

高通<sup>®</sup>科技  
GENTOP<sup>®</sup>

# GT60L16M2K4 矢量汉字库芯片

— 产品规格书 —

V1.0\_A  
2016-07



## 版本修订记录

版本号	修改内容	日期	备注
V1.0I_A	芯片规格书定制	2016-7	

# 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 字形样张.....	6
<b>2 操作指令</b> .....	<b>9</b>
2.1 Instruction Parameter(指令参数).....	9
2.2 Read Data Bytes (一般读取) .....	9
2.3 Read Data Bytes at Higher Speed (快速读取点阵数据) .....	10
2.4 读芯片状态时序.....	11
2.5 读取芯片状态寄存器的命令说明.....	11
<b>3 高通矢量字库 LIB 文件 使用说明</b> .....	<b>12</b>
3.1 需构造的函数: .....	12
3.2 主要功能函数说明.....	12
3.3 字符缓冲区.....	13
3.4 软件开销.....	13
<b>4 引脚描述与电路连接</b> .....	<b>14</b>
4.1 引脚配置.....	14
4.2 引脚描述.....	14
4.3 SPI 接口与主机接口参考电路示意图.....	16
<b>5 电气特性</b> .....	<b>17</b>
5.1 绝对最大额定值.....	17
5.2 DC 特性.....	17
5.3 AC 特性.....	17
<b>6 封装尺寸</b> .....	<b>19</b>

# 1 概述

高通曲线字库模组GT60L16M2K4是以二次余弦曲线函数算法来描述汉字和字符的字型笔划轮廓。它的特点是字形压缩比高，可以任意缩放（16点到192点），变形效果（加粗，倾斜，反白，钩边，灰度等），具有字体平滑而不失真等优点，可产生多种高质量的汉字输出，适用于各种嵌入式电子产品，包括需要高分辨率的显示或打印设备。

## 1.1 芯片特点

- 数据总线：SPI 串行总线接口
- 点阵排列方式：横置横排
- 时钟频率：60MHz @3.3V
- 工作电压：2.7V~3.6V
- 电流：
  - 工作电流：12mA
  - 待机电流：10uA
- 工作温度：-40℃~85℃
- 封装：SOP8-B
- 字符集：
  - 宋体 GBK 汉字字符集
  - 黑体 GBK 汉字字符集
  - 仿宋 GBK 汉字字符集
  - 楷体 GBK 汉字字符集
  - 16 点阵宋体 GB18030 汉字字符集
  - ASCII 字符集
  - 全角汉字字符集
- 字号：16 点到 192 点阵
- 总线接口：SPI 串行总线
- 封装类型：SOP8-B

## 1.2 芯片内容

字符集	字库	字号	字符数	字体
ASCII 字符集	ASCII	8X16 点阵	96	粗体
	ASCII	12X24 点阵	96	标准
	ASCII	16X32 点阵	96	标准
	ASCII	16 点阵不等宽	96	粗体
	ASCII	16 点阵不等宽	96	标准
	ASCII	24 点阵不等宽	96	粗体
	ASCII	24 点阵不等宽	96	标准
	ASCII	32 点阵不等宽	96	粗体
	ASCII	32 点阵不等宽	96	标准
	ASCII	16~192 点阵	96	方头
	ASCII	16~192 点阵	96	白正
	ASCII	16~192 点阵	96	方斜
	ASCII	16~192 点阵	96	长黑
	ASCII	16~192 点阵	96	白斜
	ASCII	16~192 点阵	96	圆头
	ASCII	16~192 点阵	96	打字
	ASCII	16~192 点阵	96	美术
	ASCII	16~192 点阵	96	歌德
	ASCII	16~192 点阵	96	手写
	ASCII	16~192 点阵	96	黑正
数字字符集	数字	8X16 点阵不等宽	14	方块
	数字	16X16 点阵不等宽	14	白正
	数字	16X16 点阵不等宽	14	方头
	数字	8X16 点阵不等宽	14	方块
	数字	24X24 点阵不等宽	14	白正
	数字	24X24 点阵不等宽	14	方头
	数字	16X24 点阵不等宽	14	方块
	数字	24X24 点阵不等宽	14	时钟
	数字	24X32 点阵不等宽	14	白正
	数字	24X32 点阵不等宽	14	方头
	数字	16X32 点阵不等宽	14	方块
	数字	24X32 点阵不等宽	14	时钟
	数字	40X48 点阵不等宽	14	白正
	数字	40X48 点阵不等宽	14	方头
	数字	24X48 点阵不等宽	14	方块
	数字	40X48 点阵不等宽	14	时钟
	数字	40X64 点阵不等宽	14	白正
	数字	40X64 点阵不等宽	14	方头
	数字	32X64 点阵不等宽	14	方块
	数字	48X64 点阵不等宽	14	时钟

可穿戴图标	图标	32X32 点阵	15	
全角汉字字符集	汉字字符	16~192 点阵	984	宋体
转码表	BIG5 to GBK			
	UNICODE to GB18030			
GB 汉字字符集	GBK 汉字	16~192 点阵	21003	宋体
	GBK 汉字	16~192 点阵	21003	黑体
	GBK 汉字	16~192 点阵	21003	仿宋
	GBK 汉字	16~192 点阵	21003	楷体
	GB18030 汉字	16 点阵	27484	宋体

### 1.3 字形样张

16 点	32 点	48 点
啊阿埃挨哎唉哀皑癌蔼矮 艾碍爱隘鞍氨安俺按暗岸 胺案肮昂盎凹敖翱袄傲奥 懊澳芭捌扒叭吧笆八疤巴 拔跋靶把耙坝霸罢爸白柏 百摆佰败拜裨斑班搬扳般 颁板版扮拌伴瓣半办绊邦 帮梆榜膀绑棒磅蚌镑傍谤 苞胞包褒剥薄雹保堡饱宝	啊阿埃挨棋奇 高通汉字库芯 片开启鞍氨安 俺按暗岸胺案 瓶评屏坡泼普	啊阿埃挨 高通芯片 库质忠舟
64 点	80 点	96 点
啊阿高 通汉字 库芯片	啊阿 埃挨	啊阿 埃挨
112 点	128 点	144 点

啊

160 点

啊

176 点

啊

192 点

啊

192 点宋体

啊

192 点楷体

啊

192 点仿宋

宋

楷

仿

48 点阵不等宽 (Arial)

Low High	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	.	,	:	'		

48 点阵不等宽 (Times new Roman)

Low High	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	.	,	:	'		

48 点阵不等宽 (时钟体)

Low High	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	.	,	:	'		

48 点阵不等宽 (方块体)

Low High	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	.	,	:	'		



## 2 操作指令

### 2.1 Instruction Parameter(指令参数)

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 $\infty$
WREN	Write Enable	0000 0110	06 h	—	—
WRDI	Write Disable	0000 0100	04 h	—	—
PP	Page Program	0000 0010	02 h	—	1 to 256
SE	Sector Erase	0010 0000	20 h	—	—
BE	Block Erase(64K)	1101 1000	D8 h	—	—
CE	Chip Erase	0110 0000/ 1100 0111	60 H/ C7 H	—	—

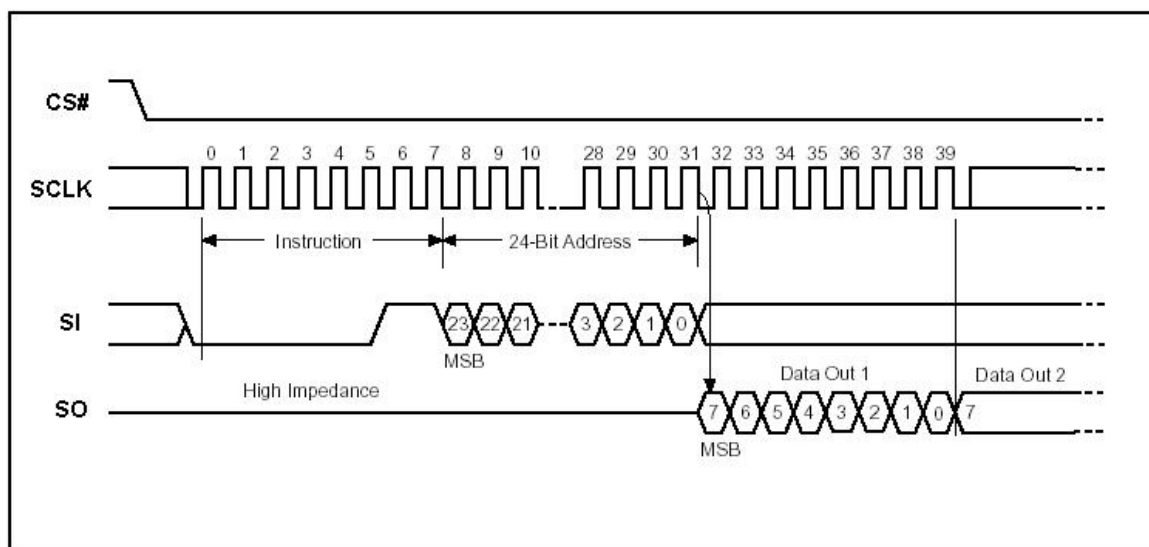
### 2.2 Read Data Bytes (一般读取)

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:



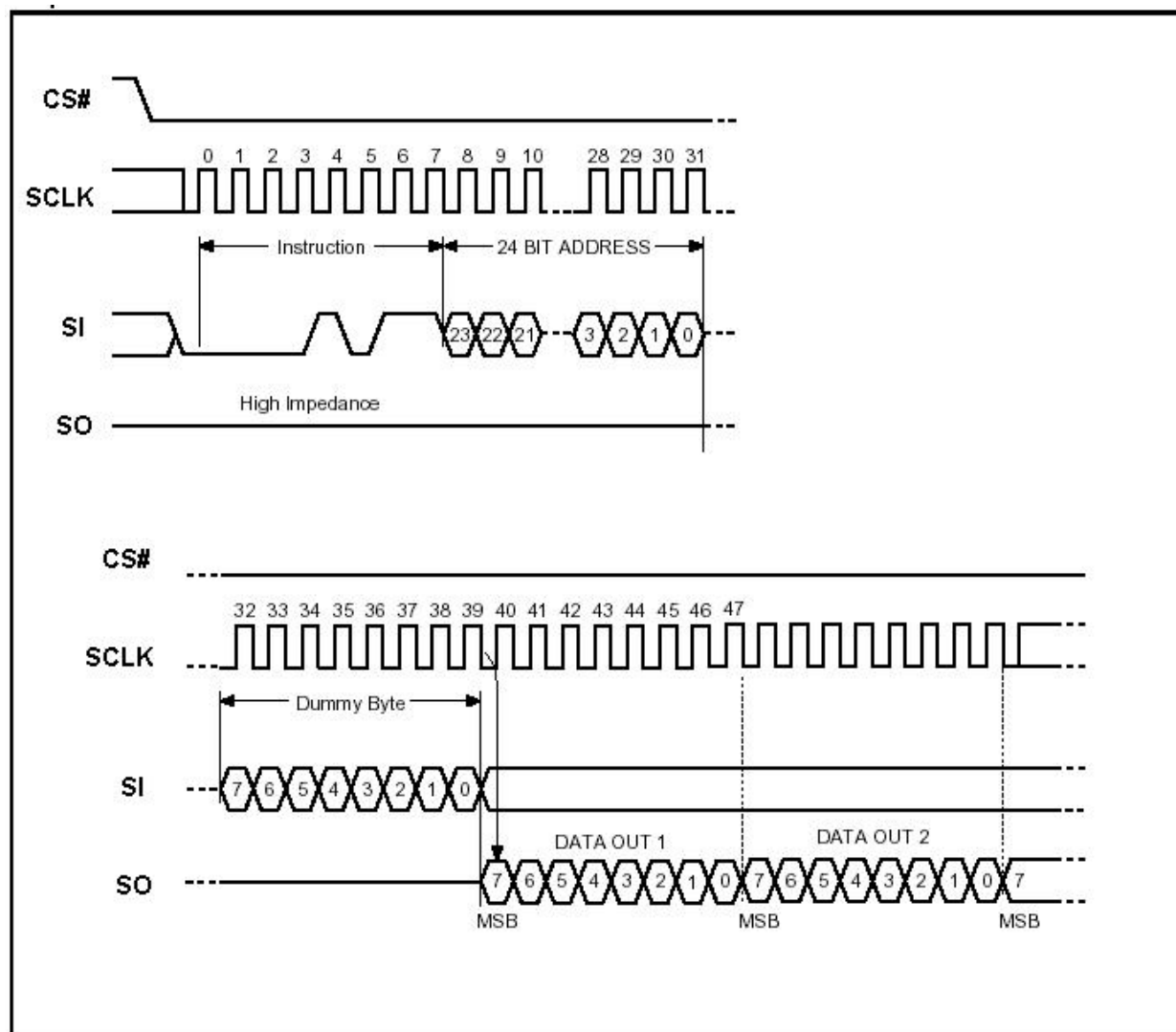
## 2.3 Read Data Bytes at Higher Speed (快速读取点阵数据)

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):

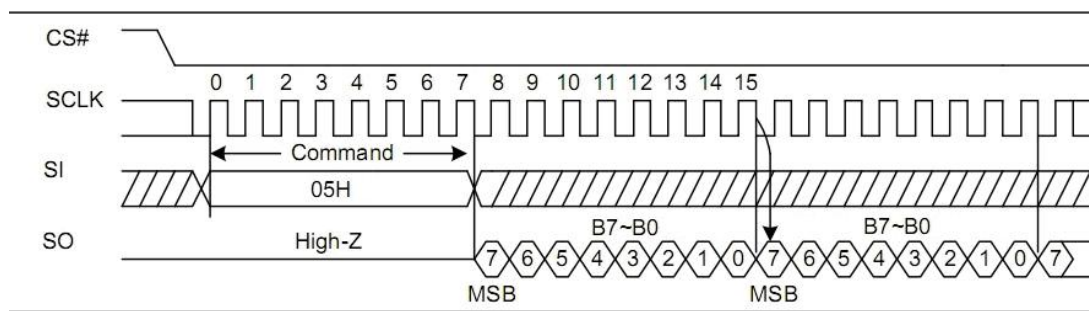
- 首先把片选信号 (CS#) 变为低, 紧跟着的是 1 个字节的命令字 (0B h) 和 3 个字节的地址以及一个字节 Dummy Byte 通过串行数据输入引脚 (SI) 移位输入, 每一位在串行时钟 (SCLK) 上升沿被锁存。
- 然后该地址的字节数据通过串行数据输出引脚 (SO) 移位输出, 每一位在串行时钟 (SCLK) 下降沿被移出。
- 如果片选信号 (CS#) 继续保持为底, 则下一个地址的字节数据继续通过串行数据输出引脚 (SO) 移位输出。例: 读取一个 15x16 点阵汉字需要 32Byte, 则连续 32 个字节读取后结束一个汉字的点阵数据读取操作。

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:



## 2.4 读芯片状态时序



## 2.5 读取芯片状态寄存器的命令说明

发送命令 05H，然后读取芯片状态寄存器的 B7-B0 位。判断 WIP 位的状态来判断芯片是否在忙状态。

## 3 高通矢量字库 LIB 文件 使用说明

### 3.1 需构造的函数:

- 1: unsigned char r\_dat(unsigned long int address);  
功能: 从字库的指定地址 address 读取一个字符并返回该字符
- 2: void r\_dat\_bat(unsigned long int address,unsigned char byte\_long,unsigned char \*p\_arr);  
功能: 从字库的指定地址 address 读取 byte\_long 个字符到 p\_arr 的缓冲区里。

### 3.2 主要功能函数说明

1,void get\_font(unsigned char \*bits,unsigned char sty,unsigned char Ms,unsigned char lsb,unsigned char len,unsigned char wth,unsigned char thick);

功能: 得到矢量字符的数据

参数说明: ①pBits 矢量字库缓冲区, ②sty, 字体风格, 只对 24 点及 24 点以上点阵字符有效。③msb, 字符编码的高位, ASCII 编码为 00, ④lsb, 字符编码的低位, ⑤len: 该字符的高度, ⑥: thick: 为字符曲线宽度, 只对 32 点以上字符有效。

- 示例: ①get\_font(pBits,0x03,0xb0,0xa1,32,32,32); 读取 32X32 宋体矢量点阵的 “啊”  
 ②get\_font(pBits,0x05,0xb0,0xa1,32,32,32); 读取 32X32 黑体矢量点阵的 “啊”  
 ③get\_font(pBits,0x06,0xb0,0xa1,32,32,32); 读取 32X32 仿宋矢量点点阵的 “啊”  
 ④get\_font(pBits,0x07,0xb0,0xa1,32,32,32); 读取 32X32 楷体矢量点点阵的 “啊”  
 ⑤sty=0x08 显示方斜矢量 ASCII 字符 sty=0x09 显示白斜矢量 ASCII 字符  
 ⑥sty=0x0A 显示长黑矢量 ASCII 字符 sty=0x0B 显示圆头矢量 ASCII 字符  
 ⑦sty=0x0C 显示打字矢量 ASCII 字符 sty=0x0D 显示美术矢量 ASCII 字符  
 ⑧sty=0x0E 显示歌德矢量 ASCII 字符 sty=0x10 显示黑正矢量 ASCII 字符  
 ⑨sty=0x11 显示白正矢量 ASCII 字符 sty=0x13 显示手写矢量 ASCII 字符  
 ⑩sty=0x12 显示方头矢量 ASCII 字符  
 ⑪sty=0x14 显示全角矢量 ASCII 字符  
 ⑫sty=0x02 显示 8X16 点阵 ASCII 字符  
 ⑬sty=0x01 显示 12X24 点阵 ASCII 字符  
 ⑭sty=0x04 显示 16X32 点阵 ASCII 字符

示例: get\_font(pBits,0x08,0x00,0xa1,32,32,32); 读取 32X32 白斜点阵的 “A” 字符

2,void hzbmp16X16( unsigned int code, unsigned int codeex, unsigned char size,unsigned char \*buf);

功能: 得到 16 点阵汉字字符的数据

参数说明: ①code 表示汉字内码低位 ②codeex 表示汉字内码高位 ③size 表示字体大小  
④buf 表示点阵缓冲区

示例: hzbmp16X16 (0xb0a1,0,16, buf); 读取 16X16 宋体点阵的 “啊” 字

3, unsigned char ASCII\_GetData\_AT(unsigned char ASCIIcode,unsigned long BaseAdd,unsigned char\*DZ\_Data);

功能: 得到不等宽的 ASCII 码字符

参数说明: ①ASCIIcode 表示 ASCII 码 ②BaseAdd 说明该套字库在芯片中的起始地址  
③DZ\_Data 是保存读出的点阵数据的数组

示例:

- ①ASCII\_GetData\_AT(0x41, ASCII\_16X16\_T, DZ\_Data)读取 16 点阵不等宽的 ASCII 字符粗体
- ②ASCII\_GetData\_AT(0x41, ASCII\_16X16\_A, DZ\_Data)读取 16 点阵不等宽的 ASCII 字符标准
- ③ASCII\_GetData\_AT(0x41, ASCII\_24X24\_T, DZ\_Data)读取 24 点阵不等宽的 ASCII 字符粗体
- ④ASCII\_GetData\_AT(0x41, ASCII\_16X16\_A, DZ\_Data)读取 24 点阵不等宽的 ASCII 字符标准
- ⑤ASCII\_GetData\_AT(0x41, ASCII\_32X32\_T, DZ\_Data)读取 32 点阵不等宽的 ASCII 字符粗体
- ⑥ASCII\_GetData\_AT(0x41, ASCII\_32X32\_T, DZ\_Data)读取 32 点阵不等宽的 ASCII 字符标准

**4, unsigned int U2GB18030(unsigned int UN\_CODE);**//unicode 转 GB18030 的函数

功能: 输入 unicode 编码得到 GB18030 编码

参数说明: UN\_CODE 表示汉字对应的 unicode 编码

unsigned int GB\_CODE;

示例: GB\_CODE= U2GB18030(0x554a); 将“啊”字的 UNICODE 编码转换成 GB 国标编码, 存在 GB\_CODE 中.如例 GB\_CODE=0xb0a1

**5, unsigned int BIG52GBK( unsigned char h, unsigned char l);**

功能: 输入 BIG5 编码得到汉字 GBK 编码

参数说明: h 表示汉字内码的高八位 l 表示汉字内码的低八位

unsigned int GBK\_CODE;

示例: GBK\_CODE=BIG52GBK(0xb0,0xda); //将“啊”字的 BIG5 编码转换成 GB 国标编码, 存在 GBK\_CODE 中.如例 GBK\_CODE=0xb0a1

**6, void Gray\_Process(unsigned char \*OutPutData ,int width,int High,unsigned char Grade);**

功能: 对点阵字库进行灰度处理

参数说明: ①OutPutData 点阵数据输出缓冲 ②width 矢量字库的宽度 ③High 矢量字库的高度 ④Grade 表示灰度级别

**7, void get\_cr\_zf(unsigned char Sequence,unsigned long BaseAdd,unsigned char\*DZ\_Data);**

功能: 得到可穿戴设备符号

参数说明: ①Sequence 表示符号的顺序(顺序参见原始图片) ②BaseAdd 说明该套字库在芯片中的起始地址 ③DZ\_Data 是保存读出的点阵数据的数组

示例: get\_cr\_zf(0,cr\_zf\_32X32\_TB, DZ\_Data);读取 32X32 可穿戴设备图标

### 3.3 字符缓冲区

unsigned char pBits[192\*192/8];

该字符缓冲区为内建, 不可删除, 更改。

### 3.4 软件开销

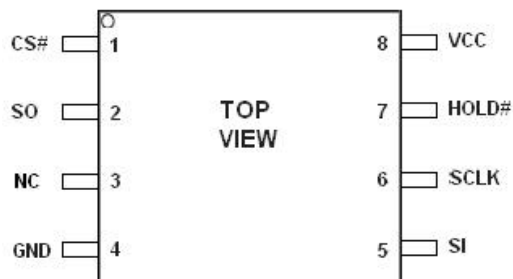
ROM 36K

RAM 28K

## 4 引脚描述与电路连接

### 4.1 引脚配置

#### SOP8-B



### 4.2 引脚描述

#### SOP8-B

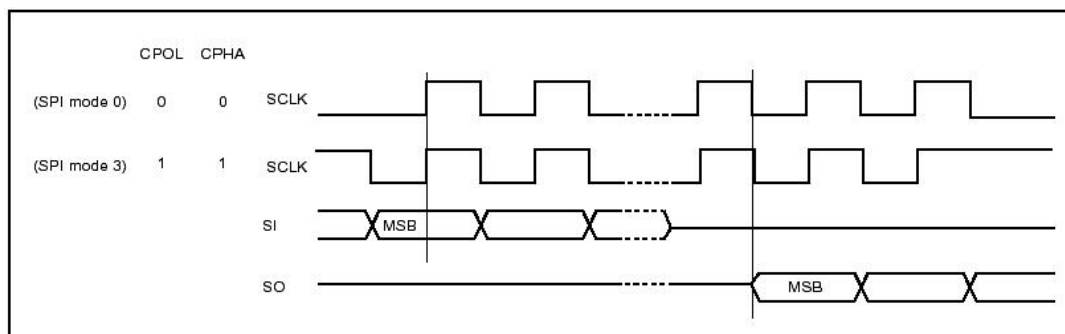
NO.	名称	I/O	描述
1	CS#	I	片选输入 (Chip enable input)
2	SO	O	串行数据输出 (Serial data output)
3	NC		悬空
4	GND		地(Ground)
5	SI	I	串行数据输入 (Serial data input)
6	SCLK	I	串行时钟输入 (Serial clock input)
7	HOLD#	I	总线挂起 (Hold, to pause the device without)
8	VCC		电源(+ 3.3V Power Supply)

**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。

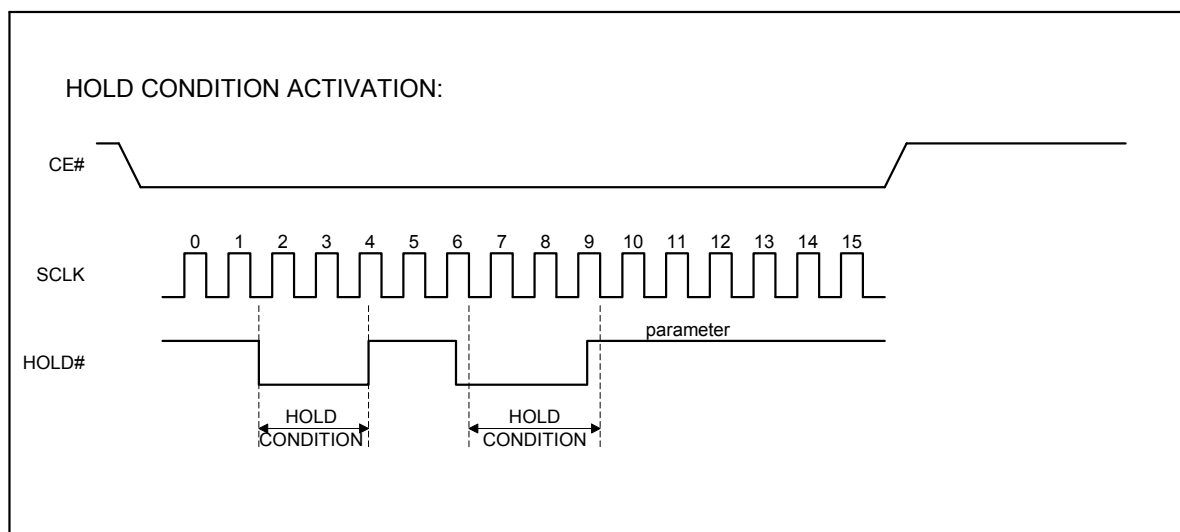


**总线挂起输入 (HOLD#):**

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。

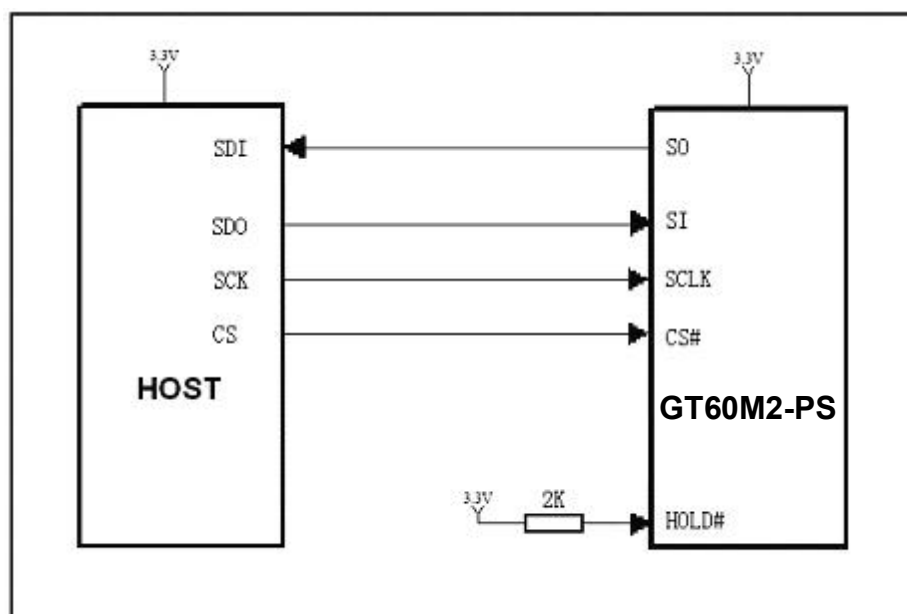
当HOLD#信号变为低并且串行时钟信号 (SCLK) 处于低电平时，进入总线挂起状态。

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



### 4.3 SPI 接口与主机接口参考电路示意图

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）。



SPI 接口与主机接口参考电路示意图



## 5 电气特性

### 5.1 绝对最大额定值

Symbol	Parameter	Min.	Max.	Unit	Condition
T <sub>OP</sub>	Operating Temperature	-40	85	°C	
T <sub>STG</sub>	Storage Temperature	-65	150	°C	
V <sub>CC</sub>	Supply Voltage	-0.3	3.6	V	
V <sub>IN</sub>	Input Voltage	-0.3	V <sub>CC</sub> +0.3	V	
GND	Power Ground	-0.3	0.3	V	

### 5.2 DC 特性

Condition: T<sub>OP</sub> = -20°C to 70°C, GND=0V

Symbol	Parameter	Min.	Max.	Unit	Condition
I <sub>DD</sub>	VCC Supply Current(active)		12	mA	
I <sub>SB</sub>	VCC Standby Current		10	µA	
V <sub>IL</sub>	Input LOW Voltage	-0.3	0.3V <sub>CC</sub>	V	V <sub>CC</sub> =2.2~3.6V
V <sub>IH</sub>	Input HIGH Voltage	0.7V <sub>CC</sub>	V <sub>CC</sub> +0.4	V	
V <sub>OL</sub>	Output LOW Voltage		0.4 (I <sub>OL</sub> =1.6mA)	V	
V <sub>OH</sub>	Output HIGH Voltage	0.8V <sub>CC</sub> (I <sub>OH</sub> =-100µA)		V	
I <sub>LI</sub>	Input Leakage Current	0	2	µA	
I <sub>LO</sub>	Output Leakage Current	0	2	µA	

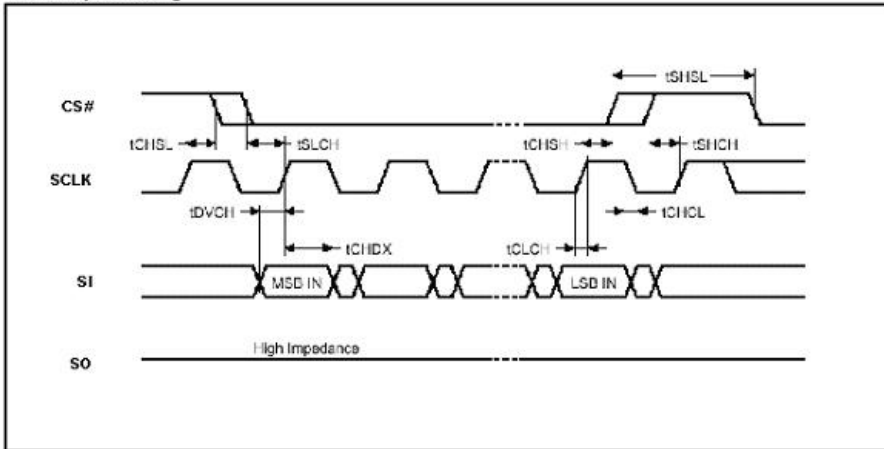
Note: I<sub>IL</sub>: Input LOW Current, I<sub>IH</sub>: Input HIGH Current,  
I<sub>OL</sub>: Output LOW Current, I<sub>OH</sub>: Output HIGH Current,

### 5.3 AC 特性

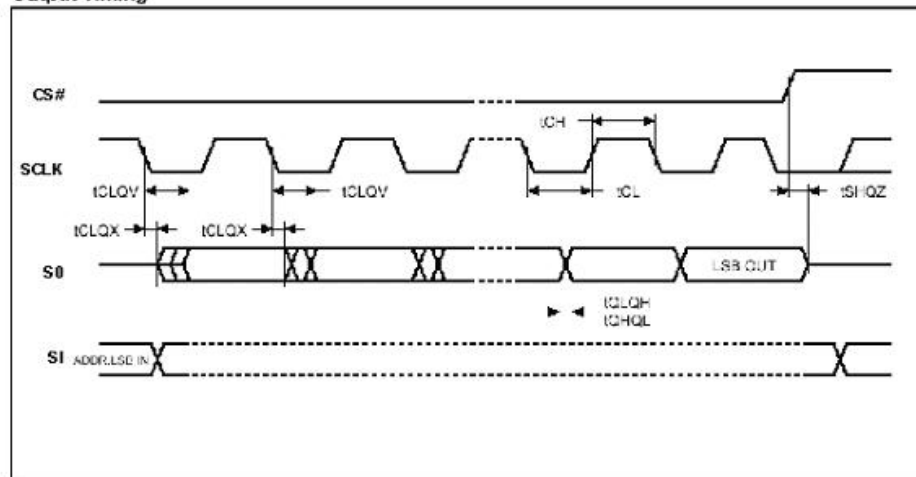
Symbol	Alt.	Parameter	Min.	Max.	Unit
F <sub>c</sub>	F <sub>c</sub>	Clock Frequency	D.C.	90	MHz
t <sub>CH</sub>	t <sub>CLH</sub>	Clock High Time	15		ns
t <sub>CL</sub>	t <sub>CLL</sub>	Clock Low Time	15		ns
t <sub>CLCH</sub>		Clock Rise Time(peak to peak)	0.1		V/ns
t <sub>CHCL</sub>		Clock Fall Time (peak to peak)	0.1		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	5		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		9	ns

$t_{CLQV}$	$t_v$	Clock Low to Output Valid		9	ns
$t_{CLQX}$	$t_{HO}$	Output Hold Time	0		ns

Serial Input Timing



Output Timing

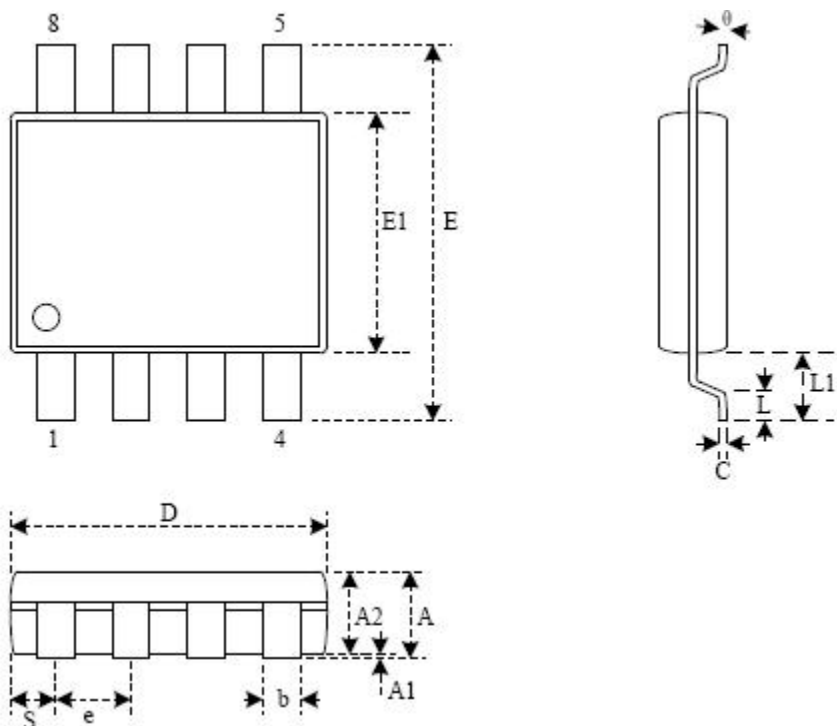


## 6 封装尺寸

封装类型	封装尺寸
SOP8-B	5.28mmX7.90mm (206milX311mil)

SOP8-B

Unit :mm



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

		A	A1	A2	b	C	D	E	E1	⌀	L	S	⊖
Mm	Min.	-	0.05	0.75	0.35	0.15	5.18	7.70	5.18		0.50	0.41	0
	Norm.	-	0.10	0.80	0.42	0.20	5.28	7.90	5.28	1.27	0.65	0.54	5
	Max.	1.0	0.15	0.85	0.48	0.25	5.38	8.10	5.38		0.80	0.67	10
inch	Min.	-	0.002	0.030	0.014	0.006	0.204	0.303	0.204		0.020	0.016	0
	Norm.	-	0.004	0.032	0.016	0.008	0.206	0.311	0.206	0.050	0.026	0.021	5
	Max.	0.04	0.006	0.034	0.020	0.010	0.210	0.319	0.210		0.031	0.026	10



创 造 文 明 智 能

**上海 OFFICE**

地址：上海徐汇区宜山路 1388 号民润大厦 2 号楼 2 层

电话：021-54451588 54451000 54452288

传真：021-54451589-810

E-mail：gtsales@genitop.com

**深圳 OFFICE**

地址：深圳市福田区车公庙泰然工贸园 213 栋 3B2

电话：0755-83453881 83453855

传真：0755-83453855-8004