

# GT30L24T3Y 标准汉字字库芯片

## 规格书 DATASHEET

- 字符集：GB2312 字符集
- GB12345 字符集
- BIG5 字符集
- 兼容 Unicode
- 字号：12x12 点阵、16x16 点阵、24x24 点阵
- 排置方式：竖置横排
- 总线接口：SPI 串行总线
- 封装类型：SOP8-B

V1.0I\_A

2013-04



# 目 录

<b>1 概述</b> .....	<b>4</b>
1.1 芯片特点.....	4
1.2 芯片内容.....	5
1.3 字型样张.....	6
<b>2 操作指令</b> .....	<b>11</b>
2.1 指令参数.....	11
2.2 Read Data Bytes（一般读取）.....	11
2.3 Read Data Bytes at Higher Speed（快速读取点阵数据）.....	12
<b>3 字符点阵字库地址表</b> .....	<b>13</b>
<b>4 字符点阵数据在芯片中的地址计算方法</b> .....	<b>14</b>
4.1 汉字字符点阵数据的地址计算.....	14
4.2 ASCII 字符的地址计算.....	26
<b>5 引脚描述与电路连接</b> .....	<b>28</b>
5.1 引脚配置.....	28
5.2 引脚描述.....	28
5.3 HOST CPU 主机接口与 SPI 接口电路示意图.....	29
<b>6 电气特性</b> .....	<b>30</b>
6.1 绝对最大额定值.....	30
6.2 DC 特性.....	30
6.3 AC 特性.....	30
<b>7 封装尺寸：SOP8-B</b> .....	<b>32</b>
<b>8 附录</b> .....	<b>33</b>
8.1 GB2312 1 区字符（470 字符）.....	33
8.2 补丁文件.....	34

## 1 概述

GT30L24T3Y是一款内含12x12点阵、16x16点阵、24x24点阵的汉字库芯片，支持GB2312国标汉字（含有国家信标委合法授权）、GB12345国标繁体汉字（含有国家信标委合法授权）、BIG5字符集汉字及ASCII字符。并兼容Unicode编码格式。排列格式为竖置横排。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。

### 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
- 访问速度： SPI 时钟频率： 80MHz(max.)
- 工作电压： 2.7V~3.6V
- 电流： 工作电流： 20mA  
待机电流： 5uA
- 封装： SOP8-B
- 尺寸（SOP8-B）： 7.90mmX5.23mm
- 工作温度： -40℃~85℃ 存储温度： -55~125℃

## 1.2 芯片内容

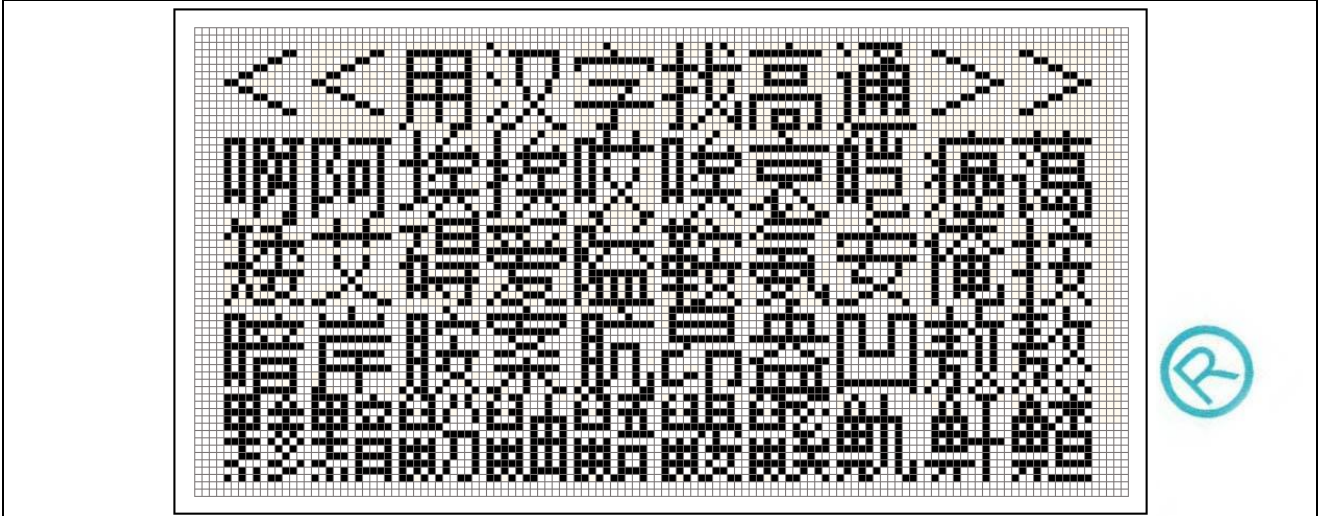
分类	字库	字号	字符数	字体	排列方式	备注
ASCII 字符集	ASCII	5x7	96	标准	Y-竖置横排	
	ASCII	7x8	96	标准	Y-竖置横排	
	ASCII	6x12	96	标准	Y-竖置横排	
	ASCII	8x16	96	标准	Y-竖置横排	
	ASCII	12点阵不等宽	96	Arial (方头)	Y-竖置横排	
	ASCII	16点阵不等宽	96	Arial (方头)	Y-竖置横排	
	ASCII	24点阵不等宽	96	Arial (方头)	Y-竖置横排	
GB 字符集	GB2312 汉字	12x12	6763	宋体	Y-竖置横排	
		16x16	6763	宋体	Y-竖置横排	
		16x16	6763	宋体	Y-竖置横排	
	GB2312 字符	12x12	470	宋体	Y-竖置横排	
		16x16	470	宋体	Y-竖置横排	
		16x16	470	宋体	Y-竖置横排	
	BIG5 汉字	12x12	5410	宋体	Y-竖置横排	
		16x16	5410	宋体	Y-竖置横排	
		16x16	5410	宋体	Y-竖置横排	
	BIG5 字符	12x12	408	宋体	Y-竖置横排	
		16x16	408	宋体	Y-竖置横排	
		16x16	408	宋体	Y-竖置横排	
GB12345 汉字 内码索引表						
BIG5 汉字内码 索引表						
Unicode 汉字内 码索引表						



### 1.3 字型样张

#### 1.3.1 汉字字符

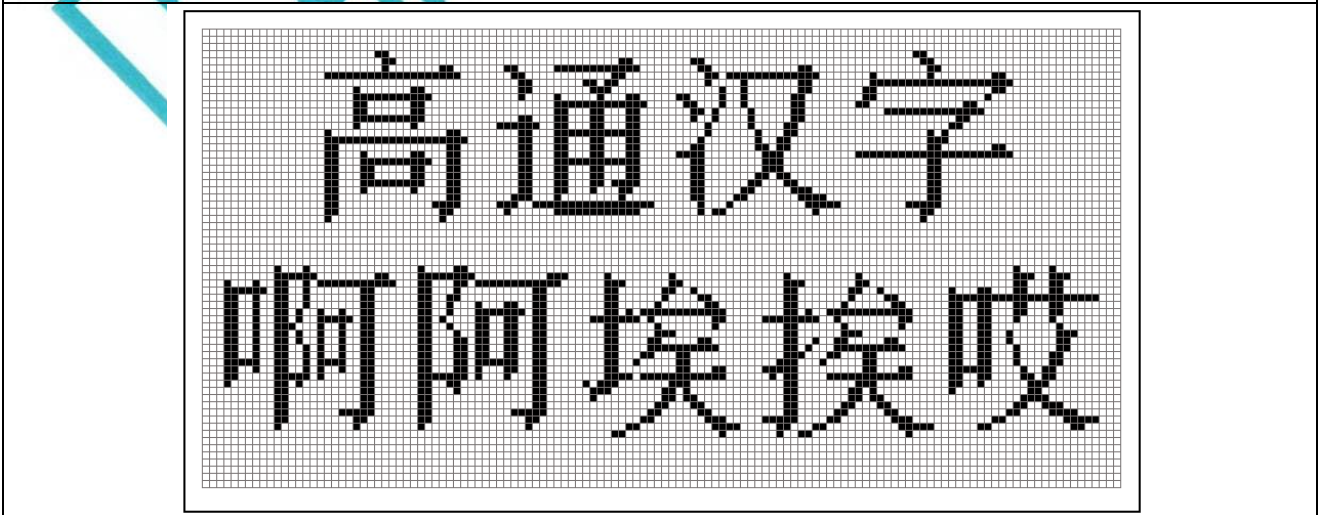
##### 12x12 点阵 GB2312 汉字



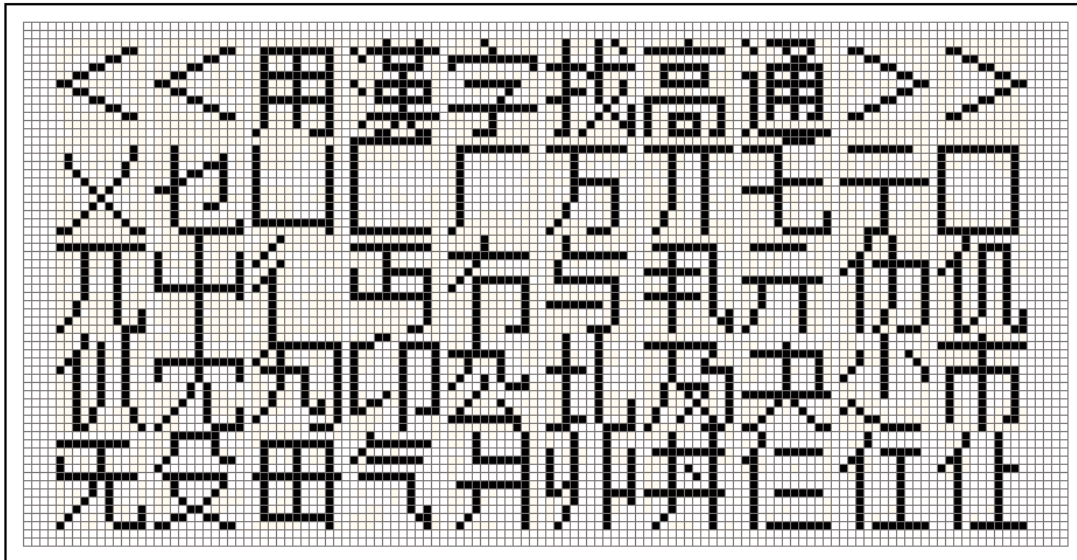
##### 16x16 点阵 GB2312 汉字



##### 24x24 点阵 GB2312 汉字



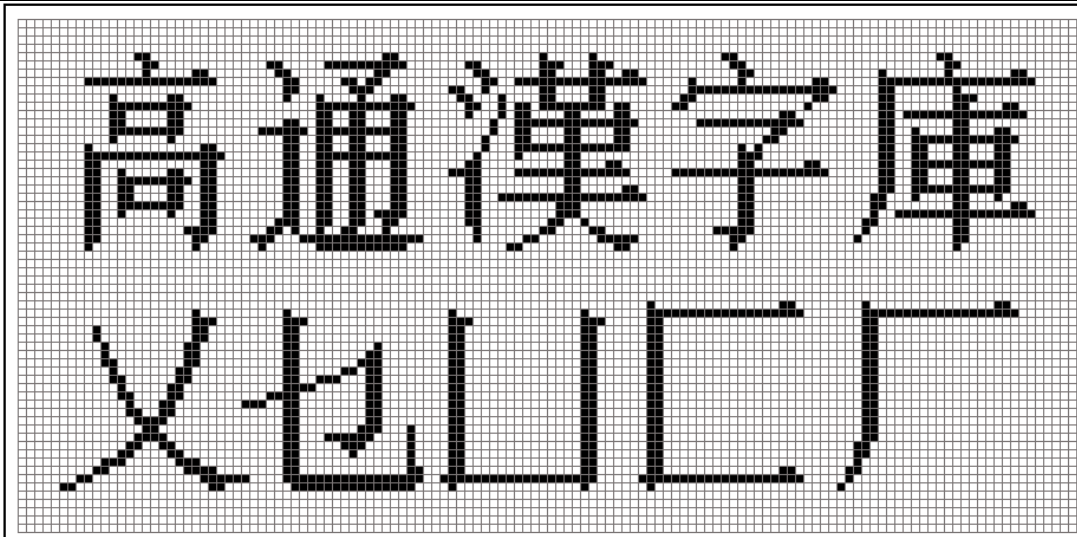
12x12 点阵 BIG5 汉字



16x16 点阵 BIG5 汉字



24x24 点阵 BIG5 汉字



1.3.2 其它点阵字符

5x7 点阵 ASCII 标准字符

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

7x8 点阵 ASCII 标准字符

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



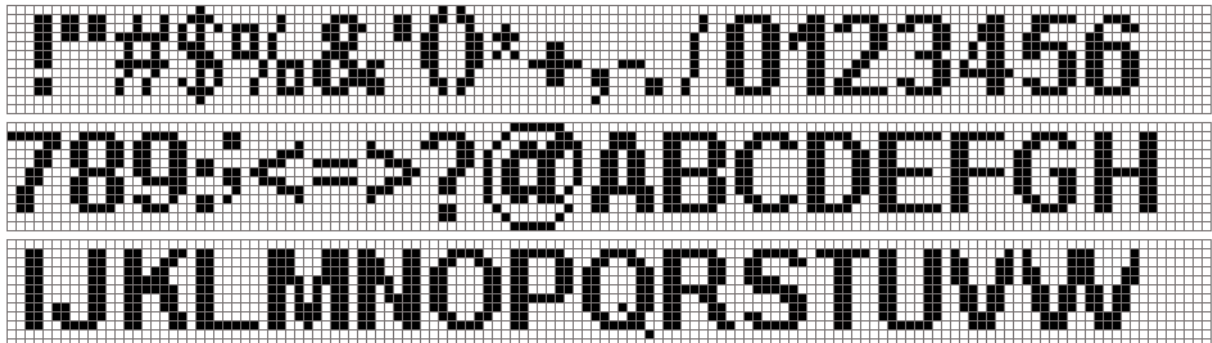
### 6x12 点阵 ASCII 标准字符

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

### 8x16 点阵 ASCII 标准字符

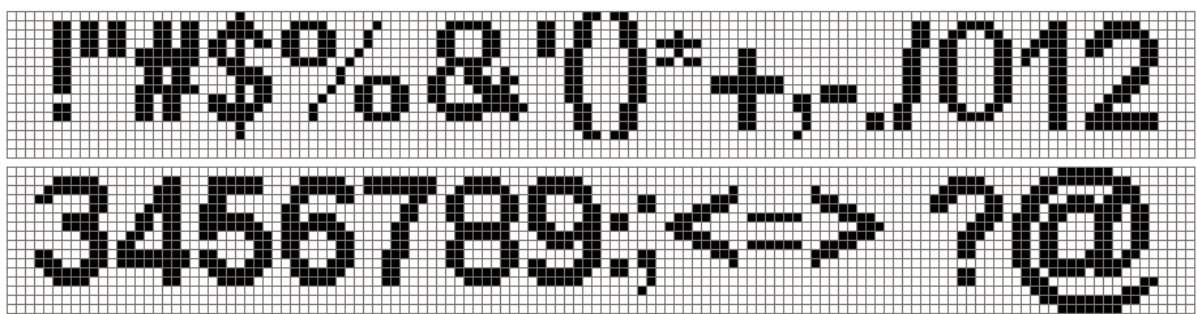
Low 4bit / High 4bit	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	

### 12 点阵不等宽 ASCII 方头 (Arial)



!"#\$%&'()\*+,-./0123456  
789:;<=>?@ABCDEFGHIJ  
KLMNOPQRSTUVWXYZ

### 16 点阵不等宽 ASCII 方头 (Arial)



!"#\$%&'()\*+,-./012  
3456789:;<=>?@

### 24 点阵不等宽 ASCII 方头 (Arial)



!"#\$%&'()\*+,-./01234567

## 2 操作指令

### 2.1 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	—	1 to $\infty$
FAST_READ	Read Data Bytes at Higher Speed	0000 1011	0B h	1	1 to $\infty$

所有对本芯片 SPI 接口的操作只有 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”)。

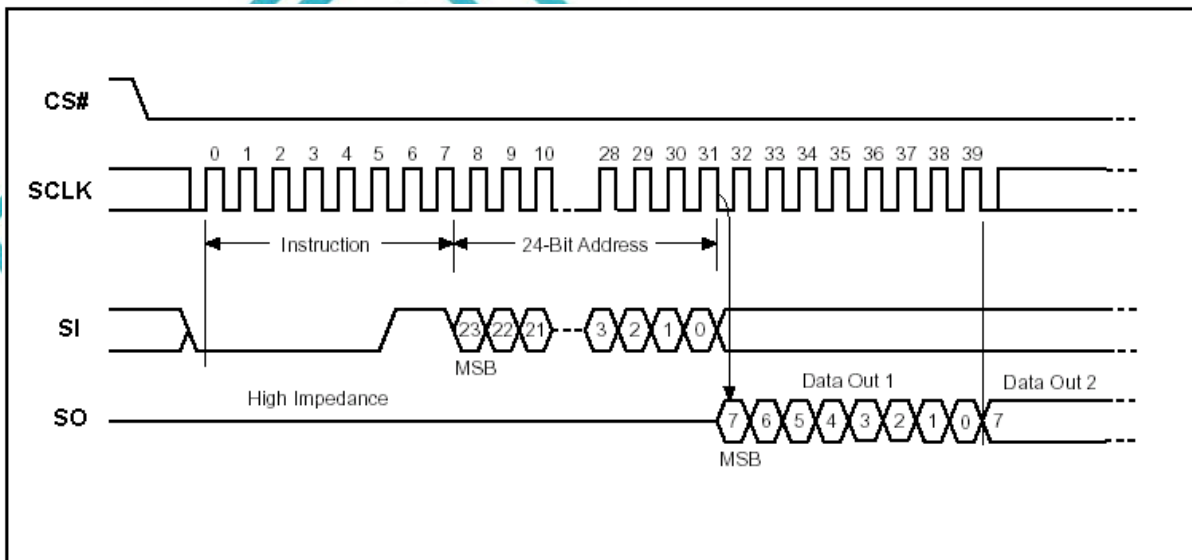
### 2.2 Read Data Bytes (一般读取)

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，紧接着的是 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



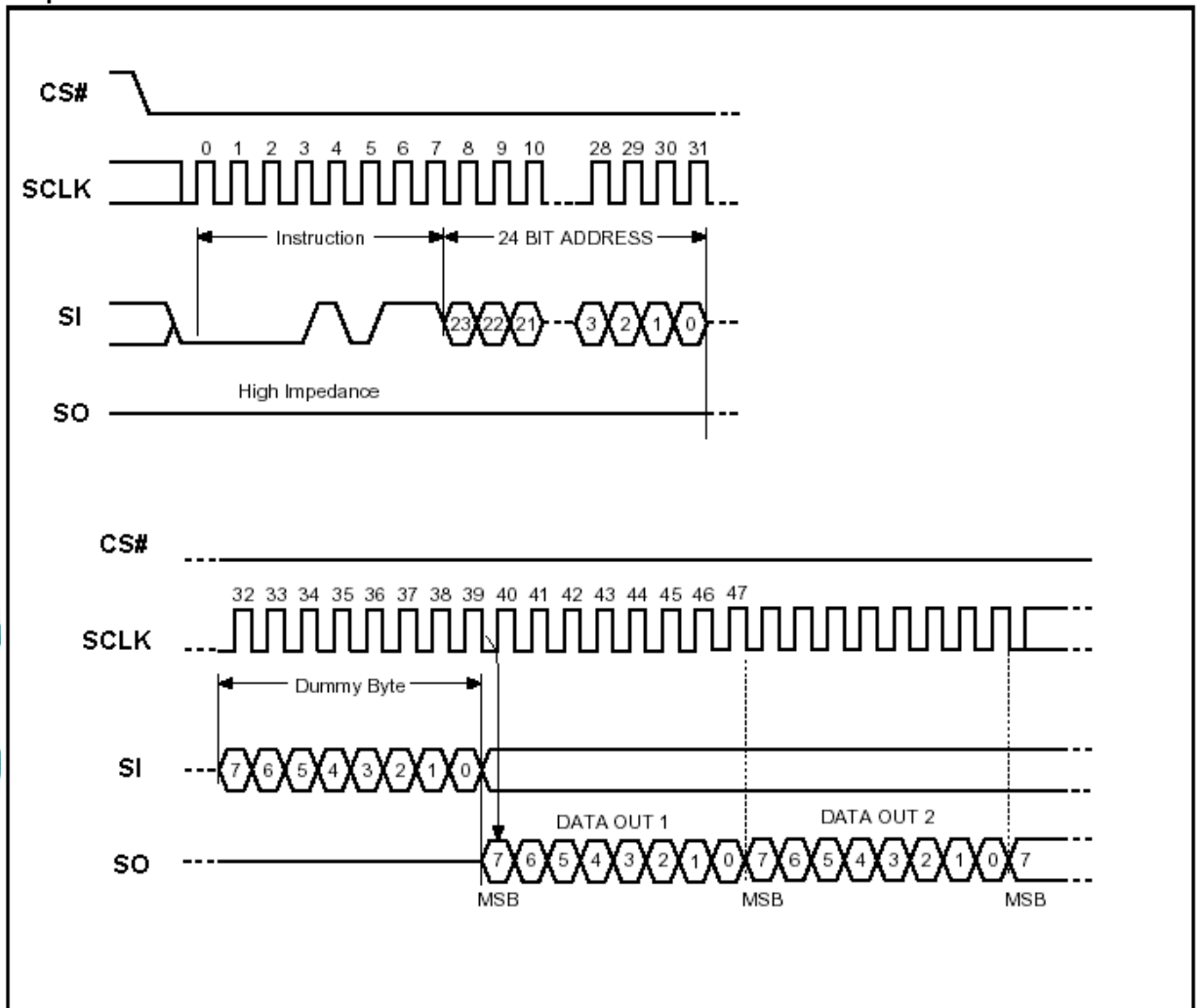
## 2.3 Read Data Bytes at Higher Speed (快速读取点阵数据)

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



### 3 字符点阵字库地址表

NO.	字库内容	编码体系	字符数	起始地址	参考算法
1	12x12 点阵 GB2312 字库	GB2312	6763+470	0x00000	<a href="#">4.1.1</a>
2	16x16 点阵 GB2312 字库	GB2312	6763+470	0x5F4C0	<a href="#">4.1.2</a>
3	24x24 点阵 GB2312 字库	GB2312	6763+470	0xDE5C0	<a href="#">4.1.3</a>
4	GB12345 索引表	GB12345	6866+662	0x190958	<a href="#">4.1.4-6</a>
5	BIG5 索引表	BIG5	5401+408	0x193F06	<a href="#">4.1.7-9</a>
6	Unicode 索引表	Unicode		0x19AC30	<a href="#">4.1.10-12</a>
7	GB12345 一对多索引表	GB12345	94	0x1A7020	<a href="#">4.1.13</a>
8	5x7 点阵 ASCII 标准字符	ASCII	96	0x1A7C33	<a href="#">4.2.1</a>
9	7x8 点阵 ASCII 标准字符	ASCII	96	0x1A76CC	<a href="#">4.2.2</a>
10	6x12 点阵 ASCII 标准字符	ASCII	96	0x1A79CC	<a href="#">4.2.3</a>
11	8x16 点阵 ASCII 标准字符	ASCII	96	0x1A7FCC	<a href="#">4.2.4</a>
12	12 点阵不等宽 ASCII 方头 (Arial) 字符	ASCII	96	0x1A87CC	<a href="#">4.2.5</a>
13	16 点阵不等宽 ASCII 方头 (Arial) 字符	ASCII	96	0x1A918C	<a href="#">4.2.6</a>
14	24 点阵不等宽 ASCII 方头 (Arial) 字符	ASCII	96	0x1A9E4C	<a href="#">4.2.7</a>
15	保留区			0x1FF6A4	



## 4 字符点阵数据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。

### 4.1 汉字字符点阵数据的地址计算

#### 4.1.1 12x12 点阵 GB2312 汉字&字符

**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=0x0;
if(MSB >=0xA1 && MSB <= 0xA3 && LSB >=0xA1)//全角字符 1 区 1 部分
    Address =( (MSB - 0xA1) * 94 + (LSB - 0xA1))*24+ BaseAdd;
else if(MSB ==0xA6 && LSB >=0xA1)//全角字符 1 区 2 部分
    Address =( (MSB - 0xA1) * 94 + (LSB - 0xA1)-94*2 ) *24+ BaseAdd;
else if(MSB ==0xA9 && LSB >=0xA1)//全角字符 1 区 3 部分
    Address =( (MSB - 0xA1) * 94 + (LSB - 0xA1)-94*4 ) *24+ BaseAdd;
else if(MSB >=0xB0 && MSB <= 0xF7 && LSB >=0xA1)//12 点阵汉字区
    Address = ((MSB - 0xB0) * 94 + (LSB - 0xA1)+ 662)*24+ BaseAdd;
```

#### 4.1.2 16x16 点阵 GB2312 汉字&字符

**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=0x5F4C0;
if(MSB >=0xA1 && MSB <= 0xA3 && LSB >=0xA1)//全角字符 1 区 1 部分
    Address =( (MSB - 0xA1) * 94 + (LSB - 0xA1))*32+ BaseAdd;
else if(MSB ==0xA6 && LSB >=0xA1)//全角字符 1 区 2 部分
    Address =( (MSB - 0xA1) * 94 + (LSB - 0xA1)-94*2 ) *32+ BaseAdd;
else if(MSB ==0xA9 && LSB >=0xA1)//全角字符 1 区 3 部分
    Address =( (MSB - 0xA1) * 94 + (LSB - 0xA1)-94*4 ) *32+ BaseAdd;
else if(MSB >=0xB0 && MSB <= 0xF7 && LSB >=0xA1)//16 点阵汉字区
    Address = ((MSB - 0xB0) * 94 + (LSB - 0xA1)+ 662)*32+ BaseAdd;
```

### 4.1.3 24x24 点阵 GB2312 汉字&字符

**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=0XDE5C0;

if(MSB >=0xA1 && MSB <= 0XA3 && LSB >=0xA1)//全角字符 1 区 1 部分

Address = (MSB - 0xA1) \* 94 + (LSB - 0xA1)\*72+ BaseAdd;

else if(MSB ==0xA6 && LSB >=0xA1)//全角字符 1 区 2 部分

Address = (MSB - 0xA1) \* 94 + (LSB - 0xA1)-94\*2)\*72+ BaseAdd;

else if(MSB ==0xA9 && LSB >=0xA1)//全角字符 1 区 3 部分

Address = (MSB - 0xA1) \* 94 + (LSB - 0xA1)-94\*4)\*72+ BaseAdd;

else if(MSB >=0xB0 && MSB <= 0xF7 && LSB >=0xA1)//24 点阵汉字区

Address = ((MSB - 0xB0) \* 94 + (LSB - 0xA1)+ 662)\*72+ BaseAdd;

### 4.1.4 12x12 点阵 GB12345 字符集汉字字符

**参数说明:**

FontCode表示汉字内码。

MSB 表示汉字内码FontCode的高8bits。

LSB 表示汉字内码FontCode的低8bits。

Address 表示汉字点阵在芯片中的字节地址。

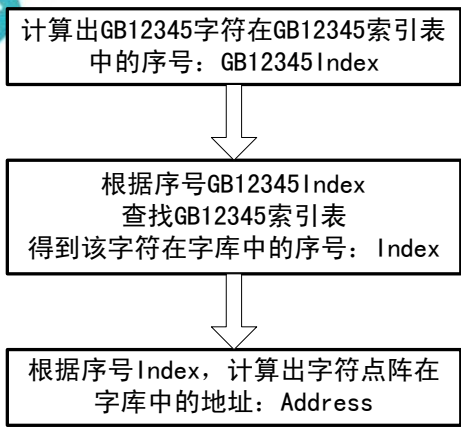
BaseAdd: 说明点阵数据在字库芯片中的起始地址。

GB12345Table: 表示 GB12345 索引表。表的起始地址为 0x190958

GB12345Index: 表示该内码汉字在 GB12345 索引表中的序号。由该序号可查 GB12345 索引表得出汉字在字库中的序号

Index: 表示该字符在字库中的序号。

**计算方法:**



BaseAdd=0x00;

if(MSB >=0xA1 && MSB <= 0XA3 && LSB >=0xA1)//全角字符 1 区 1 部分

```

Address = ( (MSB - 0xA1) * 94 + (LSB - 0xA1)) * 24 + BaseAdd;
else if (MSB == 0xA6 && LSB >= 0xA1) // 全角字符 1 区 2 部分
    Address = ( (MSB - 0xA1) * 94 + (LSB - 0xA1) - 94 * 2 ) * 24 + BaseAdd;
else if (MSB == 0xA9 && LSB >= 0xA1) // 全角字符 1 区 3 部分
    Address = ( (MSB - 0xA1) * 94 + (LSB - 0xA1) - 94 * 4 ) * 24 + BaseAdd;
else if (MSB >= 0xB0 && MSB <= 0xF9 && LSB >= 0xA1)
{
    GB12345Index = (MSB - 0xB0) * 94 + (LSB - 0xA1);
    Index = GB12345Table [GB12345Index * 2] * 256 + GB12345Table [GB12345Index * 2 + 1];
    Address = Index * 24 + BaseAdd;
}

```

#### 4.1.5 16x16 点阵 GB12345 字符集汉字字符

##### 参数说明:

FontCode 表示汉字内码。

MSB 表示汉字内码 FontCode 的高 8bits。

LSB 表示汉字内码 FontCode 的低 8bits。

Address 表示汉字点阵在芯片中的字节地址。

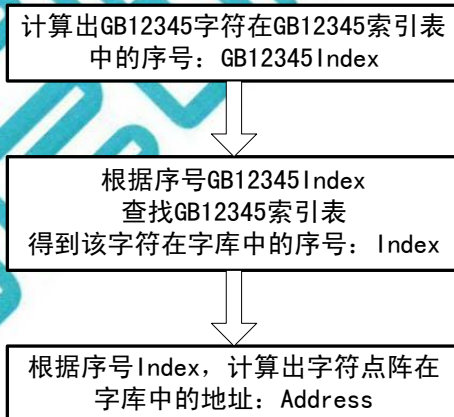
BaseAdd: 说明点阵数据在字库芯片中的起始地址。

GB12345Table: 表示 GB12345 索引表。表的起始地址为 0x190958

GB12345Index: 表示该内码汉字在 GB12345 索引表中的序号。由该序号可查 GB12345 索引表得出汉字在字库中的序号

Index: 表示该字符在字库中的序号。

##### 计算方法:



BaseAdd=0x5F4C0;

if (MSB >= 0xA1 && MSB <= 0xA3 && LSB >= 0xA1) // 全角字符 1 区 1 部分

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

else if (MSB == 0xA6 && LSB >= 0xA1) // 全角字符 1 区 2 部分

Address = ( (MSB - 0xA1) \* 94 + (LSB - 0xA1) - 94 \* 2 ) \* 32 + BaseAdd;

else if (MSB == 0xA9 && LSB >= 0xA1) // 全角字符 1 区 3 部分

Address = ( (MSB - 0xA1) \* 94 + (LSB - 0xA1) - 94 \* 4 ) \* 32 + BaseAdd;

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

{



```

GB12345Index = (MSB - 0xB0) * 94 + (LSB - 0xA1);
Index = GB12345Table[GB12345Index*2] * 256 + GB12345Table[GB12345Index*2+1];
Address =Index * 32 + BaseAdd;
}

```

#### 4.1.6 24x24 点阵 GB12345 基本集汉字字符

##### 参数说明:

FontCode表示汉字内码。

MSB 表示汉字内码FontCode的高8bits。

LSB 表示汉字内码FontCode的低8bits。

Address 表示汉字点阵在芯片中的字节地址。

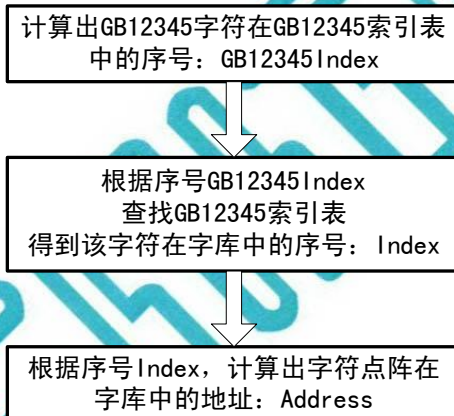
BaseAdd: 说明点阵数据在字库芯片中的起始地址。

GB12345Table: 表示 GB12345 索引表。表的起始地址为 0x190958

GB12345Index: 表示该内码汉字在 GB12345 索引表中的序号。由该序号可查 GB12345 索引表得出汉字在字库中的序号

Index: 表示该字符在字库中的序号。

##### 计算方法:



BaseAdd=0XDE5C0;

if(MSB >=0xA1 && MSB <= 0xA3 && LSB >=0xA1)//全角字符 1 区 1 部分

Address =( MSB - 0xA1) \* 94 + (LSB - 0xA1)\*72+ BaseAdd;

else if(MSB ==0xA6 && LSB >=0xA1)//全角字符 1 区 2 部分

Address =( MSB - 0xA1) \* 94 + (LSB - 0xA1)-94\*2)\*72+ BaseAdd;

else if(MSB ==0xA9 && LSB >=0xA1)//全角字符 1 区 3 部分

Address =( MSB - 0xA1) \* 94 + (LSB - 0xA1)-94\*4)\*72+ BaseAdd;

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

{

GB12345Index = (MSB - 0xB0) \* 94 + (LSB - 0xA1);

Index = GB12345Table[GB12345Index\*2] \* 256 + GB12345Table[GB12345Index\*2+1];

Address =Index \* 72 + BaseAdd;

}

#### 4.1.7 12x12 点阵 BIG5 字符集汉字字符

##### 参数说明:

FontCode表示汉字内码。

MSB 表示汉字内码FontCode的高8bits。

LSB 表示汉字内码FontCode的低8bits。

Address 表示汉字点阵在芯片中的字节地址。

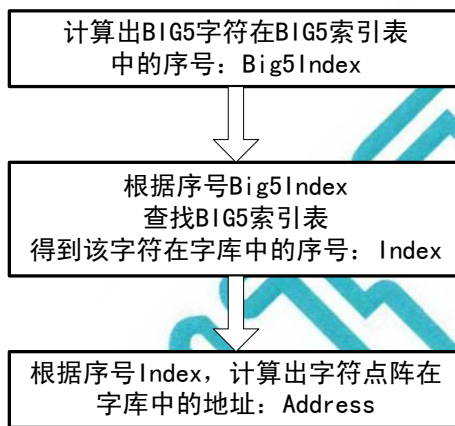
BaseAdd: 说明点阵数据在字库芯片中的起始地址。

Big5Table: 表示 BIG5 索引表。表的起始地址为 0x193F06

Big5Index: 表示该内码汉字在 BIG5 索引表中的序号。由该序号可查 BIG5 索引表得出汉字在字库中的序号

Index: 表示该字符在字库中的序号。

##### 计算方法:



```

BaseAdd=0x00;
if(MSB >=0xA1 && MSB <= 0XF9)
{
    if(LSB >=0x40 && LSB <= 0X7E)
        Big5Index =(MSB - 0xA1) * 157 + (LSB - 0x40);
    else if(LSB >=0XA1 && LSB <= 0XFE)
        Big5Index =(MSB - 0xA1) * 157 + 63 + (LSB - 0XA1));
}
Index = Big5Table[Big5Index*2] * 256 + Big5Table[Big5Index*2+1];
Address =Index * 24 + BaseAdd;
  
```

#### 4.1.8 16x16 点阵 BIG5 字符集汉字字符

##### 参数说明:

FontCode表示汉字内码。

MSB 表示汉字内码FontCode的高8bits。

LSB 表示汉字内码FontCode的低8bits。

Address 表示汉字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。

Big5Table: 表示 BIG5 索引表。表的起始地址为 0x193F06

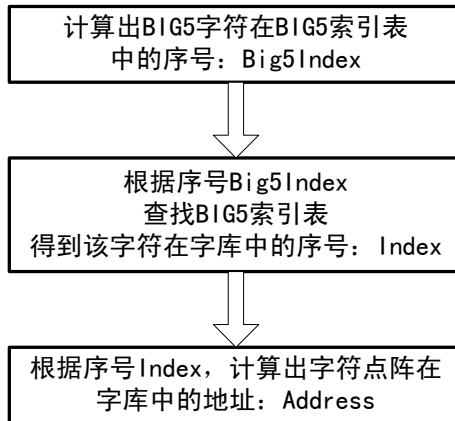
Big5Index: 表示该内码汉字在 BIG5 索引表中的序号。由该序号可查 BIG5 索引表得出汉字在字库中的序



号

Index: 表示该字符在字库中的序号。

计算方法:



```

BaseAdd=0x5F4C0;
if(MSB >=0xA1 && MSB <= 0XF9)
{
    if(LSB >=0x40 && LSB <= 0X7E)
        Big5Index =(MSB - 0xA1) * 157 + (LSB - 0x40);
    else if(LSB >=0XA1 && LSB <= 0XFE)
        Big5Index =(MSB - 0xA1) * 157 + 63 + (LSB - 0xA1));
}
Index = Big5Table[Big5Index*2] * 256 + Big5Table[Big5Index*2+1];
Address =Index * 32 + BaseAdd;
  
```

#### 4.1.9 24x24 点阵 BIG5 基本集汉字字符

参数说明:

FontCode表示汉字内码。

MSB 表示汉字内码FontCode的高8bits。

LSB 表示汉字内码FontCode的低8bits。

Address 表示汉字点阵在芯片中的字节地址。

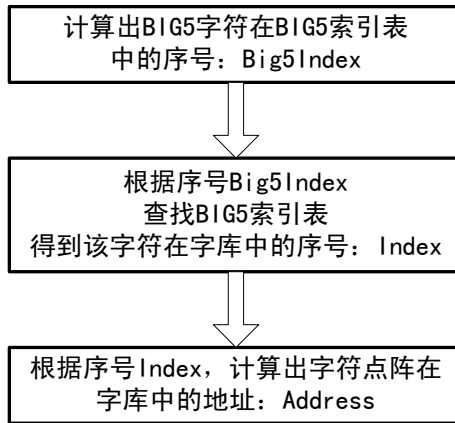
BaseAdd: 说明点阵数据在字库芯片中的起始地址。

Big5Table: 表示 BIG5 索引表。表的起始地址为 0x193F06

Big5Index: 表示该内码汉字在 BIG5 索引表中的序号。由该序号可查 BIG5 索引表得出汉字在字库中的序号

Index: 表示该字符在字库中的序号。由于 24 点字库只支持到 BIG5 基本集字符, 总共有 10139 个字符。所以当序号 Index 小于 10139 时才有效。

计算方法:



```

BaseAdd=0XDE5C0;
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 - 0xA1));
}
Index = Big5Table[Big5Index*2] * 256 + Big5Table[Big5Index*2+1];
If(Index<10139)
  Address =Index * 72 + BaseAdd;
  
```

#### 4.1.10 12x12 点阵 Unicode 中文字符

##### 参数说明:

FontCode表示中文内码。

Address 表示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。

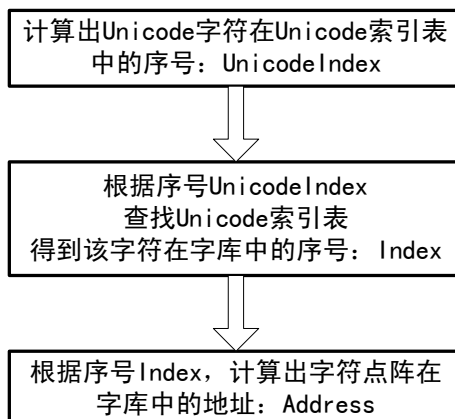
UnicodeTable: 表示 Unicode 索引表。表的起始地址为 0x19AC30

UnicodeToIndex(): 根据 Unicode 内码计算出该字符在 Unicode 索引表中的序号。

UnicodeIndex: 表示该内码字符在 Unicode 索引表中的序号。由该序号可查 Unicode 索引表得出字符在字库中的序号

Index: 表示该字符在字库中的序号。

##### 计算方法:



```
BaseAdd=0x00;
UnicodeIndex = UnicodeToIndex(FontCode);
Index = UnicodeTable [UnicodeIndex *2] * 256 + UnicodeTable [UnicodeIndex *2+1] ;
Address =Index * 24 + BaseAdd;
```

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

```
{
    BYTE result=0;
    WORD h=0;

    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< 0xfe0) result=1;
    else if(code<=0xfe5) h=code-160-7102-2493-6698-18374-4295-1030-149-129;
    else result=1;

    if(result==1)
    {
        h = 0x3000-160-7102-2493;
    }

    return h;
}
```

#### 4.1.11 16x16 点阵 Unicode 中文字符

##### 参数说明:

FontCode表示中文内码。

Address 表示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。

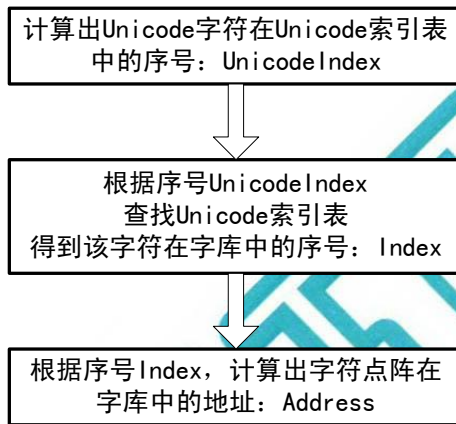
UnicodeTable: 表示 Unicode 索引表。表的起始地址为 0x19AC30

UnicodeToIndex(): 根据 Unicode 内码计算出该字符在 Unicode 索引表中的序号。

UnicodeIndex: 表示该内码字符在 Unicode 索引表中的序号。由该序号可查 Unicode 索引表得出字符在字库中的序号

Index: 表示该字符在字库中的序号。

##### 计算方法:



BaseAdd=0x5F4C0;

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=0;
```

```
    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;
}

```

#### 4.1.12 24x24 点阵 Unicode 中文字符

##### 参数说明:

FontCode表示中文内码。

Address 表示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。

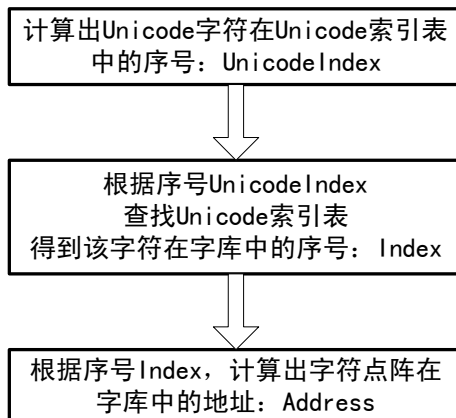
UnicodeTable: 表示 Unicode 索引表。表的起始地址为 0x19AC30

UnicodeToIndex(): 根据 Unicode 内码计算出该字符在 Unicode 索引表中的序号。

UnicodeIndex: 表示该内码字符在 Unicode 索引表中的序号。由该序号可查 Unicode 索引表得出字符在字库中的序号

Index: 表示该字符在字库中的序号。由于 24 点字库只支持到 BIG5 基本集字符，总共有 10139 个字符。所以当序号 Index 小于 10139 时才有效。

##### 计算方法:





```
BaseAdd=0XDE5C0;
UnicodeIndex = UnicodeToIndex(FontCode);
Index = UnicodeTable [UnicodeIndex *2] * 256 + UnicodeTable [UnicodeIndex *2+1] ;
If(Index<10139)
    Address =Index * 72 + BaseAdd;
```

WORD UnicodeToIndex(WORD code)

```
{
    BYTE result=0;
    WORD h=0;

    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;
}
```

### 4.1.13 GB12345 一对多索引表

GB12345 字符集中，有 94 个内码，同一个内码有不同的写法。如下表：

GB2312	GB12345	GB2312	GB12345	GB2312	GB12345	GB2312	GB12345
摆	擺擺	胡	胡鬚	蒙	蒙濛濛	系	系係繫
板	板闆	划	劃划	弥	彌瀾	咸	咸鹹
表	表錶	回	回迴	面	面麵	向	向嚮
别	別弊	汇	匯彙	蔑	蔑巖	须	須鬚
卜	卜蔔	获	獲穫	辟	辟闢	药	藥葯
才	才纜	饥	饑飢	苹	蘋苹	叶	葉叶
厂	廠厂	几	幾几	凭	憑凭	郁	鬱郁
冲	衝冲	家	家傢	扑	撲扑	御	御禦
丑	醜丑	价	價价	仆	僕仆	吁	吁籲
出	出齣	荐	薦荐	朴	樸朴	愿	願愿
当	當噹	姜	姜薑	千	千韃	云	雲去
党	黨党	尽	盡儘	签	簽籤	脏	臟髒
淀	澱淀	据	據据	纤	纖緯	症	癥症
冬	冬冬	卷	卷捲	秋	秋鞦	只	祇只隻
斗	鬥斗	克	克剋	曲	曲糲	致	致緻
恶	惡噁	夸	誇夸	确	確确	制	制製
发	發髮	困	困暈	舍	捨舍	种	種种
范	範范	蜡	蠟蜡	术	術术	朱	朱硃
丰	豐丰	腊	臘腊	松	鬆松	筑	築筑
复	復復	累	累纍	苏	蘇蘇		
干	幹干	里	裏里	台	臺台檯颱		
谷	谷穀	历	歷曆	坛	壇壇		
刮	刮颯	帘	簾帘	涂	塗涂		
广	廣广	卤	滷滷	团	團糰		
合	合閤	霉	霉霉	万	萬万		

GB12345 一对多索引表的结构：

Struct GB12345\_MultiTable

```

{
    WORD incode; //GB12345 汉字内码
    WORD index1; //该内码的第一个字在字库中的序号
    WORD index2; //该内码的第二个字在字库中的序号
    WORD index3; //该内码的第三个字在字库中的序号
    WORD index4; //该内码的第四个字在字库中的序号
}

```

## 4.2 ASCII 字符的地址计算

### 4.2.1 5x7 点阵 ASCII 标准字符

**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=0x1A73CC

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

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

### 4.2.2 7x8 点阵 ASCII 标准字符

**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=0x1A76CC

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

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

### 4.2.3 6x12 点阵 ASCII 标准字符

**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=0x1A79CC

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

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

### 4.2.4 8x16 点阵 ASCII 标准字符

**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=0x1A7FCC

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

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

#### 4.2.5 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=0x1A87CC

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

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

#### 4.2.6 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=0x1A918C

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

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

#### 4.2.7 24 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=0x1A9E4C

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

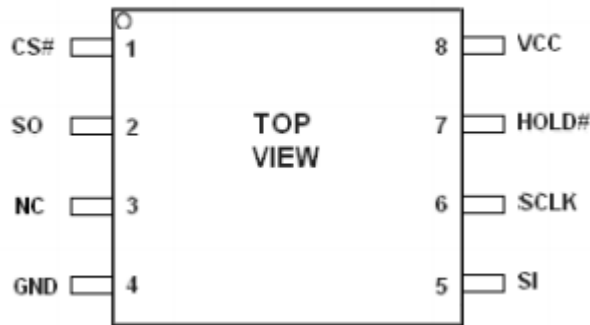
Address = (ASCIICode - 0x20) \* 74 + BaseAdd



## 5 引脚描述与电路连接

### 5.1 引脚配置

SOP8-B



SOP8-B

NO.	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)

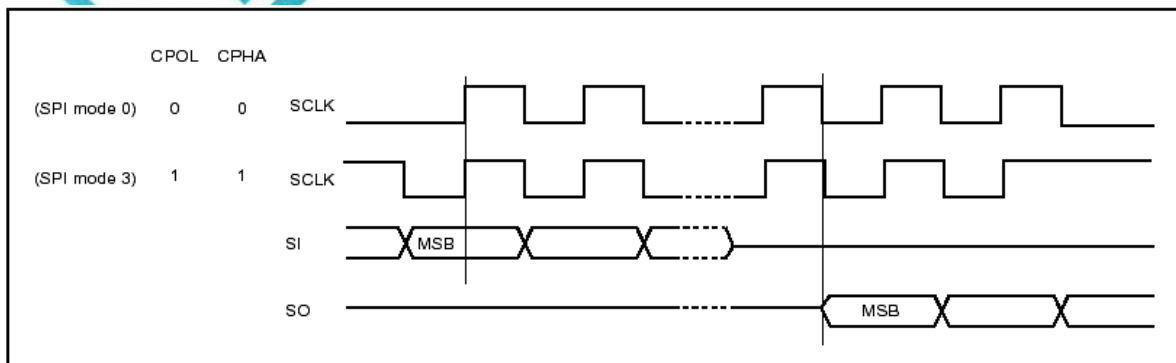
### 5.2 引脚描述

**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。



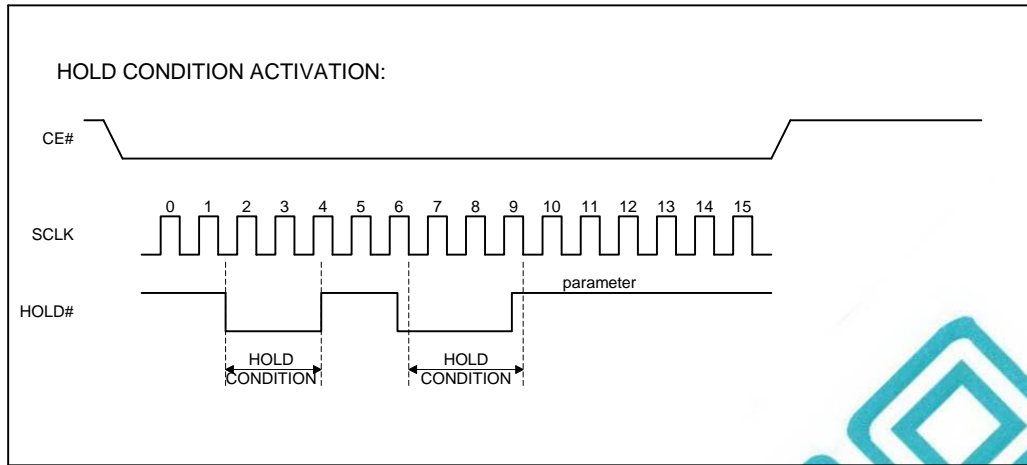
**总线挂起输入 (HOLD#):**

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。

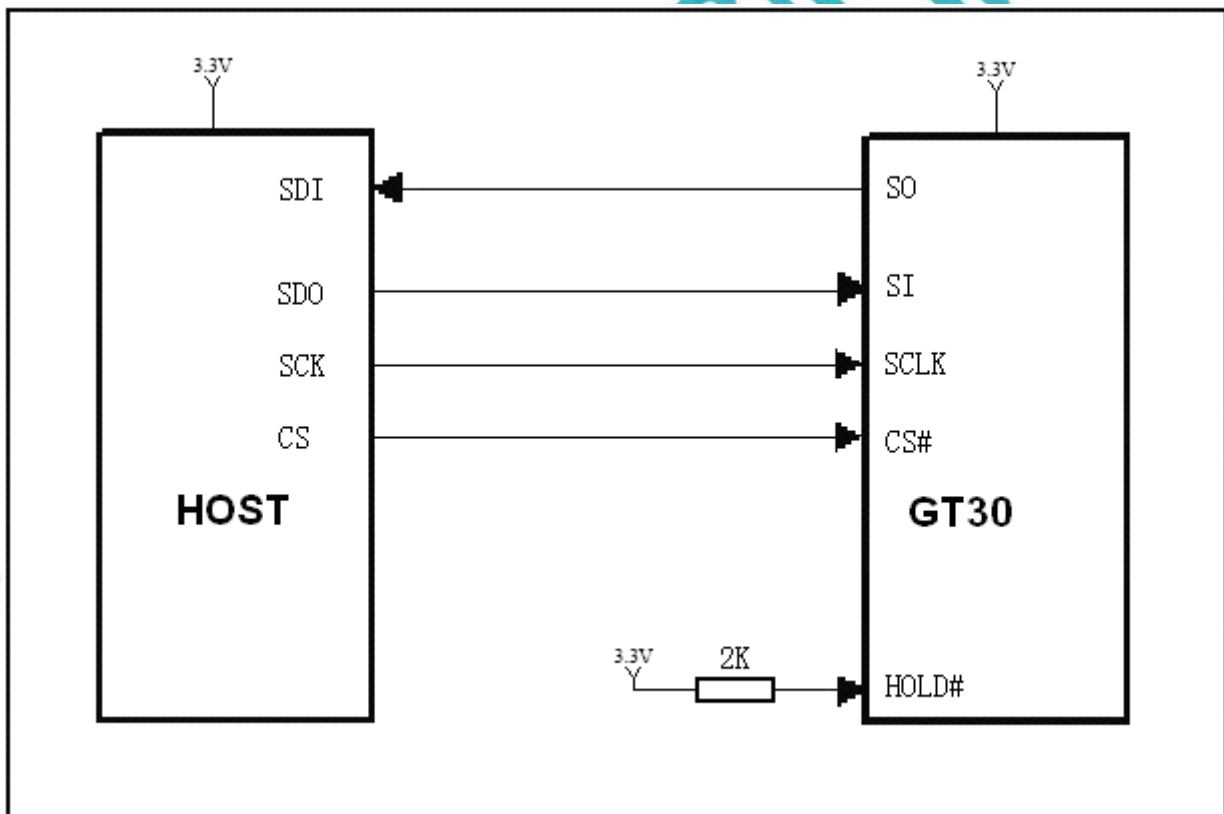
当HOLD#信号变为低并且串行时钟信号（SCLK）处于低电平时，进入总线挂起状态。

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。



### 5.3 HOST CPU 主机接口与 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

## 6 电气特性

### 6.1 绝对最大额定值

Symbol	Parameter	Min.	Max.	Unit	Condition
T <sub>OP</sub>	Operating Temperature	-40	85	°C	SPI mode
T <sub>STG</sub>	Storage Temperature	-55	125	°C	
V <sub>CC</sub>	Supply Voltage	-0.5	4.0	V	
V <sub>IN</sub>	Input Voltage	-0.5	4.0	V	
GND	Power Ground	0	0	V	

### 6.2 DC 特性

Condition: T<sub>OP</sub> = -40°C to 85°C, GND=0V in SPI mode;

Symbol	Parameter	Min.	Max.	Unit	Condition
I <sub>DD</sub>	VCC Supply Current(active)		20	mA	VCC=2.7V-3.6V
I <sub>SB</sub>	VCC Standby Current		5	uA	
V <sub>IL</sub>	Input LOW Voltage	-0.3	0.6	V	
V <sub>IH</sub>	Input HIGH Voltage	0.7VCC	VCC+0.3	V	
V <sub>OL</sub>	Output LOW Voltage		0.4 (I <sub>OL</sub> =1.6mA)	V	
V <sub>OH</sub>	Output HIGH Voltage	VCC-0.2 (I <sub>OH</sub> =100uA)		V	
I <sub>LI</sub>	Input Leakage Current	0	±2	uA	
I <sub>LO</sub>	Output Leakage Current	0	±2	uA	

Note: I<sub>IL</sub>: Input LOW Current, I<sub>IH</sub>: Input HIGH Current,  
 I<sub>OL</sub>: Output LOW Current, I<sub>OH</sub>: Output HIGH Current,

### 6.3 AC 特性

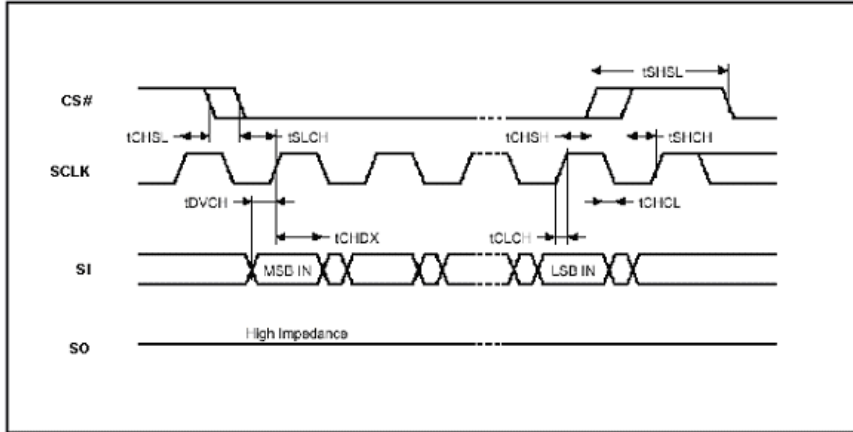
#### 6.3.1 SPI 接口模式下 AC 特性

Condition: T<sub>OP</sub> = -20°C to 85°C, VCC= 2.7V to 3.6V

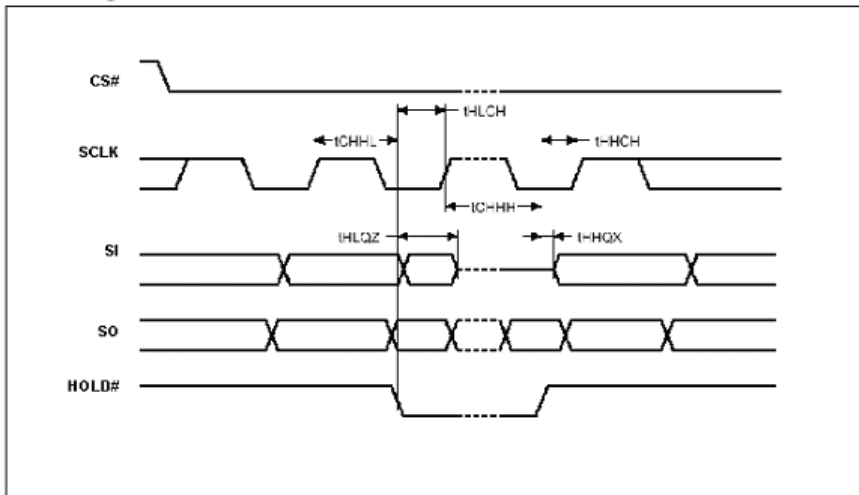
Symbol	Alt.	Parameter	Min.	Max.	Unit
F <sub>c</sub>	F <sub>c</sub>	Clock Frequency	D.C.	80	MHz
t <sub>CH</sub>	t <sub>CLH</sub>	Clock High Time	4		ns
t <sub>CL</sub>	t <sub>CLL</sub>	Clock Low Time	4		ns
t <sub>CLCH</sub>		Clock Rise Time(peak to peak)	0.2		V/ns
t <sub>CHCL</sub>		Clock Fall Time (peak to peak)	0.2		V/ns
t <sub>SLCH</sub>	t <sub>css</sub>	CS# Active Setup Time (relative to SCLK)	5		ns
t <sub>CHSL</sub>		CS# Not Active Hold Time (relative to SCLK)	5		ns
t <sub>DVCH</sub>	t <sub>dsu</sub>	Data In Setup Time	2		ns
t <sub>CHDX</sub>	t <sub>dh</sub>	Data In Hold Time	2		ns
t <sub>CHSH</sub>		CS# Active Hold Time (relative to SCLK)	5		ns
t <sub>SHCH</sub>		CS# Not Active Setup Time (relative to SCLK)	5		ns
t <sub>SHSL</sub>	t <sub>CSH</sub>	CS# Deselect Time	100		ns
t <sub>SHQZ</sub>	t <sub>dis</sub>	Output Disable Time		6	ns
t <sub>CLQV</sub>	t <sub>v</sub>	Clock Low to Output Valid		7	ns

t <sub>CLQX</sub>	t <sub>HO</sub>	Output Hold Time	0		ns
t <sub>HLCH</sub>		HOLD# Setup Time (relative to SCLK)	5		ns
t <sub>CHHH</sub>		HOLD# Hold Time (relative to SCLK)	5		ns
t <sub>HHCH</sub>		HOLD Setup Time (relative to SCLK)	5		ns
t <sub>CHHL</sub>		HOLD Hold Time (relative to SCLK)	5		ns
t <sub>HHQX</sub>	t <sub>LZ</sub>	HOLD to Output Low-Z		6	ns
t <sub>HLQZ</sub>	t <sub>HZ</sub>	HOLD# to Output High-Z		6	ns

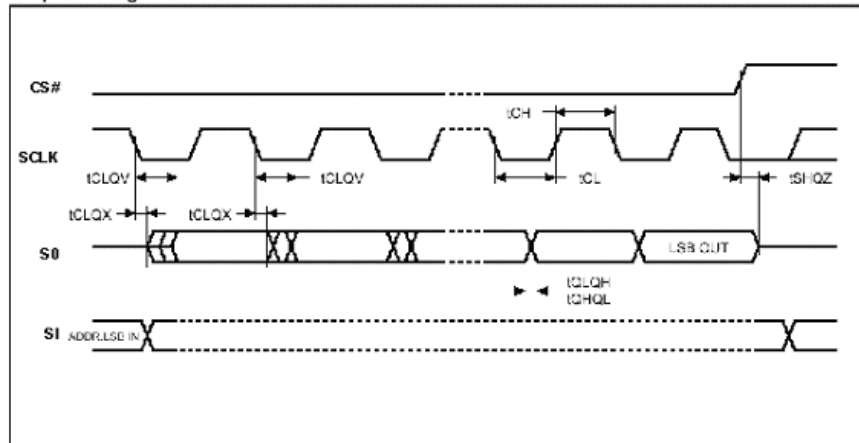
Serial Input Timing



Hold Timing



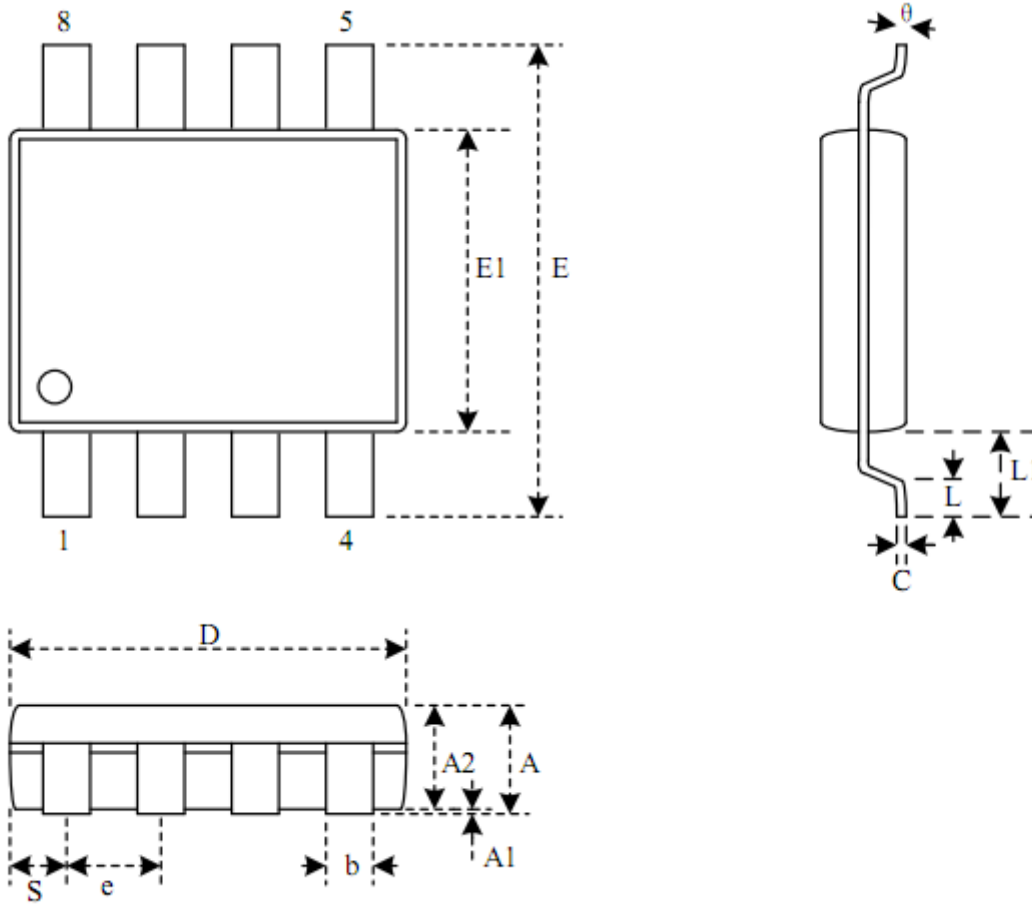
Output Timing





## 7 封装尺寸：SOP8-B

单位: mm



### Dimensions

Symbol		A	A1	A2	b	C	D	E	E1	e	L	L1	S	$\theta$
Unit														
mm	Min		0.05	1.70	0.36	0.19	5.13	7.70	5.18		0.50	1.21	0.62	0
	Nom		0.15	1.80	0.41	0.20	5.23	7.90	5.28	1.27	0.65	1.31	0.74	5
	Max	2.16	0.25	1.91	0.51	0.25	5.33	8.10	5.38		0.80	1.41	0.88	8
Inch	Min		0.002	0.067	0.014	0.007	0.202	0.303	0.204		0.020	0.048	0.024	0
	Nom		0.006	0.071	0.016	0.008	0.206	0.311	0.208	0.050	0.026	0.052	0.029	5
	Max	0.085	0.010	0.075	0.020	0.010	0.210	0.319	0.212		0.031	0.056	0.035	8

## 8 附录

### 8.1 GB2312 1 区字符 (470 字符)

GB2312 标准点阵字符 1 区对应码位的 A1A1~A9EF 共计 846 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	—	~		…	‘	’
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		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	{		}	—	

GB2312 1 区

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	ξ	η	θ	ι	κ	λ	μ	ν	ξ	ο
D	π	ρ	σ	τ	υ	φ	χ	ψ	ω	,	°	`	:	;	!	?
E	ˆ	˘	˘	˘	˘	˘	˘	˘	˘	˘	˘	˘	˘	˘	˘	˘
F	˘	˘		∴		∴										

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

8.2 补丁文件

序号	补丁描述	修补方式	对应文档	版本号	发布时间
1	12、16点 UNICODE 对应字 库表。	码本对应表	《GT30L24T3Y-12-16-UNICOD E 字库码本》	V1.00	2010-8-20
2	24点 UNICODE 对应字库表	码本对应表	《GT30L24T3Y-24-UNICODE 字库码本》	V1.00	2010-8-20
3	12、16点 BIG5 对 应字库表	码本对应表	《GT30L24T3Y-12-16-BIG5 字 库码本》	V1.00	2010-8-20
4	24点 BIG5 对应字 库表	码本对应表	《GT30L24T3Y-24-BIG5 字库码 本》	V1.00	2010-8-20
5	16点 BIG5 缺损字 符码本	码本表	《GT30L24T3Y-16-BIG5 缺损字 符码本》	V1.00	2010-8-20
6	24点 BIG5 缺损字 符码本	码本表	《GT30L24T3Y-24-BIG5 缺损字 符码本》	V1.00	2010-8-20