

GT21L16S2W 标准汉字字库芯片

用户手册 DATASHEET

- 字型：11X12 点阵、15X16 点阵
- 字符集：GB2312
- 兼容 Unicode 内码
- 输入法码本：GT 快捷拼音输入法
- 排置方式：横置横排
- 总线接口：SPI 串行总线
- 芯片形式：SO8 封装

VER 3.6

2010-Q3

版本修订记录

版本号	修改内容	日期	备注
V35	1. 15*16 点汉字算法部分	2010-7	
	2. 8X16 点国标扩展字符	2010-7	
	3. 8X16 点国标扩展字符起始地址	2010-7	
	4. 6X12 点国标扩展字符	2010-7	
	5. 6X12 点国标扩展字符起始地址	2010-7	
V36	6. 内容没有变动。	2010-8	

目 录

第一部分：硬件部分

1 概述	4
1.1 芯片特点	4
1.2 芯片内容	4
2 引脚描述与接口连接	6
2.1 引脚配置	6
2.2 引脚描述	6
2.3 HOST CPU 主机 SPI 接口电路示意图	7
3 操作指令	8
3.1 指令参数	8
3.2 Read Data Bytes（一般读取）	8
3.3 Read Data Bytes at Higher Speed（快速读取点阵数据）	9
4 电气特性	10
4.1 绝对最大额定值	10
4.2 DC 特性	10
4.3 AC 特性	10
5 封装尺寸	12

第二部分：软件部分

6 字库调用方法	13
6.1 汉字点阵排列格式	13
6.2 汉字点阵字库地址表	17
6.3 字符在芯片中的地址计算方法	18
6.4 Unicode 到 GB2312 内码转换程序	20
6.5 GB2312 到 Unicode 内码转换程序	22
7 附录	24
7.1 GB2312 1 区 (376 字符)	24
7.2 8×16 点国标扩展字符 (126 字符)	25

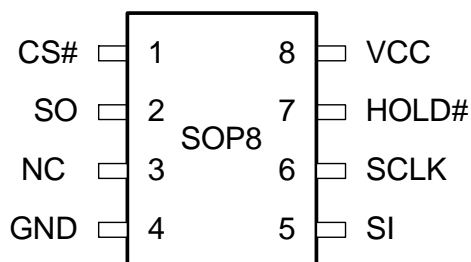
1 概述

GT21L16S2W是一款内含11X12点阵和15X16点阵的汉字库芯片，支持GB2312国标简体汉字（含有国家信标委合法授权）、ASCII字符及GB2312与Unicode编码互转表。排列格式为横置横排。用户通过字符内码，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。

本字库芯片内含GT快捷拼音输入法码本，另外配合本公司的输入法程序，可实现数字小键盘IT产品的汉字快捷输入。

1.1 芯片特点

- 数据总线：SPI 串行总线接口
- 点阵排列方式：字节横置横排
- 时钟频率：30MHz(max.) @3.3V
- 工作电压：2.7V~3.6V
- 电流：
 - 工作电流：12mA
 - 待机电流：10uA
- 封装：SOP8
- 尺寸（SOP8）：4.90mmX3.90mm（193milX154mil）
- 工作温度：-20℃~85℃



1.2 芯片内容

分类	字库内容	编码体系（字符集）	字符数
汉字及字符	11X12 点 GB2312 标准点阵字库	GB2312	6763+376
	15X16 点 GB2312 标准点阵字库	GB2312	6763+376
	6X12 点国标扩展字符	GB2312	126
	8X16 点国标扩展字符	GB2312	126
ASCII 字符	5X7 点 ASCII 字符	ASCII	96
	7X8 点 ASCII 字符	ASCII	96
	6X12 点 ASCII 字符	ASCII	96
	8X16 点 ASCII 字符	ASCII	96
	12 点阵不等宽 ASCII 方头（Arial）字符	ASCII	96
	16 点阵不等宽 ASCII 方头（Arial）字符	ASCII	96
内码转换表	Unicode 到 GB2312 内码转换表		
	GB2312 到 Unicode 内码转换表		
输入法码表	GT 快捷拼音输入法码表	GB2312	

GT21L16S2W

简体

字库芯片

标准字库

简体字库

字型样张

11X12 点 GB2312 汉字

咽阿挨挨哎唉唉皓癌蒿矮艾碍碍隘鞍
氨安俺俺暗岸胺案肮昂盎凹敖熬熬袄
傲奥懊懊芭捌叭扒吧芭疤巴拔跋跋
把把坝霸罢爸白柏百摆佰败拜拜班班
搬扳般般板版拌伴伴瓣半包钹钹钹
保堡抱抱抱暴暴豹豹煲煲煲煲北北
背背倍倍倍倍倍倍被被本本本本本

15X16 点 GB2312 汉字

啊阿埃挨哎唉哀皑癌蔼矮艾
碍爱隘隘隘隘隘隘隘隘隘隘隘
肮昂盎盎盎盎盎盎盎盎盎盎盎
芭捌扒叭吧芭八疤巴拔摆佰败
把耙坝霸罢爸白柏百摆佰败
拜裨班班搬般颁扳般般般般般

5x7 点 ASCII 字符

```
!"#$%&'()*+,-./0123456789:
=>?@ABCDEFGHIJKLMN O PQRSTU V
YZ[\]^_`abcdef ghijklmnopqr
```

7x8 点 ASCII 字符

```

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

```

6x12 点 ASCII 字符

```
!"#$%&'()*+,-./0123456789:;  
=>?@ABCDEFGHIJKLMN O PQRSTU VW  
YZ[\]^_`abcdef ghijklmnopqrs  
uvwxyz{|}~¡¢£ ¤ ¥ ¦ § ¨ © ª « ¬ ® ¯ ° ±
```

8x16 点 ASCII 字符

!"#\$%&'()*+,-./012345
6789:;<=>?@ABCDEFGHIJK
LMNOPQRSTUVWXYZ[\]^_`a

12 点阵不等宽 ASCII 方头

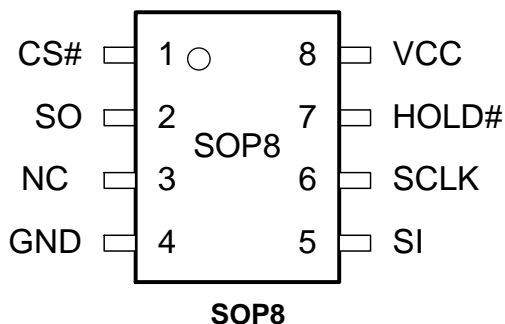
```
!"#$%&'()*+,-./0123456789;<=>?@ABC
DEFGHIJKLMNOPQRSTUVWXYZ[\]^_`
abcdefghijklmnopqrstuvwxyz{|}~
!"#$%&'()*+,-./0123456789;<=>?@ABC
```

16 点阵不等宽 ASCII 方头

```
!'#$%&'()*+,-./0123456789:;<=>  
DEFGHIJKLMNOPQRSTUVWXYZ  
abcdefghijklmnopqrstuvwxyz{
```

2 引脚描述与接口连接

2.1 引脚配置



2.2 引脚描述

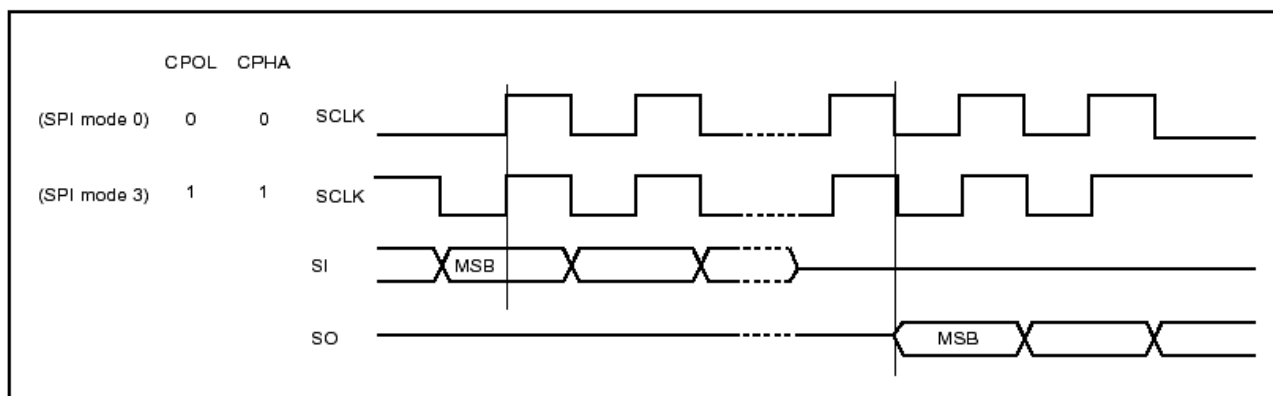
SOP8	名称	I/O	描述
1	CS#	I	片选输入 (Chip enable input)
2	SO	O	串行数据输出 (Serial data output)
3	NC		悬空
4	GND		地(Ground)
5	SI	I	串行数据输入 (Serial data input)
6	SCLK	I	串行时钟输入 (Serial clock input)
7	HOLD#	I	总线挂起 (Hold, to pause the device without)
8	VCC		电源(+ 3.3V Power Supply)

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CS#下降沿, CS#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。

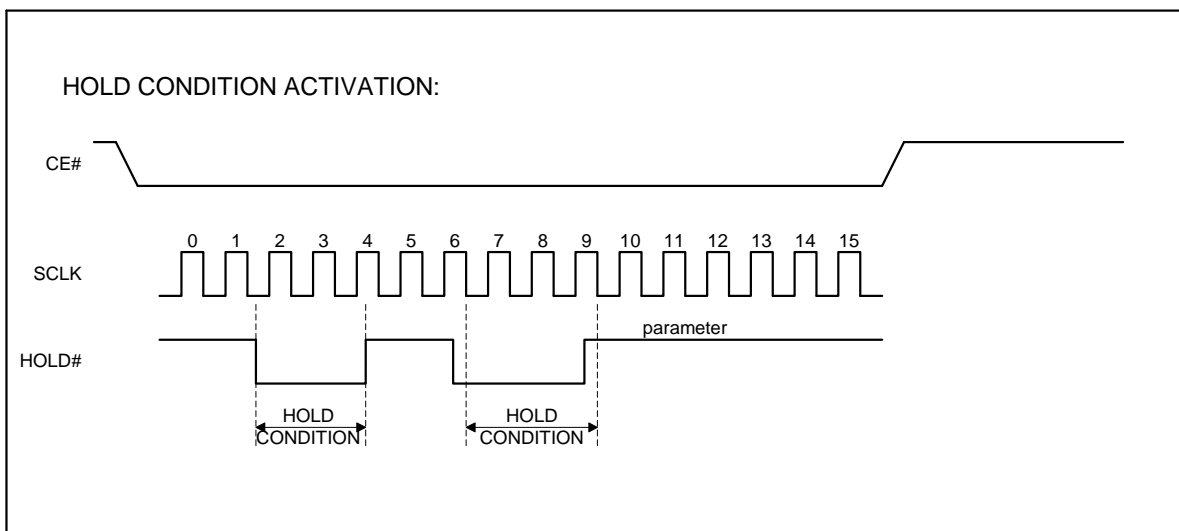


总线挂起输入 (HOLD#):

该信号用于片选信号有效期间暂停数据传输，在总线挂起期间，串行数据输出信号处于高阻态，芯片不对串行数据输入信号和串行时钟信号进行响应。

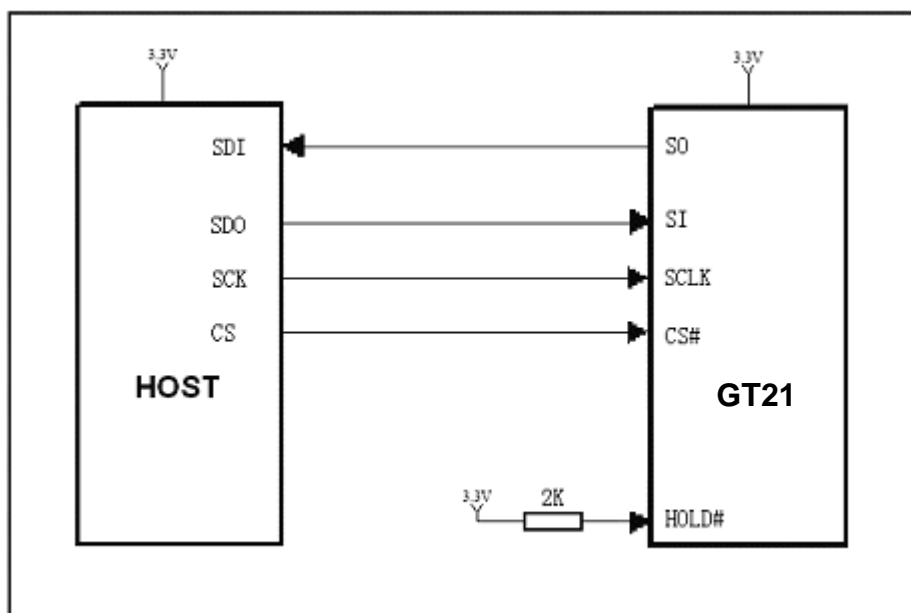
当HOLD#信号变为低并且串行时钟信号 (SCLK) 处于低电平时，进入总线挂起状态。

当HOLD#信号变为高并时串行时钟信号 (SCLK) 处于低电平时，结束总线挂起状态。



2.3 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 (#HOLD 管脚建议接 2K 电阻 3.3V 拉高)。



HOST CPU 主机 SPI 接口电路示意图

3 操作指令

3.1 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”)。

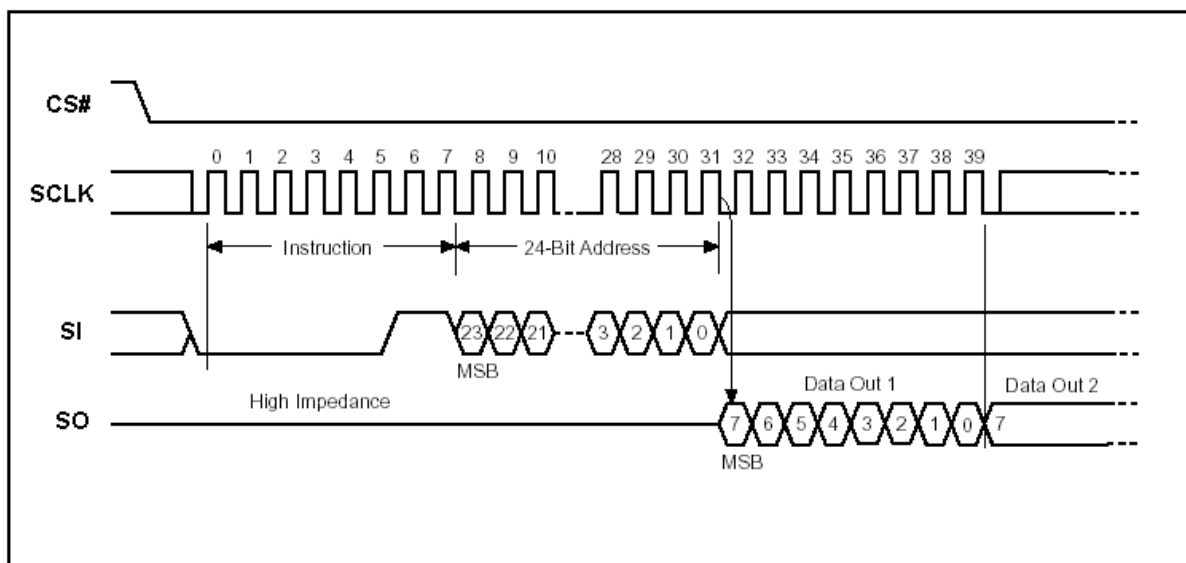
3.2 Read Data Bytes（一般读取）

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:



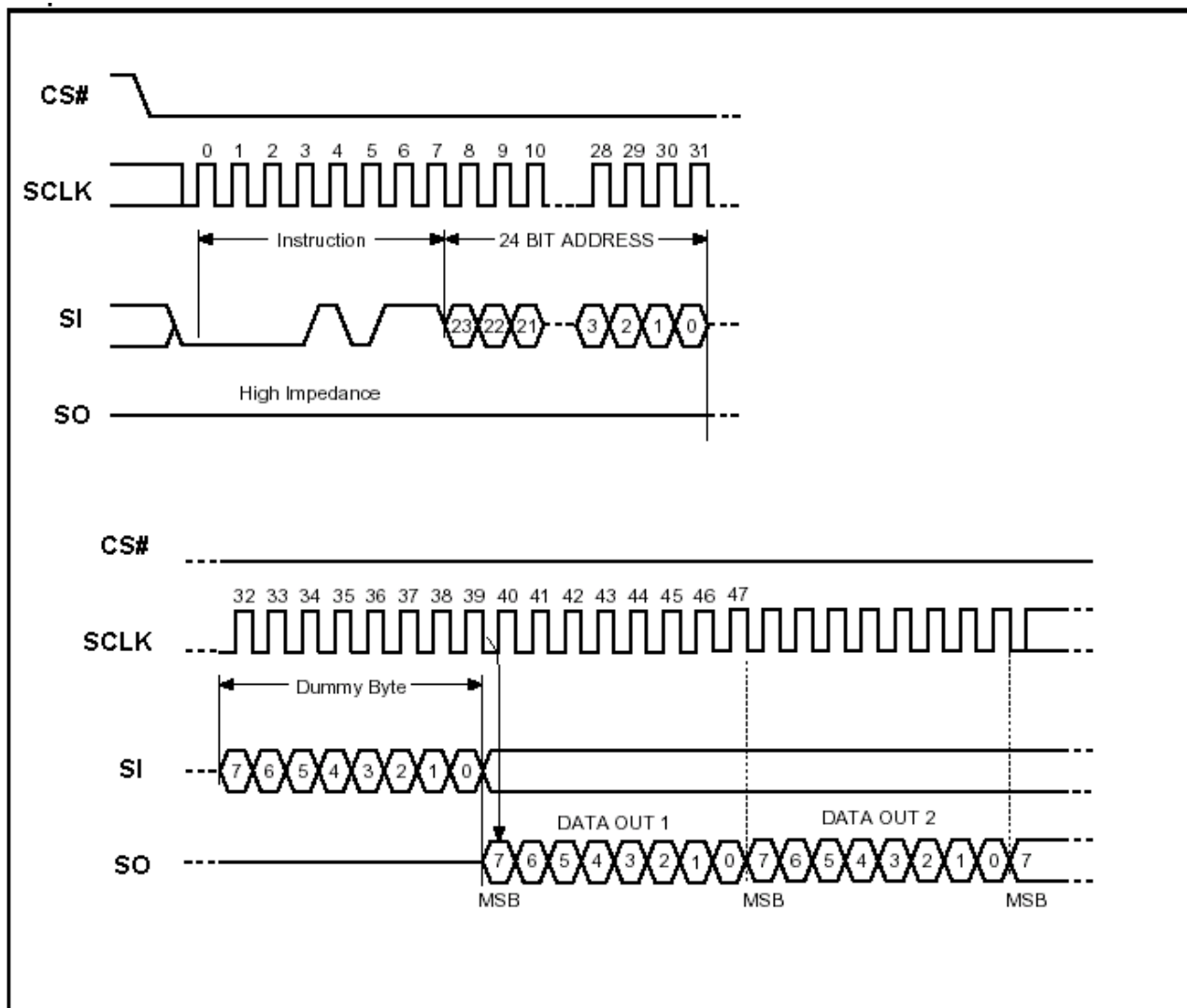
3.3 Read Data Bytes at Higher Speed (快速读取点阵数据)

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:



4 电气特性

4.1 绝对最大额定值

Symbol	Parameter	Min.	Max.	Unit	Condition
T _{OP}	Operating Temperature	-20	85	°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	

4.2 DC 特性

Condition: T_{OP} = -20°C to 85°C, GND=0V

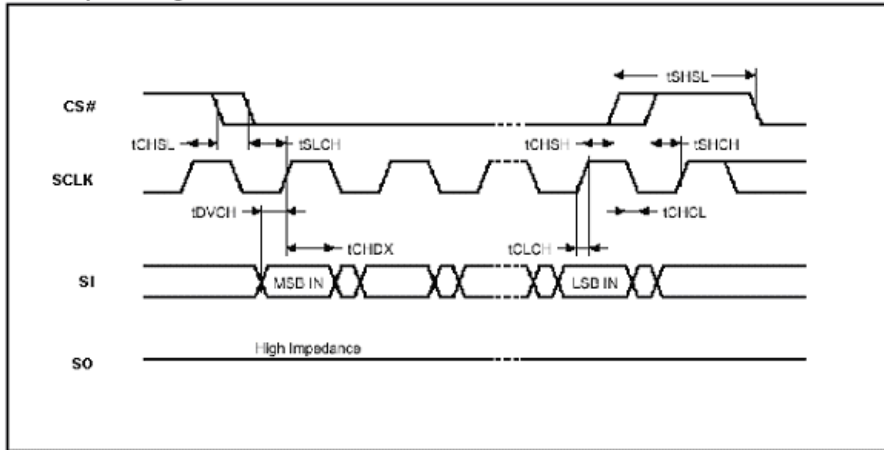
Symbol	Parameter	Min.	Max.	Unit	Condition
I _{DD}	VCC Supply Current(active)		12	mA	
I _{SB}	VCC Standby Current		10	uA	
V _{IL}	Input LOW Voltage	-0.3	0.3V _{CC}	V	V _{CC} =2.7~3.6V
V _{IH}	Input HIGH Voltage	0.7V _{CC}	V _{CC} +0.4	V	
V _{OL}	Output LOW Voltage		0.4 (I _{OL} =1.6mA)	V	
V _{OH}	Output 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,

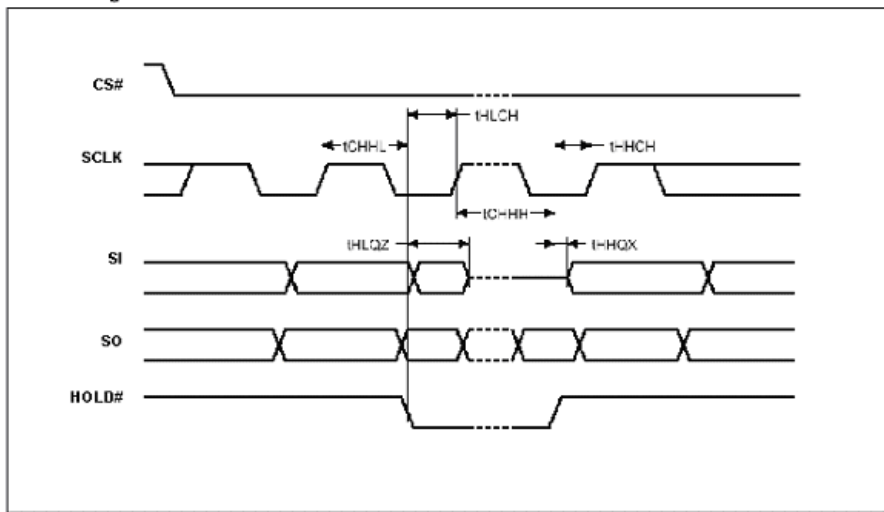
4.3 AC 特性

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

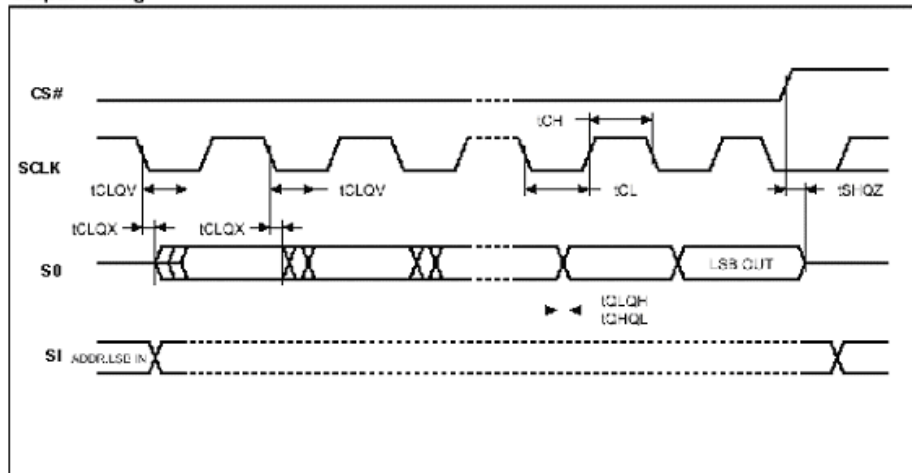
Serial Input Timing



Hold Timing



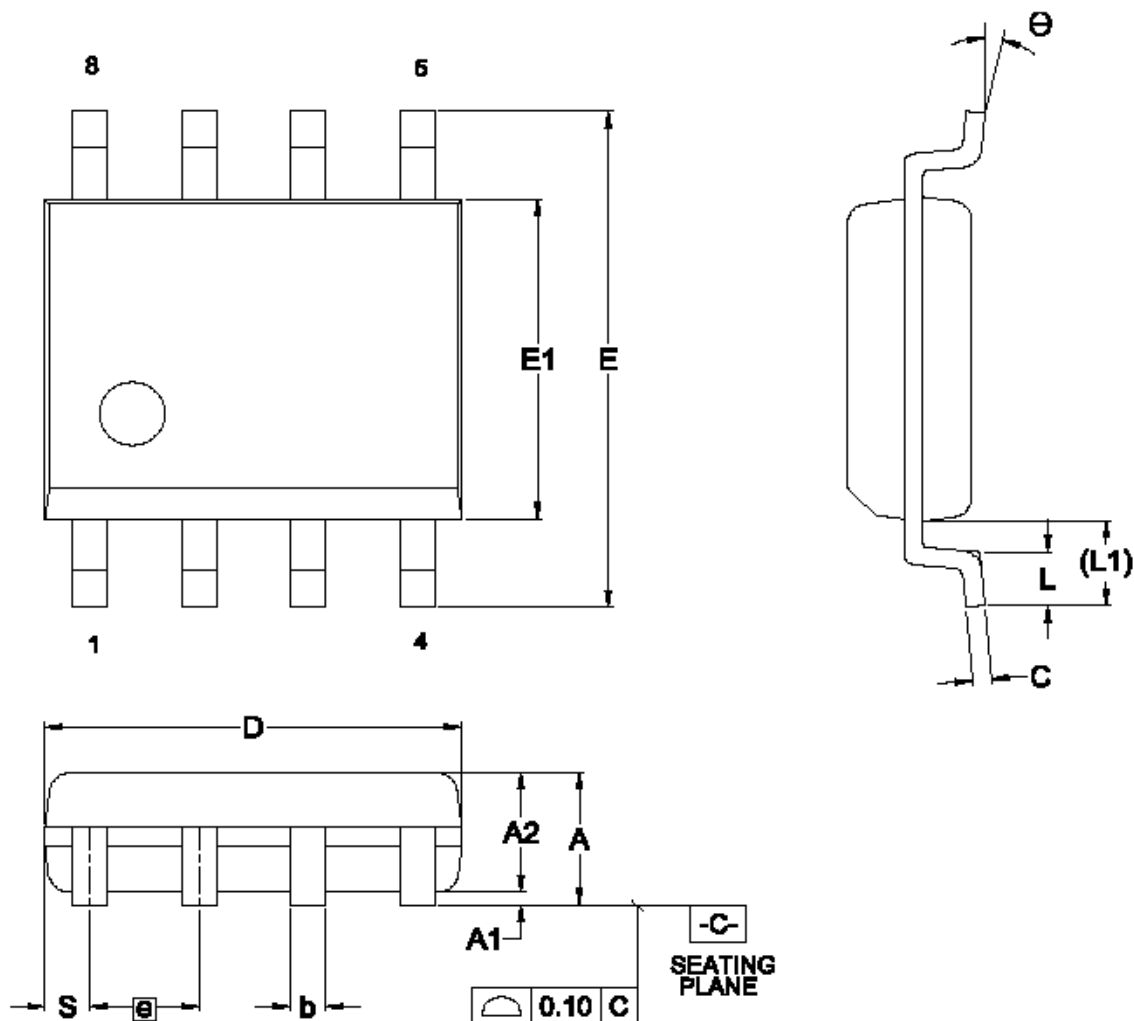
Output Timing



5 封装尺寸

SOP8 Package

Unit :mm

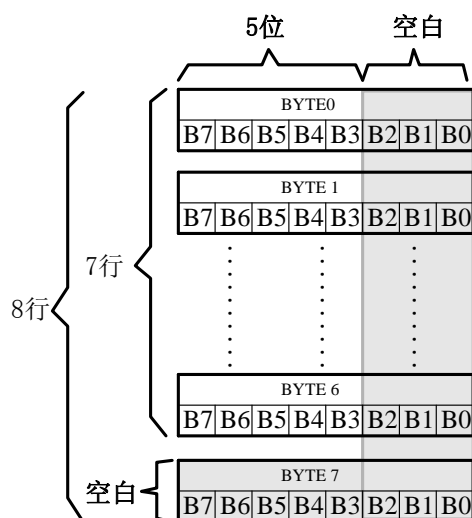


Dimensions(inch dimensions are derived from the original mm dimensions)

		A	A1	A2	b	C	D	E	E1	θ	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

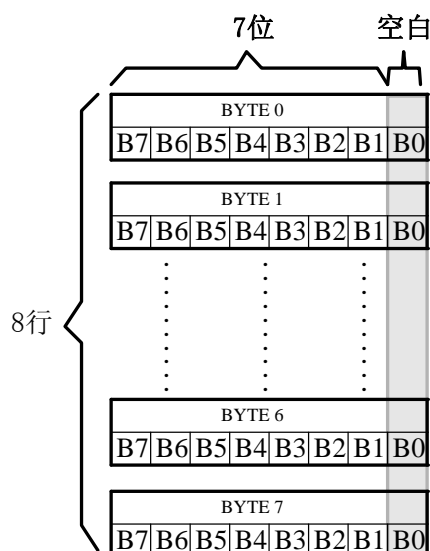
6.1.3 5X7 点 ASCII 字符排列格式

5X7 点 ASCII 的信息需要 8 个字节（BYTE 0 – BYTE7）来表示。该 ASCII 点阵数据是横置横排的，其具体排列结构如下图：



6.1.4 7X8 点 ASCII 字符排列格式

7X8 点 ASCII 的信息需要 8 个字节（BYTE 0 – BYTE7）来表示。该 ASCII 点阵数据是横置横排的，其具体排列结构如下图：



6.1.5 6X12 点字符排列格式

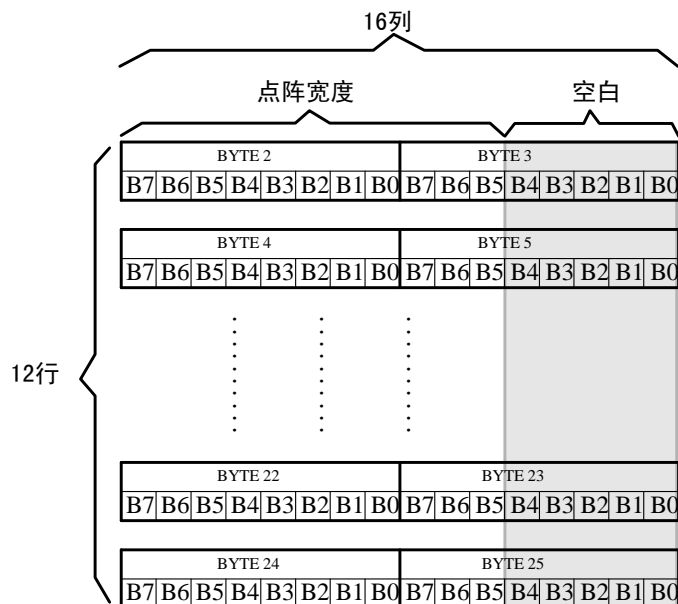
适用于此种排列格式的字体有：

6X12 点 ASCII 字符

6X12 点国标扩展字符

6X12 点 ASCII 的信息需要 12 个字节（BYTE 0 – BYTE11）来表示。该 ASCII 点阵数据是横置横排的，其具体排列结构如下图：

BYTE0~ BYTE1 所存放点阵的实际宽度数据，可以对还原下一个字的显示或排版留作参考。

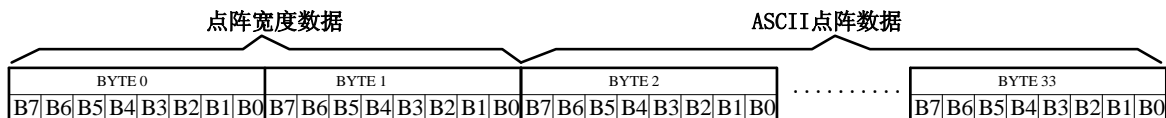


6.1.8 16 点阵不等宽 ASCII 方头（Arial）字符排列格式

16 点阵不等宽字符的信息需要 34 个字节（BYTE 0 – BYTE33）来表示。

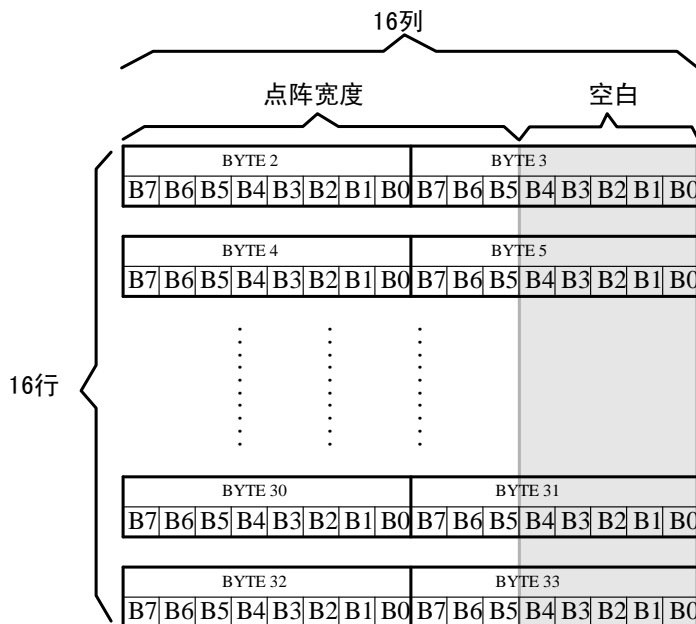
■ 存储格式

由于字符是不等宽的，因此在存储格式中 BYTE0~ BYTE1 存放点阵宽度数据，BYTE2-33 存放横置横排点阵数据。具体格式见下图：



■ 存储结构

不等宽字符的点阵存储宽度是以 BYTE 为单位取整的，根据不同字符宽度会出现相应的空白区。根据 BYTE0~ BYTE1 所存放点阵的实际宽度数据，可以对还原下一个字的显示或排版留作参考。



例如：ASCII 方头字符 B

0-33BYTE 的点阵数据是： 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

其中：

BYTE0~ BYTE1: 00 0C 为 ASCII 方头字符 B 的点阵宽度数据，即：12 位宽度。字符后面有 4 位空白区，可以在排版下一个字时考虑到这一点，将下一个字的起始位置前移。

BYTE2-33: 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 7F C0 7F 80 00 00 为 ASCII 方头字符 B 的点阵数据。

6.2 汉字点阵字库地址表

	字库内容	编码体系	码位范围	字符数	起始地址	结束地址	参考算法
1	15X16 点 GB2312 标准点阵字库	GB2312	A1A1-F7FE	6763+376	00000	3B7BF	6.3.1.2
2	GB2312 到 Unicode 内码转换表				2F00	66BF	6.4
3	7X8 点 ASCII 字符	ASCII	20~7F	96	66C0	69BF	6.3.2.2
4	8X16 点国标扩展字符	GB2312	AAA1-ABC0	126	3B7D0	3BFBF	6.3.1.4
5	8X16 点 ASCII 字符	ASCII	20~7F	96	3B7C0	3BFBF	6.3.2.4
6	5X7 点 ASCII 字符	ASCII	20~7F	96	3BFC0	3C2BF	6.3.2.1
7	16 点阵不等宽 ASCII 方头 (Arial) 字符	ASCII	20~7F	96	3C2C0	3CF7F	6.3.2.6
8	11X12 点 GB2312 标准点阵字库	GB2312	A1A1-F7FE	6763+376	3CF80	66D3F	6.3.1.3
9	6X12 点国标扩展字符	GB2312	AAA1-ABC0	126	66D4C	6733F	6.3.1.3
10	6X12 点 ASCII 字符	ASCII	20~7F	96	66D40	6733F	6.3.2.3
11	12 点阵不等宽 ASCII 方头 (Arial) 字符	ASCII	20~7F	96	67340	67CFF	6.3.2.5
12	保留区				67D00	67D6F	
13	Unicode 到 GB2312 内码转换表				67D70	7278F	6.5
14	GT 快捷拼音输入法码表				72790	7FA33	
15	保留区				7FA33	7FFFF	

GT21L16S2W

简体

字库芯片

标准字库

简体字库

6.3 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。

6.3.1 汉字字符的地址计算

6.3.1.1 11X12 点 GB2312 标准点阵字库

参数说明：

GBCode表示汉字内码。

MSB 表示汉字内码GBCode 的高8bits。

LSB 表示汉字内码GBCode 的低8bits。

Address 表示汉字或ASCII字符点阵在芯片中的字节地址。

BaseAdd: 说明点阵数据在字库芯片中的起始地址。

计算方法：

BaseAdd=0x3cf80;

if(MSB >=0xA1 && MSB <= 0xA3 && LSB >=0xA1)

Address = (MSB - 0xA1) * 94 + (LSB - 0xA1)*24+ BaseAdd;

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

Address = (282 + (LSB - 0xA1))*24+ BaseAdd;

else if(MSB >=0xB0 && MSB <= 0xF7 && LSB >=0xA1)

Address = ((MSB - 0xB0) * 94 + (LSB - 0xA1)+ 376)*24+ BaseAdd;

6.3.1.2 15X16 点 GB2312 标准点阵字库

参数说明：

GBCode表示汉字内码。

MSB 表示汉字内码GBCode 的高8bits。

LSB 表示汉字内码GBCode 的低8bits。

Address 表示汉字或ASCII字符点阵在芯片中的字节地址。

BaseAdd: 说明点阵数据在字库芯片中的起始地址。

计算方法：

BaseAdd=0;

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

Address = (282 + (LSB - 0xA1))*32+ BaseAdd;

else if(MSB >=0xA1 && MSB <= 0xA3 && LSB >=0xA1)

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

else if(MSB >=0xB0 && MSB <= 0xF7 && LSB >=0xA1)

Address = ((MSB - 0xB0) * 94 + (LSB - 0xA1)+ 846)*32+ BaseAdd;

6.3.1.3 6X12 点国标扩展字符

说明：

BaseAdd: 说明本套字库在字库芯片中的起始字节地址。

FontCode: 表示字符内码（16bits）

ByteAddress: 表示字符点阵在芯片中的字节地址。

计算方法：

BaseAdd=0x66d4c

if (FontCode>= 0xAAA1) and (FontCode<=0xAAFE) then

ByteAddress = (FontCode-0xAAA1) * 12+BaseAdd

Else if(FontCode>= 0xABA1) and (FontCode<=0xABC0) then

ByteAddress = (FontCode-0xABA1 + 95) * 12+BaseAdd

6.3.1.4 8X16 点国标扩展字符

说明:

BaseAdd: 说明本套字库在字库芯片中的起始字节地址。

FontCode: 表示字符内码 (16bits)

ByteAddress: 表示字符点阵在芯片中的字节地址。

计算方法:

BaseAdd=0x3b7d0

if (FontCode>= 0xAAA1) and (FontCode<=0xAAFE) then

ByteAddress = (FontCode-0xAAA1) * 16+BaseAdd

Else if(FontCode>= 0xABA1) and (FontCode<=0xABC0) then

ByteAddress = (FontCode-0xABA1 + 95) * 16+BaseAdd

6.3.2 ASCII 字符的地址计算

6.3.2.1 5X7 点 ASCII 字符

参数说明:

ASCIICode: 表示 ASCII 码 (8bits)

BaseAdd: 说明该套字库在芯片中的起始地址。

Address: ASCII 字符点阵在芯片中的字节地址。

计算方法:

BaseAdd=0x3bfc0

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

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

6.3.2.2 7X8 点 ASCII 字符

参数说明:

ASCIICode: 表示 ASCII 码 (8bits)

BaseAdd: 说明该套字库在芯片中的起始地址。

Address: ASCII 字符点阵在芯片中的字节地址。

计算方法:

BaseAdd=0x66c0

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

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

6.3.2.3 6X12 点 ASCII 字符

说明:

ASCIICode: 表示 ASCII 码 (8bits)

BaseAdd: 说明该套字库在芯片中的起始地址。

Address: ASCII 字符点阵在芯片中的字节地址。

计算方法:

BaseAdd=0x66d40

```
if (ASCIICode >= 0x20) and (ASCIICode <= 0x7E) then
    Address = (ASCIICode - 0x20) * 12 + BaseAdd
```

6.3.2.4 8X16 点 ASCII 字符

说明:

ASCIICode: 表示 ASCII 码 (8bits)

BaseAdd: 说明该套字库在芯片中的起始地址。

Address: ASCII 字符点阵在芯片中的字节地址。

计算方法:

BaseAdd=0x3b7c0

```
if (ASCIICode >= 0x20) and (ASCIICode <= 0x7E) then
    Address = (ASCIICode - 0x20) * 16 + BaseAdd
```

6.3.2.5 12 点阵不等宽 ASCII 方头 (Arial) 字符

说明:

ASCIICode: 表示 ASCII 码 (8bits)

BaseAdd: 说明该套字库在芯片中的起始地址。

Address: ASCII 字符点阵在芯片中的字节地址。

计算方法:

BaseAdd=0x67340

```
if (ASCIICode >= 0x20) and (ASCIICode <= 0x7E) then
    Address = (ASCIICode - 0x20) * 26 + BaseAdd
```

6.3.2.6 16 点阵不等宽 ASCII 方头 (Arial) 字符

说明:

ASCIICode: 表示 ASCII 码 (8bits)

BaseAdd: 说明该套字库在芯片中的起始地址。

Address: ASCII 字符点阵在芯片中的字节地址。

计算方法:

BaseAdd=0x3c2c0

```
if (ASCIICode >= 0x20) and (ASCIICode <= 0x7E) then
    Address = (ASCIICode - 0x20) * 34 + BaseAdd
```

6.4 Unicode 到 GB2312 内码转换程序

功能: 把 Unicode 内码转换为 GB2312 内码

参数: srcCode: 输入 Unicode 内码串。

destCode: 转换后得到的 GB2312 内码串。

BYTE * UToGb2312Buff = 0x67d70; // UToGb2312Buff 为 Unicode 到 GB2312 内码转换表在 ROM 中的起始地址

BYTE UnicodeToGB2312(BYTE *srcCode, BYTE *destCode)

```
{
    BYTE *ptr, result=0;
    int h;
    WORD code;
```

```
code = *srcCode++;
code = (code<<8) + *srcCode;

if(code<0xa0) result=1;

else if(code<=0xf7) h=code-160;
else if(code<0x2c7) result=1;

else if(code<=0x2c9) h=code-160-463;
else if(code<0x2010) result=1;

else if(code<=0x2312) h=code-160-463-7494;
else if(code<0x2460) result=1;

else if(code<=0x2642) h=code-160-463-7494-333;
else if(code<0x3000) result=1;

else if(code<=0x3017) h=code-160-463-7494-333-2493;
else if(code<0x3220) result=1;

else if(code<=0x3229) h=code-160-463-7494-333-2493-520;
else if(code<0x4e00) result=1;

else if(code<=0x9b54) h=code-160-463-7494-333-2493-520-7126;
else if(code<0x9c7c) result=1;

else if(code<=0x9ce2) h=code-160-463-7494-333-2493-520-7126-295;
else if(code<0x9e1f) result=1;

else if(code<=0x9fa0) h=code-160-463-7494-333-2493-520-7126-295-316;
else if(code<0xe76c) result=1;

else if(code<=0xe774) h=code-160-463-7494-333-2493-520-7126-295-316-18379;
else if(code<0xff00) result=1;

else if(code<=0xff5f) h=code-160-463-7494-333-2493-520-7126-295-316-18379-6027;
else if(code<0xffe0) result=1;

else if(code<=0xffe5) h=code-160-463-7494-333-2493-520-7126-295-316-18379-6027-128;
else result=1;

if(result==0)
{
    ptr = UToGb2312Buff + (h<<1);
```

```

        *destCode++ = *ptr++;
        *destCode = *ptr;
    }
    else
    {
        *destCode++ = 0xa1;
        *destCode = 0xa1;
    }

    return 0;
}

```

6.5 GB2312 到 Unicode 内码转换程序

功能： 把 GB2312 内码转换为 Unicode 内码

参数： srcCode: 输入 GB2312 内码串。

destCode: 转换后得到的 Unicode 内码串。

BYTE * Gb2312ToUBuff = 0x2f00; // Gb2312ToUBuff 为 GB2312 到 Unicode 内码转换表在 ROM 中的起始地址

BYTE GB2312ToUnicode(BYTE *srcCode, BYTE *destCode)

```

{
    BYTE *ptr;
    int h;
    h = Gb2311ToIndex(srcCode);

    ptr = Gb2312ToUBuff + (h << 1);    // +48 head
    *destCode++ = *ptr++;
    *destCode = *ptr;
    return 0;
}

```

int Gb2311ToIndex(BYTE *incode)

```

{
    int index;
    BYTE byte0, byte1;
    byte0 = incode[0];
    byte1 = incode[1];

    if (byte0 >= 0xA1 && byte0 <= 0xA3 && byte1 >= 0xA1)
    {
        index = (byte0 - 0xA1) * 94 + (byte1 - 0xA1);
    }
    else if (byte0 == 0xA9 && byte1 >= 0xA1 && byte1 <= 0xF6)
        index = 282 + (byte1 - 0xA1);
}

```

```
else if(byte0>=0xb0 && byte0<=0xf7 && byte1 >=0xA1)
    index = (byte0 - 0xB0) * 94 + (byte1 - 0xA1)+ 368;
else
    index=0;
return index;

}
```

GT21L16S2W

简体

字库芯片

标准字库

简体字库

7 附录

7.1 GB2312 1 区 (376 字符)

GB2312 标准点阵字符 1 区对应码位的 A1A1~A9EF 共计 376 个字符;

GB2312 1 区

A1	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A			、	。	·	—	√	”	々	一	~		...	‘	’	
B	“	”	{	}	<	>	《	》	「	」	『	』	【	】	【	】
C	±	×	÷	:	∧	∨	Σ	Π	U	∩	€	::	√	⊥	//	∠
D	∩	⊙	∫	∫	≡	≈	≈	≈	≈	≠	≠	≠	≠	≠	≠	∞
E	∴	↑	♀	°	'	”	℃	\$	⊗	⊗	£	%	§	No	☆	★
F	○	●	◎	◇	◆	□	■	△	▲	※	→	←	↑	↓	=	

A2	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A																
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	{		}		

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 8×16 点国标扩展字符（126 字符）

内码组成为 AAA1~ABC0 共计 126 个字符

AA	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	{		}	~	

A	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	ǐ	ì	ō	ó	ǒ
B	ò	ū	ú	ǔ	ù	ǘ	ú	ǚ	ù	ü	ê	ɑ	ǎ	ǎ	ǎ	ǎ
C	g															

GT21L16S2W

简体

字库芯片

标准字库

简体字库