



TC769HI

TC769HI 16Key + IIC规格书

产品描述

此方案提供客户一个简单的1~16键IIC输出应用。

产品特点

- 程序的独立按键输出模式有两种Multiple以及Single，当选则为Multiple时会将所有按下的按键输出，而选择Single时只会输出第一个按下的按键，当按键被放开时，才会输出下一个按键。
- 设计有省电模式，适合用在如遥控器等需要长时间待机的应用。
- 有4组Slave address，可由外部开关切换。
- 增加多组睡眠扫描，提高休眠灵敏度。

产品应用

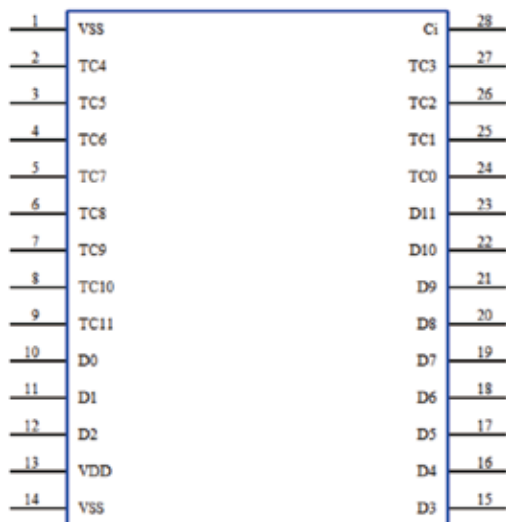
- 大小家电
- 智能家居
- 蓝牙音箱
- 滑动调光台灯
- 门禁监控设备
- 消费类电子



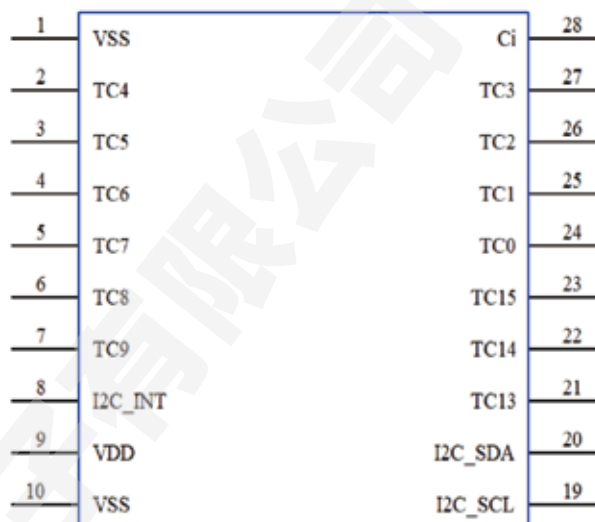
TC769HI

TC769HI 16Key + IIC规格书

封装脚位图



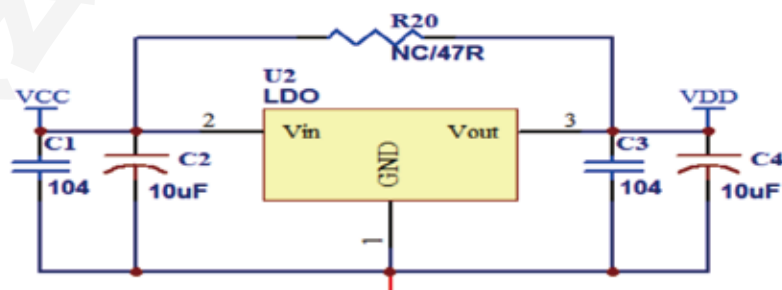
TC769HI-16 SSOP28 A



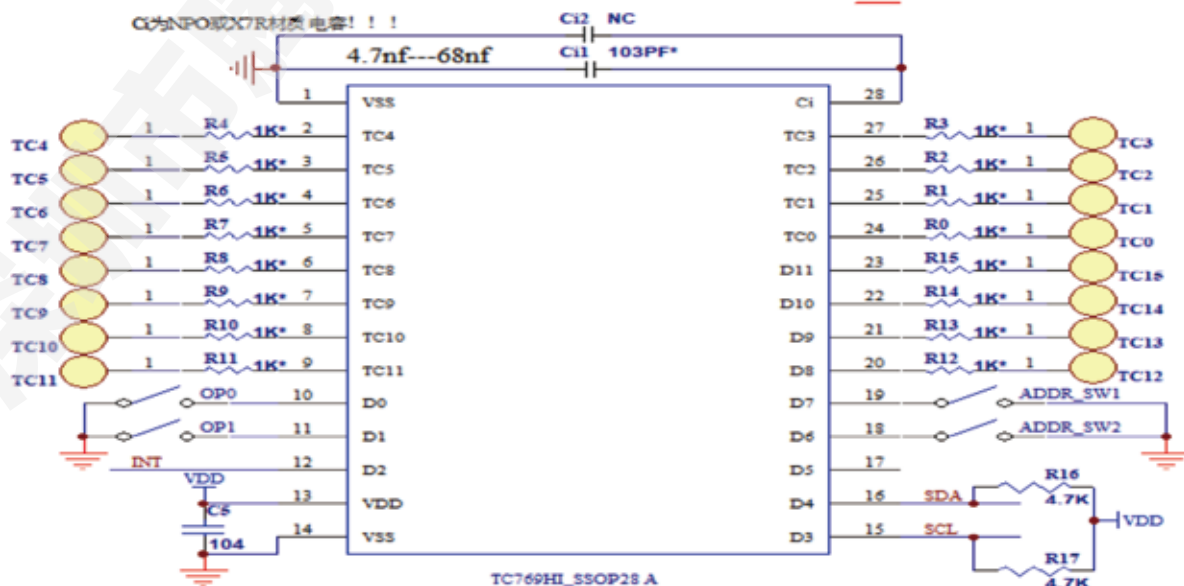
TC769HI-13 TSSOP20

应用参考原理图

OP0	OP1	灵敏度
0	0	400%
0	1	300%
1	0	200%
1	1	100%



C1为NPO或X7R材质电容!!!



TC769HI-16 SSOP28 A

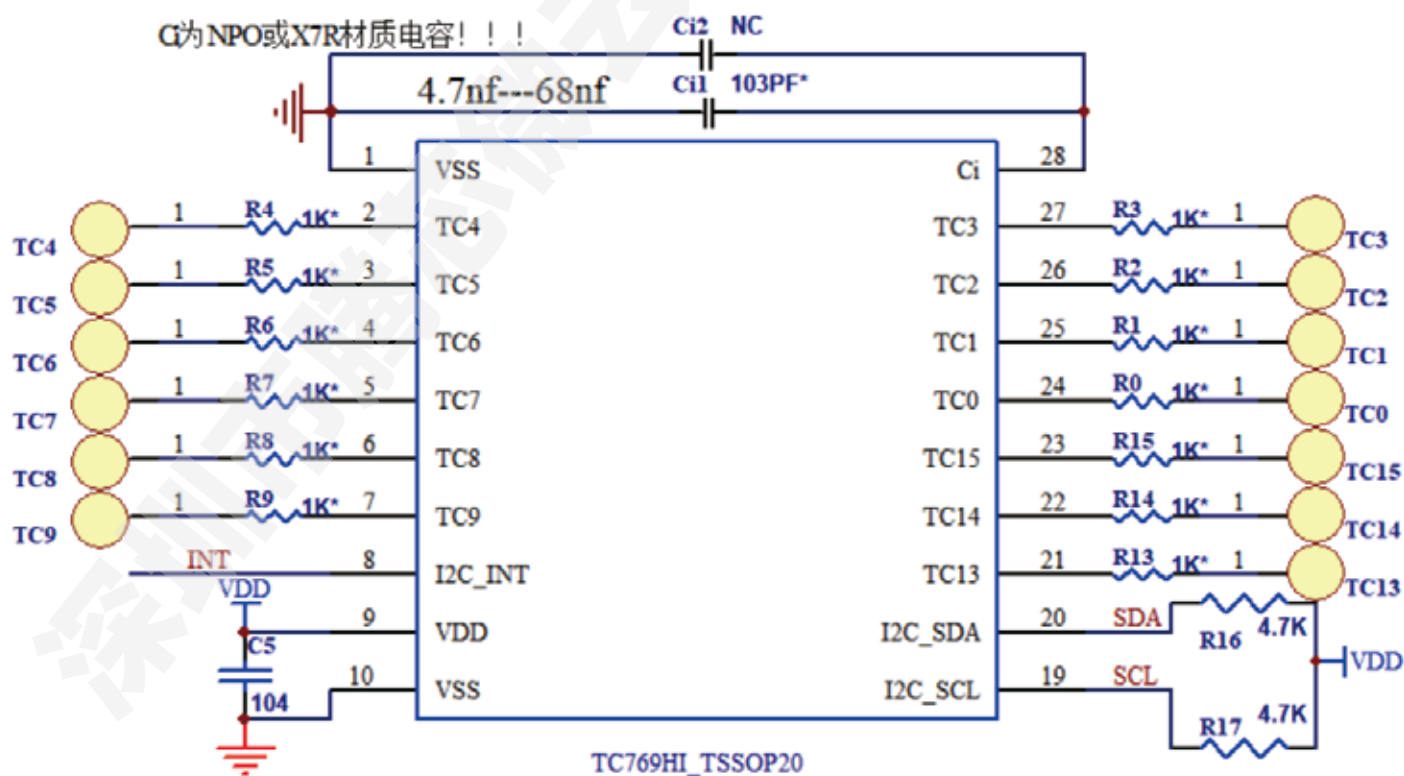
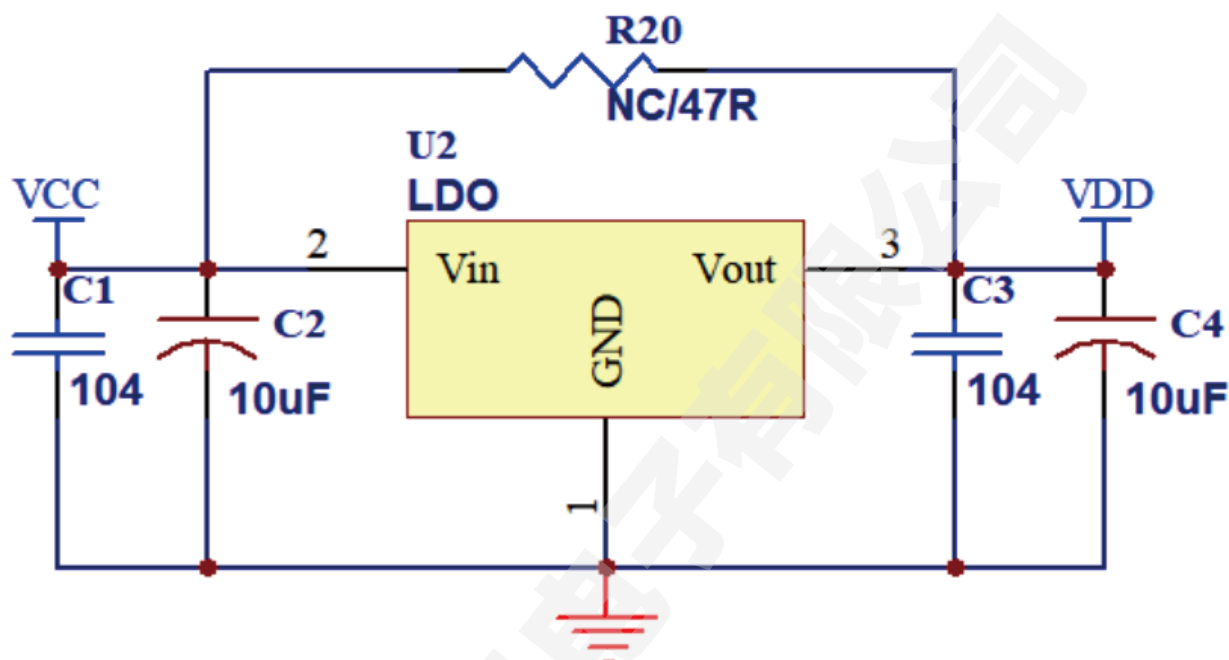
仅供参考!!!



TC769HI

TC769HI 16Key + IIC规格书

TC769HI (TSSOP20) 13KEY





TC769HI

TC769HI 16Key + IIC规格书

脚位定义:

28pin	20pin	Define	I/O	Pin Description
-		RSTB	I	External reset input, active low 50kΩ pull-up(5v)
13	9	V _{DD}	Power	Positive power supply
14	10	V _{SS}	Power	Negative power supply, ground
28	20	Ci	I	Touch sensor input
1	1	V _{SS}	Power	
10		OP0	IO	Sensitivity select option 0
11		OP1	I	Sensitivity select option 1
12	8	I2C_INT	O	I2C interrupt pin
-		-	-	
15	11	I2C_SCL	IO	I2C clock pin
16	12	I2C_SDA	IO	I2C data pin
18		ADDR_SW1	I	I2C slave address select
19		ADDR_SW2	I	I2C slave address select
24	16	TC0	IO/I	touch pad input
25	17	TC1	IO/I	touch pad input
26	18	TC2	IO/I	touch pad input
27	19	TC3	IO/I	touch pad input
2	2	TC4	IO/I	touch pad input
3	3	TC5	IO/I	touch pad input
4	4	TC6	IO/I	touch pad input
5	5	TC7	IO/I	touch pad input
6	6	TC8	IO/I	touch pad input
7	7	TC9	IO/I	touch pad input
8		TC10	IO/I	touch pad input
9		TC11	IO/I	touch pad input
20		TC12	IO/I	touch pad input
21	13	TC13	IO/I	touch pad input
23	14	TC14	IO/I	touch pad input
24	15	TC15	IO/I	touch pad input

Table1. TC769HI pin descriptio



TC769HI

TC769HI 16Key + IIC规格书

Ci为量测电容接脚，电容大小约4.7nF~39nF。

TC0~TC15是触摸按键的量测PAD，TC769HI 最多可侦测16个按键。不使用的按键请保持空接。

OP0、OP1是外部灵敏度选择脚位：

OP1	OP0	灵敏度
0	0	400%
0	1	200%
1	0	20%
1	1	100%

I2C_SDA是I2C的数据输出/输入脚。

I2C_SCL是I2C的时脉输入脚。

电气特性：

1 最大绝对额定值

参数	符号	条件	值	单位
工作温度	Top	——	-40~+85	℃
存放温度	T _{STG}	——	-50~+125	℃
电源电压	VDD	Ta=25℃	VSS-0.3~VSS+5.5	V
输入电压	V _{IN}	Ta=25℃	VSS-0.3~VDD+0.3	V
芯片抗静电强度 HBM	ESD	——	>5	KV
备注：VSS 代表系统接地				

2 DC/AC特性：（测试条件为室温=25℃）

参数	符号	测试条件	最小值	典型值	最大值	单位
工作电压	VDD		2.5	-	5.5	V
系统震荡频率	F	VDD=5V	-	4M	-	Hz
工作电流	I _{OP}	待机，VDD=3V 输出无负载	-	1.1	-	mA
	I _{OFF}	待机，VDD=3V 输出无负载	5.3	6.8	10.0	uA



TC769HI

TC769HI 16Key + IIC规格书

功能描述：

触摸按键介绍：

触摸按键是利用测量人体接近导体时产生的电容变化，转换为数值判断的一种方式。此应用中所有的触摸按键都有Threshold 设定参数，用来调整触摸按键的灵敏度。Threshold 依照按键的按压深度来做调整，数值越小越灵敏，但也越容易受到噪声干扰，需根据实际应用的数据来调整。

IIC协定：

IC使用IIC数据传输协议，两线式总线SCL、SDA来读写数据。INT脚位用来通知Master有按键状态变化。

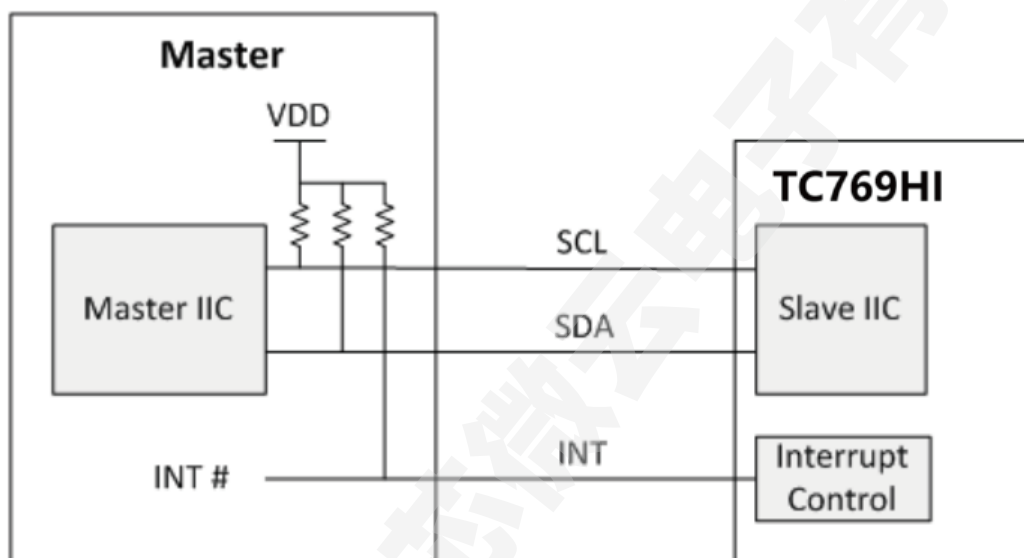
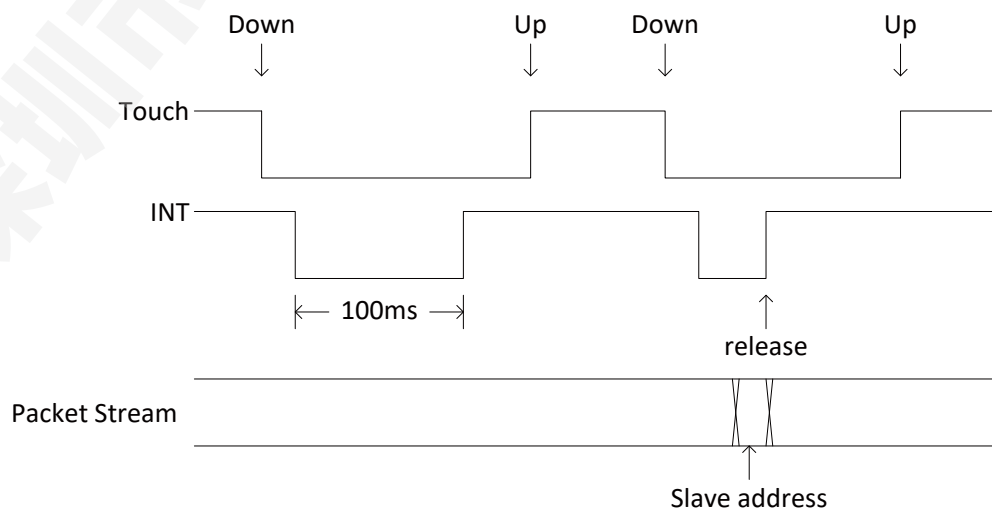


Figure6. IIC connect for master and TC769HI

INT在无按键状态变化时为High，当有按键状态变化时，INT脚位会拉Low 100ms。若Slave接收到 Slave address 则会清除回复为High。





TC769HI

TC769HI 16Key + IIC规格书

Switching Characteristics

Symbol	Description	Min	Max	Units
FSCL	SCL clock frequency.	0	100	KHz
THDSTA	Hold time(repeated) star condition. After this period, the first clock pulse is generated.	4.0	-	us
TLOW	Low period of the SCL clock.	4.7	-	us
THIGH	High period of the SCL clock.	4.0	-	us
TSUSTA	Set-up time for a repeated start condition.	4.7	-	us
THDDAT	Data hold time.	0	-	us
TSUDAT	Data set-up time.	250	-	ns
TSUSTO	Set-up time for stop condition.	4.0	-	us
TBUF	Bus free time between a stop and start condition.	4.7	-	us
TSPI	Pulse width of spikes are suppressed by the input filter.	0	50	ns
TSPT	Slave processor time	10	75	us

Table3. AC characteristics of the I2C SDA and SCL pins for vdd

Timing Waveform

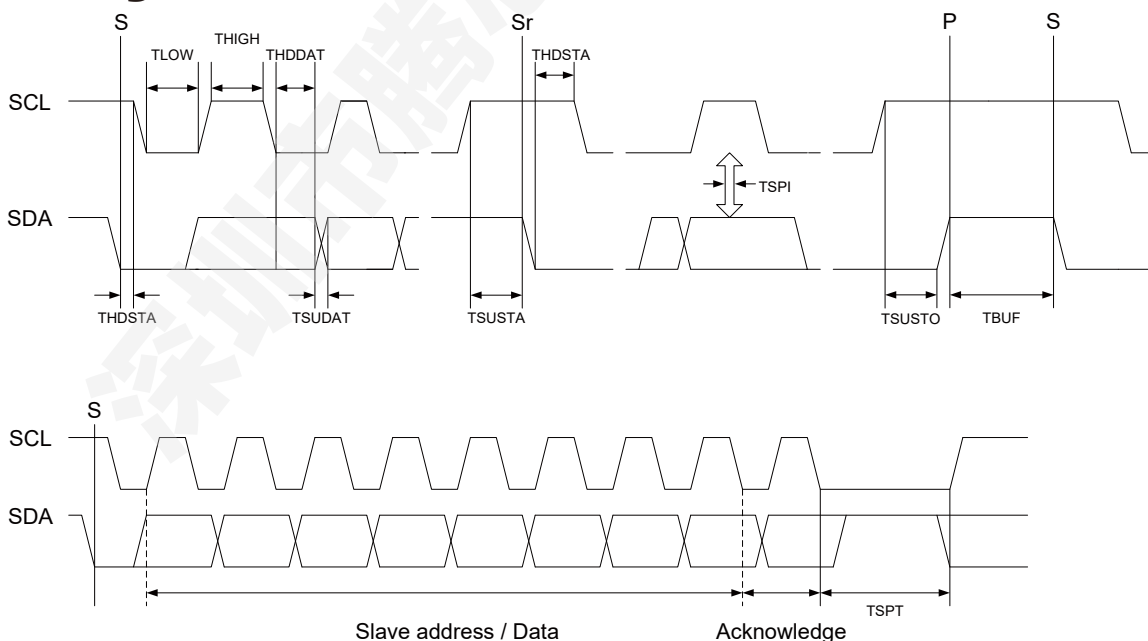


Figure8. Definition for timing for fast/standard mode on the I2C



TC769HI

TC769HI 16Key + IIC规格书

Packet Stream

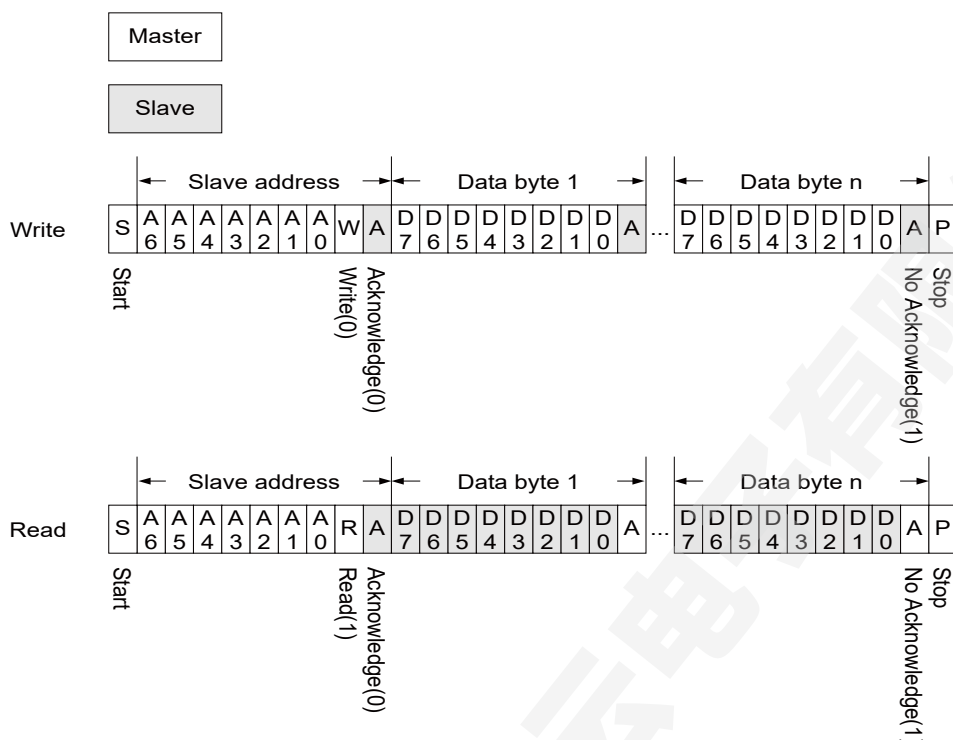


Figure9. Write / Read byte form I2Cc

Slave address

可透过ADDR_SW1, ADDR_SW2脚位做切换, 如下表

ADDR_SW1	ADDR_SW2	Slave address (A6-A0)	Write (A6-A0,R)	Read (A6-A0,R)
0	0	50H	A0H	A1H
0	1	51H	A2H	A3H
1	0	52H	A4H	A5H
1	1	53H	A6H	A7H

ADDR_SW1, ADDR_SW2空接时Slave address为53H, 如果封装没有这两个脚位默认为53H。

Data Stream:

软件设计为16键应用模式。写入数据以3 Data Bytes为一组资料串流。当一笔数据串流写入完成后, 系统会将数据覆写并进行系统重设。若写入被中断并重新写入, 则前一笔数据会被放弃。在每次写入完一组3 Data Bytes后, 若要再次写入下一组设定, 需要送出Stop讯号结束当前数据传输, 再重新写入下一组设定。



TC769HI

TC769HI 16Key + IIC规格书

16 Key Application mode

Write Data

1.Setting commands

Data byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	IICM=1	CT=0	KOM	VREFS	PSM	LDO	KRT	
2	Key Num				SA	KAT		
3	Key Off Num				-	MCKS		

IICM

IIC 数据模式选择。

IICM	IIC Mode
0	Reserve
1	Application mode (Define)

CT

在 Wheel application 模式，写入数据区分成应用设定以及阈值设定，当CT为0时是写入应用设定，当CT=1时是写入阈值设定。

CT	Custom Threshold
0	Setting commands
1	Custom threshold commands

KOM

按键输出模式，有多个按键输出以及单一按键输出两种模式。此选项是对普通按键的输出设定，滑条按键则不受影响。单一按键输出模式时只会输出第一个被按下的按键，当按键放开后才会承认其它按键。

KOM	Key Output Mode
0	Multiple
1	Single (Define)

VREFS

按键扫描的参考电压。

VREFS	
0	1/2 VDD
1	2/3VDD (Define)



TC769HI

TC769HI 16Key + IIC规格书

PSM

省电模式，无按键4秒后进入睡眠模式。

PSM	Power Save Mode
0	Disable(Define)
1	Enable

LDO

Touch IP 内部LDO开关

LDO	
0	Disable(Define)
1	Enable

KRT

长按键复位时间设定，在按键按下开始计时，时间到自动复位。
若按键输出发送变化则重新开始计时。

KRT		Key Reset Time
0	0	Disable
0	1	15 second (Define)
1	0	30 second
1	1	60 second



TC769HI

TC769HI 16Key + IIC规格书

SA

扫描平均模式

SA	Scan Average
0	Disable(Define)
1	Enable

KAT

按键消抖时间

KAT	Key Acknowledge Times		
2 1 0			
0 0 0	3 times		
0 0 1	4 times (Define)		
0 1 0	5 times		
0 1 1	6 times		
1 0 0	7 times		
1 0 1	8 times		
1 1 0	9 times		
1 1 1	10 times		

Key Num

按键数设定

Key Num	Key Number		
3 2 1 0			
0 0 0 0	1 key		
0 0 0 1	2 keys		
0 0 1 0	3 keys		
0 0 1 1	4 keys		
0 1 0 0	5 keys		
0 1 0 1	6 keys		
0 1 1 0	7 keys		
0 1 1 1	8 keys		
1 0 0 0	9 keys		
1 0 0 1	10 keys		
1 0 1 0	11 keys		
1 0 1 1	12 keys (Define)		
1 1 0 0	13 keys		
1 1 0 1	14 keys		
1 1 1 0	15 keys		
1 1 1 1	16 keys		



TC769HI

TC769HI 16Key + IIC规格书

Key Off Num

多按键重置设定，最多为16 Keys

Key Off Num				Key Off Number
3	2	1	0	
0	0	0	0	Disable (Define)
0	0	0	1	2key reset
0	0	1	0	3 keys reset
0	0	1	1	4 keys reset
0	1	0	0	5 keys reset
0	1	0	1	6 keys reset
0	1	1	0	7 keys reset
0	1	1	1	8 keys reset
1	0	0	0	9 keys reset
1	0	0	1	10 keys reset
1	0	1	0	11 keys reset
1	0	1	1	12 keys reset
1	1	0	0	13 keys reset
1	1	0	1	14 keys reset
1	1	1	0	15 keys reset
1	1	1	1	16 keys reset

MCKS

MCKS			Modulation clock selector	
2	1	0		当开启睡眠时，如设定的是偶数除频，则程序自动切换为奇数除频。
0	0		RC8M/2(Define)	RC8M/3
0	0	1	RC8M/3	
0	1	0	RC8M/4	RC8M/5
0	1	1	RC8M/5	
1	0	0	RC8M/8	
1	0	1	RC8M/16	
1	1	0	RC8M/32	
1	1	1	OSCL	OSCL

Note:开启睡眠时，MCKS需要选择RC8M/3，RC8M/5，才能保证睡眠稳定。



TC769HI

TC769HI 16Key + IIC规格书

2.Custom threshold commands

阈值则是设定按键承认的门坎。分为按键阈值，睡眠唤醒阈值两种。

Item

选择切换不同的写入参数的设定。

Item	Item
0 0	TPx setting
0 1	Sleep setting
1 0	Reference Adjust Speed
1 1	-

TPx Setting

Data byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	IICM=1	CT=1	Item=0		TP Num			
2	TPx Threshold M				TPx Threshold L			
3					TPx Threshold H			

TPx Threshold：按键承认阈值。(Define：010H) (max: FFFH ,min: 001H)

按键承认阈值越小灵敏度越高，越大灵敏度越低。预设的阈值为010H，建议的最小值为008H，若调整到008H按键灵敏度仍然不够，则建议加大Ci电容，Ci电容的值则建议小于39nF。

TP Num

数据写入的按键编号。

TP NUM				TC Number
3	2	1	0	
0	0	0	0	TC0
0	0	0	1	TC1
0	0	1	0	TC2
0	0	1	1	TC3
0	1	0	0	TC4
0	1	0	1	TC5
0	1	1	0	TC6
0	1	1	1	TC7
1	0	0	0	TC8
1	0	0	1	TC9
1	0	1	0	TC10
1	0	1	1	TC11
1	1	0	0	TC12
1	1	0	1	TC13
1	1	1	0	TC14
1	1	1	1	TC15



TC769HI

TC769HI 16Key + IIC规格书

· Sleep Setting Group 1

Data byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	IICM=1	CT=1	Item=1		0			
2	TPSLP Threshold M				TPSLP Threshold L			
3					TPSLP Threshold H			

TPSLP Threshold : 省电模式唤醒阈值。(Define : 002H)(max: FFFH ,min: 001H)

Data byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	IICM=1	CT=1	Item=1		1			
2	TC7	TC6	TC5	TC4	TC3	TC2	TC1	TC0
3	TC15	TC14	TC13	TC12	TC11	TC10	TC9	TC8

TC0~TC15:睡眠扫描通道设定(Define:000FH)

0000H=扫描所有使用中的按键。

Group 2

Data byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	IICM=1	CT=1	Item=1		2			
2	TPSLP Threshold M				TPSLP Threshold L			
3					TPSLP Threshold H			

TPSLP Threshold : 省电模式唤醒阈值。(Define : 002H) (max: FFFH ,min: 001H)

Data byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	IICM=1	CT=1	Item=1		3			
2	TC7	TC6	TC5	TC4	TC3	TC2	TC1	TC0
3	TC15	TC14	TC13	TC12	TC11	TC10	TC9	TC8

TC0~TC15:睡眠扫描通道设定(Define:00F0H) (max: FFFH ,min: 001H)

0000H=不使用。

Group 3

Data byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	IICM=1	CT=1	Item=1		4			
2	TPSLP Threshold M				TPSLP Threshold L			
3					TPSLP Threshold H			

TPSLP Threshold : 省电模式唤醒阈值。(Define : 002H) (max: FFFH ,min: 001H)

Data byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	IICM=1	CT=1	Item=1		5			
2	TC7	TC6	TC5	TC4	TC3	TC2	TC1	TC0
3	TC15	TC14	TC13	TC12	TC11	TC10	TC9	TC8

TC0~TC15:睡眠扫描通道设定(Define:0F00H)

0000H=不使用。



TC769HI

TC769HI 16Key + IIC规格书

Group 4

Data byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	IICM=1	CT=1	Item=1		6			
2	TPSLP Threshold M				TPSLP Threshold L			
3					TPSLP Threshold H			

TPSLP Threshold : 省电模式唤醒阈值。(Define : 002H) (max: FFFH ,min: 001H)

Data byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	IICM=1	CT=1	Item=1		7			
2	TC7	TC6	TC5	TC4	TC3	TC2	TC1	TC0
3	-	-	-	TC12	TC11	TC10	TC9	TC8

TC0~TC15:睡眠扫描通道设定(Define:0000H)

0000H=不使用。

Reference Adjust Speed :环境值校准速度

Data byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	IICM=1	CT=1	Item=2		0			
2	Up adjust speed							
3	Down adjust speed							

Up adjust speed (Define:80H@8ms) (max: FFH ,min: 01H)

Down adjust speed (Define:80H@8ms) (max: FFH ,min: 01H)

Read Data

Data byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	C	WSET						
2	Key 8	Key 7	Key 6	Key 5	Key 4	Key 3	Key 2	Key 1
3	Key 16	Key 15	Key 14	Key 13	Key 12	Key 11	Key 10	Key 9

C
系统校正标志，当值为0时，表示系统校正中，键值读取无效。当值为1时，键值有效。

C	Calibrate
0	Calibrating
1	Calibrate Finish

WSET

系统写入标志，上电为1，写入设定后该标志设置为0。

WSET	Have write setting
0	Have write setting
1	No write setting



TC769HI

TC769HI 16Key + IIC规格书

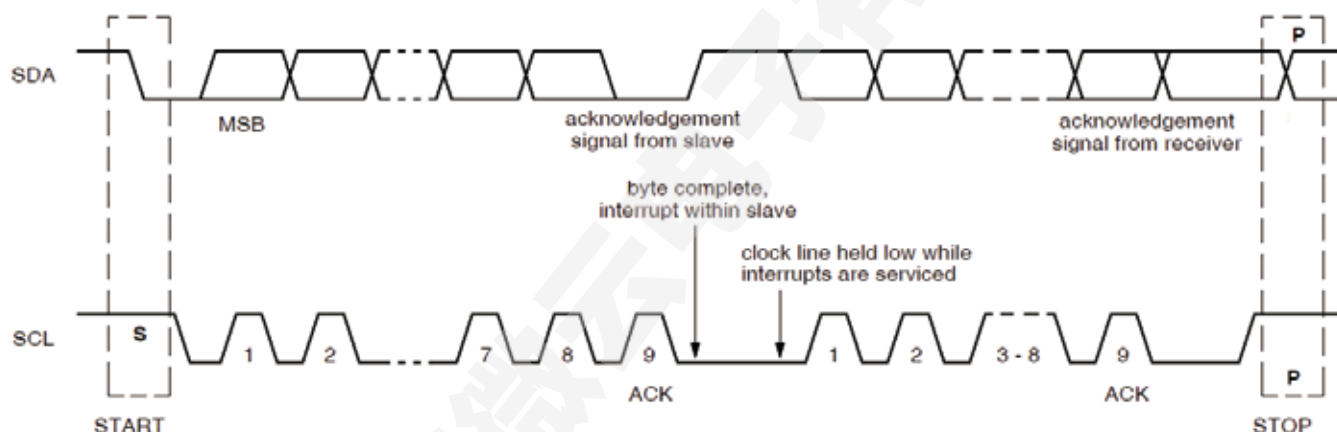
K1...K16

触摸按键标志，无按键为0，有按键为1

K1...K16	Key1...Key16
0	No touch
1	Touch

特别说明：

TC769HI的I2C接口有硬件的支持SCL可支持100KHz，但是译码为软件处理，处理约需20~100us视处理的情况而定，所以当Master发送完一组数据封包，应增加大于200us的延时，再发送下一组数据封包。



所以Master写程序时，需注意SCL拉Low的动作，若由硬件控制大多会支持此标准，若由程序控制IO脚，请增加对SCL输出High时要读回确认为High，才可让程序继续进行，若为Low应等待SCL为High后才可继续进行。C的程序如下：

```
SCL=1;
while(SCL!=1) {};
```

- 若需要连续读取键值，建议读取完后暂停10ms以上，再读取下一次键值。否则会影响按键的反应速度。
- 若开启睡眠模式，则禁止连续读取键值，因为每次读取键值时，都会清除进入睡眠的计时，会导致系统无法睡眠。
- 在系统进入睡眠模式时，会将I2C功能关闭。此时重新下I2C指令可以唤醒系统，但是会收到no ACK的响应，需要等待系统唤醒后再重新下读写命令。
- 按键阈值调整的步骤：

Step1. 选择初始测试用的Ci电容(见建议线路图):

先确定设计中是否使用滑条功能，若使用滑条功能，则建议33nF作为Ci充放电电容，若仅做一般按键使用，则建议使用10nF作为初始测试电容。

Step2. 每个按键做按压测试：



TC769HI

TC769HI 16Key + IIC规格书

以正常速度轻触按键或使用金属棒做测试条件，若在触摸到按键之前有按键输出，表示灵敏度太高，需要调高阈值，若触摸按键没有按键输出，或是要重压才有按键输出，表示灵敏度太低，需要降低阈值。

滑条按键因为锯齿状设计，在不同位置灵敏度也会不同，故建议做灵敏度测试时，以两个锯齿按键中间位置做灵敏度测试，避免滑动效果不佳。

Step3. 测试按键反应速度：

在判断按键灵敏度的时候，若觉得按键“不够灵敏”，需要进一步判断是按键响应速度不够快，还是按键灵敏度不够。判断方法是触摸停留一段时间(约1秒)，并检查是否有按键输出。若没有按键输出，则是按键不够灵敏，重新进行Step2调整，若有按键输出，则是按键响应速度不够快，则进行下一步。

Step4. 调整按键反应速度：

按键消抖时间(KAT)预设4，若按键反应速度不够快，可以下修到3。

若下修到3反应速度仍然不够，则建议将Ci电容减小。

选择好适当的Ci电容后需要回到Step2重新调整灵敏度。

需要注意的是选择较小的Ci电容，同时会降低滑条按键的精细度。



TC769HI

TC769HI 16Key + IIC规格书

示范程序:

项目名称:主控端对TC769HI透过I2C控制的范例程序

项目目的:1.透过软件模拟I2C主控端对TC769HI写入设定参数

2.透过软件模拟I2C主控端对TC769HI读取按键状态

主控MCU:AT89C51

Date & Version: 2025/03/18 v1.0

```
*/  
//-----  
#include<reg51.h>  
#include<intrins.h>  
#define uint unsigned int  
#define uchar unsigned char  
#define address_W 0xa6 //从机的地址和写入标志  
#define address_R 0xa7 //从机的地址和读取标志
```

```
sbit SINT=P0^0; //主控端与从机的IIC接口  
sbit SDA=P0^1; //主控端与从机的IIC接口  
sbit SCL=P0^2; //主控端与从机的IIC接口  
uchar Write_Buffer[3]; //主控端的写入资料缓存  
uchar Read_Buffer[3]; //主控端的读取资料缓存  
//-----
```

```
//函数名称: void delay(uint x)  
//函数功能: 程序延时  
//函数输入: x  
//函数输出: 无  
//中间变量: i, j  
//-----
```

```
void delay(uint x)  
{  
    uint i, j;  
    for(i=x; i>0; i--)  
        for(j=0x40; j>0; j--);  
}
```

```
//-----  
//函数名称: void sendStart()  
//函数功能: IIC的起始位  
//函数输入: 无
```



TC769HI

TC769HI 16Key + IIC规格书

```
//函数输出: 无
//中间变量: 无
//-----
void sendStart() //开始位
{
    SDA=1; /*发送起始条件的数据信号*/
    SCL=1;
    while(SCL!=1) {};
    SDA=0; /*发送起始信号*/
    _nop_();
    SCL=0; /*此位置只需要将SCL输出为0之后等待4US即可*/
}
//-----
//函数名称: void sendStop()
//函数功能: IIC的结束位
//函数输入: 无
//函数输出: 无
//中间变量: 无
//-----
void sendStop() //停止位
{
    SCL=0;
    SDA=0; /*发送结束条件的数据信号*/
    _nop_();
    SCL=1;
    while(SCL!=1) {};
    _nop_();
    SDA=1;
}
//-----
//函数名称: bit readACK()
//函数功能: 读取IIC的acknowledge标志位
//函数输入: 无
//函数输出: IIC的ACK信号 返回1表示无acknowledge, 0表示有acknowledge
//中间变量: 无
//-----
bit readACK() //读取应答信号
{
    SCL=0;
    SDA=1; /*此处为释放SDA总线, 由从从机发出低电平应答*/
    _nop_();
```



TC769HI

TC769HI 16Key + IIC规格书

```
SCL=1;
_nop_();
if(SDA)
    return 1; //no ACK
else
    return 0; //ACK
}
//-----
//函数名称: void sendACK()
//函数功能: 主控端送出应答信号
//函数输入: 无
//函数输出: 无
//中间变量: 无
//-----
void sendACK() //输出应答信号
{
    SCL=0;
    SDA=0;
    _nop_();
    SCL=1;
}
//-----
//函数名称: void sendNOACK()
//函数功能: 主控端送出无应答信号
//函数输入: 无
//函数输出: 无
//中间变量: 无
//-----
void sendNOACK() //输出无应答信号
{
    SCL=0;
    SDA=1;
    _nop_();
    SCL=1;
}
//-----
//函数名称: void sendByte(uchar dat)
//函数功能: 主控端写一个字节到从机
//函数输入: dat = 发送的字节
//函数输出: 无
//中间变量: i
```



TC769HI

TC769HI 16Key + IIC规格书

```
//-----  
void sendByte(uchar dat) //写一个字节  
{  
    uchar i;  
    for(i=0;i<8;i++)  
    {  
        SCL=0; /*钳住I2C总线, 准备发送数据 */  
        if(dat&0x80)  
            SDA=1;  
        else  
            SDA=0;  
        _nop_(); /*如果需要在SDA,SCL,INT上串接电阻, 根据电阻大小不同, 电阻越大建议将该时间  
适当加长, 100KHZ以内即可; */  
        _nop_();  
        SCL=1; /*此处由于51单片机的特性不需要做输入输出设置,  
但如果是其他单片机需要先将其IO口改为输入上拉的设置, 读到高之后, SCL转为输出为高。  
在读写完ACK后的第一个clock下降缘从机会钳住SCL脚做资料处理,  
所以将SCL脚置为输入上拉, 并等待SCL被释放。*/  
        while(SCL!=1) {};  
        dat<<=1;  
    }  
}  
//-----  
//函数名称: uchar readByte()  
//函数功能: 主控端对从机读取一个字节  
//函数输入: 无  
//函数输出: 读取完成的字节  
//中间变量: i, dat  
//-----  
uchar readByte() //读一个字节  
{  
    uchar i, dat=0;  
    for(i=0;i<8;i++)  
    {  
        SCL=0;  
        SDA=1;  
        _nop_(); /*如果需要在SDA,SCL,INT上串接电阻, 根据电阻大小不同, 电阻越大建议将该时间  
适当加长, 100KHZ以内即可; */  
        dat<<=1;  
        SCL=1; /*此处由于51单片机的特性不需要做输入输出设置,  
但如果是其他单片机需要先将其IO口改为输入上拉的设置, 读到高之后, SCL转为输出为高。
```



TC769HI

TC769HI 16Key + IIC规格书

在读写完ACK后的第一个clock下降缘从机会钳住SCL脚做资料处理，
所以将SCL脚置为输入上拉，并等待SCL被释放。*/

```
while(SCL!=1) {};  
if(SDA==1)  
    dat|=0x01;  
}  
return dat;  
}  
//-----  
//函数名称: bit writellC(uchar addrW, uchar *writeData, uchar length)  
//函数功能: 主控端对从机数据写入  
//函数输入: addrW = 从机地址及写入旗帜  
//          *writeData = 预备写入数据的首个地址  
//          length = 写入数据的长度(字节数)  
//函数输出: 返回IIC通讯的acknowledge状态，若为1，则停止并返回。若为0，则完成通讯后返回  
//中间变量: i, ACK  
//-----  
bit writellC(uchar addrW, uchar *writeData, uchar length)  
{  
    uchar i;  
    bit ACK;  
    sendStart();  
    sendByte(addrW); //传送地址与写入标记  
    ACK = readACK();  
    if (ACK)  
    {  
        sendStop(); //地址不正确或装置未连接，送出停止信号  
        return ACK;  
    }  
  
    for(i = 0; i<length; i++)  
    {  
        sendByte(writeData[i]);  
        ACK = readACK();  
        if (ACK)  
        {  
            sendStop(); //未接收到ACK，送出停止信号  
            return ACK;  
        }  
    }  
}  
sendStop(); //资料写入完成，送出停止信号
```



TC769HI

TC769HI 16Key + IIC规格书

```
    return ACK;
}
//-----
//函数名称: bit readIIC(uchar addrR, uchar *readData, uchar length)
//函数功能: 主控端对从机数据读取
//函数输入: addrR = 从机地址及读取旗帜
//      *readData = 预备读取后存放数据的首个地址
//      length = 读取数据的长度(字节数)
//函数输出: 返回IIC通讯的acknowledge状态, 若为1, 则停止并返回。若为0, 则完成通讯后返回
//中间变量: i, ACK
//-----
bit readIIC(uchar addrR, uchar *readData, uchar length)
{
    uchar i;
    bit ACK;
    sendStart();
    sendByte(addrR); //传送地址与读取标记
    ACK = readACK();
    if (ACK)
    {
        sendStop(); //地址不正确或装置未连接, 送出停止信号
        return ACK;
    }

    for(i = 0; i<length; i++)
    {
        readData[i] = readByte();
        if(i<length-1)
            sendACK();
        else
            sendNOACK(); //读取最后一笔资料, 送出No ACK
    }
    sendStop(); //资料读取完成, 送出停止信号
    return ACK;
}
//-----
//函数名称: void setWrite_Buffer_3(uchar byte1, uchar byte2, uchar byte3)
//函数功能: 写入3个字节到写入缓存寄存器
//函数输入: byte1
//      byte2
//      byte3
```



TC769HI

TC769HI 16Key + IIC规格书

```
//函数输出: 无
//中间变量: 无
//-----
void setWrite_Buffer_3(uchar byte1, uchar byte2, uchar byte3)
{
    Write_Buffer[0] = byte1;
    Write_Buffer[1] = byte2;
    Write_Buffer[2] = byte3;
}
void main()
{
    bit ACK;
    SINT = 1;
    setWrite_Buffer_3(0xB1, 0xB9, 0x00);
    ACK = writeIIC(address_W, &Write_Buffer, 3); //MCU Setting
    delay(1); //delay 1ms
    setWrite_Buffer_3(0xC0, 0x10, 0x00);
    ACK = writeIIC(address_W, &Write_Buffer, 3); //TC0 Threshold
    delay(1); //delay 1ms
    setWrite_Buffer_3(0xC1, 0x10, 0x00);
    ACK = writeIIC(address_W, &Write_Buffer, 3); //TC1 Threshold
    delay(1); //delay 1ms
    setWrite_Buffer_3(0xC2, 0x10, 0x00);
    ACK = writeIIC(address_W, &Write_Buffer, 3); //TC2 Threshold
    delay(1); //delay 1ms
    setWrite_Buffer_3(0xC3, 0x10, 0x00);
    ACK = writeIIC(address_W, &Write_Buffer, 3); //TC3 Threshold
    delay(1); //delay 1ms
    setWrite_Buffer_3(0xC4, 0x10, 0x00);
    ACK = writeIIC(address_W, &Write_Buffer, 3); //TC4 Threshold
    delay(1); //delay 1ms
    setWrite_Buffer_3(0xC5, 0x10, 0x00);
    ACK = writeIIC(address_W, &Write_Buffer, 3); //TC5 Threshold
    delay(1); //delay 1ms
    setWrite_Buffer_3(0xC6, 0x10, 0x00);
    ACK = writeIIC(address_W, &Write_Buffer, 3); //TC6 Threshold
    delay(1); //delay 1ms
    setWrite_Buffer_3(0xC7, 0x10, 0x00);
    ACK = writeIIC(address_W, &Write_Buffer, 3); //TC7 Threshold
    delay(1); //delay 1ms
    setWrite_Buffer_3(0xC8, 0x10, 0x00);
```



TC769HI

TC769HI 16Key + IIC规格书

```
ACK = writeIIC(address_W, &Write_Buffer, 3); //TC8 Threshold
delay(1); //delay 1ms
setWrite_Buffer_3(0xC9, 0x10, 0x00);
ACK = writeIIC(address_W, &Write_Buffer, 3); //TC9 Threshold
delay(1); //delay 1ms
setWrite_Buffer_3(0xCA, 0x10, 0x00);
ACK = writeIIC(address_W, &Write_Buffer, 3); //TC10 Threshold
delay(1); //delay 1ms
setWrite_Buffer_3(0xCB, 0x10, 0x00);
ACK = writeIIC(address_W, &Write_Buffer, 3); //TC11 Threshold
delay(1); //delay 1ms
setWrite_Buffer_3(0xD0, 0x02, 0x00);
ACK = writeIIC(address_W, &Write_Buffer, 3); //Sleep Group 1 Threshold
delay(1); //delay 1ms
setWrite_Buffer_3(0xD1, 0x0F, 0x00);
ACK = writeIIC(address_W, &Write_Buffer, 3); //Sleep Group 1
delay(1); //delay 1ms
setWrite_Buffer_3(0xD2, 0x02, 0x00);
ACK = writeIIC(address_W, &Write_Buffer, 3); //Sleep Group 2 Threshold
delay(1); //delay 1ms
setWrite_Buffer_3(0xD3, 0xF0, 0x00);
ACK = writeIIC(address_W, &Write_Buffer, 3); //Sleep Group 2
delay(1); //delay 1ms
setWrite_Buffer_3(0xD4, 0x02, 0x00);
ACK = writeIIC(address_W, &Write_Buffer, 3); //Sleep Group 3 Threshold
delay(1); //delay 1ms
setWrite_Buffer_3(0xD5, 0x00, 0x0F);
ACK = writeIIC(address_W, &Write_Buffer, 3); //Sleep Group 3
delay(1); //delay 1ms
setWrite_Buffer_3(0xD6, 0x02, 0x00);
ACK = writeIIC(address_W, &Write_Buffer, 3); //Sleep Group 4 Threshold
delay(1); //delay 1ms
setWrite_Buffer_3(0xD7, 0x00, 0x00);
ACK = writeIIC(address_W, &Write_Buffer, 3); //Sleep Group 4
delay(50);
while(1)
{
    if(!SINT) /*等待读取请求，若关闭省电模式可以不用读取SINT，但是每次读取按键建议间隔30ms*/
        ACK = readIIC(address_R, &Read_Buffer, 3); //读取按键状态
}
```



TC769HI

TC769HI 16Key + IIC规格书

触摸PAD 到IC pin 串1K Ω -4.7K Ω 是可以增强抗手机和对讲机的干扰。

Ci外接电容与压克力厚度关系：(预设阈值010H)

以铁片弹簧键，圆型实心直径 12 MM为例，压克力厚度与Ci 电容的关系如下：

压克力厚度(mm)	Ci
1	472
2	882
4	103
6	153
8	223
10	223

此表格仅供参考，不同的 PAD 大小，PCB layout 皆会影响。

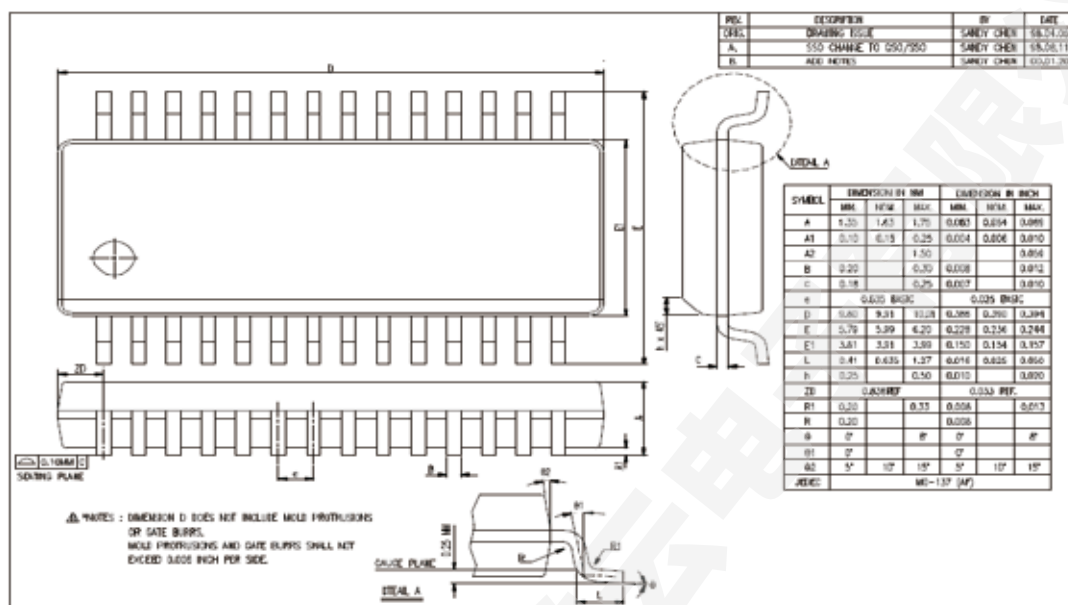


TC769HI

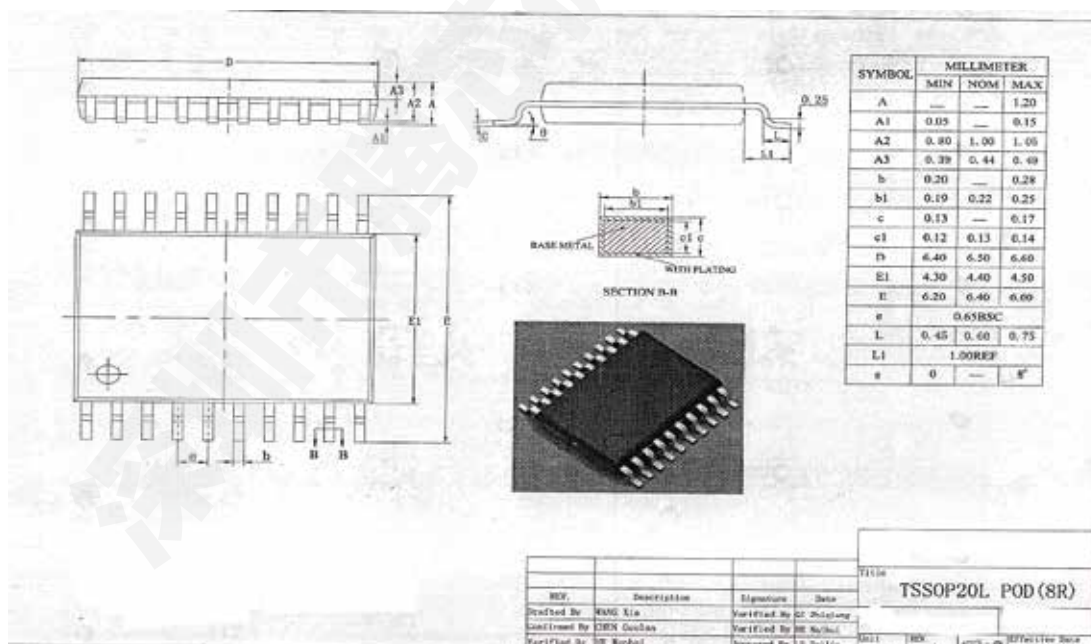
TC769HI 16Key + IIC规格书

封装说明:

(28-SSOP A)



(20-TSSOP)





TC769HI

TC769HI 16Key + IIC规格书

订购信息：

TC769HI

a. 封装型号：TC769HI-ASFN (SSOP28 A)

b. 封装型号：TC769HI-DTDN (TSSOP20)