

GT20L16J1Y 标准字库芯片

数据手册 DATASHEET

- JIS0208 日本文汉字（6355 字符）：15x16 点阵
- JIS0208 日本文字符（1644 字符）：15x16 点阵
- 日文假名（128 字）：半角 8x16 点阵
- ASCII（128 字）：半角 8x16 点阵
- ASCII（96 字）：粗体 8x16 点阵

- 排置方式：竖置横排
- 总线接口：SPI 串行总线
- 芯片形式：SOT23-6 封装

VER 4.0

2011-3

目 录

第一部分：硬件部分

| | |
|--|-----------|
| 1. 概述 | 3 |
| 芯片特点..... | 3 |
| 2. 引脚描述 | 3 |
| 3. 芯片内容 | 4 |
| 3.1 字库内容列表..... | 4 |
| 3.2 字型样张..... | 4 |
| 3.3 字库内容分配表..... | 5 |
| 4. HOST CPU 主机 SPI 接口电路示意图 | 5 |
| 5. 操作指令 | 6 |
| 5.1 指令参数..... | 6 |
| 5.2 Read Data Bytes（一般读取）..... | 6 |
| 5.3 Read Data Bytes at Higher Speed（快速读取点阵数据）..... | 7 |
| 6. 电气特性 | 8 |
| 6.1 绝对最大额定值..... | 8 |
| 6.2 DC 特性..... | 8 |
| 6.3 AC 特性..... | 9 |
| 7. 封装尺寸 | 10 |

第二部分：软件部分

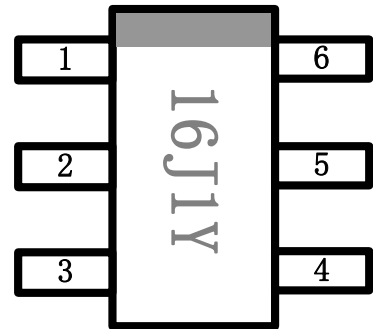
| | |
|------------------------|-----------|
| 8. 字库调用方法 | 11 |
| 8.1 字符点阵排列格式..... | 11 |
| 8.2 字符在芯片中的地址计算方法..... | 12 |
| 9. 附录 | 15 |

1. 概述

GT21L16J1Y是一款支持日本文的字库芯片，支持JIS0208字符集、支持8x16 ASCII粗体、支持256的8x16 ASCII日本假名字符。排列格式为竖置横排。用户通过JIS0208字符内码，利用本手册提供的方法计算出该字符点阵在芯片中的地址，可从该地址连续读出字符点阵信息。

芯片特点

- 数据总线：SPI 串行总线接口
- 点阵排列方式：字节竖置横排
- 时钟频率：30MHz(max.) @3.3V
- 工作电压：2.2V~3.6V
- 电流：
 - 工作电流：8mA
 - 待机电流：8uA
- 封装：SOT23-6
- 尺寸 SOT23-6：2.9mmX1.6 mm x1.10mm
工作温度：-20℃~70℃



2. 引脚描述

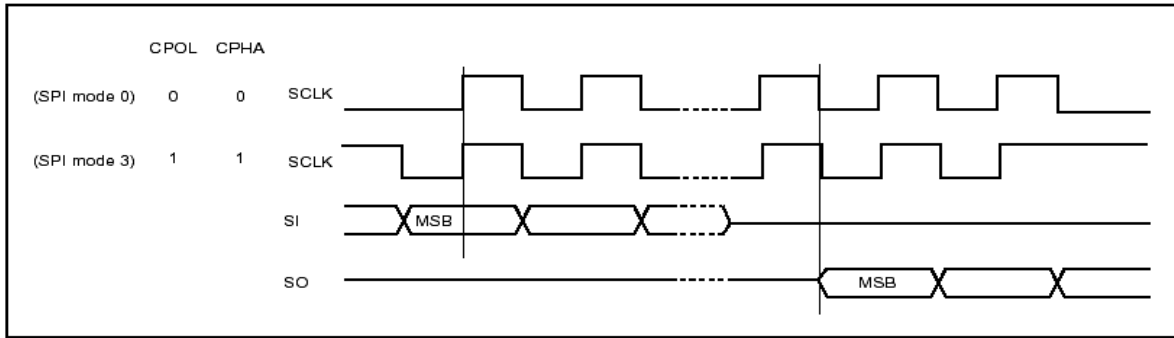
| SOT23-6 | 名称 | I/O | 描述 |
|---------|------|-----|-----------------------------|
| 1 | SCLK | I | 串行时钟输入 (Serial clock input) |
| 2 | GND | | 地(Ground) |
| 3 | CS# | I | 片选输入 (Chip enable input) |
| 4 | VCC | | 电源(+ 3.3V Power Supply) |
| 5 | SO | O | 串行数据输出 (Serial data output) |
| 6 | SI | I | 串行数据输入 (Serial data input) |

串行数据输出 (SO)：该信号用来把数据从芯片串行输出，数据在时钟的下降沿移出。

串行数据输入 (SI)：该信号用来把数据从串行输入芯片，数据在时钟的上升沿移入。

串行时钟输入 (SCLK)：数据在时钟上升沿移入，在下降沿移出。

片选输入 (CS#)：所有串行数据传输开始于CE#下降沿，CE#在传输期间必须保持为低电平，在两条指令之间保持为高电平。



3. 芯片内容

3.1 字库内容列表

| 字符集 | 点阵 | | |
|-------------|------|---------|-------|
| | 8x16 | 8x16 粗体 | 15x16 |
| JIS0208日文符号 | | | 1644 |
| JIS0208日文汉字 | | | 6355 |
| 日文假名 (半角) | 128 | | |
| ASCII (半角) | 128 | | |
| ASCII (半角) | | 96 | |

3.2 字型样张

JIS0208 日文字符

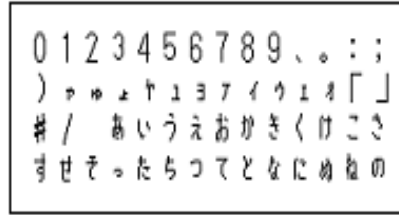
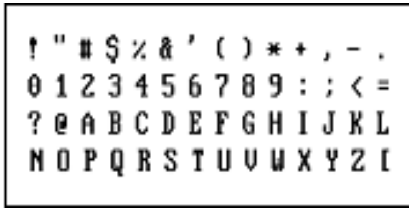
◆□■▲△※→※←↑↓≡€
 ああいううええおおかがきぎく
 亜啞娃阿哀愛挨始逢葵茜穉惡握渥
 院陰隱韻吋右宇鳥羽迂雨卯鵝窺丑

8x16 ASCII 粗体字符

JIS0208 日文汉字

亜啞娃阿哀愛挨始逢葵茜穉惡握渥
 伊位依偉田夷委威尉惟意慰易椅為
 印咽員因姻引飲淫胤蔭院陰隱韻吋
 荏餌馥宮嬰影映曳榮永泳洩瑛盈穎

8x16 ASCII 日本假名

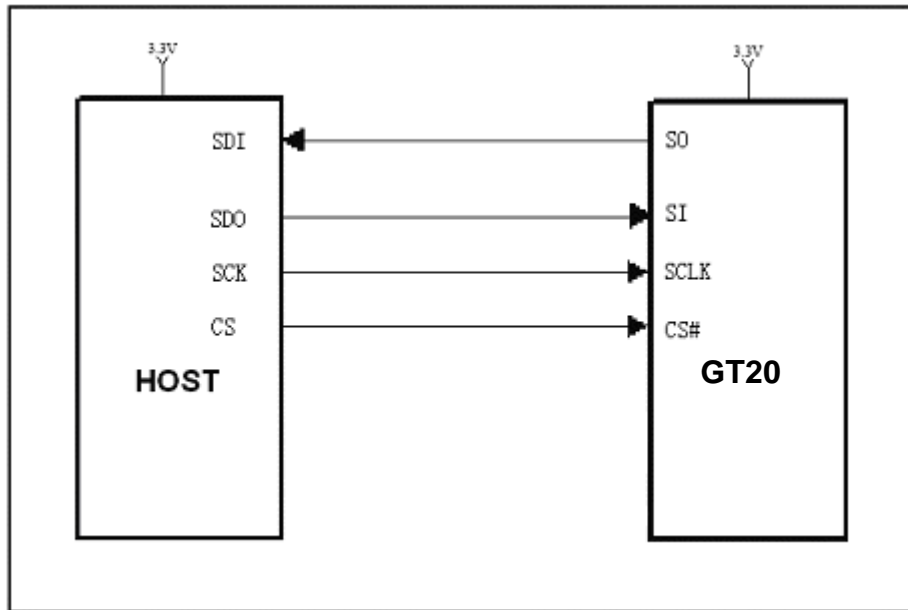


3.3 字库内容分配表

| 分类 | 字库内容 | 起始地址 | 编码体系(字符集) | 字符数 |
|------------|---------------------|--------|-----------|------|
| JIS0208 字符 | JIS0208 符号 | 0 | JIS0208 | 1644 |
| | JIS0208 日本字 | 43584 | JIS0208 | 6355 |
| ASCII 字符 | 8x16 ASCII 粗体字符(半角) | 255968 | ASCII | 96 |
| | 8x16 ASCII 日文假名(半角) | 257504 | ASCII | 256 |

4. HOST CPU 主机 SPI 接口电路示意图

SPI 与主机接口电路连接可以参考下图。



HOST CPU 主机 SPI 接口电路示意图

5. 操作指令

5.1 指令参数

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 ∞ |

所有对本芯片的操作只有 2 个，那就是 Read Data Bytes (READ “一般读取”)和 Read Data Bytes at Higher Speed (FAST_READ “快速读取点阵数据”)。

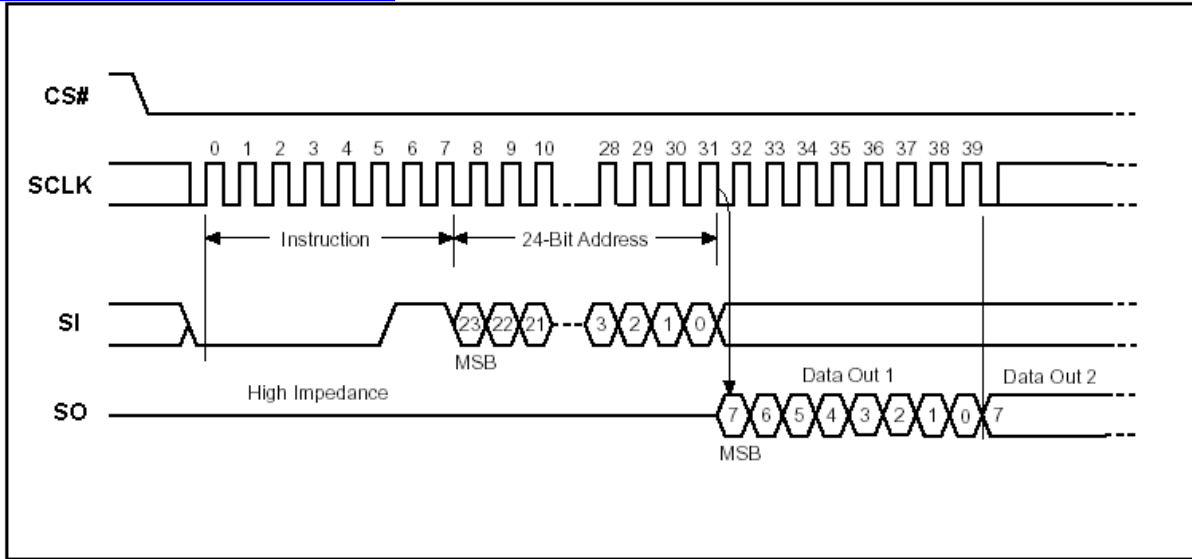
5.2 Read Data Bytes（一般读取）

Read Data Bytes 需要用指令码来执行每一次操作。READ 指令的时序如下(图):

- 首先把片选信号 (CS#) 变为低，紧跟着的是 1 个字节的命令字 (03 h) 和 3 个字节的地址和通过串行数据输入引脚 (SI) 移位输入，每一位在串行时钟 (SCLK) 上升沿被锁存。
- 然后该地址的字节数据通过串行数据输出引脚 (SO) 移位输出，每一位在串行时钟 (SCLK) 下降沿被移出。
- 读取字节数据后，则把片选信号 (CS#) 变为高，结束本次操作。

如果片选信号 (CS#) 继续保持为底，则下一个地址的字节数据继续通过串行数据输出引脚 (SO) 移位输出。

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

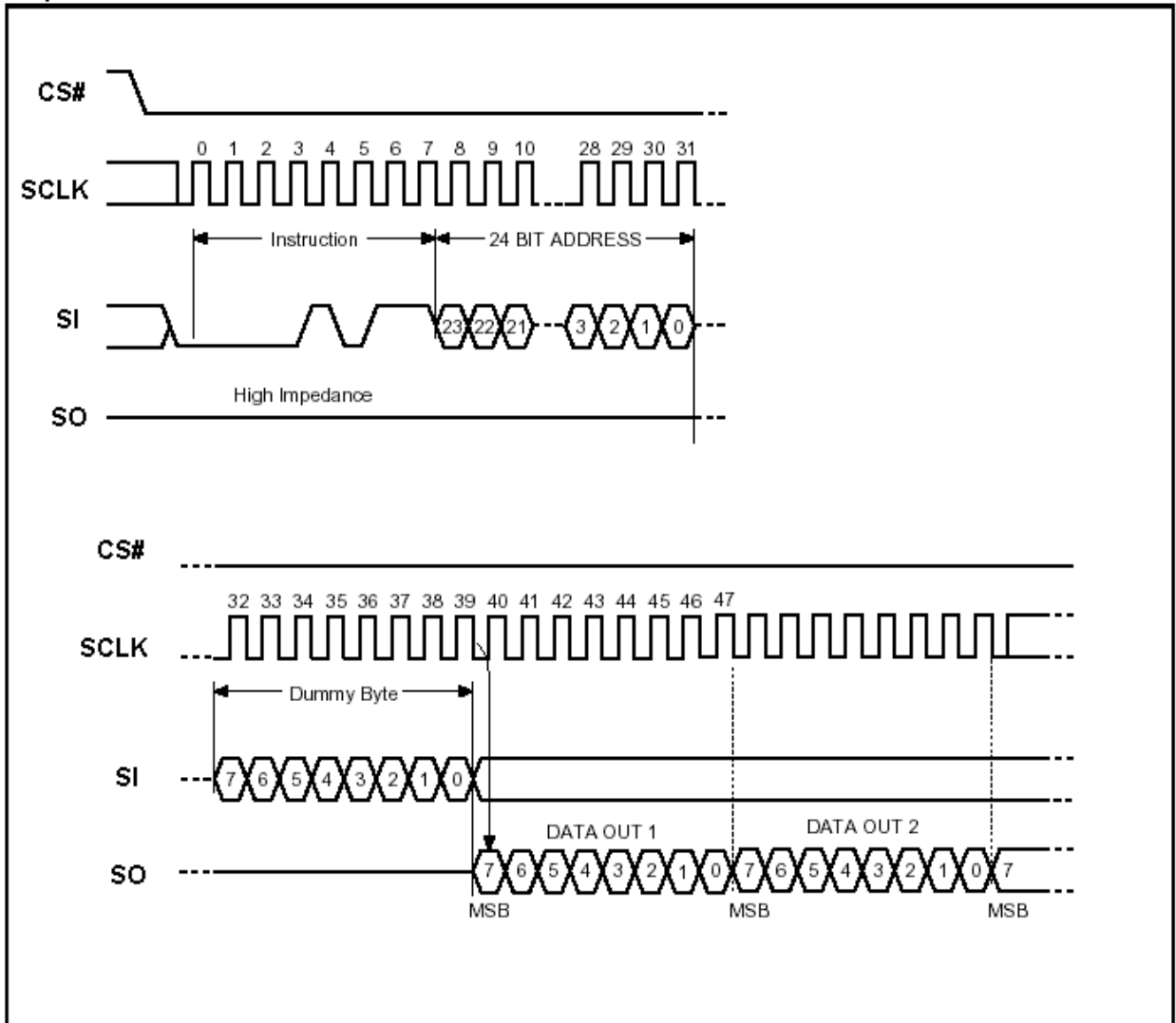


5.3 Read Data Bytes at Higher Speed (快速读取点阵数据)

Read Data Bytes at Higher Speed 需要用指令码来执行操作。READ_FAST 指令的时序如下(图):

- 首先把片选信号 (CS#) 变为低, 紧跟着的是 1 个字节的命令字 (0B h) 和 3 个字节的地址以及一个字节 Dummy Byte 通过串行数据输入引脚 (SI) 移位输入, 每一位在串行时钟 (SCLK) 上升沿被锁存。
- 然后该地址的字节数据通过串行数据输出引脚 (SO) 移位输出, 每一位在串行时钟 (SCLK) 下降沿被移出。
- 如果片选信号 (CS#) 继续保持为底, 则下一个地址的字节数据继续通过串行数据输出引脚 (SO) 移位输出。例: 读取一个 15x16 点阵汉字需要 32Byte, 则连续 32 个字节读取后结束一个汉字的点阵数据读取操作。
如果不需要继续读取数据, 则把片选信号 (CS#) 变为高, 结束本次操作。

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



6. 电气特性

6.1 绝对最大额定值

| Symbol | Parameter | Min. | Max. | Unit | Condition |
|------------------|-----------------------|------|----------------------|------|-----------|
| T _{OP} | Operating Temperature | -20 | 70 | °C | |
| T _{STG} | Storage Temperature | -65 | 150 | °C | |
| 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 | |

6.2 DC 特性

Condition: T_{OP} = -20°C to 70°C, GND=0V

| Symbol | Parameter | Min. | Max. | Unit | Condition |
|-----------------|--|------|------|------|-----------|
| I _{DD} | V _{CC} Supply Current(active) | | 8 | mA | |
| I _{SB} | V _{CC} Standby Current | | 8 | uA | |

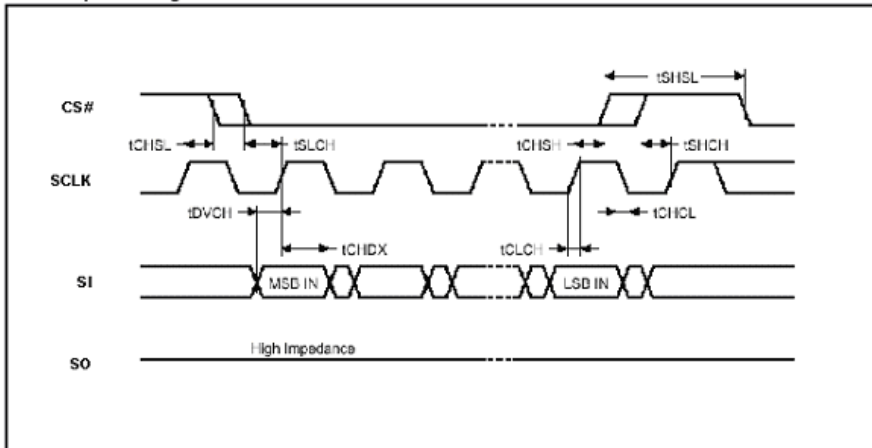
| | | | | | |
|----------|------------------------|-------------------------------|---------------------------|---------|--------------|
| V_{IL} | Input LOW Voltage | -0.3 | 0.3VCC | V | VCC=2.2~3.6V |
| V_{IH} | Input HIGH Voltage | 0.7VCC | VCC+0.4 | V | |
| V_{OL} | Output LOW Voltage | | 0.4 ($I_{OL}=1.6mA$) | V | |
| V_{OH} | Output HIGH Voltage | 0.8VCC ($I_{OH}=-100uA$) | | V | |
| I_{LI} | Input Leakage Current | 0 | 2 | μA | |
| I_{LO} | Output Leakage Current | 0 | 2 | μA | |

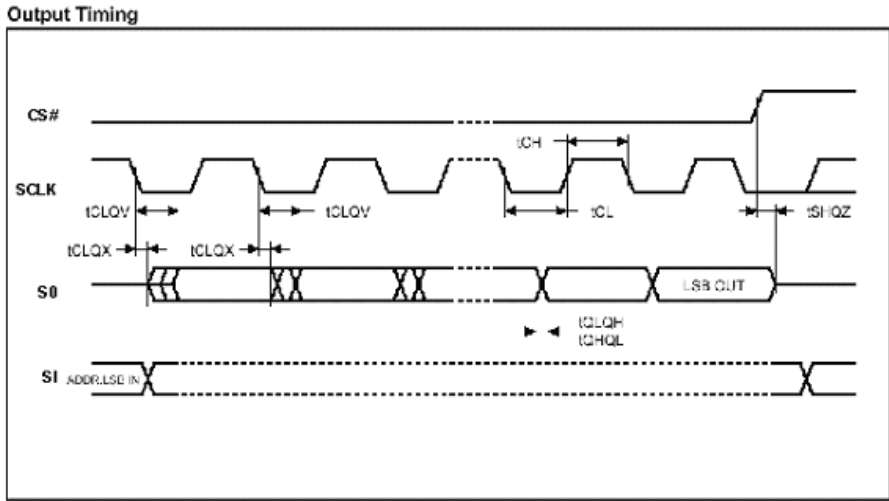
Note: I_{IL} : Input LOW Current, I_{IH} : Input HIGH Current,
 I_{OL} : Output LOW Current, I_{OH} : Output HIGH Current,

6.3 AC 特性

| 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 |
| tCHSH | | CS# Active Hold Time (relative to SCLK) | 5 | | ns |
| tSHCH | | CS# Not Active Setup Time (relative to SCLK) | 5 | | ns |
| tSHSL | tCSH | CS# Deselect Time | 100 | | ns |
| tSHQZ | tdIS | Output Disable Time | | 9 | ns |
| tCLQV | tv | Clock Low to Output Valid | | 9 | ns |
| tCLQX | tHO | Output Hold Time | 0 | | ns |

Serial Input Timing



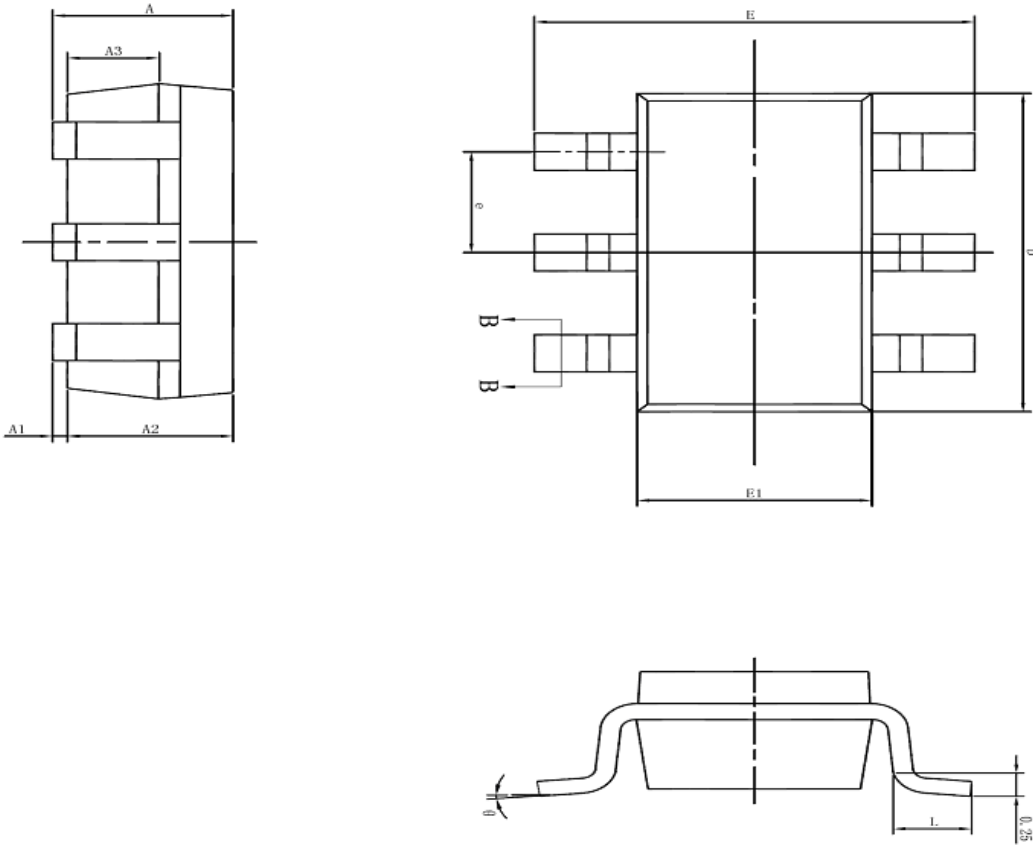


字库芯片

标准字库

7. 封装尺寸

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 封装

8. 字库调用方法

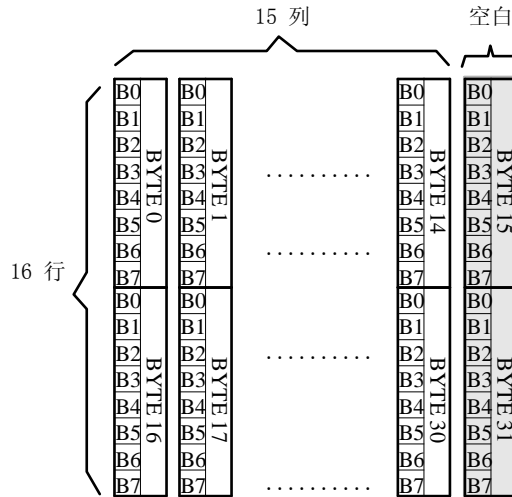
8.1 字符点阵排列格式

每个字符在芯片中是以字符点阵字模的形式存储的，每个点用一个二进制位表示，存 1 的点，当显示时可以在屏幕上显示亮点，存 0 的点，则在屏幕上不显示。点阵排列格式为竖置横排：即一个字节的高位表示下面的点，低位表示上面的点（如果用户按 16bit 总线宽度读取点阵数据，请注意高低字节的顺序），排满一行后再排下一行。这样把点阵信息用来直接在显示器上按上述规则显示，则将出现对应的字

符。

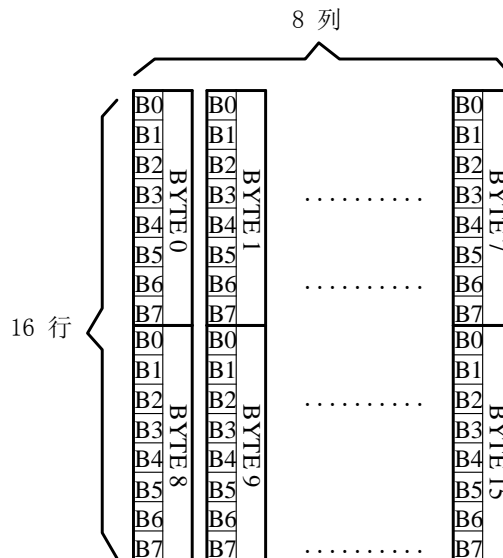
8.1.1 15X16 点日本字排列格式

15X16 点日本字的信息需要 32 个字节 (BYTE 0 – BYTE 31) 来表示。该 15X16 点日本字的点阵数据是竖置横排的，其具体排列结构如下图：



8.1.2 8X16 点字符排列格式

8X16 点字符的信息需要 16 个字节 (BYTE 0 – BYTE 15) 来表示。该 8X16 点字符的点阵数据是竖置横排的，其具体排列结构如下图：



8.2 字符在芯片中的地址计算方法

用户只要知道字符的内码，就可以计算出该字符点阵在芯片中的地址，然后就可从该地址连续读出点阵信息用于显示。

8.2.1 15x16 点阵 JIS 字符的地址计算

参数说明：

JIS0208表示汉字内码。

MSB 表示汉字内码JIS0208的高8bits。

LSB 表示汉字内码JIS0208的低8bits。

例如：

“<”的 JIS 编码为 0x0167 则：MSB 为 0x01, LSB 为 0x67。

“色”的 JIS 编码为 0x3107 则：MSB 为 0x31, LSB 为 0x07。

Address 表示汉字或 ASCII 字符点阵在芯片中的字节地址。

计算方法：

```
if(MSB >=1 && MSB <= 15 && LSB >=1 && LSB <= 94)
  Address = (MSB - 1) * 94 + (LSB - 01)*32;
else if(MSB >=16 && MSB <= 47 && LSB >=1 && LSB <= 94)
  Address = (MSB - 16) * 94 + (LSB - 1)*32+43584;
else if(MSB >=48 && MSB <=84 && LSB >=1 && LSB <= 94)
  Address = ((MSB - 48) * 94 + (LSB - 1))*32+ 138463;
else if(MSB ==85 && LSB >=0x01 && LSB <= 94)
  Address = ((MSB - 85) * 94 + (LSB - 1))*32+ 246944;
else if(MSB >=88 && MSB <=89 && LSB >=1 && LSB <= 94)
  Address = ((MSB - 88) * 94 + (LSB - 1))*32+ 249952;
```

8.2.2 8x16 点粗体 ASCII

参数说明：

ASCII CODE 表示该字符的ASCII码。

Address: 说明点阵数据在**字库芯片**中的地址。

例如：A 的 ASCII CODE 为 0x20

计算方法：

```
if(ASCII CODE >=0x20 && ASCII CODE <=0x7F)
  Address = (ASCII CODE - 0x20)*16+255968;
```

字库芯片

标准字库

8.2.3 8x16 ASCII 定制字符

参数说明：

ASCII CODE 表示该字符的ASCII码。

Address: 说明点阵数据在**字库芯片**中的地址。

计算方法：

```
if(ASCII CODE >=0x00 && ASCII CODE <=0xFF)
  Address = ASCII CODE *16+257504;
```

8.2.4 8x16 点 ASCII 定制补充字符

参数说明:

ASCII CODE 表示该字符的ASCII码。

Address: 说明点阵数据在字库芯片中的地址。

计算方法:

```
if(ASCII CODE >=0x00 && ASCII CODE <=0xFF)
```

```
    Address =ASCII CODE*16+261600;
```

GT20L16J1Y

日文

字库芯片

标准字库

9. 附录

JIS0208 字符集编码

Table with 94 columns and 94 rows containing JIS0208 character set encoding data, including character codes and corresponding symbols.

GT20L16J1Y

日文 字库芯片 标准字库