
HC89F3xx1/ HC89F30xC/HC89F1431/ HC89F24x1/HC89F34x1C系列 应用注意事项

By Home Appliance Product Line

简介

本应用笔记主要针对HolyChip的家电系列芯片在应用过程中常遇到的一些问题进行讲述及相关使用注意事项说明。

- 本应用笔记适用芯片：HC89F3xx1、HC89F30xC、HC89F1431、HC89F24x1、HC89F34x1C 系列芯片。
- 注：本文档加粗字体请详细阅读
- 相关数据手册、工具及技术文档下载网址：<http://www.holychip.cn/>。

目录

1	电源 (PWR)	3
2	时钟 (CLK)	3
3	复位 (RST)	4
4	输入/输出 (I/O)	5
5	FLASH 操作 (IAP)	7
6	定时器/计数器 (TIMER)	9
7	脉宽调制 (PWM)	10
8	模数转换 (ADC)	11
9	看门狗 (WDT)	12
10	串行外部设备接口 SPI	12
11	低功耗 (PD&IDL)	13
12	电容触摸按键 (CTK)	14
12.1	触摸软件部分开发流程建议	14
12.2	相关寄存器	14
12.3	软件抗干扰	15
12.4	常见问题分析及相关注意事项	20
13	工具使用(HC-TOOL)	28
14	版本说明	29

1 电源（PWR）

1. 为保证系统稳定性，建议在芯片 VDD、GND 之间接一个等于或大于 0.1uF 的电容。
2. MCU 正常工作电压在 2.0V~5.5V 之间，避免出现电源抖动或不在正常工作电压范围内的电压。
3. 在正常工作电压范围内，如果芯片工作电压剧烈抖动，请适当开启 BOR 消抖功能，以防止造成芯片 BOR 复位。

2 时钟（CLK）

1. Fosc 频率最高可工作在 32MHz。但 CPU 时钟不能超过 20M，若 Fcpu 高于 20MHz，程序存在跑飞的可能。最好使 Fcpu=Fosc。
2. 若 Fcpu 小于 200KHz 时，无法程序仿真。
3. 在进行 FLASH 的 IAP 擦写或者进入掉电模式之前，需要配置扩展 SFR 里 FREQ_CLK 寄存器，指明目前 CPU 时钟的频率，且 Fcpu 为 1-16MHz 范围内的正整数。如果 Fcpu 频率被修改，则需要重新配置 FREQ_CLK 寄存器指明当前时钟频率。
4. IO 输出信号的频率最高可以达到 8MHz。
5. 内部低频 RC 做 Fosc，高频 RC 可以在 Fosc 切换低频 RC 之后关闭。如果不关闭内部高频 RC，内部高频 RC 会消耗部分功耗。
6. 内部高频时钟 RC 调整
 - a) 使能 RCTRMEN 寄存器后必须立即配置 RCTRMV 寄存器。否则这个使能寄存器再执行完下一条指令后会被清零，内部高频 RC 调整就会失效。
 - b) RCTRMV 的调节曲线见数据手册电气特性章节，调节步长为 0.128MHz，内部高频 RC 调节曲线在 RCTRMV=0x3F 时出现拐点。且调整后其他模块时钟也会改变。

3 复位（RST）

外部复位和 BOR 复位消抖注意事项：掉电模式下自动关闭 BOR 消抖功能和外部 RST 消抖功能，退出掉电模式又自动打开。

外部复位注意事项：

1. 使用外部复位引脚时需要在代码选项中配置，如果 P2.4 被配置为复位脚，其端口是施密特输入上拉状态。

BOR 注意事项：

1. 掉电模式下自动关闭 BOR 消抖功能，退出掉电模式自动打开。
2. 需要进行 IAP 操作前。建议使能 BOR 功能且 BOR 电压设置为 2.4V 及 2.4V 以上，且配置合适的 BOR 消抖时间以保证 VDD 存在干扰的情况下 IAP 操作正常。

4 输入/输出（I/O）

1. MCU 上电复位后，在 7ms+option 选择等待时间（1/4/8/16ms）内，P2.0、P2.3 端口为数字输入状态，P2.2 端口为数字输入下拉状态，其他端口为模拟输入状态。正常工作时，没有这个限制。
2. **HC89F30xC/ HC89F3xx1 /HC89F1431/HC89F24x1/HC89F34x1C 系列的芯片在仿真时，P2.0/P2.3 只能作为数字输入。**
3. IO 驱动能力
 - a) HC89F3xx1 系列的芯片有四级驱动能力，配置方法对寄存器 DRENx[1:0]写入“00/01/10/11”。
 - b) HC89F30xC/HC89F1431/HC89F24x1/HC89F34x1C 系列的芯片仅有二级驱动能力，寄存器 DRENx[1:0]只能写入“00/01”，写入其他值无效。
4. 外设功能引脚全映射控制
 - a) 无映射功能引脚：大多数外设功能口可以映射到任意 I/O 口上，但 PWM 故障检测脚、ADC 输入、INTx 功能口、晶振脚等除外。
 - b) 映射引脚的输出要求：系统将禁止其多对一映射。比如：CLKO_MAP 配置为 0x01 选择 P0.1 口作为 CLKO 的输出口，T0_MAP 也配置为 0x01，这个时候硬件会按默认的优先级（引脚全映射默认优先级如[表 4-1 引脚全映射默认优先级](#)），P0.1 将配置为 CLKO 的输出口，而 T0_MAP 的配置无效。
 - c) 映射引脚的输入要求：系统将允许其多对一映射。比如：T0_MAP 配置为 0x23，则选择 P2.3 作为 T0 的输入口，T1_MAP 也配置为 0x23，这样从 P2.3 端口进入的信号同时作用于 T0 和 T1。
 - d) 如果将 TXD 和 RXD 都配置到一个端口上时，并且此端口设置为输出，则 TXD 和 RXD 将内部连接起来。

优先级顺序	复用端口功能
1	PWM3
2	CLKO
3	T0_OUT
4	T1_OUT
5	T4_OUT
6	TXD
7	RXD

8	SCK
9	MOSI
10	MISO
11	SCL
12	SDA

表 4-1 引脚全映射默认优先级

5. 端口高压注意

- a) HC89F30xC/ HC89F3xx1/HC89F1431 / HC89F24x1/HC89F34x1C 系列进行过零检测时建议用示波器看下过零检测口的电压值，如果电压值超过 VDD 电压，建议增加串联的电阻阻值，确保传入 I/O 口的电压不超过 VDD 电压。

如外围电路无法修改，建议在检测过零信号前将 I/O 口设置为数字输入状态，采集完状态后再将 I/O 设置为推挽输出低。不过此方式不够稳定，建议还是从硬件上解决此问题。

6. 流过 IO 的总电流

- a) 流过 VDD 的最大电流值在 5.0V，25°C 下须小于 100mA。
- b) 流过 GND 的最大电流值在 5.0V，25°C 下须小于 150mA。

5 Flash操作（IAP）

用户程序代码可对 FLASH 进行读、擦、写操作，作为用户更新代码或存储数据使用，为保证用户对 FLASH 操作的安全性，使用过程中请注意：

1. 只能对其他扇区进行读、擦、写操作，但无法擦除代码自身所在的扇区。
2. 如果需要写入数据，必须先擦除后再写。
3. Flash 以 128 字节为 1 个扇区，8 个扇区为 1 页，4 页为 1 个保护单位，扇区擦除时间约 5ms。
4. 在执行 IAP 操作时，不可避免的会遇到数据擦除结束后，尚未写数据就掉电的情况，所以建议采用双区域保存数据的方式，即使一个区域的数据被擦除，也可以保证另一个区域的数据被正常读取。详细内容可以在《AN0003_HA_通用系列 IAP 使用》找到。
5. 数据存放地址需要在程序存放地址之后。
6. 在进行 FLASH 的 IAP 擦写之前，需要配置扩展 SFR 里 FREQ_CLK 寄存器，指明目前 CPU 时钟的频率，FREQ_CLK 寄存器配置的值等于 CPU 时钟的频率值，最小为 1MHz。假如目前 CPU 的运行频率为 16MHz，那就配置寄存器 FREQ_CLK=0x10。建议在 IAP 擦写之前，将 CPU 时钟频率分频为整数。当 CPU 时钟频率低于 1MHz 时，不能进行 FLASH 的 IAP 擦写操作。
7. 系统在进行 IAP 操作时，CPU 进入空闲模式，此时不响应任何中断。建议在 IAP 操作前关闭中断或采用“#pragma disable”语句，确保在 IAP 操作期间不会被中断影响，待 IAP 操作完成后，再将中断恢复。如果在 IAP 操作前并未关闭中断。在配置 IAP 操作时，产生一个中断，会导致对 Flash 操作失败。
8. 系统的正常工作电压是 2.0V~5.5V，当系统电压低于该电压范围时，可能会导致 IAP 误操作。建议此时使能 BOR 功能且 BOR 电压设置为 2.4V 及 2.4V 以上，且配置合适的 BOR 消抖时间以保证 VDD 存在干扰的情况下 IAP 操作正常。
9. IAP 读擦写 FLASH 保护
 - a) IAP 通过 MOVC 指令来读 FLASH，IAP 读保护以 4K 字节为单位。
 - 如果一个 4K 字节空间设置了读保护，其他 4K 字节空间的 MOVC 指令读这个 4K 字节空间，读出来的数据为零，但这个 4K 字节空间的 MOVC 指令可以读取自身的数据。
 - 如果一个 4K 字节空间读保护被使能，但被允许擦除与写入，则可先擦除后获得该 4K 字节空间的读允许，直至复位或掉电。
 - b) IAP 的擦写保护以 4K 字节为单位，IAP 擦写之前需要先看相应扇区的擦写保护是否使能，没有使能擦写保护才能进行 IAP 的擦写。

- c) 在 Option 中设置相关的 IAP 擦写保护，使能 IAP 操作程序所在扇区保护位，可以有效的保证程序区不会被改写或误擦除。
 - d) IAP 的读擦写保护通过上位机软件来配置，详细情况请参见《HC-LINK 用户手册》。
 - e) 当 code 修饰的变量所在的 4K 存储区被 IAP 读保护后，非本保护区的代码无法正常读取这个被 code 修饰变量。
10. Option 的 Memory mode 设置

- a) 当将 Memory Mode 设置为非“Small: variables in DATA”（Memory Mode 设置方法如图 5-1 Memory mode 设置），此时定义的变量默认存储在拓展 RAM 中。单片机访问拓展 RAM 所需要的时间要比访问内部 RAM 时间长。而使用 IAP 命令对 Flash 解锁，经过 22 个 CPU 时钟后 Flash 会自动锁定，锁定后无法再对 Flash 进行擦写。当定义的变量的存储位置为拓展 RAM 时，将数据备份到 Flash 的过程中可能会因为单片机访问拓展 RAM 时间较长导致数据保存失败。建议将定义的变量的位置设置为内部 RAM，可以避免这样的情况。

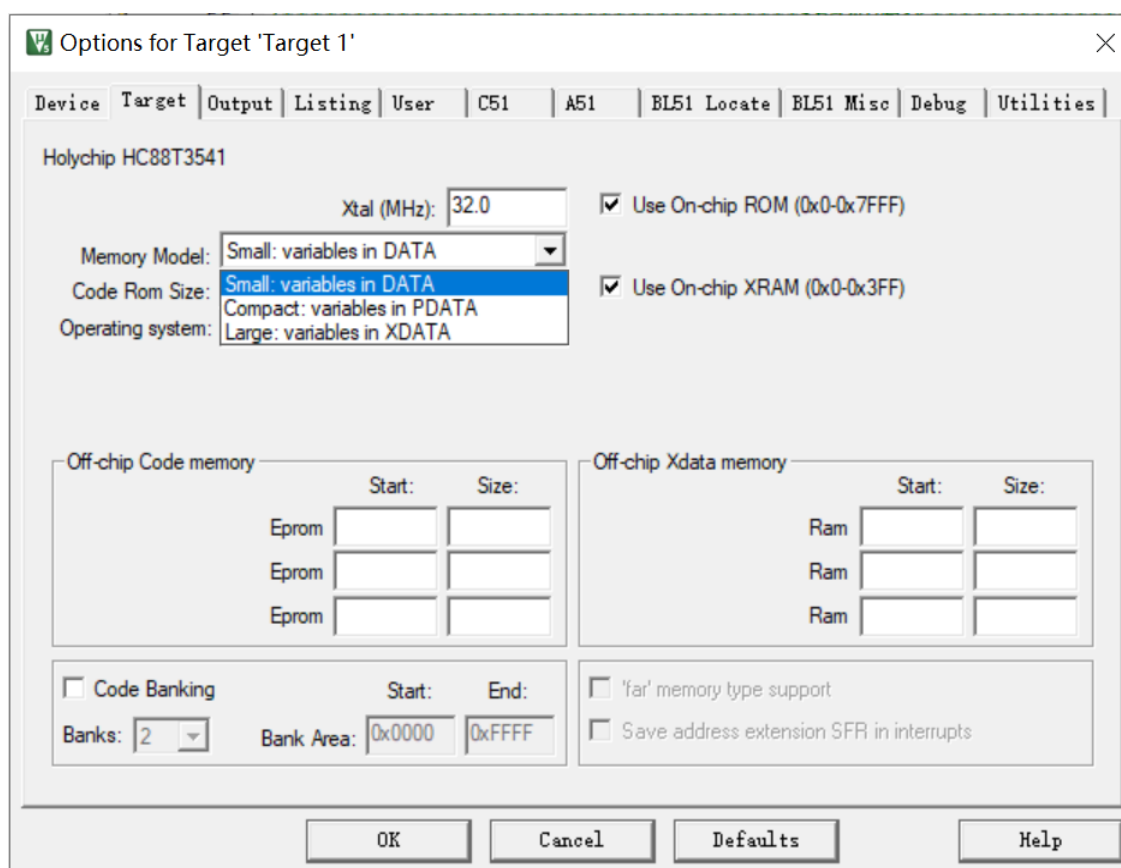


图 5-1 Memory mode 设置

6 定时器/计数器（Timer）

1. 定时器/计数器 T0

- a) 方式 3 时 T0 占用 T1 的 TR1、TF1 及中断源，由于 TR1 被 T0 占用，此时需要关闭 T1 可将 T1 设为工作方式 3。
- b) 在方式 0（Mode0）时 THx(x = 0,1)和 TLx(x = 0,1)读写操作遵循先高位后低位。并且在修改重载数据时，低位无论有无改变，只要高位被修改，低位都必须再写入一次，方式 1、2、3 时无此要求。

2. 定时器/计数器 T3

- a) T3 外部时钟输入口必须设置为数字输入。
- b) T3 时钟使用内部时钟时，T3 无法工作在掉电模式下。
- c) 必须在使能前给 TH3，TL3 赋值。
- d) 当 T3 计数时钟源选择端口输入外部时钟，TR3 和 T3CLKS 的配置需要同时进行，即使用一条指令完成配置。

3. 定时器/计数器 T4

- a) T4 外部时钟必须输入一个方波。
- b) T4 外部时钟输入口必须设置为数字输入。
- c) 不可以在中断中判断 T4 中断标志位。

4. 定时器/计数器 T5（仅 HC89F3xx1 系列的芯片有此模块）

- a) T5 设置为捕获功能，需要配置 I/O 章节里面的复用端口 T5_MAP 寄存器，将 timer5 配置到任意的 IO 上面并且对应 I/O 必须设为输入模式。
- b) 当捕获低频内部 RC 或 RXD 引脚时，也需要使能 EXEN5，同时还要配置上升沿捕获或下降沿捕获。

7 脉宽调制 (PWM)

1. 单路 8 位 PWM3 模块

- a) 关闭 PWM3 模块时, PWM 计数停止, 输出立即关闭。再打开 PWM3 模块时, PWM 计数器都重新从 1 开始计数, 输出受 PWM3OEN 控制。
- b) PWM3 允许输出, 必须在 PWM3EN 置 1 下才有效, 否则为 PWM3 输出关闭状态 (输出时对应端口必须设为输出模式); 即使都禁止输出, 只要相关位被使能, PWM3 都可以溢出中断, 即此时 PWM3 可以作为定时器/计数器使用, 此控制位修改立即生效。

2. 12 位 PWM 模块

- a) 互补输出模式及独立输出模式都可受故障检测脚控制。
- b) 修改 PWMx 工作模式时建议先关闭 PWMx 模块 (x=0、1、2)。
- c) 关闭时, PWMx 计数停止, 输出立即关闭。打开时, PWMx 计数器都重新从 1 开始计数, 输出受 PWMx_OEN 和 PWMx1_OEN 控制 (x=0、1、2)。
- d) PWMx 允许输出, 必须在 PWMx_EN 置 1 下才有效, 否则 PWMx 输出关闭状态 (输出时对应端口必须设为输出模式); 即使都禁止输出, 只要 PWMx_EN 位被使能, PWMx 都可以溢出中断, 即此时 PWMx 可以作为定时器使用, 此控制位修改立即生效 (x=0、1、2)。
- e) 互补模式 PWMx 的有效期间为占空比期间, PWMx1 的有效期间为 PWMx 占空比的互补期间 (x=0、1、2)。
- f) 修改 PWMx 周期时先修改高位, 后修改低位, 读时不受限制。

➤ 例如:

PWMxPH = 0x05;

PWMxPL = 0x08; //此时 PWM 计数器溢出, 则下一个周期开始周期计算数据 0x0508

PWMxPH = 0x06; //此时 PWM 计数器溢出, 则下一个周期开始周期计算数据 0x0508

PWMxPL = 0x08; //此时 PWM 计数器溢出, 则下一个周期开始周期计算数据 0x0608

PWM1PL = 0x09; //此时 PWM 计数器溢出, 则下一个周期开始周期计算数据 0x0609

显然只要修改 PWM 周期, 无论低位寄存器是否需要修改, 低位都必须写入一次, 且周期修改都只会在下一个 PWM 周期才会生效 (PWM 关闭时修改不受限制)。

8 模数转换（ADC）

硬件注意事项：

1. 尽量保证系统电压的稳定性，如 VDD 和输入电压的动态噪声。
2. ADC 采样输入阻抗尽量小，建议输入阻抗<10K Ω 。
3. 注意 IO 之间的串扰信号，如相邻端口或 ADC 走线是否有高压（高于 VDD 电压）、高频信号等。

软件注意事项：

1. ADCEN 置 1 或通道切换后需延迟 20us 再启动 ADC 转换。
2. 启动转换时，ADCIF 需要先软件清 0，ADCIF 位为 1 时，置 ADCST 不能启动新的转换。在转换过程中，若 ADCST 位软件清 0 将终止转换。
3. 在进行内部通道选择时，需要注意：HC89F3xx1/HC89F1431/HC89F24x1/HC89F34x1C/HC89F30xC 系列的芯片在使用内部 1/4VDD 检测通道时，外部通道选择 XCHS[3:0]无论写入什么值，都会造成使能的外部通道与内部通道同时打开的情况，但只要使能的外部通道对应的 I/O 口浮空输入且外围无电路则内部通道可正常使用（需舍弃一个带有 ADC 通道的 I/O 口）。
 - a) 如用户需要采集内部 1/4VDD 电压，ADCC1 写入值为 0x4E（01001110B），使能的外部通道为 AN14，那么 AN14 对应的 P2.6 口设置为浮空输入且无外围电路，则采集的内部通道值 1/4VDD 为正确值。如 P2.6 口未设置为浮空输入或存在外围电路，那么采集的内部通道值 1/4VDD 的值不正确。
4. ADC 外部通道除设置 XCHS[3:0]，还需设置对应管脚的功能为模拟输入。
5. 掉电模式下，ADC 停止工作。
6. 系统进入掉电模式前，ADC 参考电压选择。HC89F3xx1/HC89F1431/ HC89F24x1/HC89F34x1C /HC89F30xC 系列的芯片在系统进入掉电模式前，建议将 ADC 参考电压选择非 VDD，可以进一步降低系统功耗。
7. 内部参考电压选择 2V 时，VDD 工作电压须高于 2.7V。内部参考电压选择 3V/4V 时，VDD 工作电压须高于内部参考电压 0.5V 以上。
8. ADC 转换频率尽量不要大于 1MHz。
9. 对 ADC 采样结果软件增加适当的滤波。
10. 一般在电机启动、引擎点火等时，会对 ADC 输入电压造成一定的干扰，建议 ADC 的启动采样避开电压抖动时刻。

11. 为精确定位问题，建议在程序中只启动 ADC，验证是否程序中是否有其他动作影响 ADC 采样效果。

9 看门狗（WDT）

1. 看门狗定时器是一个递增计数器，其时钟源为独立的内部低频 RC 时钟（44KHz），可以通过寄存器选择在空闲/掉电模式下是否运行。
2. 因 WDT 时钟（内部低频 RC）允许偏差为±50%，所以 WDT 时钟源未进行出厂前校准前，不可直接作为精确定时使用。
3. WDT 看门狗复位模式。使用 WDT 复位时需要在代码选项中配置取消“看门狗复位禁止”。
4. WDT 定时器模式。使用 WDT 做定时器时需要在代码选项中配置勾选“看门狗复位禁止”。
5. 在使用 WDT 进行掉电模式唤醒时请注意：HC89F3xx1/HC89F1431/ HC89F24x1/HC89F34x1C 系列/HC89F30xC 系列的芯片在清狗动作的指令和进入掉电模式的指令需要间隔大于等于 3 个 wdt_clk 的周期（约 100us）。

10 串行外部设备接口SPI

1. SSIG=0&CPHA=0 时，数据在 SS 为低被驱动；CPHA=1 时，数据在 SCK 的前时钟沿驱动。
2. 接收为双 BUFF，接收溢出发生在第二个数据传送完成前仍未清除之前接收数据产生的 SPIF 志，故每次准备接收下一个数据前必须先清除 SPIF，否则 RXOV 将置 1，RXOV 置 1 不会影响 SPI 接收。
3. 当 OSC_CLK 的频率 Fosc 和 CPU 的频率 Fcpu 不一致时，标志位 WCOL 会错误置位，但不影响正常的 SPI 通信。

11 低功耗（PD&IDL）

1. 空闲模式能够降低系统功耗，在此模式下，程序中止运行，CPU 时钟停止，但外部设备时钟可继续运行。
2. 掉电模式可以使 MCU 进入功耗非常低的状态。掉电模式将停止 CPU 和外围设备的所有时钟信号，但如果 WDT 和 TIMER3（此时 TIMER3 时钟源为外部时钟）使能且允许在掉电模式下工作，则 WDT 和 TIMER3 模块将继续工作。
3. 若同时置 PD&IDL，系统将进入掉电模式，唤醒后标志被同时清除。且退出掉电模式后，CPU 也不会进入空闲模式。
4. 在涉及低功耗的操作时，配置完 Fcpu 后需要配置 **FREQ_CLK** 寄存器指明当前时钟频率，时钟频率如低于 1MHz 时，则配置为 0x01。如果 Fcpu 频率被修改，则需要重新配置 FREQ_CLK 寄存器指明当前时钟频率。
5. **BOR** 上电时默认打开，进入掉电模式前可将其关闭，进一步降低功耗。
6. 系统进入掉电模式前，ADC 参考电压选择：**HC89F3xx1/HC89F30xC/HC89F1431**
/HC89F24x1/HC89F34x1C 系列的芯片在系统进入掉电模式前，建议将 ADC 参考电压选择非 **VDD**，可以进一步降低系统功耗。
7. 进入低功耗前，建议将 IO 设置为固定状态（如：推挽输出低/高、数字输入带上拉/带下拉）或模拟输入，可以进一步降低系统功耗。
8. 掉电模式下 MCU 自动关闭 BOR 消抖功能和外部 RST 消抖功能，退出掉电模式又自动打开。
9. 在使用 **WDT** 进行掉电模式唤醒时请注意：**HC89Fxx1/HC89F30xC/HC89F1431 / HC89F24x1**
/HC89F34x1C 系列的芯片在清除动作的指令和进入掉电模式的指令需要间隔大于等于 3 个 **wdt_clk** 的周期（约 100us）。
10. 内部低频 **RC** 时钟频率存在偏差，因此使用 **WDT** 设置的定时唤醒时间会有偏差。

12 电容触摸按键（CTK）

12.1 触摸软件部分开发流程建议

用户在使用触摸产品开发时，建议在程序中先调试触摸部分，其余功能程序待触摸调试完成后添加。这样的好处是：

- 反映最真实的触摸硬件环境，如电源、PCB 布板对触摸的影响有多大，可以做出最佳的参数设置。
- 触摸部分完成后，在此基础上添加其余功能程序，可以减少过程中的调试时间。

12.2 相关寄存器

下图为触摸库使用时，需要配置的相关寄存器。

```
/******设置触摸相关寄存器******/
// 设置数据采样时钟个数 数据采样时钟个数 = (HCTouchLib_DSCR+1)*256 - 1
#define HCTouchLib_DSCR 48
// 设置CTK采样频率 0->Fosc/2, 1->Fosc/4, 2->Fosc/8, 3->Fosc/16, 4->Fosc/32, 5->Fosc/64, 6->Fosc/128
// 注：单个通道扫描时间 ≈ ((HCTouchLib_DSCR+1)*256 - 1)/HCTouchLib_CTKDS
#define HCTouchLib_CTKDS 0
// 设置充放电频率 0->Fosc/2, 1->Fosc/4, 2->Fosc/8, 3->Fosc/16, 4->Fosc/32, 5->Fosc/64, 6->Fosc/128, 7->Fosc/256
// 注：建议设置CTK采样频率 >= 2*充放电频率
#define HCTouchLib_PRSDIV 2
// 设置充电电源电压 0->2v 1->2.5v 2->3v 3->4v
// 注：必须保证IC VDD > 设置充电电源电压+0.5V
#define HCTouchLib_VD 2
// 设置放电电阻 0~7, 0->放电电阻最小, 7->放电电阻最大, 方向0(小)-->7(大), 放电电阻越小, 计数值越小
#define HCTouchLib_RBS 6
// 设置CTK扫描补偿是否打开 0->关闭, 1->打开 一般使用关闭即可
#define HCTouchLib_OFFSET_SW 1
/*******/
```

图 12-1 需要配置的相关寄存器

12.2.1 充电电源电压（HCTouchLib_VD）

充电电压共有 4 档，分别为 2V、2.5V、3V、4V。充电电压越高，触摸计算值越大。但必须保证 MCU 的 VDD 电压>所设的充电电压+0.5V，否则会导致触摸计数值不正常。

举例：芯片 VDD 的正常工作电压为 5V，在驱动外部器件时导致 VDD 电压有抖动，且最低电压可到 4V，那用户所设的充电电压不能选为 4V，应选择为 3V、2.5V 或 2V。

12.2.2 CTK 扫描补偿（HCTouchLib_OFFSET_SW）

CTK 扫描补偿为应对某些特殊情况所设，用户在配置时必须将其配置为关闭状态，即如下配置

```
#define HCTouchLib_OFFSET_SW 0
```

注：部分版本存在此选项，如使用的库中无此选项可忽略。

12.3 软件抗干扰

强悍的抗干扰性能离不开硬件、软件的合理设计。本节主要介绍在软件上可以有效提高抗干扰性的方法。

对芯片的干扰路径主要是 MCU 的电源口、IO，不论干扰源是什么，表现在触摸的计数值上则为不正常的跳变，表现在芯片的程序运行上就是跑飞或者死机。因此，在软件上主要针对这两点来进行处理。

- MCU VDD 的有效监测
- BOR 设置
- WDT 的合理使用
- IO 口的状态处理
- 触摸阈值的合理设置
- 触摸基线更新的及时性、有效性
- 触摸按键的消抖

12.3.1 MCU VDD 的有效监测

如果在应用环境不可避免会导致 MCU 的 VDD 产生较大的抖动，系统应及时监测到该情况，以便程序作出应对处理操作（VDD 抖动造成的影响详见本文 12.4.10 章节 [《芯片 VDD 变化注意事项》](#)）。

芯片触摸产品提供可以检测 VDD 的方式有：LVD（低电压监测电路）、ADC（内部 1/4VDD 检测通道），用户可在不增加外围电路的情况下使用芯片内部资源完成对 VDD 的监测。使用 ADC（内部 1/4VDD 检测通道）检测 VDD 时请先阅读本文 12.4.14 章节 [《通过 ADC 内部通道检测 VDD 电压注意事项》](#)。

12.3.2 BOR 设置

BOR 为芯片内部低电压复位电路，当 VDD 电压下降到 V_{BOR} 以下，且持续时间超过 T_{BOR} 时，系统产生欠压复位。BOR 复位时，RSTFR 寄存器的 BORF 位将被置 1，用户可以判断此标志以来确定是否发生 BOR 复位。

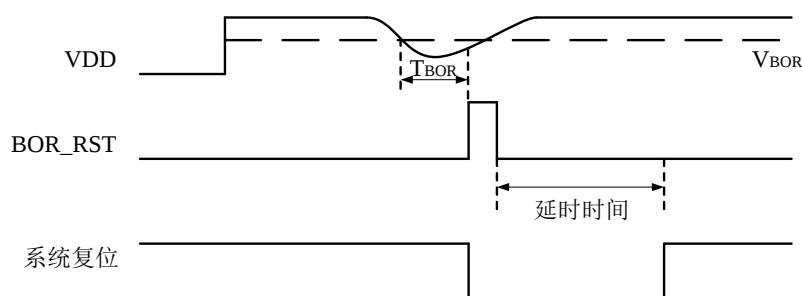


图 12-2 BOR 复位时序图

用户可以通过代码选项或寄存器来选择 BOR 检测的电压档位，客户在使用时可根据情况选择合适的 BOR 检测电压。BOR 档位：4.2V/3.9V/3.6V/3.0V/2.6V/2.4V/2.0V/1.8V。

T_{BOR} 也可以通过寄存器配置，用来进行电压消抖，从而提高或者降低 BOR 对 VDD 电压响应的灵敏度。原则上要根据用户实际电源情况进行 BOR 档位和消抖时间的设置。

12.3.3 WDT 设置

为了防止系统在异常情况下受到干扰，MCU 程序跑飞，导致系统长时间异常工作，通常是引进看门狗，如果 MCU 程序中不在规定的时间内按要求喂狗，就认为 MCU 处于异常状态，看门狗就会强制 MCU 复位，芯片重新从 0000H 开始运行。

WDT 复位功能可以通过 OPTION 进行禁止，WDT 复位功能禁止后可以使用 WDT 中断唤醒掉电模式。

在使用 WDT 进行掉电模式唤醒时请注意：**清狗动作的指令和进入掉电模式的指令需要间隔大于 3 个 wdt_clk 的周期（约 100us）。**

12.3.4 IO 口的状态处理

IO 口作为 MCU 和外部交互信息的主要窗口，很多干扰的路径也是通由 IO 口进入 MCU 内部，因此在对 IO 口处理上应格外注意。

在使用时请遵守以下注意事项：

- IO 口不要接入高于芯片 VDD 或低于芯片 GND 的电压，如需接入请参考本文 12.4.7 章节 [《端口过零信号处理》](#)。
- 未使用到的 IO 口应设置为固定状态：输入带上/下拉或推挽输出高/低。
- 作为 CTK 检测口时，需要将 IO 口设置为**模拟输入状态**
- 在不影响正常应用功能的前提下，对于做输入功能的 IO 口，当程序中未处理到该 IO 口时，应将该 IO 口设为固定状态，当做输入检测功能前再将 IO 口设为输入模式。

举例：程序中，每隔一段时间检测某 IO 口端的电平状态，针对类似的功能要求，可以在检测该 IO 口电平判断完成后即将该 IO 口设为输出模式，当要进行检测时再将该 IO 口设为输入模式。

12.3.5 触摸阈值的合理设置

如下图红色通道所示，用户在设置触摸阈值时应根据实际应用环境中，手指触摸后的 Differ 值情况来设置。

下图中，手指触摸后 Differ 值约为 230，为保证灵敏度和可靠性，用户可将触摸阈值设为 190，触摸迟滞设为 20。如此设置的意思是：当手指触摸后 Differ 大于 210（190+20）后，则认为是有有效触摸；当 Differ 小于 170（190-20）则认为无效触摸。

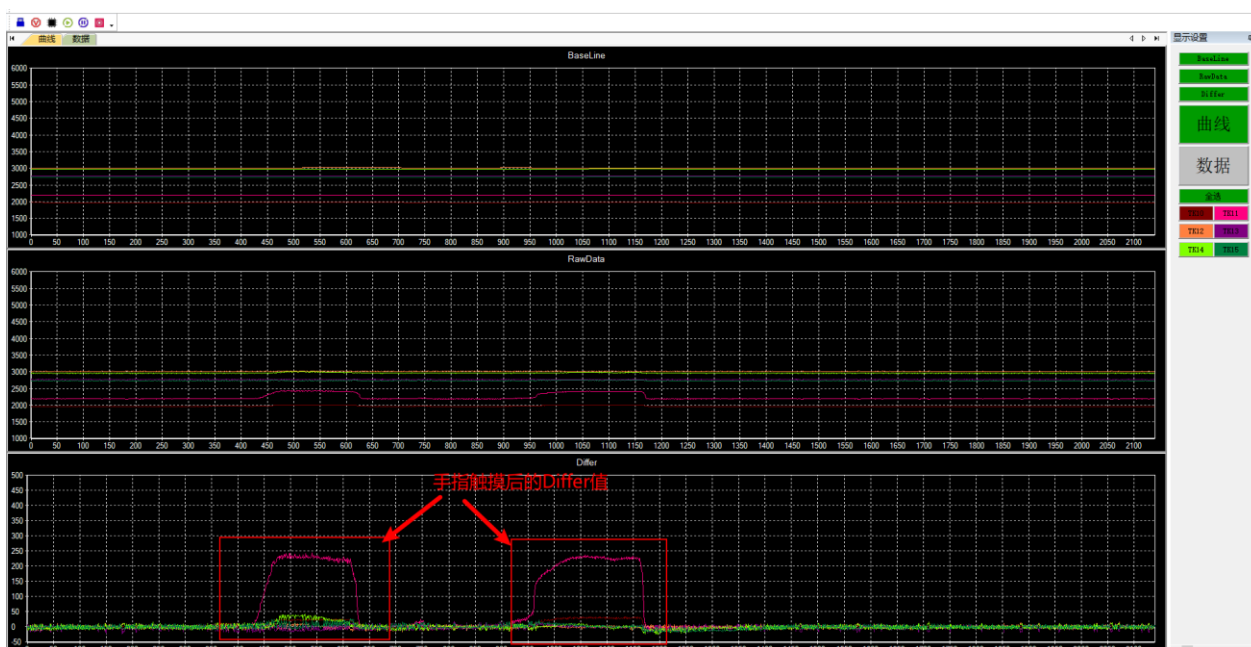


图 12-3 手指触摸按键 Differ 值

切不可随意设置该值，否则会造成灵敏度过高导致误触发情况，应通过触摸调试工具查看实际触摸值变化。

12.3.6 触摸基线更新的及时性、有效性

不同用户对于触摸基准线更新速度有不同要求，因此设置一个合适的触摸基准线更新时间对于用户来说极其重要。关于触摸基准线更新的原理及时间计算公式本应用笔记将不做具体阐述，用户可以参考《AN0101_HA_HC89F3xx1_HC89F30xC 系列按键触摸库使用》的 3.3.2 章节《基准线更新》。

12.3.6.1 触摸自动复位计数值及低基准线复位计数值

触摸自动复位计数值原理：当 $\text{Differ} \geq \text{正噪声}$ （用户设置的 $\text{HCTouchLib_TKx_N_THRESHOLD}$ 正值）时，内部计数器开始累加，如果累加值超过设置值时， $\text{Baseline} += \text{Differ}$ 。即按键按下时长超过设置的触摸自动复位计数值（该值对应的是时长），基线会更新，按键表现为无效，此值根据用户需求配置。

低基准线复位计数值原理：当 $\text{Differ} \leq \text{负噪声}$ （用户设置的 $\text{HCTouchLib_TKx_N_THRESHOLD}$ 负值）时，内部计数器开始累加。如果累加值超过设置值时， $\text{Baseline} += \text{Differ}$ 。即如果 Differ 小于负噪声值且时长超过设置的低基准线复位计数值（该值对应的是时长），基线会更新，此值根据用户需求配置。

举例：如下图所示，整体一共分为五个阶段，对应状态描述详见表 12-1

- 触摸自动复位计数值参数对应时长记为 $T_{\text{自动复位}}$
- 低基准线复位计数值参数对应时长记为 $T_{\text{低基准线复位}}$



图 12-4 自动复位和低基准线复位

阶段	按键状态	对应值	状态描述
阶段一	无效	Baseline = 2850; RawData = 2850; Differ = 0;	本阶段无按键按下，为正常状态，Baseline 不更新，Differ 为 0。
阶段二	有效	Baseline = 2850; RawData = 3120; Differ = 270;	手指持续按下，Differ ≥ 触摸阈值+迟滞，且时长 < T _{自动复位} ，按键有效，Baseline 不更新。
阶段三	无效	Baseline = 3120; RawData = 3120; Differ = 0;	手指持续按下，由于按下时长已经超过 T _{自动复位} ，Baseline 更新，Baseline 更新后 Differ 为 0，按键无效。
阶段四	无效	Baseline = 3120; RawData = 2850; Differ = -270;	手指离开，Differ 值 ≤ 负噪声，且时长 < T _{低基准线复位} ，Baseline 不更新，Differ 为负值，按键无效。
阶段五	无效	Baseline = 2850; RawData = 2850; Differ = 0;	手指离开，由于时长已经超过 T _{低基准线复位} ，Baseline 更新，整体为正常状态。

表 12-1

12.3.6.2 毛刺消除计数次数

毛刺消除计数次数需和判断为毛刺的条件（HCTouchLib_LIMIT_FACTOR）参数配合使用。

- HCTouchLib_TKx_F_THRESHOLD* HCTouchLib_LIMIT_FACTOR 记为 NUM_{MAX}
- HCTouchLib_OVERHIGH_TIME（毛刺消除计数次数）对应时长记为 T_{毛刺消除}

可以将毛刺理解为一个上限值，如果 Differ ≥ NUM_{MAX} 则判断为无效按键。并且如果毛刺持续时间超过 T_{毛刺消除}，会更新 Baseline，更新原理可以参考触摸自动复位计数值。

12.3.6.3 噪声桶容量大小

- 内部定义一个变量为 Num_{虚拟噪声桶}

当负噪声 < Differ < 正噪声时，Num_{虚拟噪声桶} += Differ，如 Num_{虚拟噪声桶} ≥ 正噪声桶容量或 Num_{虚拟噪声桶} ≤ -负噪声桶容量，Baseline ± 2。

本参数主要用于微调 Baseline，当外界出现温度、湿度等变化时，Differ 值可能会有一些微小变化，此参数可以用来微调这些变化。建议将此值设置为噪声的 5-10 倍，用户可以根据自己的需求调整本值。

12.3.7 触摸按键的消抖

目前版本的库函数中，有两个参数与按键消抖有关，分别是确认手指触摸次数和确认手指离开次数。

- 确认手指触摸次数参数对应时长记为 T_{确认触摸}

- 确认手指离开次数参数对应时长记为 $T_{\text{确认离开}}$

确认手指触摸次数：当 $\text{Differ} \geq \text{触摸阈值} + \text{迟滞}$ ，且时长 $\geq T_{\text{确认触摸}}$ ，对应通道的 TouchKeyFlag 标志位置 1。

确认手指离开次数：当 $\text{Differ} \leq \text{触摸阈值} - \text{迟滞}$ ，且时长 $\geq T_{\text{确认离开}}$ ，对应通道的 TouchKeyFlag 标志位清 0。

这两位参数主要用于按键按下和离开的消抖时间，常规情况下，按键消抖时间与机械按键相似，消抖会带来稳定性，也会减少按键响应灵敏度。按键计算方式与 12.3.6 章节中的几个参数计算方式相同，用户可以参考《AN0101_HA_HC89F3xx1_30xC 系列按键触摸库使用》的 3.3.2 章节《基准线更新》。

12.4 常见问题分析及相关注意事项

12.4.1 测试或使用时的串键

问题情景：

有触摸按键 K1、K2，在触摸按键 K1 时，却触发了 K2 按键或者再去触摸 K2，K2 表现为无效状态。

问题分析：

- ? K1、K2 按键的检测线在 PCB 上相邻过近。手指触摸某个按键时，因为和其他通道相邻过近，电容效应会影响到其他通道，导致串扰现象。
 - ✓ 解决办法：修改 PCB，保证相邻触摸检测线之间的距离。
- ? 如果在配置中打开了 CTK 扫描补偿（即 `#define HCTouchLib_OFFSET_SW 1`），也有可能造成该问题的发生。因为 CTK 扫描补偿打开以后，按下 K1 按键后，K2 按键值会下降（这是芯片本身的一个缺陷）过了一段时间以后，K2 Baseline 会更新（下降），K1 松开后，K2 RawData 会上升，此时 K2 Differ 值有可能会超过触摸阈值，按键一直有效，导致再按 K2 会无效。
 - ✓ 解决办法：关闭 CTK 扫描补偿（即 `#define HCTouchLib_OFFSET_SW 0`）
- ? 用户检查自己程序，是否有对触摸按键判断不合理的代码存在。

12.4.2 上电触摸无反应

问题情景：

应用板在量产测试时，如果手指先放在触摸按键弹簧上，然后上电，此时触摸按键，发现触摸无任何反应，要等一段时间才能正常触摸。

问题分析：

- ? 此类问题发生的原因是手指按在触摸按键上再上电后，由于触摸库在上电后初始化时将此时的触摸计数值经过基线处理算法后设为初始基线，很明显，这种情况下的基线是高于正常值的。

等一段时间才能正常触摸，是因为触摸库对基线进行实时更新，但这个更新过程需要一定的时间（此时间为用户设置）。

- ✓ 解决办法：参考本文 12.3.6 章节 [《触摸基线更新的及时性、有效性》](#) 将触摸低基准线复位计数值设置的更小一些，建议设置的时间为 0.5s。

12.4.3 灵敏度不够（带产品外壳）

问题情景：

触摸按键时灵敏度较低，有时需要长时间的按或者用大拇指按才有效。

问题分析：

- ？ 没有设置一个合适的触摸阈值，不同人或者不同外界情况的电容值并不相同，也许调试时手指测试灵敏度是正常的，但是换了一个人可能电容量较小，所以按下去无反应。

解决办法：参考本文 12.3.5 章节 [《触摸阈值的合理设置》](#) 以及《AN0101_HA_HC89F3xx1_

- ✓ HC89F30xC 系列按键触摸库使用》3.3.1 章节的《阈值设置》中提供的参考值设置一个合适的触摸阈值。另外为保证不同人测试的环境相同，建议调试时使用 10mm 的手持铜柱。

- ？ 没有设置一个合理的触摸按键消抖时间，手指需要按下一段时间后才会确认为按键有效。

- ✓ 解决办法：参考本文 12.3.7 章节 [《触摸按键的消抖》](#)，设置一个合理的按键消抖时长，注意此时长除了与确认手指触摸次数有关，也与 CTK 扫描时间等有关，计算方式详见

《AN0101_HA_HC89F3xx1 _ HC89F30xC 系列按键触摸库使用》。

- ？ 除了库函数中的按键消抖，用户程序中额外增加了按键消抖处理。

- ✓ 解决办法：检查用户程序，删除不必要的按键消抖处理。

12.4.4 灵敏度不够（不带产品外壳）

问题情景：

触摸按键时灵敏度较低，按键按下去无反应，或按的速度比较慢的时候有效，按的比较快的时候无效。

问题分析：

- ？ 设置了毛刺判断的条件，手指按下去以后， $\text{Differ} > \text{HCTouchLib_LIMIT_FACTOR} * \text{HCTouchLib_TKx_F_THRESHOLD}$ ，所以按键判断为无效。

- ✓ 解决办法：设置为不消除毛刺，或者确保消毛刺的值大于按键按下的 Differ 值。

- ? 由于没有接外壳，手指在举例触摸 PAD 较远时便被判断为有效，当按键尚未判断为无效时，手指又按了下去，所以表现为按键无效。
- ✓ 解决办法：测试时放上外壳或者测试时每次按下时手指离开远一些，确认按键已经释放再按下。

12.4.5 按键误触发或太过灵敏

问题情景：

使用时发现手指没有按到按键，按键就触发了，或者按键太过灵敏，手指距离很远时就触发了按键。

问题分析：

- ? 触摸阈值设置的过低，导致手指离得很远，或者外界一个意外的干扰就触发了按键。

解决办法：参考本文 12.3.5 章节 [《触摸阈值的合理设置》](#) 以及《AN0101_HA_HC89F3xx1_

- ✓ HC89F30xC 系列按键触摸库使用》3.3.1 章节的《阈值设置》中提供的参考值设置一个合适的触摸阈值。
- ? 按键确认次数设置对应时长过短，导致偶然性的一个毛刺便触发了按键
- ✓ 解决办法：参考本文 12.3.7 章节 [《触摸按键的消抖》](#) 中提供的参考值设置一个合适的按键确认次数。

12.4.6 Differ 出现周期性高毛刺

问题情景：

Differ 出现周期性高毛刺，且此时外界无干扰。

问题分析：

- ? $F_{osc} \neq F_{cpu}$ ，且设置了中断优先级。

F_{osc} 及 F_{cpu} 介绍：以下以 HC89F3541 为例（HC89F3xx1 系列、HC89F1431、HC89F24x1、HC89F34x1C 系列与 HC89F30xC 系列均适用），如图下图所示，HC89F3541 内部有一个 32M 高精度时钟，通过时钟选择寄存器 CLKSWR 可以设置外设时钟，即 F_{osc} 时钟， F_{osc} 用于系统各外设使用， F_{osc} 经过时钟分频寄存器 CLKDIV 分频后作为 CPU 时钟，即 F_{cpu} 时钟。

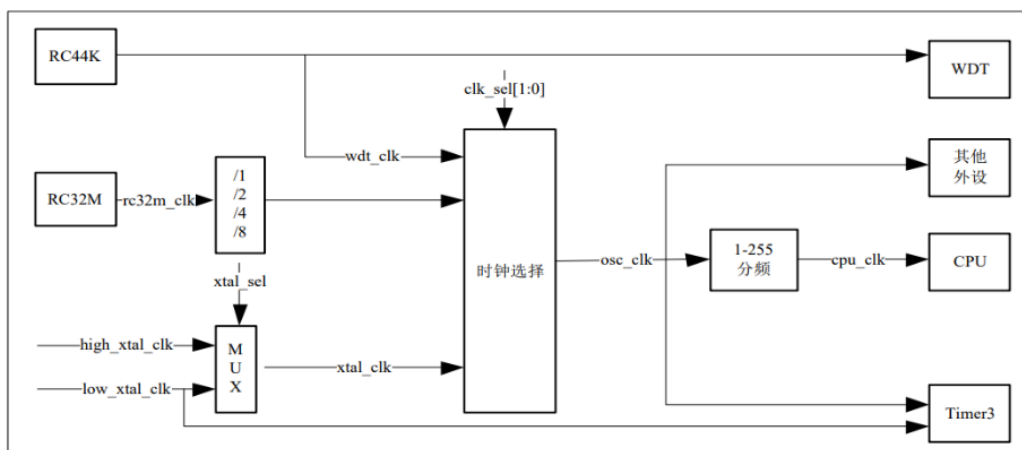


图 12-5 系统时钟框图

中断优先级介绍：以下以 HC89F3541 为例（HC89F3xx1 系列、HC89F1431、HC89F24x1、HC89F34x1C 系列与 HC889F3xx1 系列均适用），本处所指的中断优先级即抢断优先级，HC89F3541 通过 IP0、IP1、IP2、IP3 可以设置抢断优先级。

- ✓ 解决办法：将 Fosc 以及 Fcpu 时钟设置的一样，或者不设置中断优先级。

12.4.7 端口过零信号处理注意事项

情况描述：

针对小家电等市电应用时，经常性需要检测过零信号。大部分用户采用的方法为串联一个大电阻后接到 MCU，电阻阻值常为 1M。

注意事项：

- 1、进行过零检测时建议用示波器看下过零检测口的电压值，如果电压值超过 VDD 电压，建议增加串联的电阻阻值，确保传入 I/O 口的电压不超过 VDD 电压。
- 2、如外围电路无法修改，建议在检测过零信号前将 I/O 口设置为数字输入状态，采集完状态后再将 I/O 设置为推挽输出低。不过此方式不够稳定，建议还是从硬件上解决此问题。

12.4.8 通过 CS/对讲机干扰测试注意事项

情况描述：

对于大部分用户来说，对讲机或 CS 干扰测试是经常要面对的测试。应对这些干扰时，受干扰的非单单是 MCU，而是整个系统，此处将供几个简单并有效的解决办法。

注意事项：

- 1、检查系统电源，CS 测试绝大部分干扰是从电源导入 MCU 的，一个好的电源能够隔绝大部分干扰信号。
- 2、适当增加触摸 PAD 上串联电阻的阻值，触摸 PAD 上串联的电阻相当于一个 RC 滤波，如果增加阻值，可以增加滤波效果，带来影响是手指触摸之降低。
- 3、使用 HC Touch VLAB 触摸调试工具，观察干扰时的波形，设置合理的消毛刺条件及低基准线复位时长。

12.4.9 低功耗处理注意事项

情况描述:

针对部分低功耗应用场景，HolyChip 提供了触摸库《HC89F3xx1_Touch_Low_Power_Demo_Code》、《HC89F30xC_Touch_Low_Power_Demo_Cod》，用户可以在 HolyChip 官网下载到最新版本，此处将简单介绍使用低功耗时一些注意事项。

注意事项:

- 1、例程中使用了 WDT 进行掉电唤醒，注意清除动作的指令和进入掉电模式的指令需要间隔大于等于 3 个 wdt_clk 的周期（约 100us）。
- 2、在 IAP 擦写和进入掉电模式之前，需要配置 `FREQ_CLK` 寄存器，指明目前 CPU 时钟的频率。
- 3、当 Option 中 WDT 复位使能关闭，且 `EA=1`，`EWDT=1`，那么 WDT 溢出中断则会唤醒 MCU 掉电模式。如果 OPTION 中 WDT 复位使能关闭，且 `EA = 0` 或 `EWDT=0`，则 WDT 溢出值会唤醒部分系统，此时进入了掉电模式，系统会工作异常。因此，应保证再 WDT 溢出中断唤醒掉电模式时，WDT 溢出可以产生有效中断。禁止在进入掉电模式后，WDT 的溢出无法产生中断。
体现在库的则需注意:①确保 OPTION 中 WDT 复位使能关闭。②WDT 初始化中确认 EA=1, EWDT=1。③在 WDT 初始化后需在 WDT 溢出中断前（WDT 唤醒时间内）进入掉电模式。
- 4、进入低功耗之前建议关闭 BOR 以节省功耗。
- 5、系统进入掉电模式前，建议将 ADC 参考电压选择非 VDD，可以进一步降低系统功耗。
- 6、HC89F30xC 系列内部由于没有 T5 定时器，所以上电后没有对 WDT 的定时时间进行校准，因此《HC89F30xC_Touch_Low_Power_Demo_Code》中 WDT 设置的定时唤醒时间可能会有偏差。

12.4.10 芯片 VDD 变化注意事项

情况描述:

在部分特殊应用情况下, MCU VDD 可能会发生变化, 这种变化可能会对 CTK 模块造成一些影响, 用户使用时建议注意以下事项。

注意事项:

- 1、 确保 VDD 电压大于 CTK 充电电压 0.5V 以上, 否则触摸计数值将不稳定。
- 2、 CTK RawData 值从 5V 降低至 3.3V 时会上升 2%-3%。
- 3、 应避免 VDD 有较大抖动的情况。

12.4.11 滑条/滑轮应用

情况描述:

针对部分需要使用滑条或者滑轮的用户, HolyChip 提供了例程供与用户参考。如用户需要自行写相关算法函数, 用户可以直接调用“HC89F3xx1_GeneralKey_Touch_Lib_CFG.c”文件中的 XDATA_RawData 数组以及 XDATA_BaseLine 进行计算。

12.4.12 提升系统的健壮性

情况描述:

一个健壮的程序能够提升整个系统的稳定性, 因此在保证程序功能实现的前提下, 用户可以参考以下注意事项从而提升系统的健壮性。

注意事项:

- 1、 合理使用 WDT 看门狗。
- 2、 不使用悬空的 IO 口建议将其设置为推挽输出低。
- 3、 类似死循环等待时增加超时处理
- 4、 使用外部中断时避免连续外部中断打断系统正常执行

12.4.13 触摸端口复用

情况描述:

在一些特殊情况下, 用户可能需要将 TK 口与其他功能复用。一般情况下, 不建议用户使用端口复用的形式, 这种方式可能会导致触摸按键迟钝, Differ 噪声较大等情况。

注意事项:

如果必不可免发生 TK 口与其他功能口复用的情况，在复用时应注意 IO 口状态的切换与处理。

12.4.14 通过 ADC 内部通道检测 VDD 电压注意事项

情况描述：

部分使用场景下，用户需要对 MCU VDD 进行监控，HolyChip HC89F3xx1、HC89F1431、HC89F24x1、HC89F34x1C 及 HC89F30xC 系列均提供了内部 1/4 VDD 电压采集，使用时需注意以下事项。

注意事项：

目前版本的 MCU 在使用内部 1/4VDD 检测通道时会导致内部外部通道同时打开的情况。为规避此问题，用户可以选择舍弃一个带 ADC 通道的 I/O 口实现内部通道采集功能，使用方法如下：

ADCC1

位编号	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	ICHS[1:0]		-		XCHS[3:0]			

位编号	位符号	说明
7-6	ICHS[1:0]	ADC 内部输入通道选择 00: 禁止内部通道接入 01: 1/4VDD 作为 ADC 输入通道 10: 保留位 11: GND 接入 注：在进行内部通道选择时，外部通道选择 XCHS[3:0]应配置为 1111，否则可能会造成内部通道和外部通道同时打开的情况。
5-4	-	保留位（读为 0，写无效）
3-0	XCHS[3:0]	ADC 外部输入通道选择 XCHS[3:0] = x(x=0...15)，表示当前检测通道为 ANx，如 XCHS[3:0] = 2，表示当前检测通道为外部通道 AN2。 注：外部通道除设置 XCHS[3:0]，还需设置对应管脚的功能为模拟输入。

表 12-2

如表 12-2 所示，ADCC1 寄存器分为 ICHS[1:0]及 XCHS[3:0]两部分，ICHS[1:0]用于选择使能的内部通道，XCHS[3:0]用于选择使能的外部通道。当 ICHS[1:0]选择 01（使能 1/4VDD 作为 ADC 输入通道）时，XCHS[3:0]无论写入什么值，都会造成使能的外部通道与内部通道同时打开的情况，但只要使能的外部通道对应的 I/O 口浮空输入且外围无电路，则内部通道可正常使用。

举例：如用户需要采集内部 1/4VDD 电压，ADCC1 写入值为 0x4E（01001110B），使能的外部通道为 AN14，那么 AN14 对应的 P2.6 口设置为浮空输入且无外围电路，则采集的内部通道值 1/4VDD 为正确值。如 P2.6 口未设置为浮空输入或存在外围电路，那么采集的内部通道值 1/4VDD 的值不正确。

12.4.15 IAP 操作注意事项

情况描述:

在一些需要用到掉电数据存储的情况下，用户在进行 IAP（Flash 做类 EEPROM）操作时，需要注意以下注意事项。

注意事项:

- 1、 在进行扇区擦除时，建议先建议先暂停 CTK 扫描（Scan_Pause 函数），等到操作完成后再继续进行 CTK 扫描（Scan_Continue 函数）。
- 2、 其他注意事项建议阅读《AN0003_HA_通用系列 IAP 使用》。

12.4.16 ADC 使用注意事项

情况描述:

在一些需要用到 ADC 的情况下，需要注意以下注意事项。

注意事项:

- 1、 建议在 CTK 初始化之前初始化 ADC，并且需要在 ADC 初始化内配置好需要使用的 ADC 通道，确保不会和触摸通道产生冲突。

错误示例：ADC_Init 在 CTK_Init 之前，并且未对 ADCC1 寄存器进行配置（此时 ADC 通道默认 AN0），然后使能 CTK，并且使能了 TK0，由于 AN0 与 TK0 为同一个端口，于是 AN0 和 TK0 同时打开，导致初始化的 RawData 的数据错误。

- 2、 ADC_EN 使能以后，建议保持打开，不要关闭。

13 工具使用(HC-TOOL)

请参考《HC-LINK_工具用户手册》、《HC-PM51_工具用户手册》。

14 版本说明

版本	日期	描述
V1.00	2020/05/09	初版
V1.01	2020/08/25	文档规范化 将《触摸应用注意事项》内容并入到第 12 章电容触摸按键（CTK）
V1.02	2020/10/01	增加了 HC89F3xx1 系列
V1.03	2021/11/03	删除 HC88T3xx1、HC89F30xB 系列，新增 HC89F30xC 系列
V1.04	2024/11/21	增加 HC89F1431、HC89F24x1、HC89F34x1C 系列

HOLYCHIP公司保留对以下所有产品在可靠性、功能和设计方面的改进作进一步说明的权利。

HOLYCHIP不承担由本手册所涉及的产品或电路的运用和使用所引起的任何责任，HOLYCHIP的产品不是专门设计来应用于外科植入、生命维持和任何HOLYCHIP产品产生的故障会对个体造成伤害甚至死亡的领域。如果将HOLYCHIP的产品用于上述领域，即使这些是由HOLYCHIP在产品设计和制造上的疏忽引起的，用户应赔偿所有费用、损失、合理的人身伤害或死亡所直接或间接所产生的律师费用，并且用户保证HOLYCHIP及其雇员、子公司、分支机构和销售商与上述事宜无关。

芯圣电子

2024 年 11 月