

# MC30P8160-触摸汇编程序库说明

## V1.1

技术服务部

2023-11

sino**mcu**

## 目录

|    |                                  |                   |
|----|----------------------------------|-------------------|
| 1、 | <a href="#">概述.....</a>          | <a href="#">3</a> |
| 2、 | <a href="#">程序占用资源.....</a>      | <a href="#">3</a> |
| 3、 | <a href="#">寄存器定义.....</a>       | <a href="#">3</a> |
| 4、 | <a href="#">程序段落含义.....</a>      | <a href="#">4</a> |
| 5、 | <a href="#">调用程序库时的参数设定.....</a> | <a href="#">5</a> |
| 6、 | <a href="#">辅助串口调试工具使用.....</a>  | <a href="#">5</a> |
| 7、 | <a href="#">注意事项.....</a>        | <a href="#">6</a> |

## 1、概述

该程序库使用汇编方式编写，在少占用芯片资源的情况下完成触摸响应动作，工程师可以在此基础上添加其它用户定义功能的程序。程序库提供简单的程序出入口方式，可以在程序表头定义触摸端口和各个触摸点的灵敏度等参数，用一个寄存器或相应的 I/O 输出触摸响应动作。程序库包含抗干扰的数据处理，可以在一定的干扰环境下正常工作。

## 2、程序占用资源

程序库在触摸方面占用：

RAM: 15+4\*N (N 表示使用的触摸通道数量)

ROM: <0.4K

## 3、寄存器定义

MC30P8160 寄存器有 64 个字节，从 0X00-0X4F，其中触摸功能使用的变量在 MC30P8160\_TK.H 里定义，从 0X10 开始，为节省寄存器的资源，可以根据使用的通道数取消那些没有用到的触摸通道数据寄存器单元（见注释）。

```
cblock      0x20
;;
TKCNT0_SAVEH          ;触摸按键0是否使能;没有使用时取消定义
TKCNT0_SAVEL
TKCNT0_BASEH
TKCNT0_BASEL

TKCNT1_SAVEH          ;触摸按键1是否使能;没有使用时取消定义
TKCNT1_SAVEL
TKCNT1_BASEH
TKCNT1_BASEL

TKCNT2_SAVEH          ;触摸按键2是否使能;没有使用时取消定义
TKCNT2_SAVEL
TKCNT2_BASEH
TKCNT2_BASEL

TKCNT3_SAVEH          ;触摸按键3是否使能;没有使用时取消定义
TKCNT3_SAVEL
TKCNT3_BASEH
TKCNT3_BASEL
;;
;cblock      0x30
TOUCH_SCAN_TONE
TOUCH_SCAN_LIST
TKCNTH_NEW
TKCNTL_NEW
TOUCH_FLAG            ;对应哪个触摸通道被触发（相应的位置1则表示有触摸）
TOUCH_DEBOUNCE1       ;触摸按下消抖计数
TOUCH_DEBOUNCE2       ;触摸松开消抖计数
TOUCH_DEBOUNCE3       ;触摸环境消抖计数
TOUCH_DEBOUNCE4       ;触摸环境错误消抖计数
TOUCH_TEMP1           ;暂存
TOUCH_TEMP2           ;暂存
TOUCH_TEMP3           ;暂存
TOUCH_TEMP4           ;暂存
TOUCH_TEMP5           ;暂存
TOUCH_TEMP6           ;暂存
;cblock      0x3F
ENDC
```

\*注：主程序从 0X10-0X20 定义了部分公用寄存器，MC30P8160-TK.H 定义了触摸功能的寄存器，MC30P8160\_UART.ASM 定义了串口使用的寄存器，暂时余下 0X46-0X4F 地址的寄存器，如触摸通道数有减少，或者串口没有使用，要注意节省出来的寄存器地址，以便挪给其它程序使用。

## 4、程序段落含义

A、SET\_TOUCH\_OPTION 段落，用于初始化触摸端口、设定触摸定时器等操作；

B、SET\_TOUCH\_OPTION2 段落，用于初始化触摸寄存器等操作，在上电或休眠唤醒时使用；

```
SET_TOUCH_OPTION:
    MOVAI    TOUCH_KEY          ;00001111B
    MOVRA    PIADCR             ;设定对应的TK口为非数字功能，1为非数字功能
    ORRA     DDR1               ;只设定TK脚为输入模式，其它不变。输入/输出：1为输入，0为输出
    MOVRA    TOUCH_FLAG
    SWAPR    TOUCH_FLAG        ;将对应的TK 使能触摸点要刷新基准数据（TOUCH_NEW_FLAG）
    BCLRR    T2IE               ;不需要T2中断，TK采用扫描方式
    MOVAI    11001000B          ;定时器0配置,BIT7为定时器2开启,BIT6为定时器2连续/单次计数模式,BIT4-3:时钟源选定,BIT0-2:定时器分频
    ORRA     T2CR               ;开启T2定时器，并且为单次计时模式
    MOVAI    00000000B
    MOVRA    TKCR0              ;TK触摸配置,BIT7=1开启TK,BIT6=1允许触摸充放电,BIT5/4为TK时钟滤波,BIT3:TK跳频,BIT2-0:TK振荡频率选择
    MOVAI    11111111B         ;选择通道0
    MOVRA    TKCR2              ;TK触摸配置,BIT7=1使能TK计数,BIT6:TKM计数中断,BIT2/0:选定触摸通道
    RETURN

SET_TOUCH_OPTION2:            ;休眠唤醒后重新设定触摸参数
    CLRR     TOUCH_SCAN_LIST
    CLRR     TOUCH_SCAN_TONE
    CLRR     TOUCH_DEBOUNCE1
    CLRR     TOUCH_DEBOUNCE2
    CLRR     TOUCH_DEBOUNCE3
    CLRR     TOUCH_DEBOUNCE4
    BCLRR    TK_SCANEND_FG      ;所有TK的扫描完成
    BSET     WAKEUP_FG          ;看门狗唤醒
    RETURN
```

C、SET\_TOUCH 段落，扫描触摸通道、获取触摸读数、处理数据、响应触摸动作等功能；

```
SCAN_TOUCH:
    MOVAR    TOUCH_SCAN_LIST    ;扫描触摸的步骤
    ANDAI    0X03
    ADDRA    PCL
    GOTO     SCAN_TOUCH_STEP1    ;设定扫描通道和扫描参数，并开启触摸
    GOTO     SCAN_TOUCH_STEP2    ;等待触摸计数完成，并屏蔽杂波
    GOTO     SCAN_TOUCH_STEP3    ;分析每个通道的读数（滤波和消抖），确认是否触发触摸按键
    CLRR     TOUCH_SCAN_LIST
    GOTO     SCAN_TOUCH_STEP1    ;设定扫描通道和扫描参数，并开启触摸
;-----
SCAN_TOUCH_STEP1:
    MOVAR    TOUCH_SCAN_TONE    ;扫描通道数
    ANDAI    0X03
    ADDRA    PCL
    #if SET_TOUCH0_ENABLE        ;第0个触摸点是否使能
        GOTO SCAN_TOUCH_STEP1_TONE0
    #else
    #endif
    #if SET_TOUCH1_ENABLE        ;第1个触摸点是否使能
        GOTO SCAN_TOUCH_STEP1_TONE1
    #else
    #endif
```

D、SUB\_2BYTE 子程序，2个字节的减法子程序；

```
SUB_2BYTE:
    ;减法子程序（SUB_AH/L-SUB_BH/L=SUB_CH/L）(适用于2个数据相差不过15BIT)
    ;SUB_AH/L为被减数高/低8位，SUB_BH/L为减数高/低8位，差值存在SUB_CH/L，且SUB_CH-bit7为零时表示减法运算结果无溢出，为1时表示运算结果有溢出
    ;如果两数相差超过0xFF，则以0xFF为最大差值
    BCLRR    _C
    MOVAR    SUB_AL
    ASUBAR    SUB_BL
    MOVRA    SUB_CL
    MOVAR    SUB_AH
```

## 5、调用程序库时的参数设定和步骤

A、调用触摸程序库需包含 MC30P8160\_TK.ASM 和 MC30P8160\_TK.H 两个文件。

B、先确定需要使用的通道数量和对应的端口，定义好下面参数，注意对应关系

```
TOUCH_KEY EQU 00001111B ;定义触摸按键使能的通道，
TOUCH_KEY_NUM EQU 4 ;定义有效的触摸按键数量（0-3）
#define SET_TOUCH3_ENABLE 1 ;触摸按键TK3是否使能P13
#define SET_TOUCH2_ENABLE 1 ;触摸按键TK2是否使能P12
#define SET_TOUCH1_ENABLE 1 ;触摸按键TK1是否使能P11
#define SET_TOUCH0_ENABLE 1 ;触摸按键TK0是否使能P10

#define TOUCH_SLEEP_FG 1 ;是否需要触摸休眠唤醒,0=不休眠，非零=休眠
;-----
#define SET_TOUCH_ENV_LEVEL 0X10 ;通道环境重适应阈值（第一级）
#define SET_TOUCH_ENV2_LEVEL 0X20 ;通道环境重适应阈值（第二级）
#define SET_TOUCH_DELAY_LEVEL 0X20 ;触摸重适应时间（缓慢靠近按键允许响应时间，设定时间短时

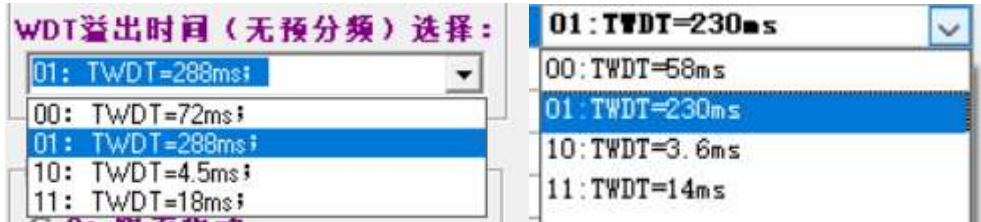
#define SET_SENS_LEVEL0 0xEE ;TK0触摸灵敏度级别
#define SET_TOUCH0_PRE_LEVEL 0X80 ;通道0按下触摸通道的阈值(单独定义值)
#define SET_TOUCH0_RELEA_LEVEL 0X80 ;通道0放开触摸通道的阈值
#define SET_TOUCH0_DEBOUNCEC 0X10 ;通道消抖处理次数,即连续多少次响应按键为有效
```



同时，也要注意主程序中对 TK 各端口的定义

#### C、确定是否需要休眠功能。

休眠功能通过看门狗休眠唤醒来实现低功耗和重新唤醒扫描，（注意烧录芯片选项和仿真器选项的时间有差异），看门狗时钟有 4 种烧录选项，分别是 288ms，72ms，18ms 和 4.5ms，时间越短，触摸响应速度越快，相对待机功耗也会越大，同时待机功耗还与使能触摸的通道数量和灵敏度（SET\_SENS\_LEVEL）有关。



```
#DEFINE TOUCH_SLEEP_FG 0 ;是否需要触摸休眠唤醒,0=不休眠,非零=休眠
```

#### D、灵敏度设定。

根据实际模具的介质厚度，设定合适的触摸灵敏度，每个 TK 触摸通道都有对应的灵敏度设定值 SET\_SENS\_LEVELx 和响应阈值来确定各 TK 通道的触摸灵敏度，并以 IC 放在实际产品上调试为准。

**注：仿真器灵敏度和实际 IC 的灵敏度会有很大差异，所以一般是在仿真器上先把触摸逻辑功能调试完毕，然后烧录好 IC，在实际产品上通过串口读到的数据来验证触摸灵敏度的响应情况，然后再微调灵敏度。**

另外，通过外部串口设备读取触摸读数时会因实际产品和外部串口设备共地而造成触摸读数失真，不能如实反映产品的触摸状况，所以建议通过串口隔离板来读取产品上的触摸读数（详见“串口数据传送隔离板使用说明”文件）。

```
#DEFINE SET_SENS_LEVEL0 0xEE ;TK0触摸灵敏度级别
#DEFINE SET_TOUCH0_PRE_LEVEL 0x80 ;通道0按下触摸通道的阈值(单独定义值)
#DEFINE SET_TOUCH0_RELEA_LEVEL 0x80 ;通道0放开触摸通道的阈值
#DEFINE SET_TOUCH0_DEBOUNCEC 0x10 ;通道消抖处理次数,即连续多少次响应按键为有效
```

#### E、设定环境适应值、环境适应时间参数：

```
#DEFINE SET_TOUCH_ENV_LEVEL 0x10 ;通道环境重适应阈值（第一级）
#DEFINE SET_TOUCH_ENV2_LEVEL 0x20 ;通道环境重适应阈值（第二级）
#DEFINE SET_TOUCH_DELAY_LEVEL 0x20 ;触摸重适应时间（缓慢靠近按键允许响应时间，设
```

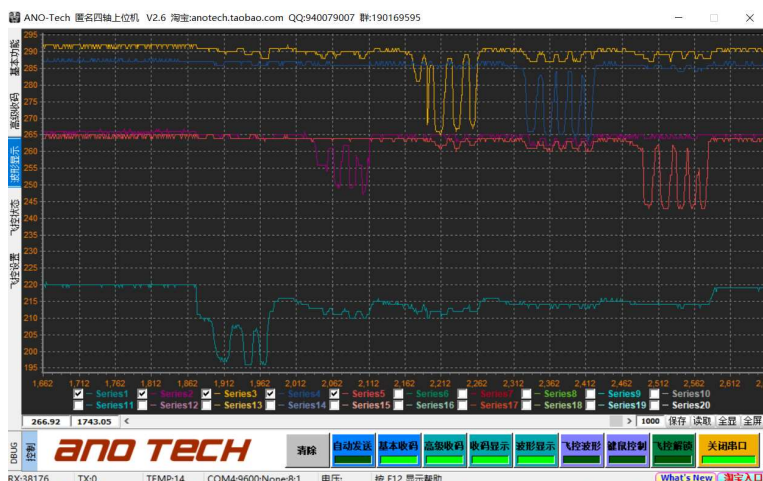
#### F、触摸功能实现步骤：

- 1、在主程序中，需要初始化所有寄存器，然后调用“SET\_TOUCH\_OPTION”和“SET\_TOUCH\_OPTION2”子程序进行触摸数据初始化
- 2、在程序主循环中不断调用“SET\_TOUCH”子程序进行触摸扫描，并根据返回的全局变量“TOUCH\_FLAG”的数值确定哪个 TK 通道有触摸响应。SET\_TOUCH 子程序内部使用 T0 扫描，但不产生中断，所以主程序可以根据实际功能来定时调用或者不间断调用 SET\_TOUCH。
- 3、如需要某个触摸通道重新刷新，适应当前环境，可以将 TOUCH\_FLAG 寄存器对应的位进行清零，并将 TOUCH\_NEW\_FLAG 对应的位进行置位，当 TOUCH\_NEW\_FLAG 对应的位再次清零则表示该触摸通道已重新刷新。
- 4、如果需要触摸休眠唤醒，要在唤醒后再次调用“SET\_TOUCH\_OPTION2”子程序进行触摸数据初始化。
- 5、程序功能在仿真器上调试完毕后，建议打开串口数据输出功能，烧录 IC 后在产品上再次确认触摸响应情况和触摸读数的变化，以便找到最合适的触摸设定参数，实现最合适的触摸效果。

**\*通过上述参数设定，可以灵活使能每个触摸通道及调整各个触摸通道灵敏度，以适应不同的触摸介质和不同通道之间的灵敏度差异，一般通过调整这些参数来适用于各种触摸应用；**

## 6、辅助串口调试工具使用

在使能串口数据输出后,通过串口数据监测工具对输出的数据进行监控,可以直观地判断出该触摸通道平时的触摸读数和按下触摸时的触摸读数,并根据两者之间的差值对第 5 项里的触摸响应/松开阈值进行设定,方便调试。(工具使用详细见“匿名四轴显示工具使用说明”文件)



## 7、注意事项

- 为了防止在向外传送数据时,因芯片和外部检测设备共地而造成触摸环境发生变化,使得触摸数据参照产生偏差,建议用光耦等隔离方式将触摸芯片和外部设备做电气隔离。  
(详细见“串口数据传送隔离板使用说明”文件)
- 程序库里的寄存器定义没有做条件汇编,所用到的寄存器已全部定义,如选用的通道数低于 4 个,没有使用到的寄存器要手动将其屏蔽,以节省 RAM 空间。
- C 语言版本暂时以汇编程序库内嵌编译出来,程序处理方式基本相同。
- 如有任何疑问及建议,敬请联系我们。

建议开发流程:

- 烧录有打开串口功能的触摸芯片,并放到实际产品上运行,通过**串口隔离板**读取各个触摸点的触摸读数,根据读数调整各个 TK 的触摸灵敏度和阈值,建议每个触点的按下和松开的读数差值在 60 以上;  
然后设定对应 TK 的按下阈值 (SET\_TOUCHx\_PRE\_LEVEL),建议取刚刚获取的读数差值的一半,如差值是 60,则 SET\_TOUCHx\_PRE\_LEVEL =30;  
然后再设定 TK 松开阈值 (SET\_TOUCHx\_RELEA\_LEVEL),如按键有长按功能,建议松开阈值和按下阈值相同,如果是短按功能,可以比按下阈值略小,如 SET\_TOUCHx\_RELEA\_LEVEL=28;  
最后,根据调整好的参数烧好芯片在实际产品上测试各个 TK 的触摸效果,直至达到预期效果。另外,关于按键响应速度、消抖参数不在此描述,可以参考例程注释或来电咨询。
- 根据产品要求在仿真器上将所有功能逻辑开发完成,触摸灵敏度及个触摸点的阈值以上一步骤获取的参数设定(仿真器的触摸灵敏度会和实际芯片有差异,所以仿真时可以适当调整灵敏度参数,烧录芯片时再将参数改回去)。
- 在最终完成产品开发时,建议再次打开串口,测试一下各个 TK 的触摸读数有没有符合设计值、测试程序其它功能是否会影响到触摸效果。