

GT30L32S4W 标准汉字字库芯片

规格书 DATASHEET

- 字符集：GB2312
- 字号：12x12、16x16、24x24、32x32 点阵
- 排置方式：横置横排
- 总线接口：SPI 串行总线
- 封装类型：SOP8-B

V 1.0 I _A

2013-04

版本修订记录

[illegible]

目 录

1 概述	4
1.1 芯片特点	4
1.2 芯片内容	5
1.3 字型样张	6
2 操作指令	12
2.1 指令参数	12
2.2 Read Data Bytes（一般读取）	12
2.3 Read Data Bytes at Higher Speed（快速读取点阵数据）	13
3 字符点阵字库地址表	14
4 字符点阵数据在芯片中的地址计算方法	15
4.1 汉字字符点阵数据的地址计算	15
4.2 ASCII 字符的地址计算	18
5 引脚描述与接口连接	22
5.1 引脚配置	22
5.2 引脚描述	22
5.3 HOST CPU 主机接口与 SPI 接口电路示意图	23
6 电气特性	24
6.1 绝对最大额定值	24
6.2 DC 特性	24
6.3 AC 特性	24
7 封装尺寸: SOP8-B	26
8 附录	27
8.1 GB2312 1 区字符 (846 字符)	27
8.2 8×16 点国标扩展字符 (126 字符)	30
8.3 8×16 点特殊字符 (64 字符)	31

1 概述

GT30L32S4W是一款内含12x12点阵、16x16点阵、24x24点阵、32x32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）及ASCII字符。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。

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℃

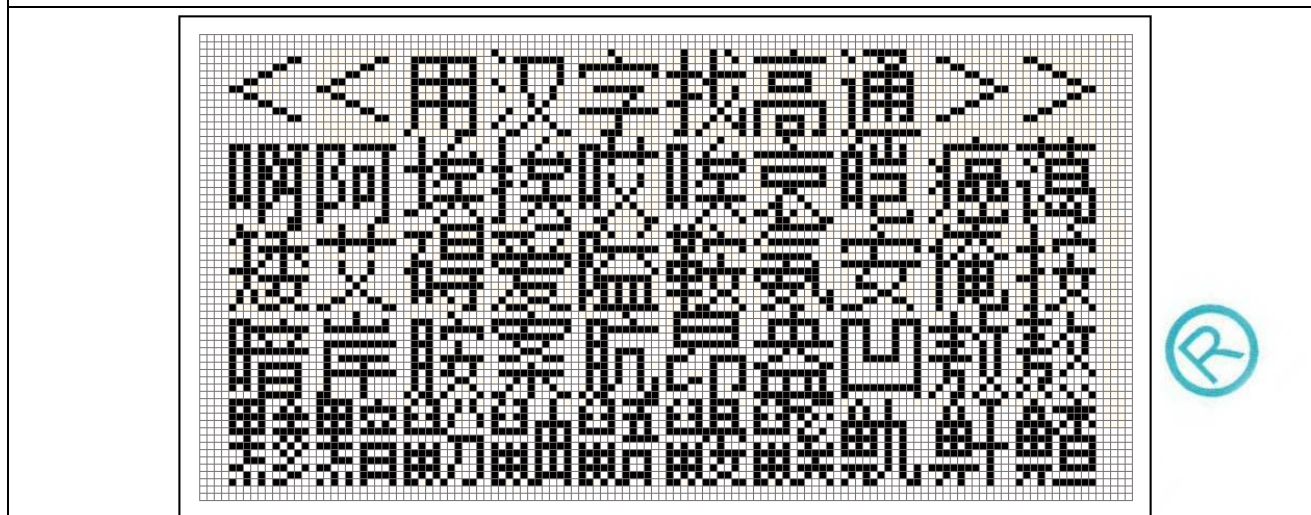
1.2 芯片内容

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	W-横置横排	
	ASCII	7x8	96	标准	W-横置横排	
	ASCII	6x12	96	标准	W-横置横排	
	ASCII	8x16	96	标准	W-横置横排	
	ASCII	12x24	96	标准	W-横置横排	
	ASCII	16x32	96	标准	W-横置横排	
	ASCII	12 点阵不等宽	96	Arial (方头)	W-横置横排	
	ASCII	12 点阵不等宽	96	Times new Roman (白正)	W-横置横排	
	ASCII	16 点阵不等宽	96	Arial (方头)	W-横置横排	
	ASCII	16 点阵不等宽	96	Times new Roman (白正)	W-横置横排	
	ASCII	24 点阵不等宽	96	Arial (方头)	W-横置横排	
	ASCII	24 点阵不等宽	96	Times new Roman (白正)	W-横置横排	
	ASCII	32 点阵不等宽	96	Arial (方头)	W-横置横排	
	ASCII	32 点阵不等宽	96	Times new Roman (白正)	W-横置横排	
汉字 字符集	GB2312 汉字	12x12	6763	宋体	W-横置横排	
		16x16	6763	宋体	W-横置横排	
		24x24	6763	宋体	W-横置横排	
		32x32	6763	宋体	W-横置横排	
	GB2312 字符	12x12	846	宋体	W-横置横排	
		16x16	846	宋体	W-横置横排	
		24x24	846	宋体	W-横置横排	
		32x32	846	宋体	W-横置横排	
	国标扩 展字符	6x12	126	宋体	W-横置横排	
		8x16	126	宋体	W-横置横排	
		12x24	126	宋体	W-横置横排	
		16x32	126	宋体	W-横置横排	

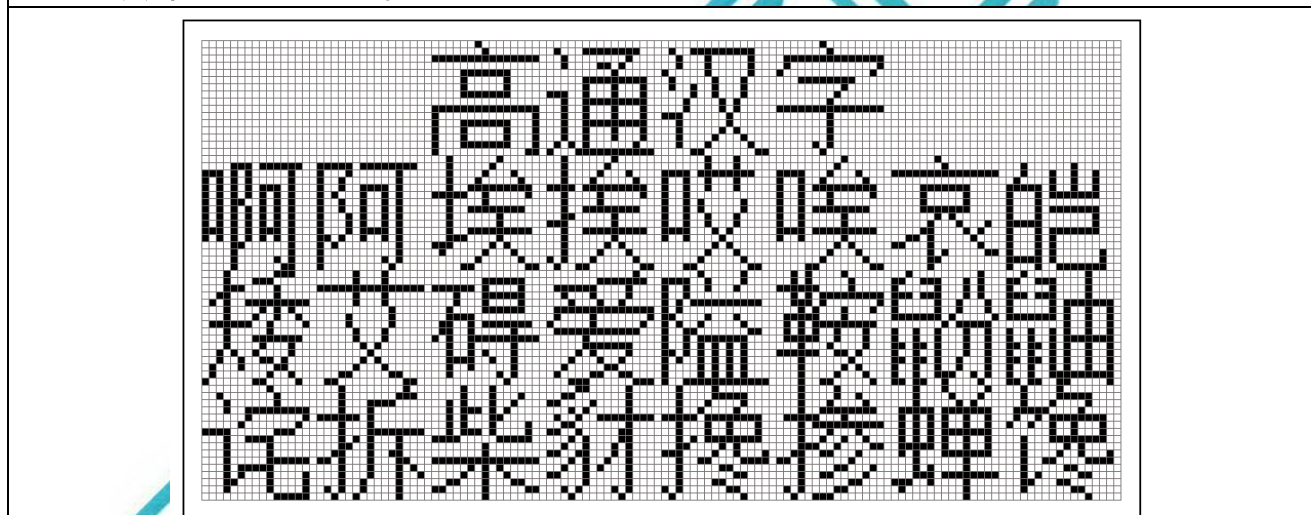
1.3 字型样张

1.3.1 汉字字符

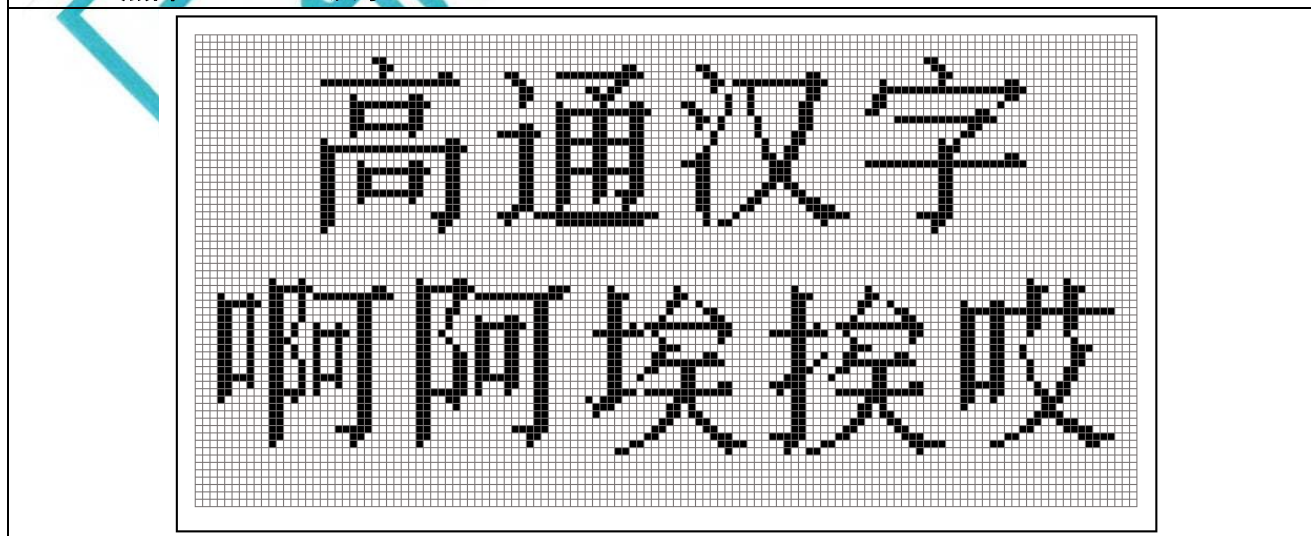
12x12 点阵 GB2312 汉字



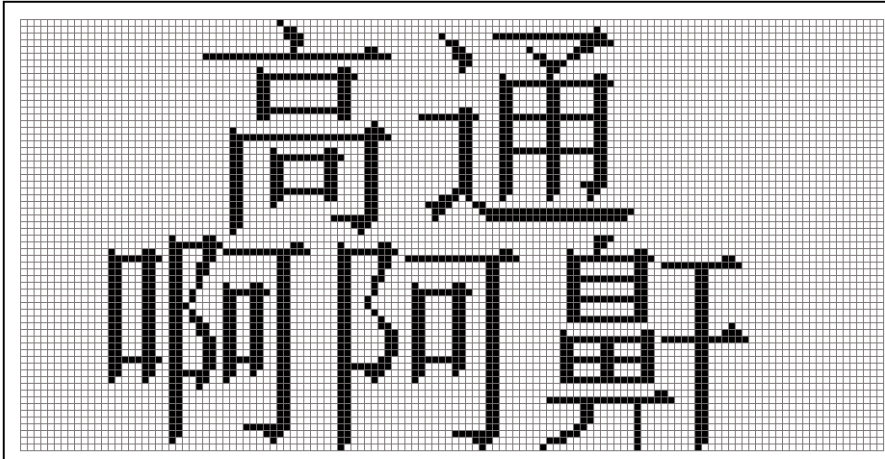
16x16 点阵 GB2312 汉字



24x24 点阵 GB2312 汉字



32x32 点阵 GB2312 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	{		}	~	

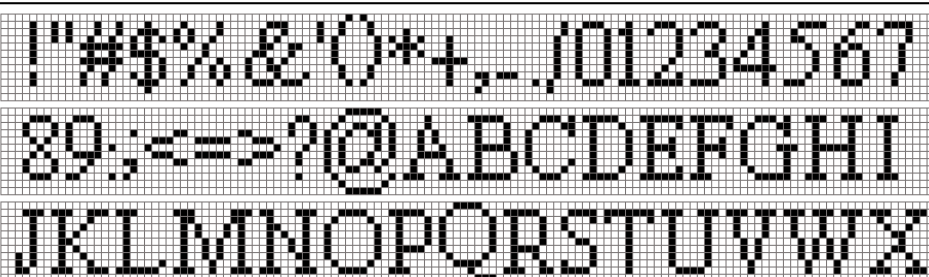
12x24 点阵 ASCII 标准字符

Low Shift High Shift	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
3		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
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	{		}	~	

16x32 点阵 ASCII 标准字符

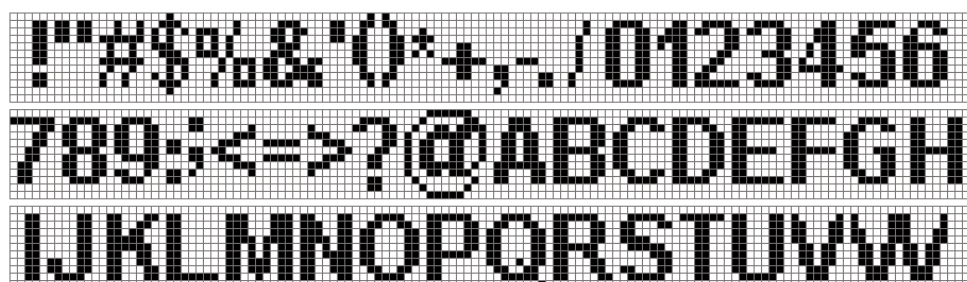
Low Shift High Shift	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
3		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	

12 点阵不等宽 ASCII 白正(Times new Roman)



!"#\$%&'()*+,-./01234567
89:;<=>?@ABCDEFGHI
JKLMNOPQRSTUVWXYZ

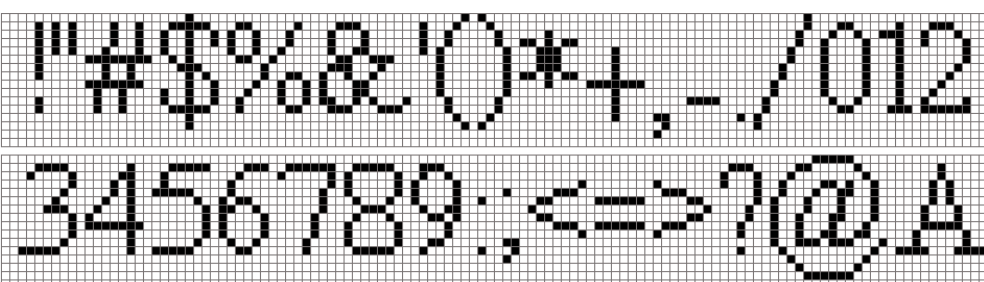
12 点阵不等宽 ASCII 方头 (Arial)



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

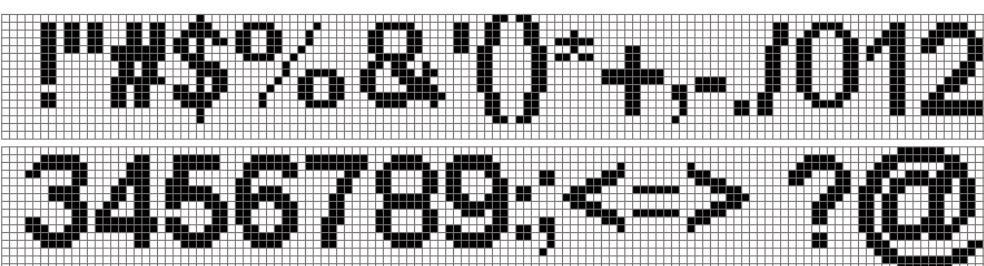


16 点阵不等宽 ASCII 白正(Times new Roman)



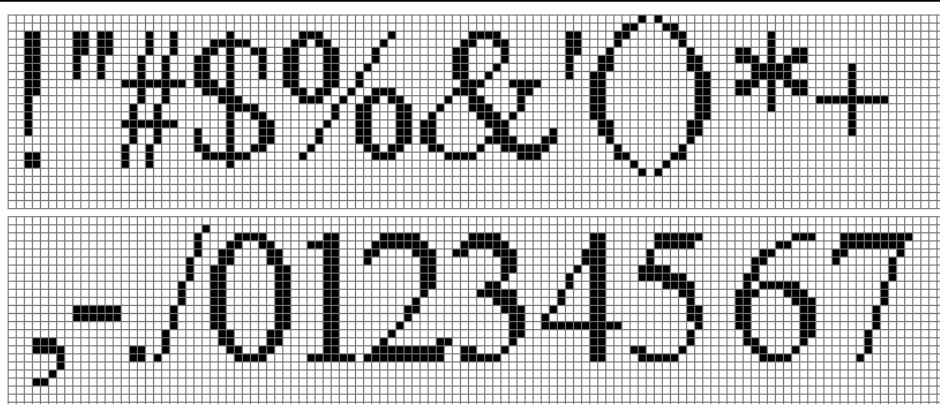
!"#\$%&'()*+,-./012
3456789:;<=>?@A

16 点阵不等宽 ASCII 方头 (Arial)



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

24 点阵不等宽 ASCII 白正(Times new Roman)



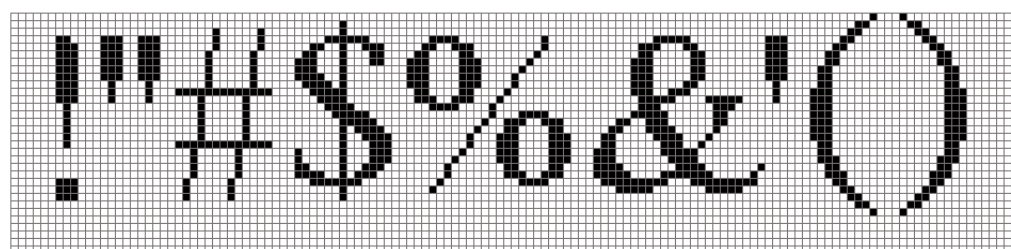
!"#\$%&'()*+
,-./01234567

24 点阵不等宽 ASCII 方头 (Arial)



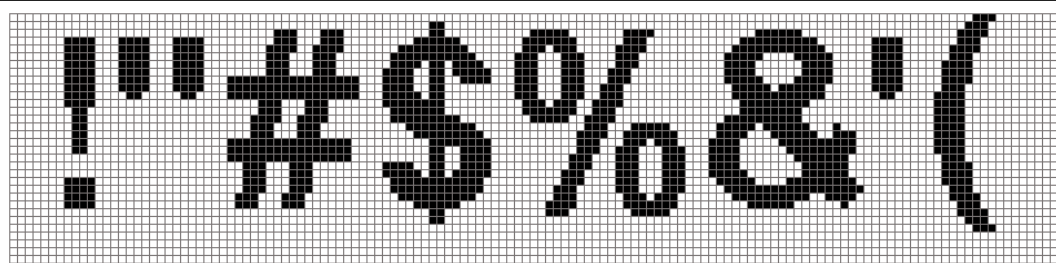
!"#\$%&'()*+
,-./01234567

32 点阵不等宽 ASCII 白正(Time new Roman)



!"#\$%&'()*

32 点阵不等宽 ASCII 方头 (Arial)



!"#\$%&'()

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 ∞
FAST_READ	Read Data Bytes at Higher Speed	0000 1011	0B h	1	1 to ∞

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”)。

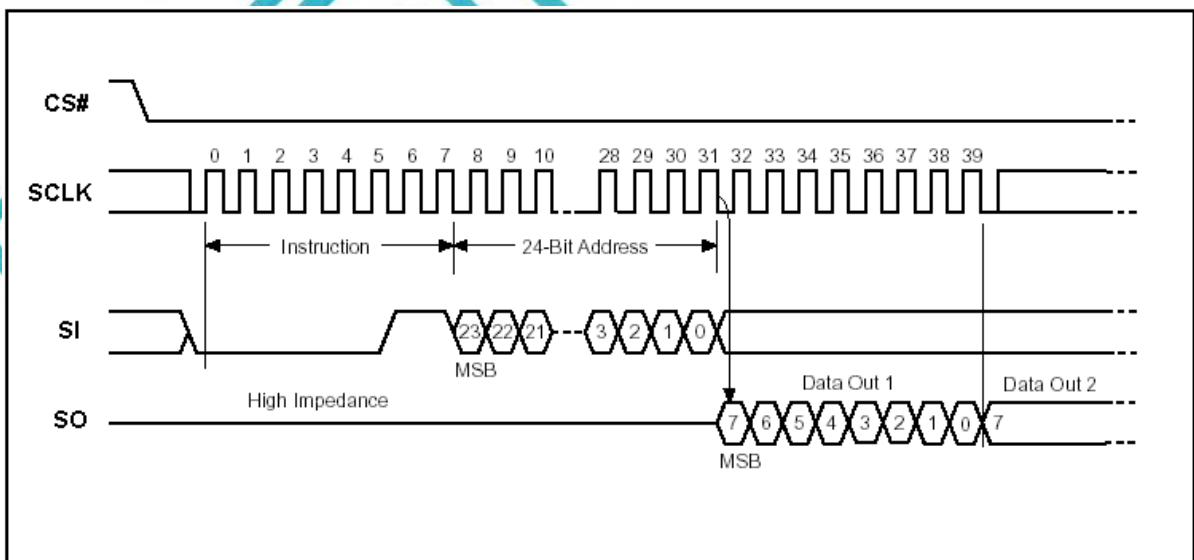
2.2 Read Data Bytes（一般读取）

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



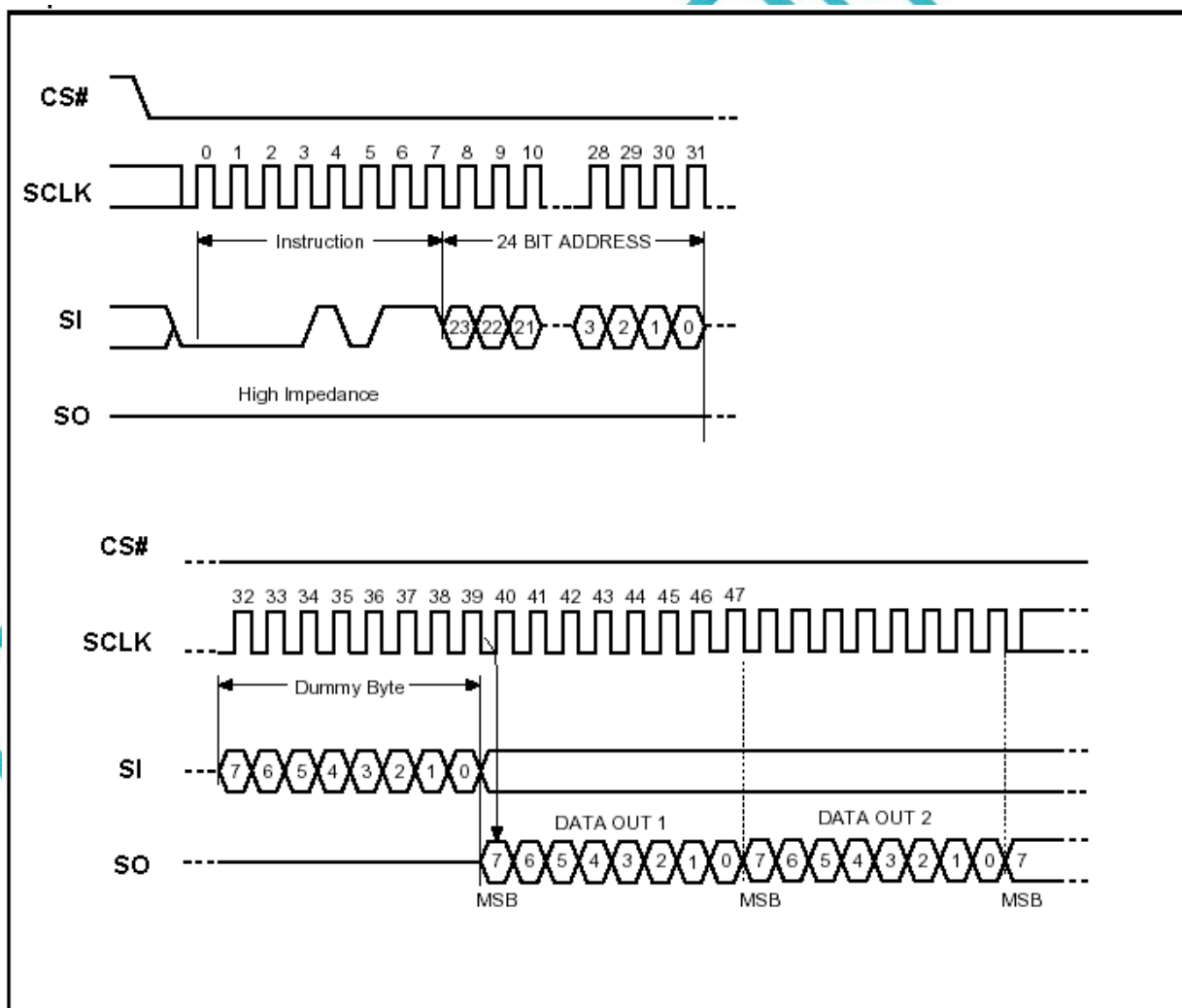
2.3 Read Data Bytes at Higher Speed（快速读取点阵数据）

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+846	0x00000	4.1.1
2	15x16 点阵 GB2312 字库	GB2312	6763+846	0x2C9D0	4.1.2
3	24x24 点阵 GB2312 字库	GB2312	6763+846	0x68190	4.1.3
4	32x32 点阵 GB2312 字库	GB2312	6763+846	0xEDF00	4.1.4
5	6x12 点阵国标扩展字符	GB2312	126	0x1DBE0C	4.1.5
6	6x12 点阵 ASCII 标准字符	ASCII	96	0x1DBE00	4.2.3
7	12 点阵不等宽 ASCII 方头 (Arial) 字符	ASCII	96	0x1DC400	4.2.7
8	12 点阵不等宽 ASCII 白正 (Time) 字符	ASCII	96	0x1DCDC0	4.2.8
9	8x16 点阵国标扩展字符	GB2312	126	0x1DD790	4.1.6
10	8x16 点阵 ASCII 标准字符	ASCII	96	0x1DD780	4.2.4
11	5x7 点阵 ASCII 标准字符	ASCII	96	0x1DDF80	4.2.1
12	7x8 点阵 ASCII 标准字符	ASCII	96	0x1DE280	4.2.2
13	16 点阵不等宽 ASCII 方头 (Arial) 字符	ASCII	96	0x1DE580	4.2.9
14	16 点阵不等宽 ASCII 白正 (Time) 字符	ASCII	96	0x1DF240	4.2.10
15	12x24 点阵国标扩展字符	GB2312	126	0x1DFF30	4.1.8
16	12x24 点阵 ASCII 字符	ASCII	96	0x1DFF00	4.2.5
17	24 点阵不等宽 ASCII 方头 (Arial) 字符	ASCII	96	0x1E22D0	4.2.11
18	24 点阵不等宽 ASCII 白正 (Time) 字符	ASCII	96	0x1E3E90	4.2.12
19	16x32 点阵国标扩展字符	GB2312	126	0x1E5A90	4.1.9
20	16x32 点阵 ASCII 字符	ASCII	96	0x1E5A50	4.2.6
21	32 点阵不等宽 ASCII 方头 (Arial) 字符	ASCII	96	0x1E99D0	4.2.13
22	32 点阵不等宽 ASCII 白正 (Time) 字符	ASCII	96	0x1ECA90	4.2.14
23	保留区			0x1EFB50	
24	8x16 点阵 GB2312 特殊字符	GB2312	64	0x1F2880	4.1.7
25	保留区			0x1F7CC8	

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 <= 0xA9 && LSB >=0xA1)

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

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

Address = ((MSB - 0xB0) * 94 + (LSB - 0xA1)+ 846)*24+ BaseAdd;

4.1.2 15x16 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=0x2C9D0;

if(MSB >=0xA1 && MSB <= 0xA9 && 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;

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=0x68190;

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

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

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

Address = ((MSB - 0xB0) * 94 + (LSB - 0xA1)+ 846)*72+ BaseAdd;

4.1.4 32x32 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=0XEDF00;

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

Address = (MSB - 0xA1) * 94 + (LSB - 0xA1)*128+ BaseAdd;

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

Address = ((MSB - 0xB0) * 94 + (LSB - 0xA1)+ 846)*128+ BaseAdd;

4.1.5 6x12 点阵国标扩展字符

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=0x1DBE0C

if (FontCode>= 0xAAA1 && FontCode<=0xAAFE)

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

Else if(FontCode>= 0xABA1 && FontCode<=0xABC0)

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

4.1.6 8x16 点阵国标扩展字符

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=0x1DD790

if (FontCode>= 0xAAA1 && FontCode<=0xAAFE)

```
ByteAddress = (FontCode-0xAAA1) * 16+BaseAdd;
Else if(FontCode>= 0xABA1 && FontCode<=0xABC0 )
    ByteAddress = (FontCode-0xABA1 + 95) * 16+BaseAdd;
```

4.1.7 8x16 点阵 GB2312 特殊字符

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=0x1F2880

if (FontCode >= 0xACA1 && FontCode <=0xACDF)

ByteAddress = (FontCode-0xACA1) * 16+BaseAdd;

4.1.8 12x24 点阵国标扩展字符

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=0x1DFF30

if (FontCode>= 0xAAA1 && FontCode<=0xAAFE)

ByteAddress = (FontCode-0xAAA1) * 48+BaseAdd;

Else if(FontCode>= 0xABA1 && FontCode<=0xABC0)

ByteAddress = (FontCode-0xABA1 + 95) * 48+BaseAdd;

4.1.9 16x32 点阵国标扩展字符

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=0x1E5A90

if (FontCode>= 0xAAA1 && FontCode<=0xAAFE)

ByteAddress = (FontCode-0xAAA1) * 64+BaseAdd;

Else if(FontCode>= 0xABA1 && FontCode<=0xABC0)

ByteAddress = (FontCode-0xABA1 + 95) * 64+BaseAdd;

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=0x1DDF80

if (ASCIICode >= 0x20 && ASCIICode <= 0x7E)

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=0x1DE280

if (ASCIICode >= 0x20 && ASCIICode <= 0x7E)

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=0x1DBE00

if (ASCIICode >= 0x20 && ASCIICode <= 0x7E)

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=0x1DD780

if (ASCIICode >= 0x20 && ASCIICode <= 0x7E)

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

4.2.5 12x24 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=0x1DFF00

if (ASCIICode >= 0x20 && ASCIICode <= 0x7E)

Address = (ASCIICode - 0x20) * 48 + BaseAdd;

4.2.6 16x32 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=0x1E5A50

if (ASCIICode >= 0x20 && ASCIICode <= 0x7E)

Address = (ASCIICode - 0x20) * 64 + BaseAdd;

4.2.7 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=0x1DC400

if (ASCIICode >= 0x20 && ASCIICode <= 0x7E)

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

4.2.8 12 点阵不等宽 ASCII 白正 (Times New Roman) 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=0x1DCDC0

if (ASCIICode >= 0x20 && ASCIICode <= 0x7E)

$Address = (ASCII Code - 0x20) * 26 + BaseAdd;$

4.2.9 16 点阵不等宽 ASCII 方头 (Arial) 字符

说明:

ASCII Code: 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=0x1DE580

if (ASCII Code \geq 0x20 && ASCII Code \leq 0x7E)

$Address = (ASCII Code - 0x20) * 34 + BaseAdd;$

4.2.10 16 点阵不等宽 ASCII 白正 (Times New Roman) 字符

说明:

ASCII Code: 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=0x1DF240

if (ASCII Code \geq 0x20 && ASCII Code \leq 0x7E)

$Address = (ASCII Code - 0x20) * 34 + BaseAdd;$

4.2.11 24 点阵不等宽 ASCII 方头 (Arial) 字符

说明:

ASCII Code: 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=0x1E22D0

if (ASCII Code \geq 0x20 && ASCII Code \leq 0x7E)

$Address = (ASCII Code - 0x20) * 74 + BaseAdd;$

4.2.12 24 点阵不等宽 ASCII 白正 (Times New Roman) 字符

说明:

ASCII Code: 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=0x1E3E90

if (ASCII Code \geq 0x20 && ASCII Code \leq 0x7E)

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

4.2.13 32 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=0x1E99D0

if (ASCIICode >= 0x20 && ASCIICode <= 0x7E)

Address = (ASCIICode - 0x20) * 130 + BaseAdd;

4.2.14 32 点阵不等宽 ASCII 白正 (Times New Roman) 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=0x1ECA90

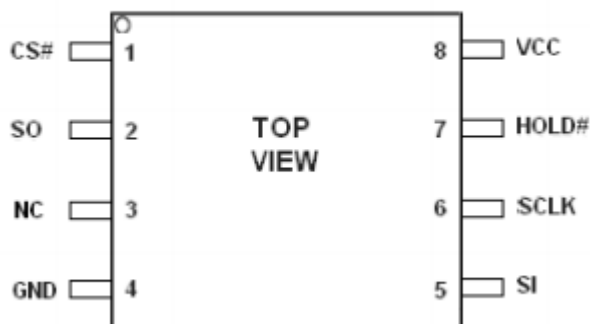
if (ASCIICode >= 0x20 && ASCIICode <= 0x7E)

Address = (ASCIICode - 0x20) * 130 + BaseAdd;

5 引脚描述与接口连接

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)

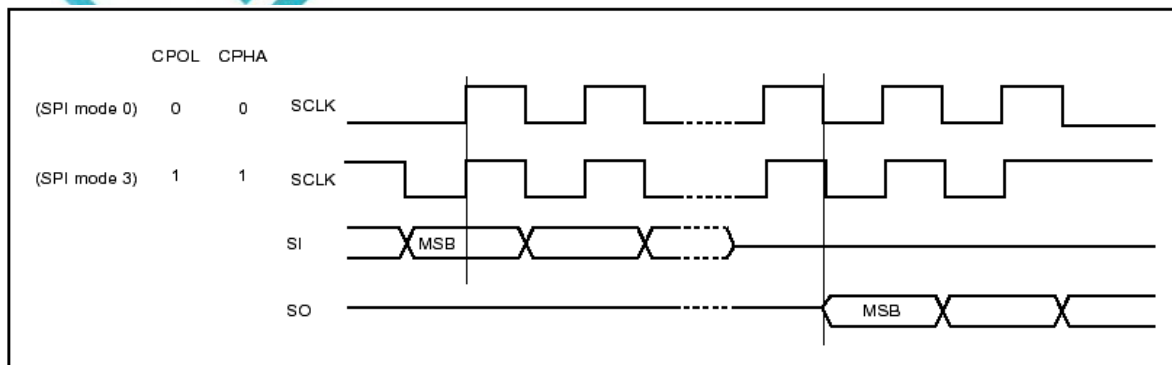
5.2 引脚描述

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。



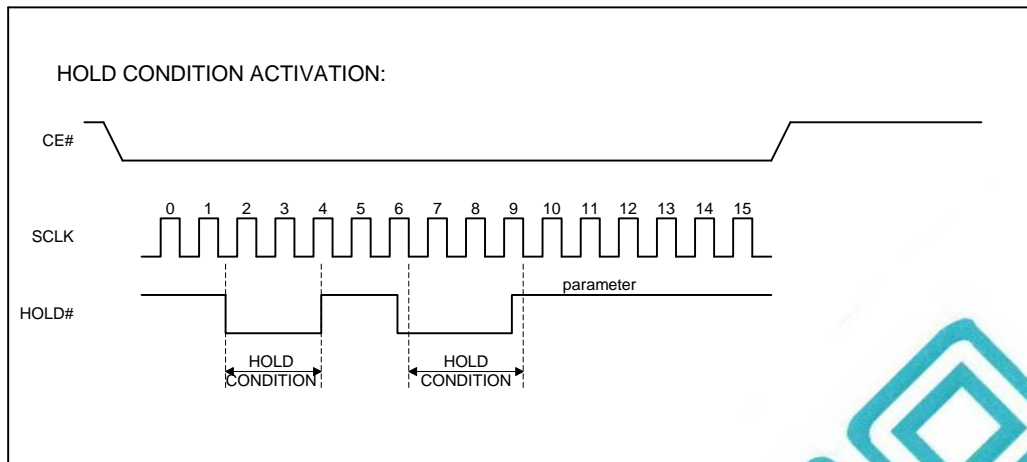
总线挂起输入 (HOLD#):

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。

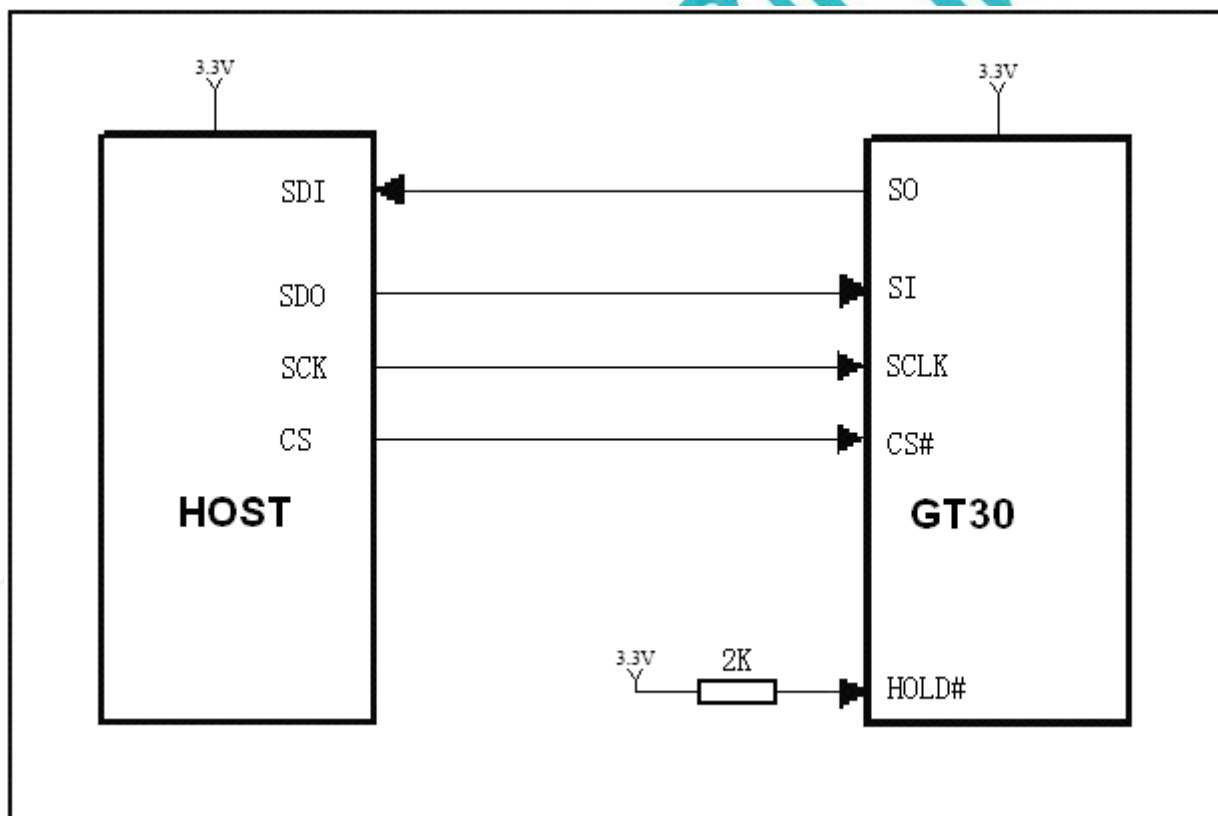
当HOLD#信号变为低并且串行时钟信号（SCLK）处于低电平时，进入总线挂起状态。

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 _{OP}	Operating Temperature	-40	85	°C	SPI mode
T _{STG}	Storage Temperature	-55	125	°C	
V _{CC}	Supply Voltage	-0.5	4.0	V	
V _{IN}	Input Voltage	-0.5	4.0	V	
GND	Power Ground	0	0	V	

6.2 DC 特性

Condition: T_{OP} = -40°C to 85°C, GND=0V in SPI mode;

Symbol	Parameter	Min.	Max.	Unit	Condition
I _{DD}	VCC Supply Current(active)		20	mA	
I _{SB}	VCC Standby Current		5	uA	
V _{IL}	Input LOW Voltage	-0.3	0.6	V	VCC=2.7V-3.6V
V _{IH}	Input HIGH Voltage	0.7VCC	VCC+0.3	V	
V _{OL}	Output LOW Voltage		0.4 (I _{OL} =1.6mA)	V	
V _{OH}	Output HIGH Voltage	VCC-0.2 (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,

6.3 AC 特性

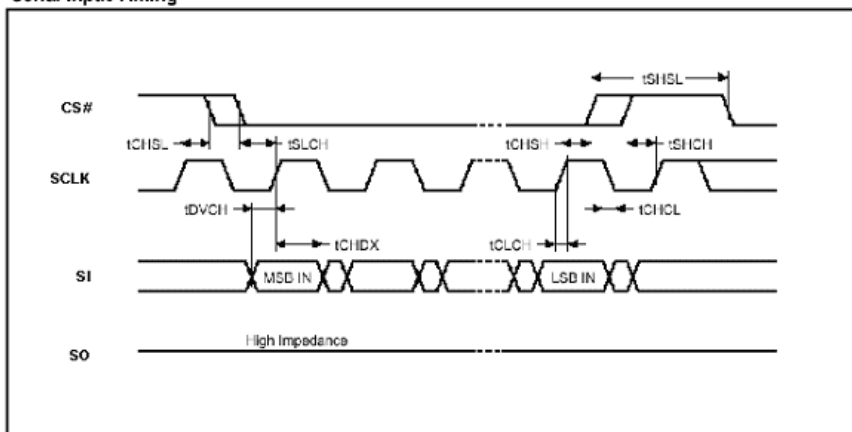
6.3.1 SPI 接口模式下 AC 特性

Condition: T_{OP} = -20°C to 85°C, VCC= 2.7V to 3.6V

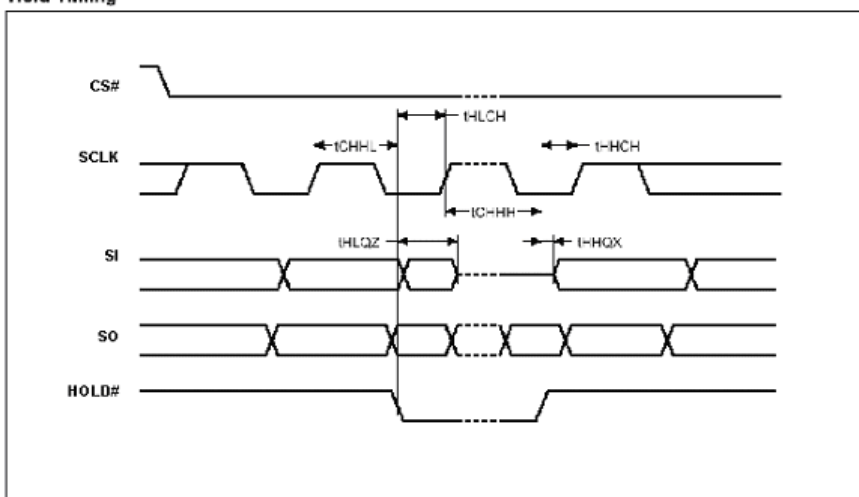
Symbol	Alt.	Parameter	Min.	Max.	Unit
F _C	F _C	Clock Frequency	D.C.	80	MHz
t _{CH}	t _{CLH}	Clock High Time	4		ns
t _{CL}	t _{CLL}	Clock Low Time	4		ns
t _{CLCH}		Clock Rise Time(peak to peak)	0.2		V/ns
t _{CHCL}		Clock Fall Time (peak to peak)	0.2		V/ns
t _{SLCH}	t _{css}	CS# Active Setup Time (relative 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	2		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		6	ns
t _{CLQV}	t _v	Clock Low to Output Valid		7	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		6	ns
t _{HLQZ}	t _{HZ}	HOLD# to Output High-Z		6	ns

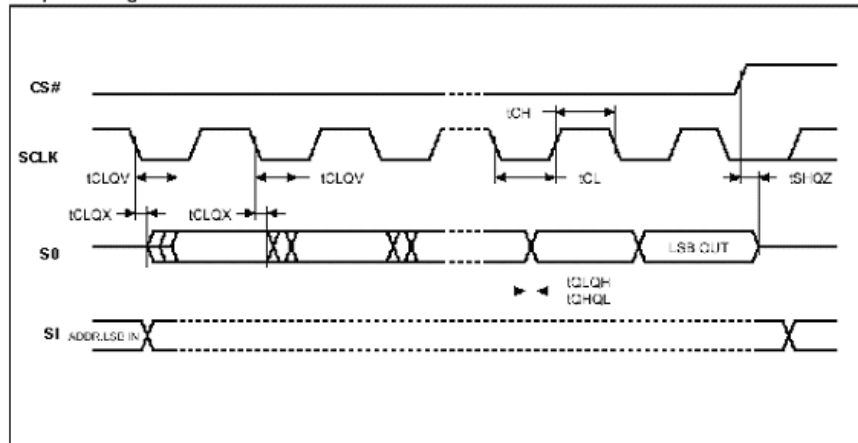
Serial Input Timing



Hold Timing

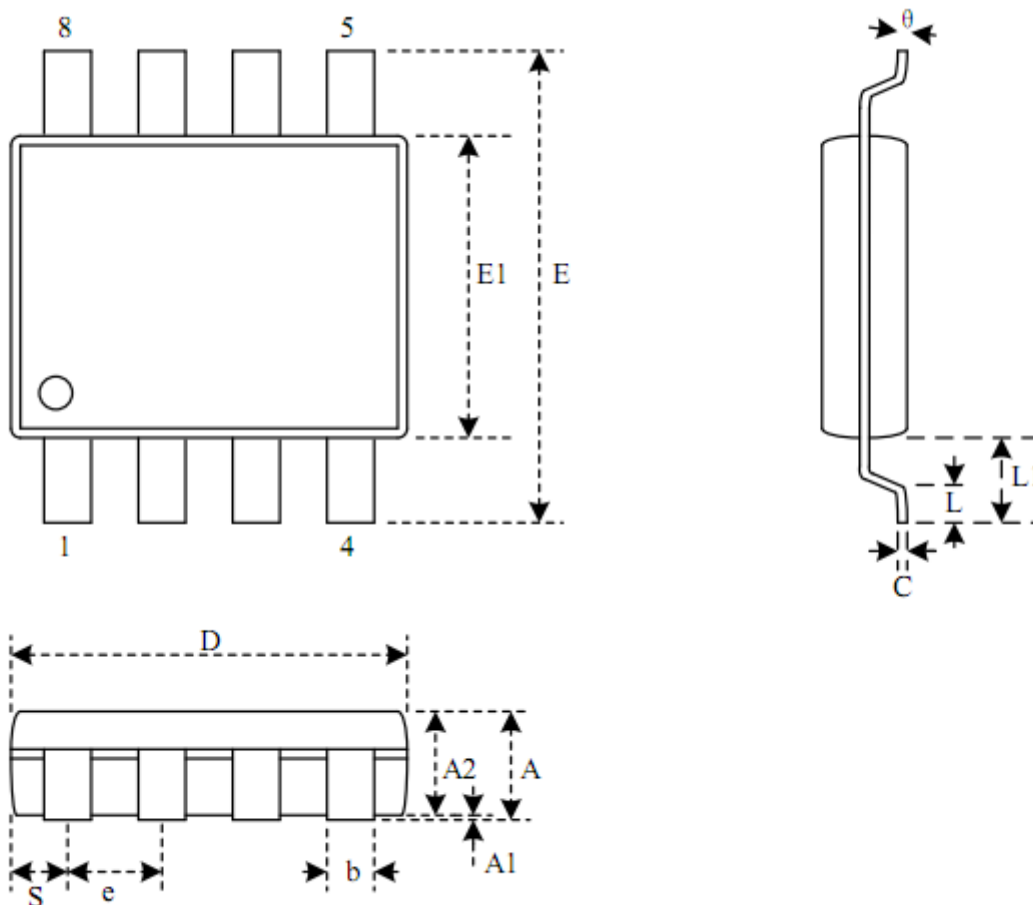


Output Timing



7 封装尺寸：SOP8-B

单位: mm



Dimensions

Symbol		A	A1	A2	b	C	D	E	E1	e	L	L1	S	θ
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 (846 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	一	~		...	‘	’
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	{		}	—	

GB2312 1 区

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	て	ど	な	ぬ	の	は						
D	ば	び	び	ふ	ぶ	へ	べ	ほ	ま	み						
E	む	め	も	や	ゆ	よ	ら	り	る	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	ス	セ	ソ	タ							
C	ダ	チ	ツ	テ	ト	ナ	ニ	ノ	ハ							
D	バ	ビ	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	ξ	η	θ	ι	κ	λ	μ	ν	ξ	ο
D	π	ρ	σ	τ	υ	φ	χ	ψ	ω	,	°	`	:	;	!	?
E	（	）	〔	〕	＜	＞	≪	≫	┐	┌	└	┘	≡	≡	≡	≡
F	～	ゝ	丨	丨	丨	丨										

GB2312 1 区

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	Э
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	э
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	ǐ	ì	ō	ó	ǒ
B	ò	ū	ú	ǔ	ù	ǖ	ǘ	ǚ	ǜ	Ǘ	Ǚ	Ǜ	ǝ	ǎ	ǎ	ǎ
C	g				ㄅ	ㄆ	ㄇ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ
D	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ
E	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ	ㄏ
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 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	{		}	~	

AB	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	ê	ɑ	ǎ	ǎ	ǎ	ǎ	ǎ
C	g															

8.3 8×16 点特殊字符（64 字符）

内码组成为 ACA1~ACDF 共计 64 个字符

AC	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		☺	☹	♥	♣	♣	♣	♣	♣	♣	♣	♂	♀	♪	♪	⚙
B	▶	◀	↕	!!	¶	§	■	⬆	↑	↓	→	←	└	↔	▲	▼
C	Ψ	,				⌈	⌊	⌋	⌌	⌍)))	◀	▶	ℙ
D	°	∞	∅	∈	∩	≡	≥	≤	≈	√	ⁿ	€	\$	∫	ℳ	÷