

简介

本应用笔记将以HC89F3541为例从原理上介绍HolyChip低功耗触摸库、使用方式、功耗计算方式、应用注意事项。

- 本应用笔记适用芯片：HA 全系列触摸芯片。
- 本应用笔记适用库：HC89F3xx1_Touch_Low_Power_Demo_Code_V3.0.0.0
HC89F30xC_Touch_Low_Power_Demo_Code_V3.0.0.0
- 相关数据手册、工具及技术文档下载网址：<http://www.holychip.cn/>。

目录

1	低功耗设计方案原理.....	3
2	库函数例程及功耗计算公式使用.....	5
2.1	相关参数配置.....	5
2.2	功耗计算公式使用.....	5
3	低功耗应用注意事项.....	7
4	版本说明.....	8

1 低功耗设计方案原理

如下图所示，对于电容式触摸按键，手指的触摸会改变感应电容 C_x ，当检测电路对 C_x 充放电时， C_x 值的变化会引起电路信号变化，通过一定的检测电路可以测量出该变化，从而判断手指是否存在。不过，系统整体功耗因为频繁的扫描 C_x 的大小而增加。

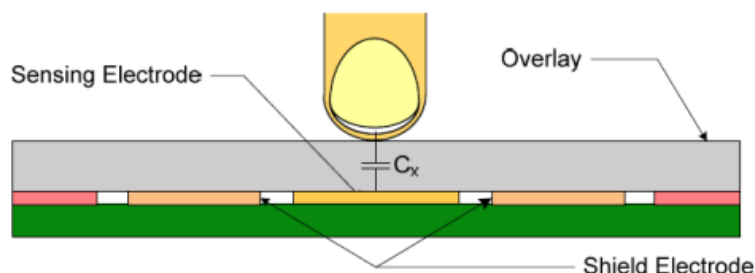


图 1-1 手指触摸产生感应电容

对于输入电压恒定的系统来说，其功耗主要取决于平均工作电流，即：

$$\text{Power} = V * I \text{ (公式1)}$$

其中， V 是系统工作电压， I 是系统平均工作电流。从公式1中可以看出，系统的功耗由系统的平均工作电流决定。降低平均工作电流的方法通常有两种：第一种是在不改变系统有效工作时间的的前提下降低系统的工作电流；第二种减少系统的有效工作时间，增加系统的休眠时间。往往只采用第一种办法不能将平均工作电流降低到一个理想的水平，所以需要结合第二种的方法。在触摸按键系统的实际工作中，相当一部分时间系统处于无任何按键按下的空闲状态。在这段时间内可以用软件将系统配置为掉电模式。

下图是一个具有休眠功能的典型系统软件流程图：

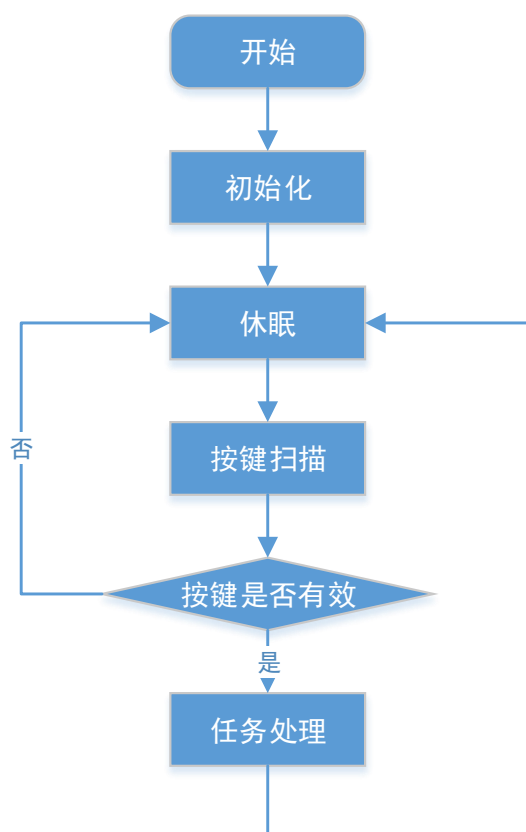


图 1-2 典型系统软件流程图

这是个典型的具有休眠功能的系统工作流程图，它的优点就是此软件流程简单易懂、容易实现，一般可以满足大多数场合的应用。

系统平均工作电流可由公式 2 计算得出：

$$I_{ave} = (T_{scan} * I_{scan} + T_{sleep} * I_{sleep}) / (T_{scan} + T_{sleep}) \quad (\text{公式2})$$

其中， T_{scan} 是一次扫描按键所需时间， I_{scan} 是按键扫描时的工作电流， T_{sleep} 是休眠时间， I_{sleep} 是休眠时的工作电流。 I_{sleep} 会远远小于 I_{scan} 。一般来说，为了保证一定的按键灵敏度， I_{scan} 可调整的空间有限，因此较快的扫描速度，较小的休眠电流，较长的睡眠时间是降低系统功耗的关键。

在实际设计中，考虑的因素更为复杂，除了上述之外，还需考虑按键的响应时间和按键的灵敏度等。最大休眠时间决定了系统的响应时间，对于相同的 I_{ave} ， I_{scan} 和 I_{sleep} ，较长的 T_{scan} 会引起 T_{sleep} 的增加，从而无法满足系统的响应时间；如果减少扫描时间，可能会无法有效减少系统噪声影响，影响按键的灵敏度。因此，低功耗触摸系统设计需要灵敏，可靠，快速的触摸按键扫描技术。

2 库函数例程及功耗计算公式使用

本笔记仅介绍低功耗下相关的参数设置及介绍，其他函数介绍请参考《AN0101_HA_HC893xx1_HC89F30xC系列按键触摸库使用》。

2.1 相关参数配置

以《HC89F3xx1_Touch_Low_Power_Demo_Code_V3.0.0.0》为例，启动Keil软件，加载HC89F3xx1_Touch_Low_Power_Demo_Code_V3.0.0.0代码。

点开文件：WDT.h，设置如下信息：

- WDT 唤醒时间

#define WDT_WAKE_TIMER	x(ms)
------------------------	-------

其中 x= 100ms~1000ms

本值即为 MCU 定时睡眠时间，时间越长，功耗越低，但是唤醒 MCU 的时长也会越长。

举例：如果将此值设置为 250，那么有可能手指按上 250ms（最长时间）以后按键才会有效

打开 Fun_Handle.c 设置如下信息：

- MCU 正常模式下，无触摸多长时间后进入掉电模式

#define T_MCU_GO_TO_LP_MODE_DF	x(s)
--------------------------------	------

其中 x=0s~25s

当 MCU 处以正常模式时，且无触摸按键按下时长超过设置时长则进入掉电模式。

举例：如果将此值设置为 5，那么无触摸按键按下 5s 后，MCU 进入掉电模式。

2.2 功耗计算公式使用

以《HC89F3xx1_Touch_Low_Power_Demo_Code_V3.0.0.0》为例，在工程文件中打开Excel表格《HC89F3xx1_HC89F30xC系列_LowPower_DemoCode功耗估算器》，如下图所示，在相应的表格中写入设置的参数即可计算出相关功耗。

用户配置区	
触摸唤醒检测通道个数	2
触摸采样时钟个数DSCR	48
触摸采样频率CTKDS	0
触摸低功耗模式WDT唤醒时间(ms)	250
关键数据估算值	
系统实际唤醒时间估算值(ms,最大值)	252
功耗估算值(uA)	27.7
其它配置条件	
系统时钟Fosc(KHz)	32000
CPU时钟Fcpu(KHz)	16000

表 2-1 功耗估算器

相关计算公式:

单通道扫描时间 $\approx ((\text{HCTouchLib_DSCR} + 1) * 256 - 1) / \text{HCTouchLib_CTKDS}$

系统实际唤醒时间估算值 = WDT唤醒时间+单通道扫描时间*触摸唤醒检测通道个数

功耗估算值 (uA) =
$$\frac{\text{WDT唤醒时间} * 250 + \text{单通道扫描时间} * \text{触摸唤醒检测通道个数} * 5000}{\text{系统实际唤醒时间估算值}}$$

3 低功耗应用注意事项

- 1、 例程中使用了 WDT 进行掉电唤醒，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等于 3 个 wdt_clk 的周期（约 100us）（HC89F30xC 系列芯片无此问题）。
- 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，应保证在 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（HC89F30xC 系列芯片无此问题）。
- 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。
- 7、 进入睡眠前，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功耗。

4 版本说明

版本	日期	描述
V1.00	2018/05/25	初版
V1.01	2018/07/24	修改相关错误描述
V1.02	2019/01/15	1、 增加第一章节 2、 第二章节删除部分描述 3、 第三章节修改相关描述 4、 增加第四章节
V1.03	2019/08/05	文档名称修改
V1.04	2020/08/27	文档规范化
V1.05	2020/10/15	删去 HC89F3xx1 系列，新增 HC88T3xx1 系列
V1.06	2020/12/30	增加 HC89F3xx1 系列
V1.07	2021/11/1	删去 HC89F30xB、HC88T3xx1 系列，新增 HC89F30xC 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

2020 年 10 月