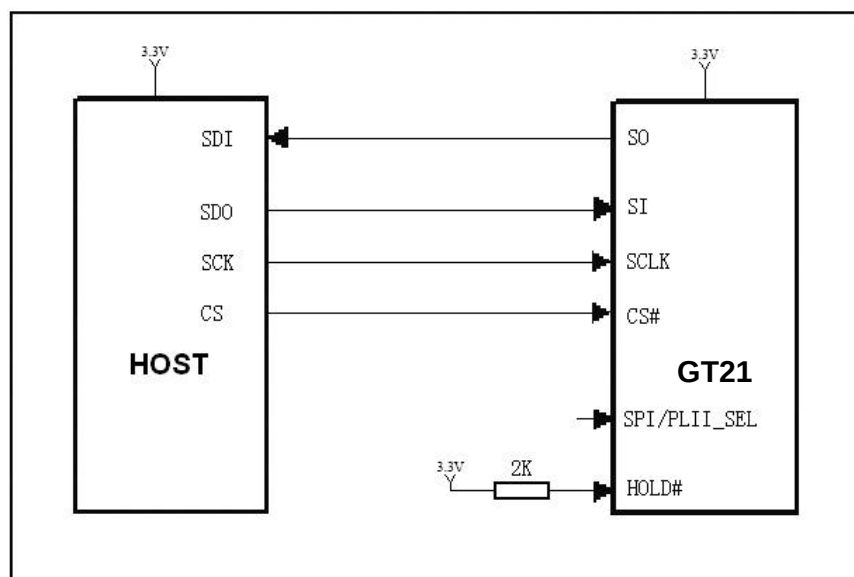


HOLD#: To temporarily stop serial communication with SPI flash memory without resetting the device.

The HOLD# mode begins when the SCK active low state coincides with the falling edge of the HOLD# signal. The HOLD mode ends when the HOLD# signal's rising edge coincides with the SCK active low state.

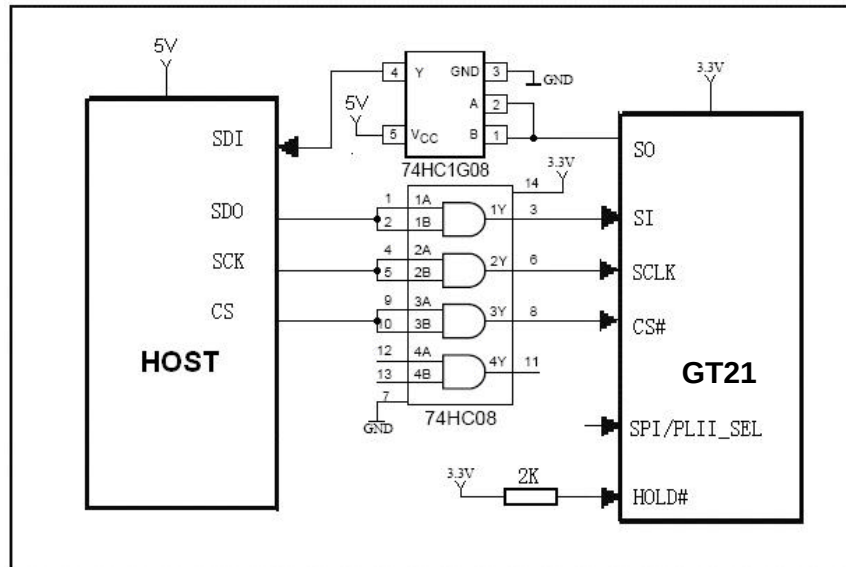


2.3 SPI connection block diagram



SPI connection block diagram

If system is supplied by 5V, the block diagram is bellowed:



3 Operating instruction

3.1 Instruction parameter

Instruction Set

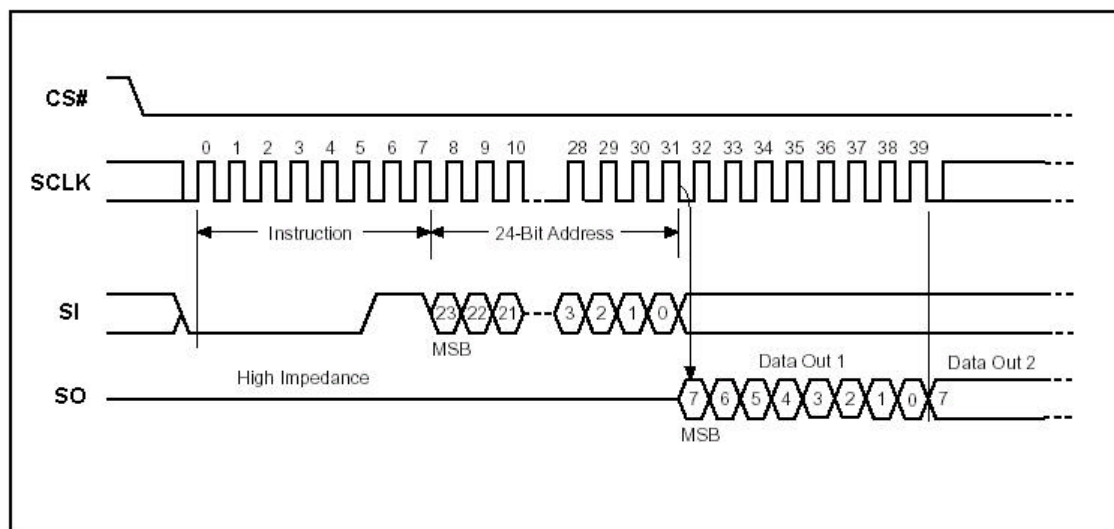
Instruction	Description	Instruction Code(One-Byte)		Address Bytes	Dummy Bytes	Data Bytes
READ	Read Data Bytes	0000 0011	03 h	3	—	1 to 8
FAST_READ	Read Data Bytes at Higher Speed	0000 1011	0B h	3	1	1 to 8

3.2 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 8bit command,03H, followed by address bits [A23-A0]. CE# must remain active low for the duration of the Read cycle.

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



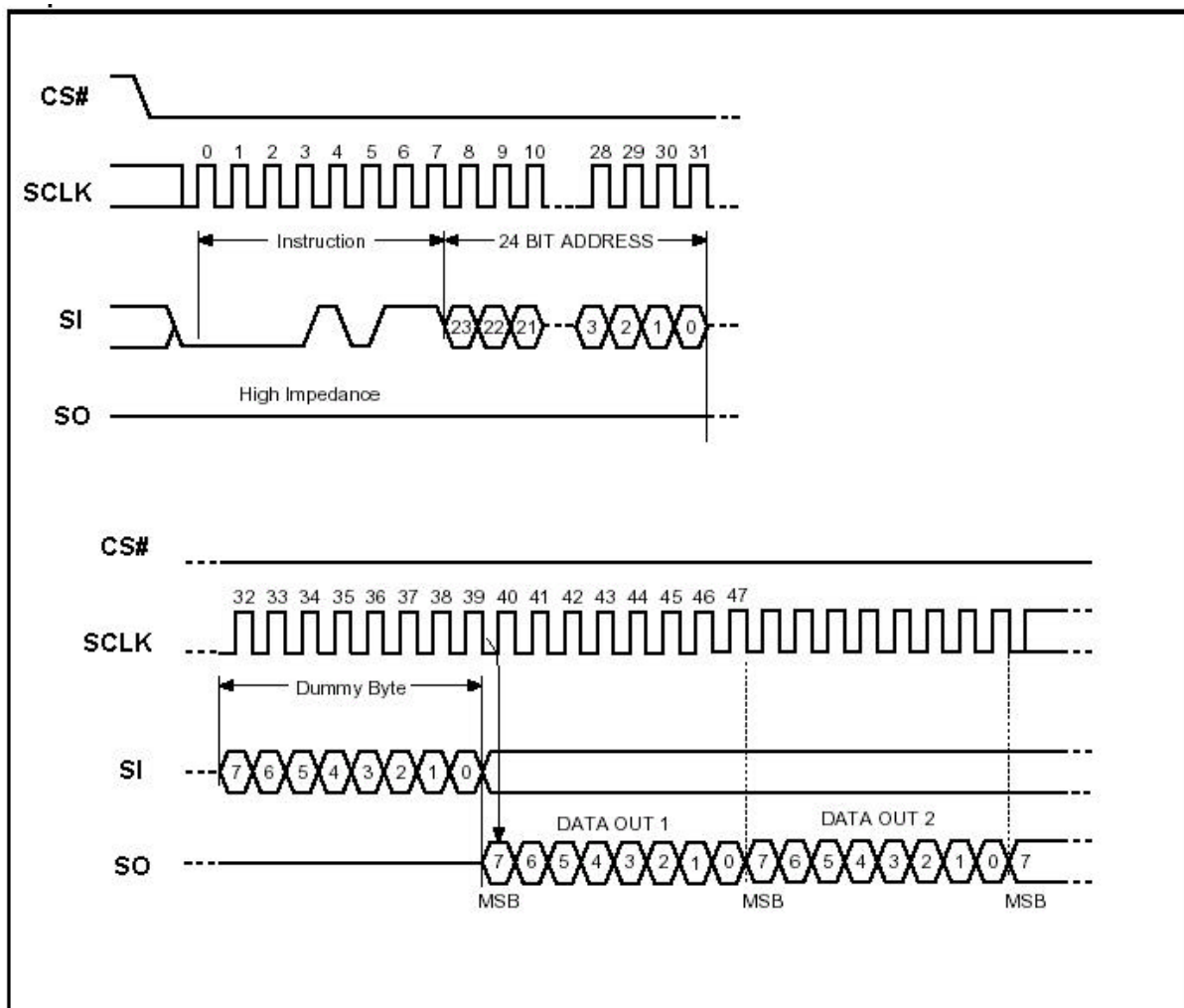
3.3 Read Data Bytes at Higher Speed

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:



4 Electric characteristic

4.1 Absolute maximum rating

Symbol	Parameter	Min.	Max.	Unit	Condition
T _{OP}	Operating Temperature	0	70		
T _{STG}	Storage Temperature	-65	150		
V _{CC}	Supply Voltage	-0.3	3.6	V	
V _{IN}	Input Voltage	-0.3	V _{CC} +0.3	V	
GND	Power Ground	-0.3	0.3	V	

4.2 DC characteristic

Condition : T_{OP} = 0 to 70 , GND=0V

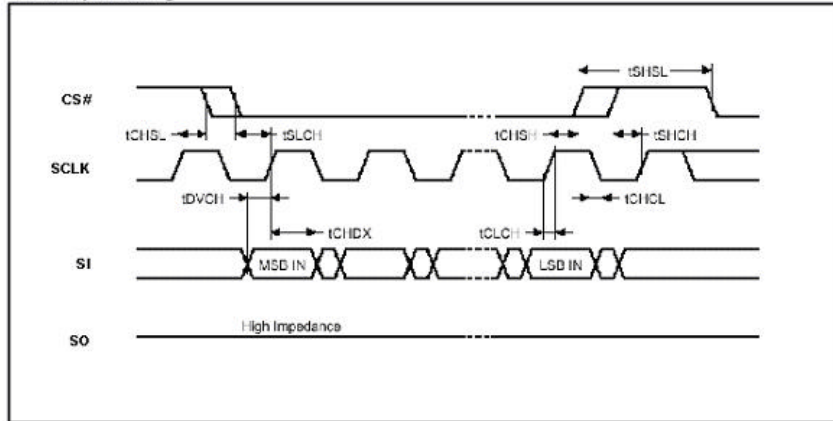
Symbol	Parameter	Min.	Max.	Unit	Condition
I _{DD}	V _{CC} Supply Current(active)		10	mA	
I _{SB}	V _{CC} Standby Current		10	uA	
V _{IL}	Input LOW Voltage	-0.3	0.3V _{CC}	V	V _{CC} =3.0-3.6V
V _{IH}	Input HIGH Voltage	0.7V _{CC}	V _{CC} +0.4	V	
V _{OL}	Output LOW Voltage		0.4 (I _{OL} =1.6mA)	V	
V _{OH}	Output HIGH Voltage	0.8V _{CC} (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 ,

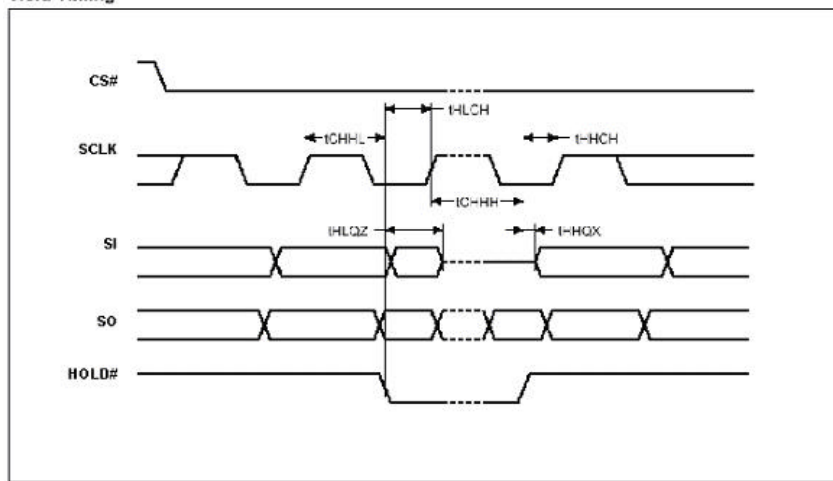
4.3 AC characteristic

Symbol	Alt.	Parameter	Min.	Max.	Unit
F _c	F _c	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}	t _v	Clock Low to Output Valid		9	ns
t _{CLQX}	t _{HO}	Output Hold Time	0		ns
t _{HLCH}		HOLD# Setup Time (relative to SCLK)	5		ns
t _{CHHH}		HOLD# Hold Time (relative to SCLK)	5		ns
t _{HHCH}		HOLD Setup Time (relative to SCLK)	5		ns
t _{CHHL}		HOLD Hold Time (relative to SCLK)	5		ns
t _{HHQX}	t _{LZ}	HOLD to Output Low-Z		9	ns
t _{HLQZ}	t _{HZ}	HOLD# to Output High-Z		9	ns

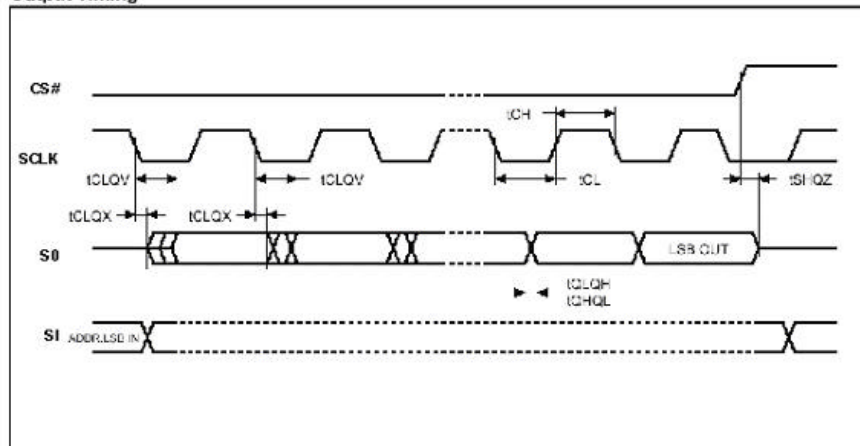
Serial Input Timing



Hold Timing



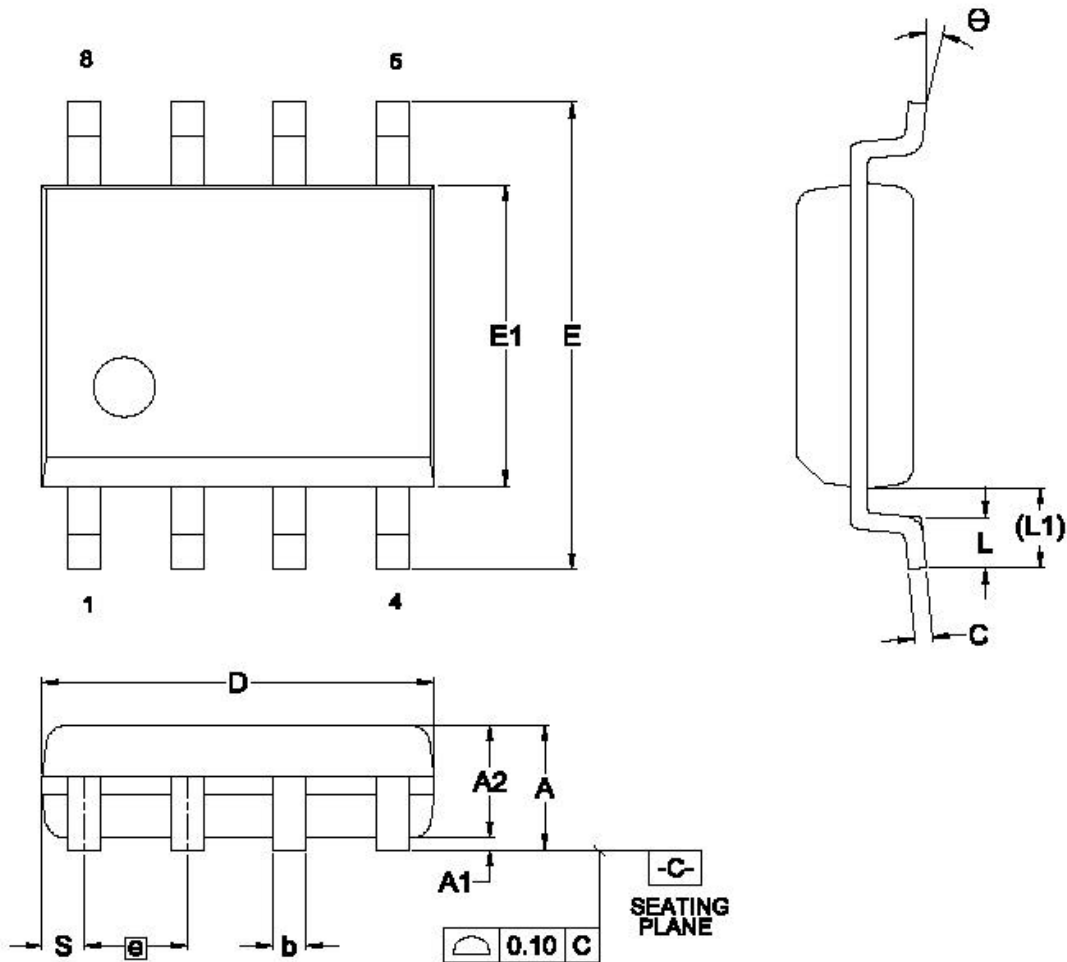
Output Timing



5 Package size

SOP8 Package

Unit :mm



Dimensions(inch dimensions are derived from the original mm dimensions)

		A	A1	A2	b	C	D	E	E1	⊗	L	L1	S	θ
Mm	Min.	-	0.10	1.35	0.36	0.15	4.77	5.80	3.60		0.46	0.65	0.41	0
	Norm.	-	0.15	1.45	0.41	0.20	4.90	5.99	3.90	1.27	0.66	1.05	0.54	5
	Max.	1.75	0.20	1.55	0.51	0.25	5.03	6.20	4.00		0.86	1.25	0.67	8
inch	Min.	-	0.004	0.053	0.014	0.006	0.188	0.228	0.150		0.018	0.033	0.016	0
	Norm.	-	0.006	0.057	0.016	0.008	0.193	0.236	0.154	0.050	0.026	0.041	0.021	5
	Max.	0.069	0.008	0.061	0.020	0.010	0.198	0.244	0.156		0.034	0.049	0.026	8

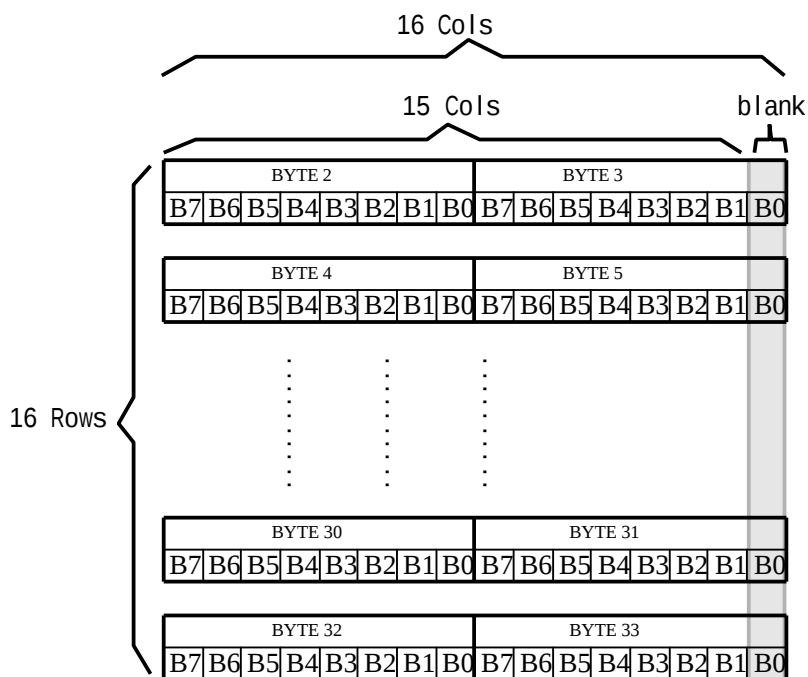
6 Font Read Method

6.1 Character dots arrangement

The data arrangement is byte horizontal, string horizontal. The highest Bit of BYTE represent left point, the lowest Bit of BYTE represent right point.

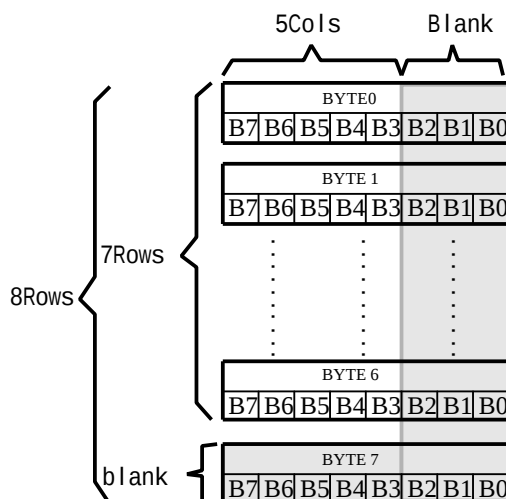
6.1.1 15X16 dots font

15X16 dots font has 32 bytes (BYTE 0 – BYTE 31) data.



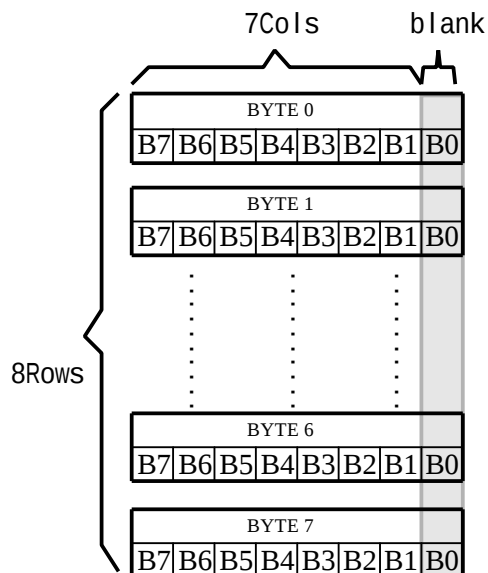
6.1.2 5X7dots ASCII font

5X7 dots ASCII font has 8 bytes (BYTE 0 – BYTE7) data.



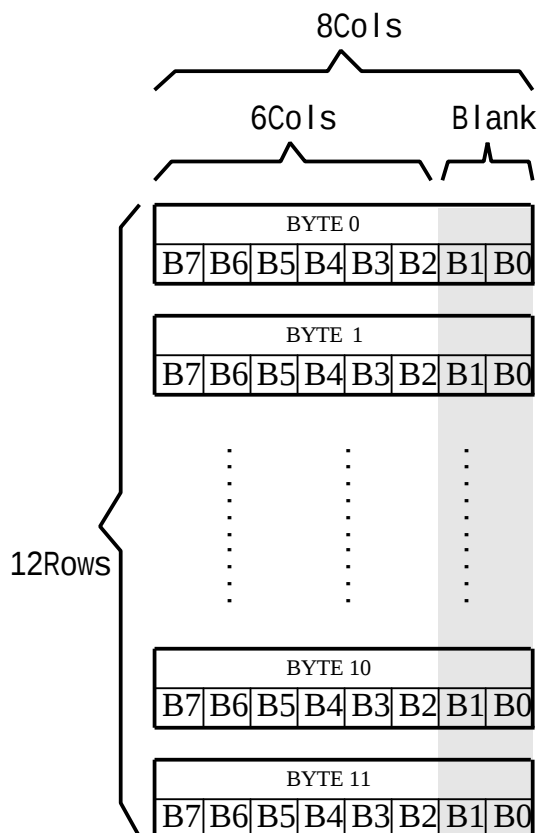
6.1.3 7X8 dots ASCII font

7X8 dots ASCII font has 8 bytes (BYTE 0 – BYTE7) data.



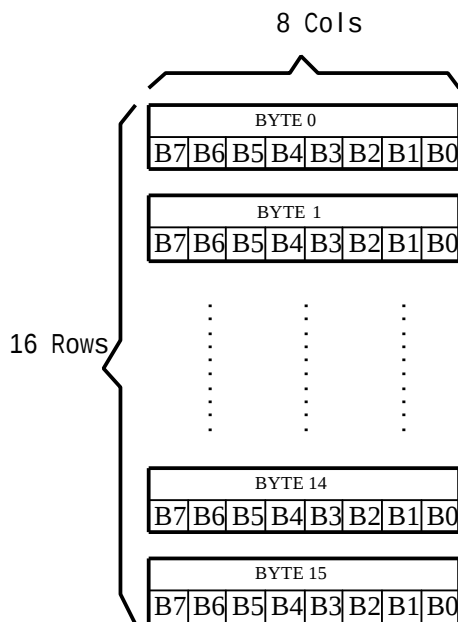
6.1.4 6X12 dots ASCII font

6X12 dots ASCII font has 12 bytes (BYTE 0 – BYTE11) data.



6.1.5 8X16 dots font

8X16 dots font has 16 bytes (BYTE 0 – BYTE15) data:



6.1.6 12 dots proportional font

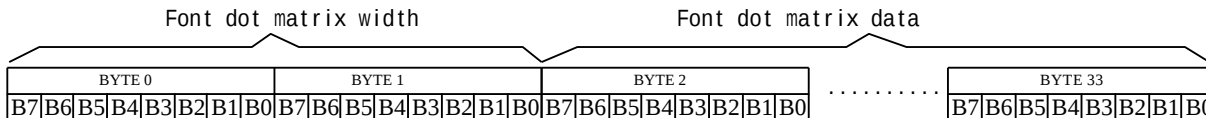
Including:

12 dots Arial font

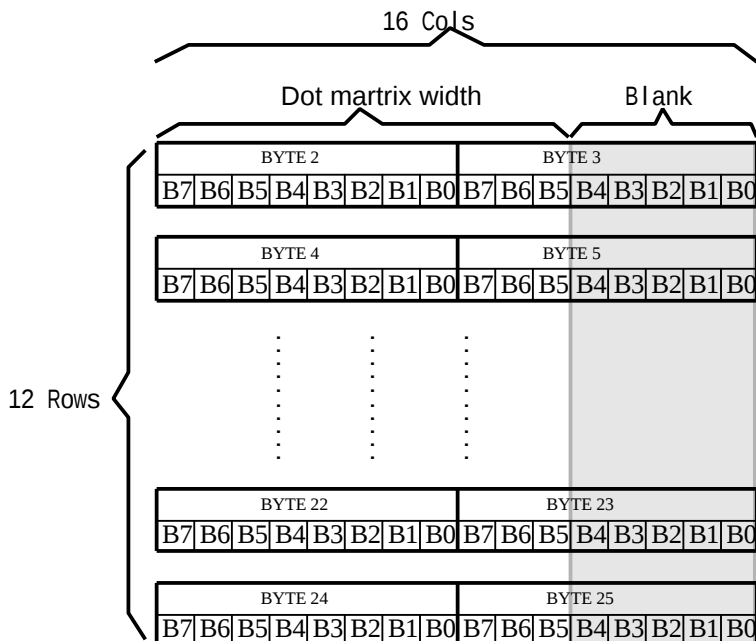
12 dots Unicode font

12 dots proportional font has 26 bytes (BYTE 0 – BYTE25) data.

For the font is variant width, BYTE0~ BYTE1 are stored font width data,.BYTE2-25 are stored dots matrix data.



The dots matrix width of proportional font use 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.



6.1.7 16 dots proportional font

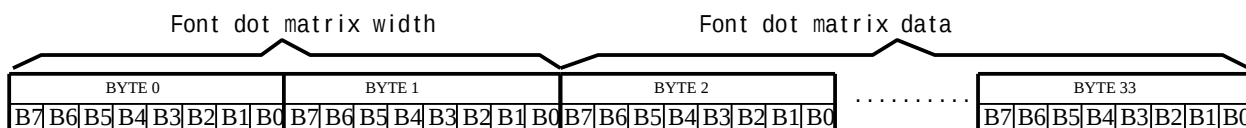
Including:

16 dots Arial font

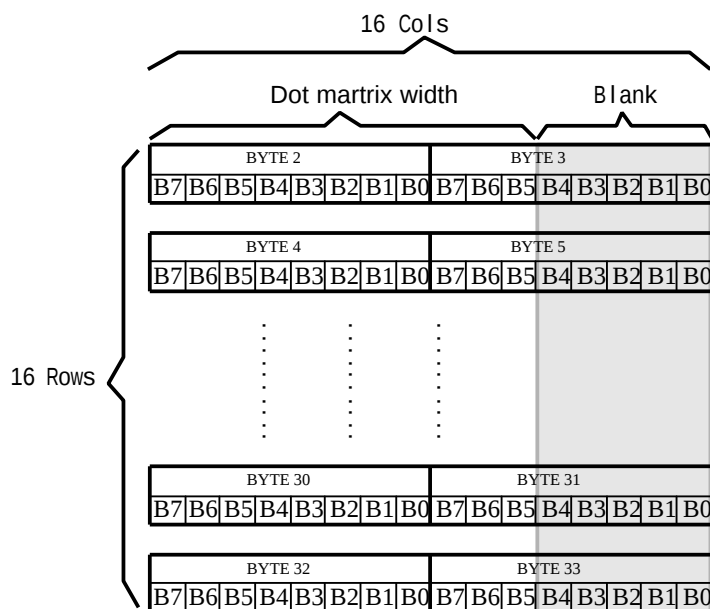
16 dots Unicode font

16 dots proportional font has 34 bytes (BYTE 0 – BYTE33) data.

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



The dot matrix width of proportional font use 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.



6.2 Dots font address table

	content	charset	Code scope	Characters	Address	Reference method
1	15X16dots GB12345 font	GB12345	A1A1-F9A9	6866+846	00000	6.3.1.1
2	5X7 dots ASCII font	ASCII	20~7F	96	51980	6.3.2.1
3	7X8 dots ASCII font	ASCII	20~7F	96	51C80	6.3.2.2
4	6X12 dots ASCII font	ASCII	20~7F	96	51F80	6.3.2.3
5	8X16 dots ASCII font	ASCII	20~7F	96	52580	6.3.2.4
6	8X16 dots bold ASCII font	ASCII	20~7F	96	52D80	6.3.2.5
7	12 dots Arial font	ASCII	20~7F	96	53380	6.3.2.6
8	12 dots Unicode font (Latin, Greek, Cyril)	Unicode	0020-04E9	555	53380	6.3.3.4- 6.3.3.6
9	16 dots Arial font	ASCII	20~7F	96	56BDE	6.3.2.7
10	16 dots Unicode font (Latin, Greek, Cyril)	Unicode	0020-04E9	555	56BDE	6.3.3.7- 6.3.3.9
11	8X16 dots Latin font	Unicode	00A0-0217	376	5B594	6.3.3.1
12	8X16 dots Greek font	Unicode	0370-03CF	96	5CD14	6.3.3.2
13	8X16 dots Cyril font	Unicode	0400-04F9	250	5D314	6.3.3.3
14	JIS0208 index table	JIS0208	0101-8794	6398+1644	5E2B4	6.3.1.3
15	BIG5 index table	BIG5	A140-C67E	5401+408	62410	6.3.1.2
16	Unicode index table	Unicode	A0-FF50		651F0	6.3.1.4
17	16 dots Arabia font	Unicode	0600~06F9	250	715E0	6.3.3.12
18	16 dots Arabia extendable font	Customize	B000-B1F1	498	73714	6.3.3.13
19	12 dots Arabia font	Unicode	0600~06F9	250	78790	6.3.3.10
20	12 dots Arabia extendable font	Customize	B000-B1F1	498	7A0F4	6.3.3.11
21	reserved				7DE80	

6.3 Calculation of character address

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

6.3.1 Chinese font and Japanese font

6.3.1.1 15X16 dots GB12345 font

Parameters:

GBCode: character code.

MSB: high byte of GBCode.

LSB: low byte of GBCode.

Address: address of character data in chip.

BaseAdd: the base address of the font in chip

Calculation of character address:

BaseAdd=0x00;

if(MSB >=0xA1 && MSB <= 0xA9 && LSB >=0xA1)

Address =(MSB - 0xA1) * 94 + (LSB - 0xA1) * 32 + BaseAdd;

else if(MSB >=0xB0 && MSB <= 0xF9 && LSB >=0xA1)

Address = ((MSB - 0xB0) * 94 + (LSB - 0xA1) + 1038) * 32 + BaseAdd;

6.3.1.2 15X16 dots BIG5 basic font

Parameters:

FontCode: character code.

MSB: high byte of FontCode.

LSB: low byte fo FontCode.

Address: address of character data in chip.

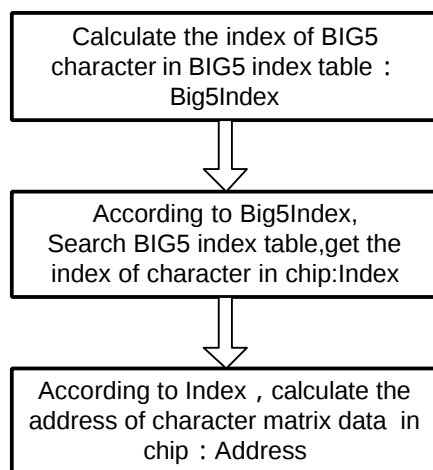
BaseAdd: the base address of the font in chip

Big5Table: BIG5 index table. The table address is 0x62410

Big5Index: index of the character in BIG5 index table.

Index : index of the character in font sets.

Calculation of character address:



BaseAdd=0x00;

if(MSB >=0xA1 && MSB <= 0xC6)

{

if(LSB >=0x40 && LSB <= 0x7E)

Big5Index =(MSB - 0xA1) * 157 + (LSB - 0x40);

else if(LSB >=0xA1 && LSB <= 0xFE)

Big5Index =(MSB - 0xA1) * 157 + 63 + (LSB - 0x40));

}

Index = Big5Table[Big5Index*2] * 256 + Big5Table[Big5Index*2+1];

Address =Index * 32 + BaseAdd;

6.3.1.3 15X16 dots JIS font

Parameters:

FontCode: character code.

MSB: high byte of FontCode.

LSB: low byte fo FontCode.

Address: address of character data in chip.

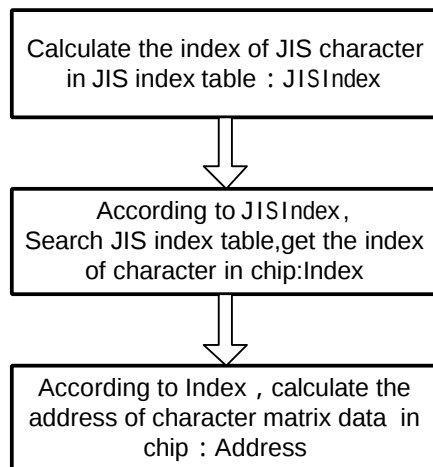
BaseAdd: the base address of the font in chip

JISTable: JIS index table. The table address is 0x5E2B4

JISIndex: index of the character in JIS index table.

Index : index of the character in font sets

Calculation of character address:



```

BaseAdd=0x00;
if ((MSB >=1 && MSB <= 89) && (LSB >=1 && LSB <= 94) )
{
    JISIndex = (MSB -1)*94 +LSB-1;
}
Index = JISTable[JISIndex *2] * 256 + JISTable[JISIndex *2+1] ;
Address =Index * 32 + BaseAdd;
  
```

6.3.1.4 15X16 dots Unicode Japanese font

Parameters:

FontCode: character code.

Address: address of character data in chip.

BaseAdd: the base address of the font in chip

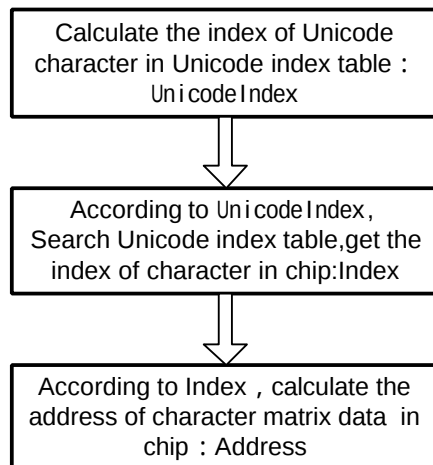
UnicodeTable: Unicode index table. The table address is 0x651F0

UnicodeToIndex(): calculate index of the character in Unicode index table.

UnicodeIndex: index of the character in JIS index table.

Index : index of the character in font sets

Calculation of character address:



```

BaseAdd=0x00;
  
```

```
UnicodeIndex = UnicodeToIndex(FontCode);
```

```
Index = UnicodeTable [UnicodeIndex *2] * 256 + UnicodeTable [UnicodeIndex *2+1] ;
```

```
Address =Index * 32 + BaseAdd;
```

```
WORD UnicodeToIndex(WORD code)
```

```
{
```

```
    BYTE result=0;
```

```
    WORD h;
```

```
    if(code<=0x20) code = 0x3000;
```

```
    else if(code<0x7f)
```

```
    {
```

```
        code=0xfe57-code-0x21;
```

```
    }
```

```
    if(code<=0xa0) result=1;
```

```
    else if(code<=0x0451) h=code-160;
```

```
    else if(code< 0x2010) result=1;
```

```
    else if(code<=0x2642) h=code-160-7102;
```

```
    else if(code< 0x3000) result=1;
```

```
    else if(code<=0x33d5) h=code-160-7102-2493;
```

```
    else if(code< 0x4e00) result=1;
```

```
    else if(code<=0x9fa5) h=code-160-7102-2493-6698;
```

```
    else if(code< 0xe76c) result=1;
```

```
    else if(code<=0xe864) h=code-160-7102-2493-6698-18374;
```

```
    else if(code< 0xf92c) result=1;
```

```
    else if(code<=0xfa29) h=code-160-7102-2493-6698-18374-4295;
```

```
    else if(code< 0xfe30) result=1;
```

```
    else if(code<=0xfe6b) h=code-160-7102-2493-6698-18374-4295-1030;
```

```
    else if(code< 0xff01) result=1;
```

```
    else if(code<=0xff5e) h=code-160-7102-2493-6698-18374-4295-1030-149;
```

```
    else if(code< 0xffe0) result=1;
```

```
    else if(code<=0xffe5) h=code-160-7102-2493-6698-18374-4295-1030-149-129;
```

```
    else result=1;
```

```
    if(result==1)
```

```
    {
```

```
        h = 0x3000-160-7102-2493;
```

```
    }
```

```
    return h;
```

```
}
```

6.3.2 ASCII font

6.3.2.1 5X7 dots ASCII font

Parameters:

ASCIICode: ASCII code(8 bits)

BaseAdd: the base address of the font in chip

Address: address of character data in chip.

Calculation of character address:

BaseAdd=0x51980

if (ASCIICode >= 0x20) and (ASCIICode <= 0x7E) then

Address = (ASCIICode - 0x20) * 8 + BaseAdd

6.3.2.2 7X8 dots ASCII font

Parameters:

ASCIICode: ASCII code(8 bits)

BaseAdd: the base address of the font in chip

Address: address of character data in chip.

Calculation of character address:

BaseAdd=0x51C80

if (ASCIICode >= 0x20) and (ASCIICode <= 0x7E) then

Address = (ASCIICode - 0x20) * 8 + BaseAdd

6.3.2.3 6X12 dots ASCII font

Parameters:

ASCIICode: ASCII code(8 bits)

BaseAdd: the base address of the font in chip

Address: address of character data in chip.

Calculation of character address:

BaseAdd=0x51F80

if (ASCIICode >= 0x20) and (ASCIICode <= 0x7E) then

Address = (ASCIICode - 0x20) * 12 + BaseAdd

6.3.2.4 8X16 dots ASCII font

Parameters:

ASCIICode: ASCII code(8 bits)

BaseAdd: the base address of the font in chip

Address: address of character data in chip.

Calculation of character address:

BaseAdd=0x52580

if (ASCIICode >= 0x20) and (ASCIICode <= 0x7E) then

Address = (ASCIICode - 0x20) * 16 + BaseAdd

6.3.2.5 8X16 dots ASCII bold font

Parameters:

ASCIICode: ASCII code(8 bits)

BaseAdd: the base address of the font in chip

Address: address of character data in chip.

Calculation of character address:

BaseAdd=0x52D80

if (ASCIICode >= 0x20) and (ASCIICode <= 0x7E) then

Address = (ASCIICode - 0x20) * 16 + BaseAdd

6.3.2.6 12 dots arial font

Parameters:

ASCIICode: ASCII code(8 bits)

BaseAdd: the base address of the font in chip

Address: address of character data in chip.

Calculation of character address:

BaseAdd=0x53380

if (ASCIICode >= 0x20) and (ASCIICode <= 0x7E) then

Address = (ASCIICode - 0x20) * 26 + BaseAdd

6.3.2.7 16 dots arial font

Parameters:

ASCIICode: ASCII code(8 bits)

BaseAdd: the base address of the font in chip

Address: address of character data in chip.

Calculation of character address:

BaseAdd=0x56BDE

if (ASCIICode >= 0x20) and (ASCIICode <= 0x7E) then

Address = (ASCIICode - 0x20) * 34 + BaseAdd

6.3.3 Unicode font

6.3.3.1 8X16 dots Latin font

Parameters:

BaseAdd: the base address of the font in chip

FontCode: Unicode code (16bits)

Address: address of character data in chip

Calculation of character address:

BaseAdd = 0x5B594

if (FontCode >= 0x00A0) and (FontCode <= 0x0217) then

Address = (FontCode - 0x00A0) * 16 + BaseAdd

6.3.3.2 8X16 dots Greek font

Parameters:

BaseAdd: the base address of the font in chip

FontCode: Unicode code (16bits)

Address: address of character data in chip

Calculation of character address:

BaseAdd = 0x5CD14

if (FontCode >= 0x0370) and (FontCode <= 0x03CF) then

$$\text{Address} = (\text{FontCode} - 0x00A0) * 16 + \text{BaseAdd}$$

6.3.3.3 8X16 dots Cyril font

Parameters:

BaseAdd: the base address of the font in chip

FontCode: Unicode code (16bits)

Address: address of character data in chip

Calculation of character address:

BaseAdd=0x5D314

if (FontCode >= 0x0400) and (FontCode <=0x04F9) then

$$\text{Address} = (\text{FontCode} - 0x0400) * 16 + \text{BaseAdd}$$

6.3.3.4 12 dots Latin font

Parameters:

BaseAdd: the base address of the font in chip

FontCode: Unicode code (16bits)

Address: address of character data in chip

Calculation of character address:

BaseAdd=0x53380

if (FontCode >= 0x0020) and (FontCode <=0x007F) then

$$\text{Address} = (\text{FontCode} - 0x0020) * 26 + \text{BaseAdd}$$

Else if (FontCode >= 0x00A0) and (FontCode <=0x017F) then

$$\text{Address} = (\text{FontCode} - 0x00A0) * 26 + \text{BaseAdd}$$

6.3.3.5 12 dots Breece font

Parameters:

BaseAdd: the base address of the font in chip

FontCode: Unicode code (16bits)

Address: address of character data in chip

Calculation of character address:

BaseAdd=0x53380+350*26

if (FontCode >= 0x0384) and (FontCode <=0x03CE) then

$$\text{Address} = (\text{FontCode} - 0x0384) * 26 + \text{BaseAdd}$$

6.3.3.6 12 dots Cyril font

Parameters:

BaseAdd: the base address of the font in chip

FontCode: Unicode code (16bits)

Address: address of character data in chip

Calculation of character address:

BaseAdd=0x53380+425*26

if (FontCode >= 0x0400) and (FontCode <=0x045F) then

$$\text{Address} = (\text{FontCode} - 0x0400) * 26 + \text{BaseAdd}$$

Else if (FontCode >= 0x0490) and (FontCode <=0x04a3) then

$$\text{Address} = (\text{FontCode} - 0x0490 + 96) * 26 + \text{BaseAdd}$$

Else if (FontCode >= 0x04AE) and (FontCode <=0x04B3) then

$Address = (FontCode - 0x04AE + 117) * 26 + BaseAdd$
 Else if (FontCode $\geq$ 0x04B8) and (FontCode $\leq$ 0x04BB) then
 $Address = (FontCode - 0x04B8 + 122) * 26 + BaseAdd$
 Else if (FontCode $\geq$ 0x04D8) and (FontCode $\leq$ 0x04D9) then
 $Address = (FontCode - 0x04D8 + 126) * 26 + BaseAdd$
 Else if (FontCode $\geq$ 0x04E8) and (FontCode $\leq$ 0x04E9) then
 $Address = (FontCode - 0x04E8 + 128) * 26 + BaseAdd$

6.3.3.7 16 dots Latin font

Parameters:

BaseAdd: the base address of the font in chip

FontCode: Unicode code (16bits)

Address: address of character data in chip

Calculation of character address:

BaseAdd=0x56BDE

if (FontCode $\geq$ 0x0020) and (FontCode $\leq$ 0x007F) then

$Address = (FontCode - 0x0020) * 34 + BaseAdd$

Else if (FontCode $\geq$ 0x00A0) and (FontCode $\leq$ 0x017F) then

$Address = (FontCode - 0x0040) * 34 + BaseAdd$

6.3.3.8 16 dots Greek font

Parameters:

BaseAdd: the base address of the font in chip

FontCode: Unicode code (16bits)

Address: address of character data in chip

Calculation of character address:

BaseAdd=0x56BDE + 350*34

if (FontCode $\geq$ 0x0384) and (FontCode $\leq$ 0x03CE) then

$Address = (FontCode - 0x0384) * 34 + BaseAdd$

6.3.3.9 16 dots Cyril font

Parameters:

BaseAdd: the base address of the font in chip

FontCode: Unicode code (16bits)

Address: address of character data in chip

Calculation of character address:

BaseAdd=0x56BDE + 425*34

if (FontCode $\geq$ 0x0400) and (FontCode $\leq$ 0x045F) then

$Address = (FontCode - 0x0400) * 34 + BaseAdd$

Else if (FontCode $\geq$ 0x0490) and (FontCode $\leq$ 0x04a3) then

$Address = (FontCode - 0x0490 + 96) * 34 + BaseAdd$

Else if (FontCode $\geq$ 0x04AE) and (FontCode $\leq$ 0x04B3) then

$Address = (FontCode - 0x04AE + 117) * 34 + BaseAdd$

Else if (FontCode $\geq$ 0x04B8) and (FontCode $\leq$ 0x04BB) then

$Address = (FontCode - 0x04B8 + 122) * 34 + BaseAdd$

Else if (FontCode >= 0x04D8) and (FontCode <=0x04D9) then

Address = (FontCode-0x04D8+126) * 34+BaseAdd

Else if (FontCode >= 0x04E8) and (FontCode <=0x04E9) then

Address = (FontCode-0x04E8+128) * 34+BaseAdd

6.3.3.10 12 dots Arabian font

Parameters:

BaseAdd: the base address of the font in chip

FontCode: Unicode code (16bits)

Address: address of character data in chip

Calculation of character address:

BaseAdd=0x78790

if (FontCode >= 0x0600) and (FontCode <=0x06F9) then

Address = (FontCode-0x0600) * 26+BaseAdd

6.3.3.11 12 dots Arabian extendable font

Parameters:

BaseAdd: the base address of the font in chip

FontCode: Unicode code (16bits)

Address: address of character data in chip

Calculation of character address:

BaseAdd=0x7A0F4

if (FontCode >= 0xB000) and (FontCode <=0XB1F1) then

Address = (FontCode-0xB000) * 26+BaseAdd

6.3.3.12 16 dots Arabian font

Parameters:

BaseAdd: the base address of the font in chip

FontCode: Unicode code (16bits)

Address: address of character data in chip

Calculation of character address:

BaseAdd=0x715E0

if (FontCode >= 0x0600) and (FontCode <=0x06F9) then

Address = (FontCode-0x0600) * 34+BaseAdd

6.3.3.13 16 dots Arabian extendable font

Parameters:

BaseAdd: the base address of the font in chip

FontCode: Unicode code (16bits)

Address: address of character data in chip

Calculation of character address:

BaseAdd=0x73714

if (FontCode >= 0xB000) and (FontCode <=0XB1F1) then

Address = (FontCode-0xB000) * 34+BaseAdd

7 Appendix

7.1 Character of GB12345 (846 Non-Chinese characters)

Corresponding codes: A1A1~A9E

GB12345 1 section

A1	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A			、	。	·	—	√	”	々	—	~		...	‘	’	
B	“	”	{	}	<	>	《	》	「	」	『	』	【	】		
C	±	×	÷	:	^	v	Σ	Π	U	∩	€	::	√	⊥	//	∠
D	∩	⊙	∫	ℳ	≡	≈	≈	∞	≠	≠	≠	≠	≠	≠	∞	∴
E	∴	↑	♀	°	'	”	℃	\$	⊗	⊗	£	%	§	No	☆	★
F	○	●	◎	◇	◆	□	■	△	▲	※	→	←	↑	↓	=	

A2	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		i	ii	iii	iv	v	vi	vii	viii	ix	x					
B		1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	15.
C	16.	17.	18.	19.	20.	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)
D	(12)	(13)	(14)	(15)	(16)	(17)	(18)	(19)	(20)	①	②	③	④	⑤	⑥	⑦
E	⑧	⑨	⑩	€		(一)	(二)	(三)	(四)	(五)	(六)	(七)	(八)	(九)	(十)	
F		I	II	III	IV	V	VI	VII	VIII	IX	X	XI	XII			

A3	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		!	"	#	¥	%	&	'	(	)	*	+	,	-	.	/
B	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
C	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
D	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
E	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
F	p	q	r	s	t	u	v	w	x	y	z	{		}	—	

GB12345 1 section

A4	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		あ	い	う	え	お	か	き	く							
B	ぐ	け	こ	さ	し	ず	せ	そ	た							
C	だ	ち	っ	つ	づ	て	ど	な	ぬ	ね	の	は				
D	ば	び	び	ふ	ぶ	へ	べ	ほ	ま	み						
E	む	め	も	や	ゆ	よ	ら	り	る	れ	ろ	わ				
F	ゐ	ゑ	を	ん												

A5	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		ア	イ	ウ	エ	オ	カ	キ	ク							
B	グ	ケ	コ	サ	シ	ズ	セ	ソ	タ							
C	ダ	チ	ツ	ヅ	テ	ト	ナ	ニ	ノ	ハ						
D	バ	ビ	フ	ブ	ヘ	ベ	ホ	ボ	マ	ミ						
E	ム	メ	モ	ヤ	ユ	ヨ	ラ	リ	ル	レ	ロ	ワ				
F	ヰ	ヱ	ヲ	ヅ	カ	ケ										

A6	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		A	B	Γ	Δ	E	Z	H	Θ	I	K	Λ	M	N	Ξ	Ο
B	Π	P	Σ	T	τ	Φ	X	Ψ	Ω							
C		α	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο
D	π	ρ	σ	τ	υ	φ	χ	ψ	ω	,	°	`	:	;	!	?
E	ˆ	˘	˙	˚	˛	˜	˝	˞	˟	ˠ	ˡ	ˢ	ˣ	ˤ	˥	˦
F	˧	˨	˩	˪	˫	ˬ	˭	ˮ	˯	˰	˱	˲	˳	˴	˵	˶

GB12345 1 section

A7	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н
B		О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь
C		Ю	Я													
D		а	б	в	г	д	е	ё	ж	з	и	й	к	л	м	н
E		о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь
F		ю	я													

A8	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		ā	á	ǎ	à	ē	é	ě	è	ī	í	ǎ	ì	ō	ó	ǒ
B		ò	ū	ú	ǔ	ù	ǖ	Ǘ	Ǚ	ǚ	ǜ	ê	ɑ	ǎ	ń	ň
C		g				勺	夕	冂	匚	勹	去	ㄣ	ㄤ	ㄥ	ㄨ	ㄩ
D		ㄣ	ㄤ	ㄥ	ㄨ	ㄩ	ㄨ	ㄩ	ㄨ	ㄩ	ㄨ	ㄩ	ㄨ	ㄩ	ㄨ	ㄩ
E		ㄣ	ㄤ	ㄥ	ㄨ	ㄩ	ㄨ	ㄩ	ㄨ	ㄩ	ㄨ	ㄩ	ㄨ	ㄩ	ㄨ	ㄩ
F																

A9	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A					—	—			---	---	!	!	---	---	!	!
B		┐	┐	┐	┐	┐	┐	┐	┐	┐	┐	┐	┐	┐	┐	┐
C		┌	┌	┌	┌	┌	┌	┌	┌	┌	┌	┌	┌	┌	┌	┌
D		└	└	└	└	└	└	└	└	└	└	└	└	└	└	└
E		+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
F																

7.2 Character of Unicode(Non- Chinese characters)

Contains Latin, Greek, Cyril (456 characters), and Arabian (250 characters)

7.2.1 8×16 dots Latin fonts (376 characters)

Corresponding codes: 00A0~0217(contains ASCII)

Character of Unicode (Latin)																
00	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		ı	ç	£	α	¥	ı	§	ˆ	©	ª	«	¬	-	®	-
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	ø	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

01	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	Ā	ā	Ǻ	ǻ	Ȧ	ȧ	Ć	ć	Ĉ	ĉ	Ċ	ċ	Č	č	Ď	ď
1	Đ	đ	Ē	ē	Ě	ě	Ẹ	ẹ	Ẽ	ẽ	Ĝ	ĝ	Ğ	ğ	Ġ	ġ
2	Ġ	ġ	Ģ	ģ	Ĥ	ĥ	Ħ	ħ	Í	í	Ī	ī	Ĭ	ĭ	Į	į
3	İ	ı	U	u	Ĵ	ĵ	Ʒ	Ʒ	Ʒ	Ʒ	Ʒ	Ʒ	Ʒ	Ʒ	Ʒ	Ʒ
4	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı
5	Ō	ō	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	Ů	ů	Ű	ű	Ŵ	ŵ	Ŷ	ŷ	Ÿ	Ž	ž	Ž	ž	Ž	ž	f

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

02	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	Ä	ä	Â	â	Ë	ë	Ê	ê	Ï	ï	Î	î	Ö	ö	Ô	ô
1	Ř	ř	Ŕ	ŕ	Ů	ů	Ů	ů								

7.2.2 8×16 dots Cyril fonts(250 characters)

Corresponding codes: 0400~04F9

Character of Unicode (Cyril)																
04	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		Ё	Ъ	Ѓ	Є	Ѕ	І	Ї	Ј	Љ	Њ	Ћ	Ќ	Й	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		ё	ђ	ѓ	є	ѕ	і	ї	ј	љ	њ	ћ	ќ		ў	џ
6	Ѡ	ѡ	Ѣ	ѣ	Ѥ	ѥ	Ѧ	ѧ	Ѩ	ѩ	Ѫ	ѫ	Ѭ	ѭ	Ѯ	ѯ
7	Ѱ	ѱ	Ѳ	ѳ	Ѵ	ѵ	Ѷ	ѷ	Ѹ	ѹ	Ѻ	ѻ	Ѽ	ѽ	Ѿ	ѿ

04	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8	Г	г	Х	Г	Г	Г	Г									
9	Г	Г	Г	Г	Б	Б	Ж	Ж	Э	Э	К	К	К	К	К	К
A	К	К	Н	Н	Н	Н	Ь	Ь	О	О	С	С	Т	Т	У	У
B	У	У	Х	Х	Ц	Ц	Ч	Ч	Ч	Ч	Н	Н	Е	Е	Е	Е
C	І	Ж	Ж	Б	Б			Н	Н			Ч	Ч			
D	А	а	А	а	Æ	æ	Ё	ё	Ө	ө	Ө	ө	Ж	Ж	Э	э
E	Э	э	Й	Й	Й	Й	Ö	ö	Ө	ө	Ө	ө			У	у
F	У	у	У	у	Ч	Ч			Ы	Ы						

7.2.3 8×16 dots Greek fonts (96 characters)

Corresponding codes: 0370~03CF

Character of Unicode (Greek)																
03	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
7					'	,					~				,	
8					'	"	Α	·	Ε	Η	Ι		Ο		Τ	Ω
9	ι	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο
A	Π	Ρ		Σ	Τ	Υ	Φ	Χ	Ψ	Ω	Ϊ	Ϋ	Ό	Ε	Η	ι
B	ϐ	α	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο
C	π	ρ	ς	σ	τ	υ	φ	χ	ψ	ω	ϊ	ϋ	ό	ύ	ώ	

7.2.4 16 dots Arabian fonts(250 characters)

Corresponding codes: 0600~06F9

06	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0													ء			
1												ء				؟
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	-	ف	ق	ك	ل	م	ن	ه	و	ى	ي	°	°	°	°	°
5	-	°	°													
6	+	١	٢	٣	٤	٥	٦	٧	٨	٩	/	,	ء	*		
7	'	أ	إ	ل	°	ا	و	ؤ	ئ	ث	ت	ب	پ	ت	ب	ت

06	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8	ب	خ	خ	ج	ج	خ	ج	ج	ظ	د	د	ظ	د	د	د	د
9	د	ظ	ز	ر	ر	ر	ر	ر	ز	ر	ش	ش	ش	ش	ش	ظ
A	غ	ف	ب	ف	ف	ف	ف	ق	ق	ك	ك	ه	ك	ك	ك	ك
B	ه	ه	ه	ه	ه	ه	ه	ه	ه	ه	ه	ه	ه	ه	ه	ه
C	ه	ه	ه	ه	ه	ه	ه	ه	ه	ه	ه	ه	ه	ه	ه	ه
D	ي	ي	ي	ي	ي	ي	ي	ي	ي	ي	ي	ي	ي	ي	ي	ي
E	'	°	°	°	°	°	°	°	°	°	°	°	°	°	°	°
F	+	١	٢	٣	٤	٥	٦	٧	٨	٩						

7.2.5 16 dots Arabian extendable fonts(498 characters)

B0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0																
1																
2																
3																
4																
5																
6																
7																
8																
9																
A																
B																
C																
D																
E																
F																

B1	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0																
1																
2																
3																
4																
5																
6																
7																
8																
9																
A																
B																
C																
D																
E																
F																

7.3 Character Checklist (150 countries)

No.	country	Area	language	Language serial
1	Malaysia	Asia	Malay	Latin
2	Brunei	Asia	Malay, English	Latin
3	Indonesia	Asia	Indonesian	Latin
4	Philippines	Asia	English	Latin
5	Sikkim	Asia	English	Latin
6	UK	Europe	English	Latin
7	Ireland	Europe	English	Latin
8	USA	North America	English	Latin
9	Canada	North America	English, French	Latin
10	Australia	Oceania	English	Latin
11	New Zealand	Oceania	English	Latin
12	Germany	Europe	German	Latin
13	Switzerland	Europe	German, French	Latin
14	Austria	Europe	German	Latin
15	Luxemburg	Europe	German, French	Latin
16	Liechtenstein	Europe	German	Latin
17	Italy	Europe	Italian	Latin
18	Vatican	Europe	Italian	Latin
19	San Marino	Europe	Italian	Latin
20	Denmark	Europe	Denish	Latin
21	Iceland	Europe	Icelandic	Latin
22	Norway	Europe	Norwegian	Latin
23	Sweden	Europe	Swedish	Latin
24	Finland	Europe	Finnish, Swedish	Latin
25	Netherlands	Europe	Dutch	Latin
26	Suriname	South America	Dutch	Latin
27	The Faroe Islands	Europe	Faeroese	Latin
28	Portugal	Europe	Portuguese	Latin
29	Brazil	South America	Portuguese	Latin
30	Cape Vrde	Africa	Portuguese	Latin
31	Guinea Bissau	Africa	Portuguese	Latin
32	Sao Tome&Principe	Africa	Portuguese	Latin
33	Angora	Africa	Portuguese	Latin
34	Mozambique	Africa	Portuguese	Latin
35	France	Europe	French	Latin
36	Belgium	Europe	French, Dutch	Latin
37	Monaco	Europe	French, Italian	Latin
38	Haiti	North America	French	Latin
39	Senegal	Africa	French	Latin
40	Mali	Africa	French	Latin
41	Burkina Faso	Africa	French	Latin
42	Guinea	Africa	French	Latin
43	Cote d'Ivoir	Africa	French	Latin

No.	country	Area	language	Language serial
44	Togo	Africa	French	Latin
45	Benin	Africa	French	Latin
46	Niger	Africa	French	Latin
47	Cameroon	Africa	French	Latin
48	Chad	Africa	French	Latin
49	Central Africa Rep.	Africa	French	Latin
50	Djibouti	Africa	French	Latin
51	Burundi	Africa	French	Latin
52	Congo,DR	Africa	French	Latin
53	Congo	Africa	French	Latin
54	Gabon	Africa	French	Latin
55	Comoros	Africa	French	Latin
56	Madagascar	Africa	French	Latin
57	Spain	Europe	Spanish, Catalan	Latin
58	Mexico	North America	Spanish	Latin
59	Guatemala	North America	Spanish	Latin
60	Costa Rica	North America	Spanish	Latin
61	Panama	North America	Spanish	Latin
62	Dominican Rep.	North America	Spanish	Latin
63	El Salvador	North America	Spanish	Latin
64	Honduras	North America	Spanish	Latin
65	Nicaragua	North America	Spanish	Latin
66	Puerto Rica	North America	Spanish	Latin
67	Cuba	North America	Spanish	Latin
68	Venezuela	South America	Spanish	Latin
69	Colombia	South America	Spanish	Latin
70	Peru	South America	Spanish	Latin
71	Argentina	South America	Spanish	Latin
72	Ecuador	South America	Spanish	Latin
73	Chile	South America	Spanish	Latin
74	Uruguay	South America	Spanish	Latin
75	Paraguay	South America	Spanish	Latin
76	Bolivia	South America	Spanish	Latin
77	Eq.Guinea	Africa	Spanish	Latin
78	Ceuta&Melilla	Africa	Spanish	Latin
79	Jamaica	North America	English	Latin
80	Belize	North America	English	Latin
81	Trinidad&Tobago	North America	English	Latin
82	Bahamas	North America	English	Latin
83	Antigua&Barbuda	North America	English	Latin
84	Dominica	North America	English	Latin
85	Saint Vincent&Grenadines	North America	English	Latin
86	Grenada	North America	English	Latin
87	Cayman Is.	North America	English	Latin

No.	country	Area	language	Language serial
88	St. Kitts-Nevis	North America	English	Latin
89	Tonga	Oceania	English	Latin
90	Fiji	Oceania	English	Latin
91	Solomon Is.	Oceania	English	Latin
92	Vanuatu	Oceania	English	Latin
93	Kiribati	Oceania	English	Latin
94	Nauru	Oceania	English	Latin
95	Marshall Is Rep	Oceania	English	Latin
96	Zimbabwe	Africa	English	Latin
97	Gambia	Africa	English	Latin
98	Sierra Leone	Africa	English	Latin
99	Liberia	Africa	English	Latin
100	Ghana	Africa	English	Latin
101	Nigeria	Africa	English	Latin
102	Uganda	Africa	English	Latin
103	Zambia	Africa	English	Latin
104	Malawi	Africa	English	Latin
105	Seychelles	Africa	English	Latin
106	Mauritius	Africa	English	Latin
107	Botswana	Africa	English	Latin
108	Namibia	Africa	English	Latin
109	Lesotho	Africa	English	Latin
110	South Africa	Africa	Dutch, English	Latin
111	Kenya	Africa	Swahili	Latin
112	Tanzania	Africa	Swahili	Latin
113	Egypt	Africa	Arabian	Arabian
114	Tunisia	Africa	Arabian	Arabian
115	Libyan Arab Jm	Africa	Arabian	Arabian
116	Morocco	Africa	Arabian	Arabian
117	Algeria	Africa	Arabian	Arabian
118	Sudan	Africa	Arabian	Arabian
119	Somalia	Africa	Arabian	Arabian
120	Djibouti	Africa	Arabian	Arabian
121	Mauritania	Africa	Arabian	Arabian
122	Syrian	Asia	Arabian	Arabian
123	United Arab Emirates	Asia	Arabian	Arabian
124	Lebanon	Asia	Arabian	Arabian
125	Yemen Rep.	Asia	Arabian	Arabian
126	Kuwait	Asia	Arabian	Arabian
127	Qatar	Asia	Arabian	Arabian
128	Palestine	Asia	Arabian	Arabian
129	Bahrian	Asia	Arabian	Arabian
130	Oman	Asia	Arabian	Arabian
131	Jordan	Asia	Arabian	Arabian

No.	country	Area	language	Language serial
132	Iraq	Asia	Arabian	Cyril
133	Saudi Arabia	Asia	Arabian	Cyril
134	Russia	Europe	Russian	Cyril
135	Byelorussia	Europe	Russian	Cyril
136	Ukraine	Europe	Ukrainian	Cyril
137	Bulgari	Europe	Bulgarian	Cyril
138	Macedonia Rep.	Europe	Macedonian	Cyril
139	Yugoslavia FR	Europe	Serbian	Cyril
140	Crotia Rep	Europe	Serbian	Cyril
141	Bosnia&Herzegovina	Europe	Serbian	Cyril
142	Azerbaijan	Asia	Azeri	Cyril
143	Kyrgyz Rep.	Asia	Kirghiz	Cyril
144	Tadzhikistan	Asia	Tadzhikistani	Cyril
145	Turkmenistan	Asia	Turkoman	Cyril
146	Uzbekstan	Asia	Uzbekstani	Cyril
147	Kazakhstan	Asia	Kazak	Cyril
148	Mongolia	Asia	Mongol	Cyril
149	Greek	Europe	Greek	Greek
150	Cyprus	Asia	Greek	Greek

In the 150 countries,112 countries are Latin serial language,21 countries are Arabian serial language, 15 countries are Cyril serial language, 2 countries are Greek serial language.

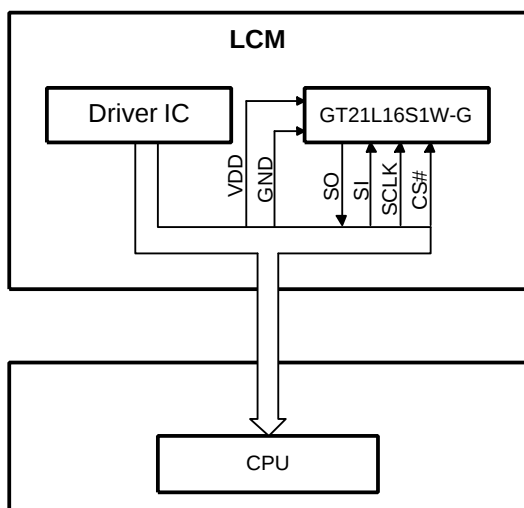
In the Latin serial countries, 39 countries are English, 22 countires are French, 22 countries are Spanish, 7 countries are Portuguese, 5 countries are German, 3 countries are Italian, 2 countries are Malay, 2 countries are Swahili, 10 countries are other Latin language,

GT21L16T1W-G

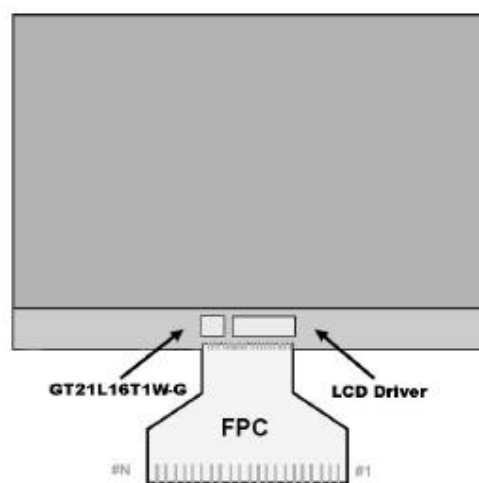
FOR LCM Application (COB/COG)

- SPI Serial Interface(4pins).
- Supports Chinese standard GB12345 traditional Chinese character set (with official license from NIT) (6866 characters)
- Supports BIG5 traditional Chinese character basic set(5401 characters)
- Japanese standard JIS0208 Japanese character set(compatible with Unicode)
- a total of 150 countries' character(compatible with Unicode)
- LCM minimize pixels : 96X32 pixels(6 Chinese characters X 2 rows).

LCM Block Diagram(COG/COB)



LCM Structure(COG)



COG adopt Gold Bump process

COG chip size

