



矽源特科技
ChipSourceTek

CSK0xT 应用手册

高性能自电容触控芯片



目录

一、 概述	1
二、 上电时序	2
三、 IIC 接口说明	3
四、 寄存器说明	5
五、 应用说明	12
六、 更新历史	17

矽源特科技
ChipSourceTek



CSK0xT 应用手册

一、CSK0xT概述

1.1、CSK 系列自容触控芯片，采用高速 MCU 内核并内嵌 DSP 电路，结合自身的快速自电容感应技术，实现极高灵敏度和极低待机功耗。

1.2、本手册针对 IIC 接口的标准品芯片的配置进行详细说明。包含的芯片型号有：CSK05T、CSK08T。
CSK05T、CSK08T 可以替换 CSK05S、CSK08S，但实际的触摸效果还需要重新验证，不可以替换后就直接量产！

1.3、CSK 系列 IC 的供电范围在：2.8V~3.6V 之间，电源纹波要求 $\leq 50\text{mv}$ ，外部接 $2.2\mu\text{F} \sim 10\mu\text{F}$ 电容滤波。不可使用电池串接电阻直接供电！推荐使用 LDO 稳压供电。

1.4、CSK 系列 IC 在上电或复位 30ms（初始化耗时）后方可进行 iic 通信。

1.5、CSK 系列 IC 的 CMOD0 脚 必须接稳压电容，大小在 $1\text{nF} \sim 10\text{nF}$ ，推荐使用 1nF 。

1.6、CSK 系列 IC 的 iic 接口除支持开漏模式外，还支持内部上拉（大约 5.6K）模式，支持 3.3V 和 1.8V 电平通信。



二、CSK0xT上电时序

CSK 系列 IC 上电的时序如下图：

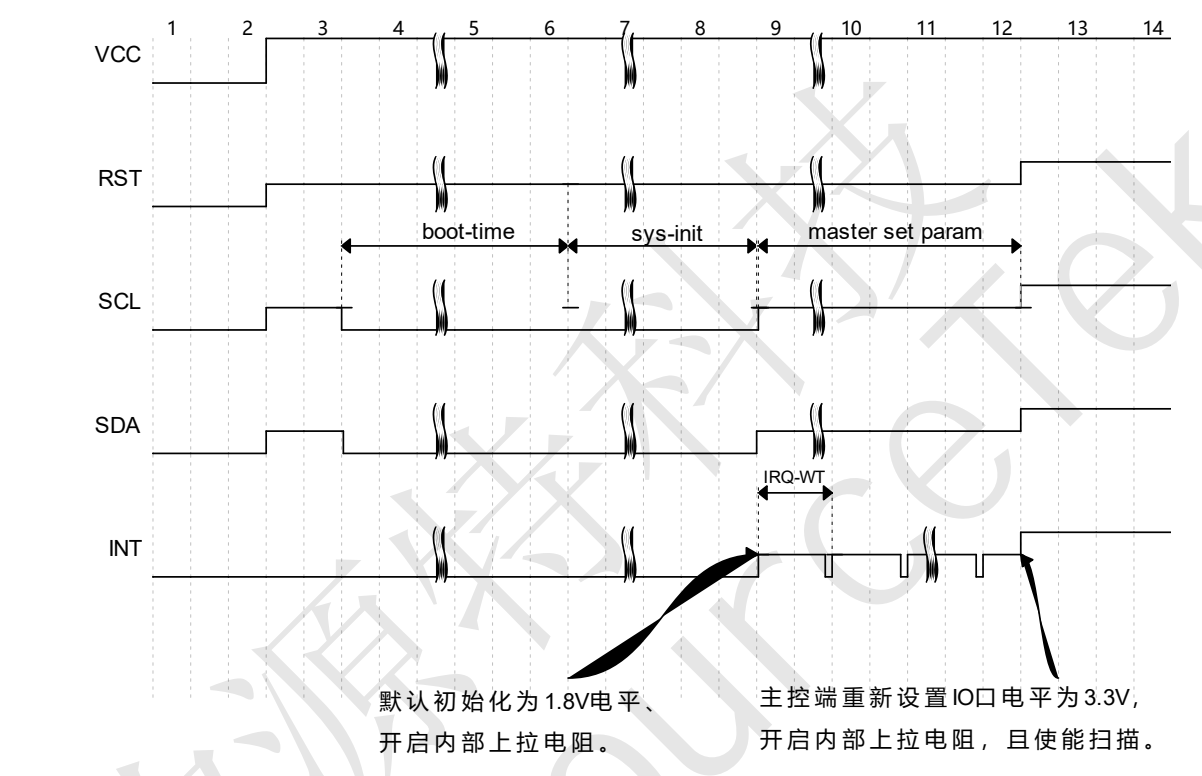


图 1： 上电时序

2.1、触摸 IC 上电 5ms 后会进入 boot 模式，在此时间窗口 (boot-time: 20ms) 内，主控可以对触摸 IC 进行固件升级，标准品 IC 可以忽略此模式。

2.2、退出 boot 模式后，触摸 IC 进入系统初始化 (sys-init)，此段时间大约耗时 10ms。初始化完成后，SCL、SDA、INT(IRQ)引脚会被初始化为 1.8V 电平，且开启内部上拉电阻。

2.3、系统初始化完成后 (master set param)，主控端可以根据实际项目需要，再对触摸 IC 进行重新初始化成 3.3V 电平 (使能触摸扫描后才会去初始化 IO 口电平)。

如果主控是 3.3V 电平的器件，建议主控将自身的 iic 接口配置成内部上拉，或将外部上拉电阻焊接上，再重新初始化触摸 IC。

2.4、触摸 IC 处于 boot-time 和 sys-init 时段内，SCL、SDA、INT、RST 都是 1.8V 电平；SCL、SDA、INT 都处于 OD 模式。

2.5、触摸 IC 的 RST 脚的内部上拉电阻一直存在，不可设置！



三、CSK0xT IIC 接口说明

CSK 芯片是使用 IIC 进行通信。本章对 IIC 的基本操作时序进行说明。

1. 器件的 IIC 地址

CSK 系列芯片的 7bit 设备地址为 0x15，即设备写地址为：0x2A，读地址为：0x2B。

2. IIC 的通信速度

为了保证通信的可靠性，建议最大使用 400Kbps 的通信速率。

3. 写入单个字节

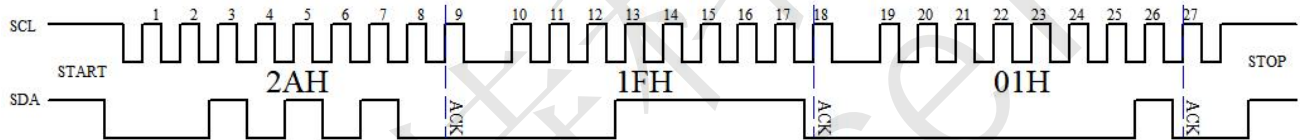


图 2: 往 0x1F 寄存器写入 0x01

4. 连续写入多个字节

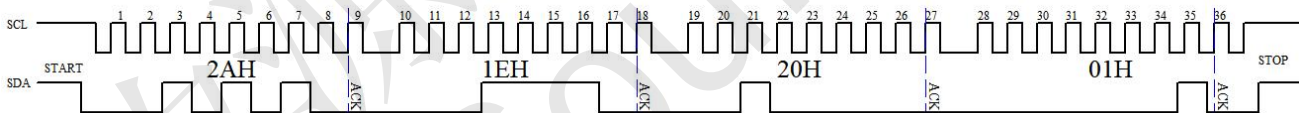


图 3: 往 0x1E、0x1F 分别写入 0x20、0x01

5. 读取单个字节

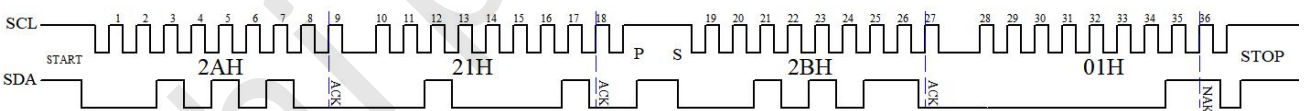


图 4: 从 0x21 读取单个字节

6. 连续读取多个字节

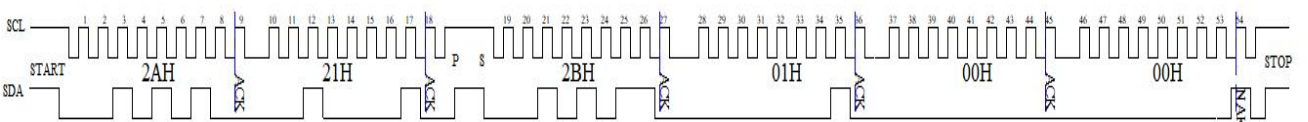


图 5: 从 0x21、0x22、0x23 读取 3 个字节



7. 关于 IIC 操作的重要说明

7.1、主控端进行读取数据时，写完寄存器地址后必须先 stop, delay50us 以上, 再 start, 再读数据。CSK 芯片不支持 restart 操作!

7.2、主控端读取一个字节后需要间隔（延时）30us 后才可以读取下一个字节!

7.3、当主控端读完数据时，务必回 NAK 后方可发送 STOP, 否则 iic 会通信异常!

7.4、当 CSK 芯片进入监控模式后，仅当 IRQ 为低时，IIC 接口才能正常使用。动态模式下 iic 通信没有限制。

矽源特科技
ChipSourceTek



四、CSK0xT寄存器说明

CSK 系列芯片的寄存器如下表所示:

寄存器	地址(Hex)	Write/Read	功能描述
Debounce	13h	W/R	消抖次数
ChannelEn[0~2]	14~16h	W/R	触摸通道使能
gFreq	18h	W/R	全局扫描频率
gWin	19h	W/R	全局扫描精度
gIdac	1Ah	W/R	全局扫描测量电流
gNoiseTH	1Bh	W/R	全局噪声阈值
gFingerTH	1C~1Dh	W/R	触摸阈值
gSensor	1Eh	W/R	全局灵敏度
ScanCR	1Fh	W/R	扫描控制
ButtonSR[0~2]	22~24h	R	触摸通道触摸状态
Checksum	27h	R	校验和
Rawdata	28~43h	R	触摸通道的原始信号
Diff	44~5Fh	R	触摸通道的信号量
IrqWidth	60h	W/R	中断脉冲宽度
IrqCR	61h	W/R	中断上报控制
IOMode	62h	W/R	IO口模式设置
Lnprk	63h	W/R	动态模式下长时间无触摸后重新校验基准值
Lprk	64h	W/R	动态模式下触摸最长保持时间
MonitorT	70~71h	W/R	监控模式下的主动唤醒周期
MonitorF	72h	W/R	监控模式下的扫描周期
ATMCR	73h	W/R	延迟进入监控模式的时间
ActiveF	74h	W/R	动态模式的扫描周期
gsEnMask[0~2]	78~7Ah	W/R	监控模式下的通道使能
gsNoiseTH	7Bh	W/R	监控模式下的阈值
gsFreq	7Ch	W/R	监控模式下的扫描频率
gsWin	7Dh	W/R	监控模式下的灵敏度
gsIDAC	7Eh	W/R	监控模式下的测量电流
FW_NUM	FFh	R	固件版本号

表1: 寄存器总表



Debounce@13H

寄存器	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	默认值
Debounce	13h	Debounce[7: 0]								03h

按下、抬起时的消抖次数寄存器

单位：一个动态模式的扫描周期，即总的消抖时间为：2*Debounce*ActiveF ms。

ChannelEn@14~16H

寄存器	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	默认值
ChannelEn[0]	14h	EnCh7	EnCh6	EnCh5	EnCh4	EnCh3	EnCh2	EnCh1	EnCh0	ffh
ChannelEn[1]	15h	EnCh15	EnCh14	EnCh13	EnCh12	EnCh11	EnCh10	EnCh9	EnCh8	ffh
ChannelEn[2]	16h	EnCh23	EnCh22	EnCh21	EnCh20	EnCh19	EnCh18	EnCh17	EnCh16	ffh

动态模式下的触摸通道使能寄存器

每个bit对应一个触摸通道，1：使能；0：失能。例如：

EnCh0=1：CS01使能

EnCh1=1：CS02使能

EnCh2=1：CS03使能

...

以此类推。

gFreq@18H

寄存器	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	默认值
gFreq	18h	gFreq[7: 0]								06h

动态模式下的全局扫描频率寄存器

一般设置6~63，值越小，灵敏度越高。f=5MHz/(gFreq+1)。

gWin@19H

寄存器	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	默认值
gWin	19h	gWin[7: 0]								ffh

动态模式下的全局扫描精度寄存器

Win=256*(gWin+1) 一般不做调整。

gIdac@1AH

寄存器	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	默认值
gIdac	1Ah	reserved	gIdac[6: 0]							1eh

动态模式下的全局检测电流寄存器

一般设置25~127，值越小，灵敏度越高。I=1uA*(gIdac)



gNoiseTh@1BH

寄存器	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	默认值
gNoiseTH	1Bh	gNoiseTH[7: 0]								50h

动态模式下的全局噪声阈值寄存器

gFingerTH@1C~1DH

寄存器	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	默认值
gFingerTH_L	1Ch	gFingerTH[7: 0]								c8h
gFingerTH_H	1Dh	gFingerTH[15: 8]								00h

动态模式下的触摸阈值

gSensor@1EH

寄存器	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	默认值
gSensor	1Eh	gSensor[7: 0]								32h

动态模式下的全局灵敏度寄存器
值越大，灵敏度越高。

ScanCR@1FH

寄存器	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	默认值
ScanCR	1Fh	EnSleep	EnAutoLP	reserved			Maxch	reserved	EnScan	00h

扫描控制寄存器

EnSleep:
1: 使能进入休眠模式。
0: 禁止进入休眠模式。

EnAutoLP:
1: 使能监控模式。在动态模式中检测到使能监控模式后，延时ATMCR*ActiveF毫秒后自动进入监控模式。
0: 禁止进入监控模式。

Maxch:
1: 使能只输出最强信号的触摸通道。
0: 同时输出多个已触发的触摸通道。

EnScan:
1: 使能动态扫描
0: 禁止动态扫描



ButtonSR@22~24H

寄存器	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	默认值
ButtonSR[0]	22h	ButtonSR[7: 0]								00h
ButtonSR[1]	23h	ButtonSR[15: 8]								00h
ButtonSR[2]	24h	ButtonSR[23: 16]								00h

触摸通道状态寄存器

每个bit对应一个触摸通道，1：已触发；0：未触发。例如：

bit0=1: CS01已触发；

bit1=1: CS02已触发；

bit2=0: CS03未触发；

...

以此类推。

Checksum@27H

寄存器	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	默认值
Checksum	27h	Checksum[7: 0]								c5h

触摸通道状态校验和寄存器

Checksum=ButtonSR[0]+ButtonSR[1]+ButtonSR[2]+0xC5

Rawdata@28~43H

寄存器	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	默认值
rawdata0	28h	rawdata0_H[15: 8]								-
	29h	rawdata0_L[7: 0]								-
...								-
rawdata13	42h	rawdata13_H[15: 8]								-
	43h	rawdata13_L[7: 0]								-

触摸通道的原始信号寄存器

每个通道的原始信号为两个字节，16位，高8位在前，低8位在后，以此类推。

Diff@44~5FH

寄存器	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	默认值
diff0	44h	diff0_H[15: 8]								-
	45h	diff0_L[7: 0]								-
...								-
diff13	5Eh	diff13_H[15: 8]								-
	5Fh	diff13_L[7: 0]								-

触摸通道的信号量寄存器

每个通道的信号量为两个字节，16位，高8位在前，低8位在后，以此类推。



IrqWidth@60H

寄存器	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	默认值
IrqWidth	60h	IrqWidth[7: 0]								01h

Irq脉冲宽度寄存器

irq引脚无触摸时为高电平，有触摸时根据IrqCR设置，输出低脉冲（低电平保持一段时间），单位：ms。

IrqCR@61H

寄存器	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	默认值
IrqCR	61h	IdleIrq	FingerIrq	SRChIrq	reserved					a0h

Irq上报控制寄存器

IdleIrq:

1: 扫描未使能时，周期性发中断，周期由5*ActiveT确定。

0: 失能

FingerIrq:

1: 动态模式下，有触摸时，周期性发出中断，周期由ActiveT确定。

0: 失能

SRChIrq:

1: 按键或滑条状态变化时，发一个中断。即按下、抬起时各发出一次低脉冲。

0: 失能

IOMode@62H

寄存器	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	默认值
IOMode	62h	reserved						EnPull	DP_V18	02h

I0口模式寄存器，IIC和IRQ的驱动模式。

EnPull:

1: 使能内部上拉电阻，约5.6K。

0: 失能

DP_V18:

1: 1.8V电平

0: VDD电平

Lnprk@63H

寄存器	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	默认值
Lnprk	63h	Lnprk[7:0]								0ah

动态模式下长时间无触摸后重新校验基准值，单位：min。置零可禁用该功能。



Lprk@64H

寄存器	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	默认值
Lprk	64h	Lprk[7:0]								0fh

动态模式下触摸最长保持时间，单位：秒（s）。置零可禁用该功能。

MonitorT@70~71H

寄存器	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	默认值
MoitorT_L	70h	MoitorT[7:0]								d0h
MoitorT_H	71h	MoitorT[15:8]								07h

监控模式下的主动唤醒周期，总时间为：MonitorF*MonitorT

MonitorF@72H

寄存器	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	默认值
MonitorF	72h	MonitorF[7:0]								1eh

监控模式下的扫描周期，单位：ms。

ATMCR@73H

寄存器	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	默认值
ATMCR	73h	ATMCR[7:0]								c8h

延迟进入监控模式的时间

在动态模式中，使能监控模式后，ATMCR*ActiveF 毫秒没有触摸后自动进入监控模式。

ActiveF@74H

寄存器	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	默认值
ActiveF	74h	ActiveF[7:0]								0ah

动态模式的扫描周期，单位：ms。



gsEnMask@78~7AH

寄存器	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	默认值
gsEnMask[0]	78h	EnCh7	EnCh6	EnCh5	EnCh4	EnCh3	EnCh2	EnCh1	EnCh0	ffh
gsEnMask[1]	79h	EnCh15	EnCh14	EnCh13	EnCh12	EnCh11	EnCh10	EnCh9	EnCh8	ffh
gsEnMask[2]	7Ah	EnCh23	EnCh22	EnCh21	EnCh20	EnCh19	EnCh18	EnCh17	EnCh16	ffh

监控模式下的触摸通道使能寄存器

每个bit对应一个触摸通道，1：使能；0：失能。例如：

EnCh0=1：CS01使能

EnCh1=1：CS02使能

EnCh2=1：CS03使能

...

以此类推。

gsNoiseTh@7BH

寄存器	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	默认值
gsNoiseTH	7Bh	gsNoiseTH[7:0]								32h

监控模式下的阈值

gsFreq@7CH

寄存器	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	默认值
gsFreq	7Ch	gsFreq[7:0]								09h

监控模式下的扫描频率， $f=5\text{MHz}/(\text{gsFreq}+1)$

gsWin@7DH

寄存器	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	默认值
gsWin	7Dh	gsWin[7:0]								06h

监控模式下的灵敏度，值越大越容易退出监控模式。

gsIdac@7EH

寄存器	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	默认值
gsIDAC	7Eh	gsIDAC[7:0]								1ch

监控模式下的测量电流， $I=1\mu\text{A}*\text{gsIDAC}$

FW_NUM@FFH

寄存器	地址	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	默认值
FW_NUM	FFh	FW_NUM[7:0]								-

固件版本号



五、CSK0xT应用说明

5.1、CSK0xT 的主要工作模式

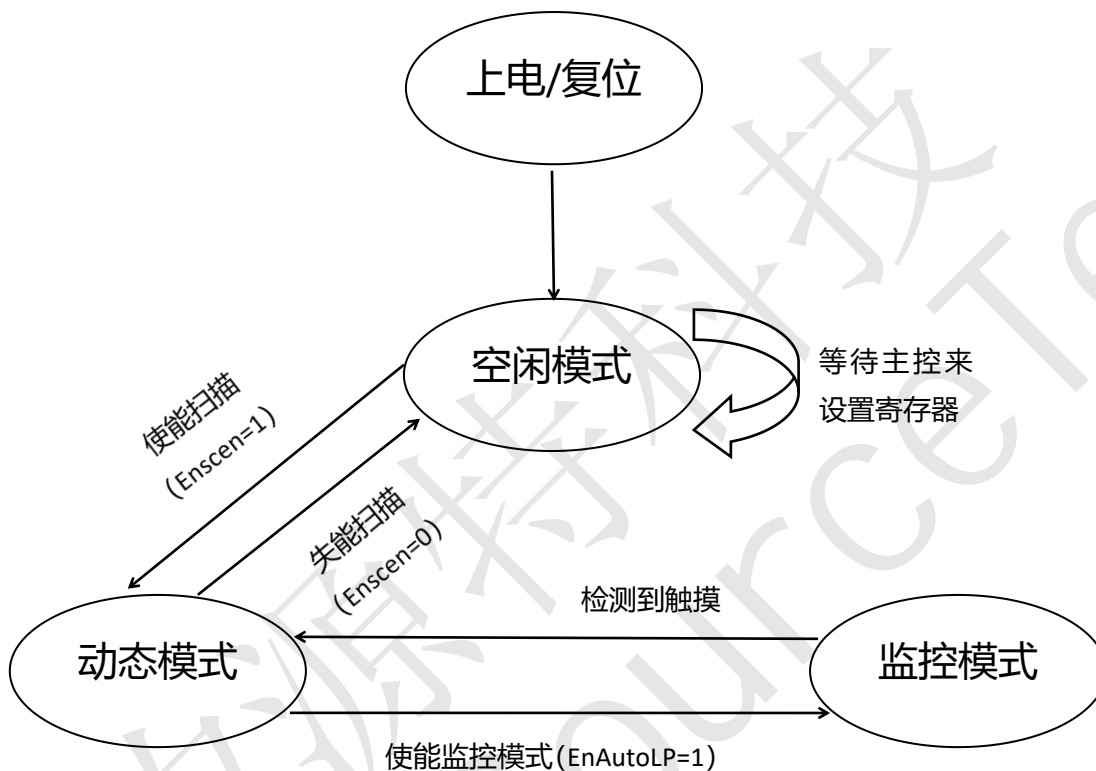


图 6: 工作模式

如上图所示，触摸 IC 上电或复位后会直接进入空闲模式。触摸 IC 在空闲模式下，会等待主控端来设置寄存器，当使能扫描后，设置的寄存器才会起作用，并进入动态模式。



5.2、IC 初始化的一般操作及说明

1、在硬件正常情况下，触摸 IC 上电 30ms 后，IRQ (int) 脚会一直周期性的发出低脉冲，周期约为 50ms (IRQ-WT)，如图 1 所示。此时主控端则可以对触摸 IC 进行参数设置。

2、设置好以下主要的寄存器后，周期性的 IRQ (int) 脚的脉冲会消失，其他寄存器可根据需求进行设置。

寄存器	说明
0x14	根据硬件所用到的通道，每一位对应一个通道。
0x15	
0x16	
0x18	全局扫描频率设置，一般设置 6~63，值越小，灵敏度越高。默认是 6。
0x19	固定是 0xFF。
0x1A	全局检测电流设置，一般设置 25~127，值越小，灵敏度越高。
0x1B	噪声门限
0x1C	信号阈值
0x1D	
0x1E	全局灵敏度设置，值越大，灵敏度越高。
0x61	IRQ 引脚模式，一般设置为 0x20。默认是 0xA0。
0x1F	使能扫描。

一般要求：

- 1) 触摸时的信号量 (diff 值)，一般取 2~3 倍的信号阈值 (触摸阈值) 为宜，具体以实际触摸感受为准。diff 值可从 44h~5fh 寄存器中读出。
- 2) 信号阈值 (触摸阈值) 要求大于等于 5 倍的最大噪声，最大噪声即是无触摸时，信号值 (diff 值) 抖动的最大幅度。
- 3) 在无触摸的情况下，连续 30 秒或以上的读取所有触摸通道的信号值 (diff 值)，选取最大值作为最大的噪声值。
- 4) 噪声门限 (1Bh) 一般取信号阈值 (触摸阈值) 的一半。

3、主控端设置好参数后，有触摸后，int 脚才会输出低脉冲，主控检测到后，可以去读取 0x22~0x24 寄存器，这三个寄存器为键值寄存器。

4、int 脚输出的低脉冲宽度是以 ms 为单位，默认为 1，即 1ms。设置 0x60 寄存可以调节 int 脚输出的低脉冲宽度，设置范围：1~5。过长的低脉冲宽度会延长整个系统周期的时间，进而降低报点率。如果主控端很难检测到 int 信号，建议改用轮询方式来读取键值寄存器！

5、触摸灵敏度的调节

一般只调节 0x1E 寄存器。如果 0x1E 调到较大 (0xF0) 后，还是不够灵敏，建议从结构上降低盖板的厚度！触摸通道的信号量 (44H<<8+45H)，即 diff 值，值越大，灵敏度越高。

设置 0x1F 寄存器的 bit2 为 1，则可以只输出最大触摸信号的通道，即：假设有 3 个通道同时触发，但只会输出最大信号对应的键值。如果产品端有组合键的操作，则 bit2 不可设置为 1！

6、用户如果中途想修改参数 (不断电再次修改参数)，需要先把 0x1F 寄存器置 0 后，延时 30ms 以上，再去设置其他寄存器，最后再把 0x1F 寄存器置回 0x01。

7、由于设置的参数都存放于 RAM 中，IC 掉电或者复位后，所设置的参数将会变为默认值！



为了防止静电、EFT 等强干扰造成异常退出扫描模式，进而造成无触。主控需要定期的去检查触摸 IC 是否处于扫描模式，即 0x1f 寄存器是否为 0x01，如若 bit0 不为 1，则需要重新设置初始化参数。

当主控接收到 int 信号后，可以从 0x1f 寄存器开始读，连同 0x22、0x23(键值)一起读回去，这样既可以检查 0x1f 的情况，又可以获取键值。

8、当触摸通道的信号量 ($44H \ll 8 + 45H$) 大于触摸阈值 ($1DH \ll 8 + 1CH$) 且稳定保持一段时间 ($13H * 74H$ 毫秒) 后，即可判定为有效按压事件。

9、设置参数的一般流程可参考以下操作：

```
#define CSK_IIC_ADDR (0x15)//7 位地址
void iic_read_data_u16(u8 iic_addr,u16 reg_addr,u8 reg_num,u8* buff,u16 len);//读数据
void iic_write_data_u16(u8 iic_addr,u16 reg_addr,u8 reg_num,u8* buff,u16 len);//写数据
//iic_addr:IIC 地址
//reg_addr: 寄存器地址
//reg_num : 寄存器长度
//buff: 数据指针
//len: 传输的字节总长度

void init_CSK_para(void)//初始化标准品参数
{
    u8 data_buff[8]={0x00}; //失能扫描
    //RST 脚拉低 delay 10ms, 再拉高。
    CSK_REST(0);
    HAL_Delay(10);
    CSK_REST(1);

    HAL_Delay(50); //unit=ms

    iic_write_data_u16(CSK_IIC_ADDR, 0x1f, 1, data_buff, 1); //写数据--禁止扫描

    HAL_Delay(20); //unit=ms

    data_buff[0]=0x1F; //5 个通道
    //data_buff[1]=0x00; //0 个通道
    //data_buff[2]=0x00; //0 个通道
    iic_write_data_u16(CSK_IIC_ADDR, 0x14, 1, data_buff, 1); //写数据

    data_buff[0]=11; //扫描分频
    data_buff[1]=0xff; //固定 0xff
    data_buff[2]=30; //扫描电流
    data_buff[3]=80; //噪声
    data_buff[4]=200; //阈值 L
    data_buff[5]=(200>>8); //阈值 H
    data_buff[6]=32; //全局灵敏度
    iic_write_data_u16(CSK_IIC_ADDR, 0x18, 1, data_buff, 7); //写数据

    // data_buff[0]=0x20;
    // iic_write_data_u16(CSK_IIC_ADDR, 0x61, 1, data_buff, 1); //写 int 模式

    // data_buff[0]=10;
    // iic_write_data_u16(CSK_IIC_ADDR, 0x74, 1, data_buff, 1); //写系统周期
```



```
// data_buff[0]=20;
// iic_write_data_u16(CSK_IIC_ADDR, (0x80 + (0) * 4 + 1), 1, data_buff, 1); //设置 CS01 通道的灵敏度

data_buff[0]=0x01;
iic_write_data_u16(CSK_IIC_ADDR, 0x1f, 1, data_buff, 1); //使能扫描

// printf("CSK0XT 标准品初始化 OK\r\n");
}
```

10、当检测到 int 信号后，可以参考以下函数，即读取寄存器的操作：

```
void get_key_data(void)
{
    if(touch_int_flg) //接收到中断后开始读按键状态
    {
        iic_read_data_u16( CSK_IIC_ADDR, 0x22, 1, read_buff, 3); //读数据
        //printf("%02x %02x %02x \r\n", read_buff[0], read_buff[1], read_buff[2]);

        touch_int_flg=0;
    }
}
```

5.3、单通道的灵敏度调试

1、实际应用中，每个触摸 pad 的大小不一定完全相同，即每个触摸通道的灵敏度（diff）会存在较大差异，所以需要单独针对某些通道设置灵敏度。

2、触摸 IC 的每个通道都可以单独设置灵敏度，所处的寄存器为： $[0x80 + (N) * 4 + 1]$ ，N 为通道序号，从 0 开始，0 即对应 CS01 通道；1 即对应 CS02 通道，以此类推。

$[0x80 + (N) * 4 + 1]$ 的值默认为 0xff，即默认使用全局灵敏度寄存器（gSensor）。当不等于 0xff 时，则使用当前的参数，触摸效果和调试方向与 gSensor 相同。

5.4、监控模式说明

1、监控模式，即在动态模式下，无触摸一定时间（ $ATMCR * ActiveF$ 毫秒）后，IC 自动进入低功耗状态。当有触摸时能立即退出，并进入到正常模式，并输出触摸键值。

2、当 IC 进入监控模式后，功耗会在 15uA 以内，此时 iic 无法正常通信。如果主控端采用了轮询的方式来读取键值，则不可以使能监控模式！

3、监控模式常用的寄存器有：触摸阈值 7BH，扫描分频 7CH，触摸灵敏度 7DH，扫描电流寄存器 7EH。一般主要调节 7DH，7EH、7CH、7BH 次之。

参数参考如下：(w 表示写，r 表示读，p 表示停止信号，0X15 表示 iic 地址)

W 0x15 0x1E 0x32 P	//配置全局灵敏度（根据调试确定）
W 0x15 0x14 0x1F 0x00 P	//使能检测通道（根据实际通道使用情况需要作相应调整）
W 0x15 0x78 0x1F 0x00 P	//配置检测通道参与低功耗扫描（建议同 14H、15H 寄存器配置）
W 0x15 0x1E 0x32 P	//配置全局灵敏度（根据调试确定）



W 0x15 0x7B 0x32 P	//配置低功耗触摸阈值
W 0x15 0x7C 0x0F P	//配置低功耗扫描分频
W 0x15 0x7D 0x0C P	//配置低功耗扫描灵敏度
W 0x15 0x7E 0x28 P	//配置低功耗扫描扫描电流
W 0x15 0x1F 0x41 P	//使能低功耗扫描，并开启扫描

7BH，一般取 30~65，默认为 50，值越大越难退出低功耗。

7CH，一般取 5~15，默认为 9，值越大越难退出低功耗。

7DH，一般取 4~20，默认为 6，值越大越容易退出低功耗。

7EH，一般取 10~60，默认为 28。值越大越难退出低功耗。

由于进入硬件低功耗模式后，iic 无法通信，这时可以尝试对地短接 IRQ 脚来强制退出低功耗，进而读取一些信号数据：

A、进入低功耗前的原始信号（无触摸）：F0H,F1H。 $((F0H<<8)+F1H)$ 要小于 2000，否则无法进入低功耗。如果大于 2000，这时可以调低 7DH 寄存器，调高 7CH、7EH 寄存器。

B、退出低功耗扫描时的原始信号（有触摸）：F4H,F5H。

C、退出低功耗扫描的高阈值： $F2H=(((F0H<<8)+F1H)+7BH)>>3$

D、退出低功耗扫描的低阈值： $F3H=(((F0H<<8)+F1H)-7BH)>>3$

当 $((F4H<<8)+F5H)>(F2H*8)$ 或 $((F4H<<8)+F5H)<(F3H*8)$ 即可触摸退出低功耗。 $((F4H<<8)+F5H)$ 应当保持在 $(F3H*8)~(F2H*8)$ 中间。

具体步骤如下：

3.1、设置低功耗参数，使能低功耗扫描，并开启扫描，经过无触摸 $ATMCR*ActiveF\ ms$ 后进入低功耗模式，使 IC 的电流稳定在 15uA 以内。

3.2、手指正常触摸着触摸区域，强行将 IRQ/int 脚短暂短接到 GND。主控立即读取 F0~F5 寄存器。

3.3、观察 F0~F5 寄存器是否符合上面的要求，如果不符合，则重新设置低功耗参数，并重复 3.1~3.3 步骤。

3.4、以上操作如果有困难，可以直接忽略；直接设置低功耗参数，并观察 IC 功耗。

4、当进入低功耗后，电流在 15uA 以下，无论怎么触摸都无响应（电流没上升）或者需要很大力气去按压才可以退出低功耗，则应该是触摸的信号远小于阈值，可调高 7DH 或调低 7EH。

5、当进入低功耗后，电流在 15uA 以下，然后立即上升，则应该是无触摸时的信号大于阈值，可调低 7DH、调高 7EH。



六、更新历史

版本	更新内容	日期
V1.0	1. 发行初始版本。	2021/10/20
V1.1	1. 完善低功耗的描述。 2. 添加寄存器的 16 进制地址描述	2022/1/5
V1.2	1. 调整 18h、1Ah 的最小值描述。 2. 在初始化说明处，加入触摸信号量、触摸阈值、噪声的选定要求。	2022/8/3