

如何利用 HT45R36 与 HT45R38 做 Touch Switch

文件编码：HA0115S

简介

本范例是利用 HT45R36 与 HT45R38 中的 C/R-F 电路模块来做 Touch Switch，这是一种利用人体电容效应的感应开关，它无需按压动作，只要在绝缘覆盖层上的开关位置轻轻地碰触即可，此种开关具有无磨损，易清洁，危险性低等特点。

芯片特性

Touch Switch 的核心电路是 C/R-F 模块，HT45R36 与 HT45R38 都具有这样的功能模块，下表列出了此两款芯片的主要片上资源：

Part No.	ROM	RAM	I/O	Timer		Interrupt		C/R-F Channel	LED Driver Output	A/D	PWM	OPA	Stack	Package
				8-bit	16-bit	Ext.	Int.							
HT45R36	2Kx14	120x8	9	1	1	2	5	16	8x8	---	---	---	4	44QFP, 52QFP
HT45R38	4kx15	192x8	13	2	1	2	5	12	8x8	12-bitx5	8-bitx2	1	6	52QFP

HT45R38 除了有 C/R-F 以外，还加入了 A/D、PWM 与 OPA 等外围组件在 MCU 里面，是 A/D + C/R-F 功能的 MCU；而 HT45R36 是 I/O + C/R-F 功能的 MCU。本文以 HT45R36 为例介绍怎样做 Touch Switch。

工作原理

Touch Switch 是利用人体电容效应改变 C/R-F 电路中电容的容量，进而改变其振荡频率，通过检测频率的改变大小即可检测出开关是否被按下。C/R-F 电路框图如下图 1。Touch Switch 框图如下图 2，圆圈阴影部分为铜片，可 layout 于 PCB 上，铜片上再覆盖一层亚克力。

IN (0 Ω), RREF (27k Ω), CREF (10pF) 引脚分别连接了固定的电阻电容 (括号中为实验时所用的电阻电容值，可用参考)，RC OSC 电路振荡产生一个频率值为参考频率值，当有电阻或电容连到 RC 引脚，就会改变 RC OSC 电路的振荡频率。当人接触 Touch Switch 时，相当于在电路中并入一个电容，振荡频率会减小，程序只需要检测实时频率与参考频率的差值大小，即可判断 Touch Switch 是否被按下。

ASON 为通道选择寄存器：

```
00000b=TG1 on, others off
00001b=TG2 on, others off
...
...
...
01111b=TG16 on, others off.
10000b=all TGs off, save mode
```

程序通过改变 ASON 轮流打开 TG1 ~ TG16, 对 RC1 ~ RC16 Touch Switch 的状态进行扫描。

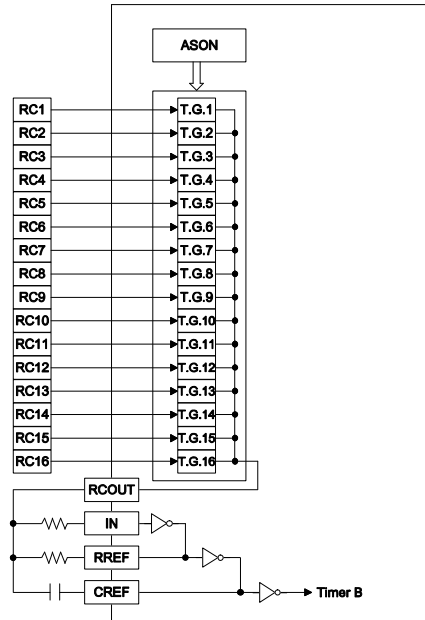


图 1: C/R-F 电路框图

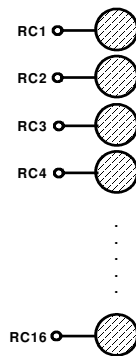


图 2: Touch Switch 框图

MCU 利用比较的方式来得到 RC OSC 的频率值，原理参考下面的框图（图 3）。假设在 Timer A 设定 1ms 产生一次 Overflow，接下来在 Timer B 读 Count 的值然后再将 Timer B 的值除以 Timer A，并且除以 1，变成频率。实际程序并不需要计算出频率值，只要计算在相同的 Timer A 定时时间内，参考频率下（Touch Switch off）Timer B Count 的值与实时的 Timer B Count 的值之间的差值，当差值达到了事先设定的 Sensitivity 值，即认为 Touch Switch 被按下。

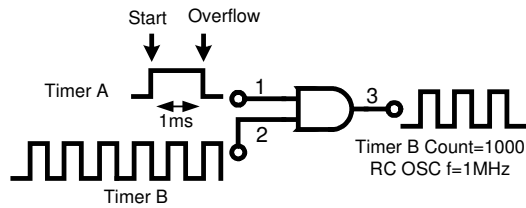
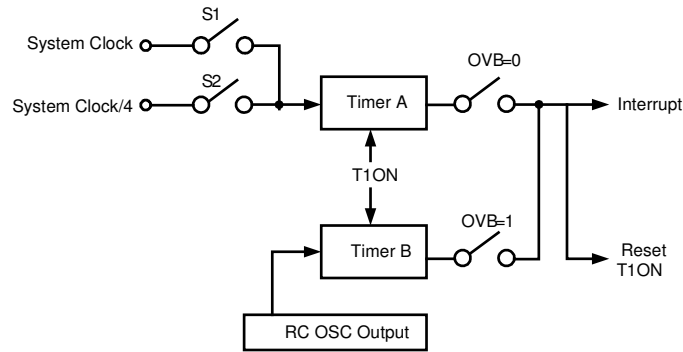


图 3

应用电路图

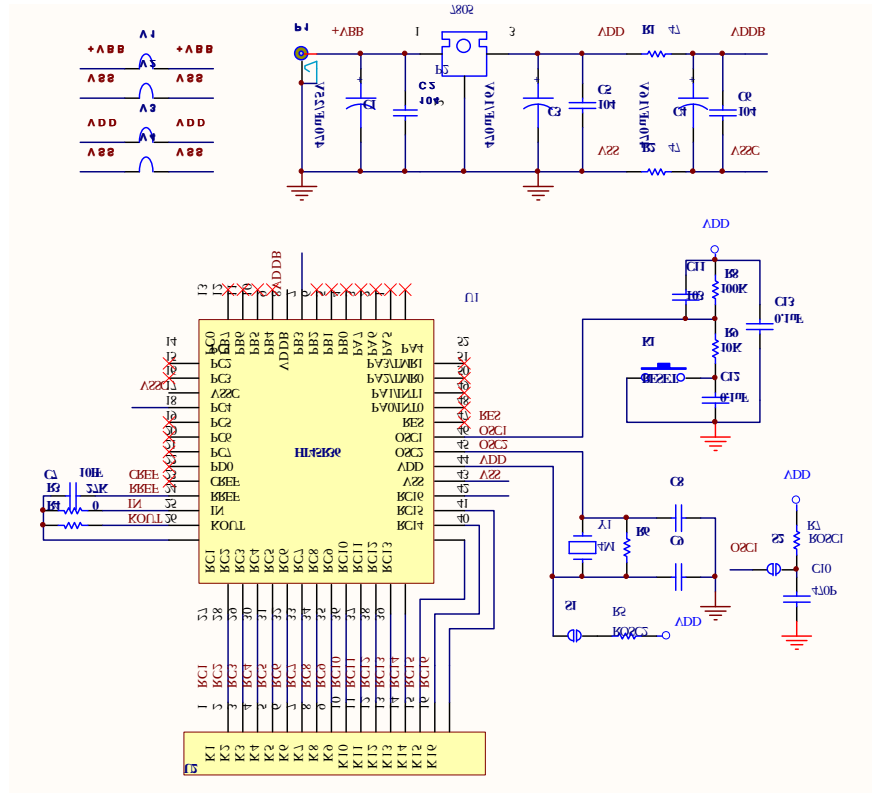


图 4

上图 4 中的 K1 ~ K16 分别连接图 2 中所示的 Touch Switch。
 注意事项：由于电阻电容受环境温度的影响较大，故 RC OSC 的振荡频率在 Touch Switch 没有被按下时也并非一个固定值，受到环境温度的影响，所以当一段时间没有 Touch Switch 按下时，应该将参考频率下 Timer B Count 的值进行更新，以保持实时性，保证 Touch Switch 不会产生误动作。

范例程序

```

=====
;;*****File Name: Main.Asm*****
;;HT_IDE Version:          V6.6 SP5
;;Main Function:          **Touch Switch **
;;MCU Body:               HT45R36
;;VDD:                    5V
;;MCU Frequency:          4MHz
;;-----
;;Mask Option:
;;-----
;Osc:                      Crystal
;Fsys:                      4MHz
;Wdt:                      Disable
;WDT Clock Source:         T1
;CLR WDT:                  One clear instruction
;Wake_Up PA.0-7:          Non-Wake-Up
;Pull-high PA:             PULL-HIGH
;Pull-high PB:             PULL-HIGH
;Pull-high PC:             PULL-HIGH
;Pull-high PD:             PULL-HIGH
;INT0 trigger edge        Disable
;INT1 trigger edge        Disable
;LVR:                      Disable
=====
#include HT45R36.INC
=====
MMOV MACRO TEMP1,TEMP2
    MOV    A,          TEMP2
    MOV    TEMP1,      A
ENDM
=====
PUSH MACRO                                ;; 进入中断服务程序时入栈

    MOV    ACC_STACK,  A
    MOV    A,          STATUS
    MOV    STATUS_STACK, A
    MOV    A,          MP0
    MOV    MP0_STATE,  A
ENDM

=====
POP MACRO                                  ;; 退出中断服务程序时出栈
    MOV    A,          MP0_STATE
    MOV    MP0,        A
    MOV    A,          STATUS_STACK
    MOV    STATUS,     A
    MOV    A,          ACC_STACK
    RETI

```

```

        ENDM
;=====
LXXXX      EQU    4167    ;;TIMERA 的定时计数初值, 对应的定时时间为 4.167ms
D_CHECK    EQU    16667   ;;TIMERA 的定时计数初值, 对应的定时时间为
                        ;;16.667ms
INITIAL_COUNT EQU    50    ;;设定的没有键按下的循环次数, 50 次循环没有键按下,
                        ;;即进行更新参考频率下的 TIMERB COUNT 值, 该值决定
                        ;;了更新的间隔时间
X_X        EQU    50      ;;设定的 Sensitivity 值
;=====
DATA_MAIN .SECTION 'DATA'
ACC_STACK DB      ?
STATUS_STACK DB    ?
MP0_STATE DB      ?
MP1_STATE DB      ?
MODE_CX   DB      ?
RF_CX     DB      ?
RF_DX     DB      ?
SAVE_RF_DX DB     ?
RF_EX     DB      ?
BUF0      DB      ?
BUF1      DB      ?
BUF2      DB      ?
BUF3      DB      ?
BUF00     DB      ?
BUF10     DB      ?
DEL0      DB      ?
DEL1      DB      ?
DEL2      DB     ?
ZERO      DB      ?
KEY_NUMBER DB  ?      ;; 用来存储被按下的 Touch Switch 的编号
INITIAL_RCOSC_FLAG DBIT ;; 16 个 Touch Switch off 时的 TIMERB COUNT 值
                        ;; 扫描完毕旗标
SCAN_RCOSC_FLAG DBIT  ;; 16 个 Touch Switch 一轮扫描完毕旗标
TIMER1_ISR_FLAG DBIT  ;; 进入 TIMER1 中断服务程序旗标位
F_KEY_ON   DBIT      ;; 有 Touch Switch 被按下的旗标
ORG 029H
RF_DATA    DB 32 DUP (?) ;; 用来存储 16 个 Touch Switch off 时的 TIMERB
                        ;; COUNT 值 (16bit)
;=====
CODE_MAIN .SECTION AT 00H 'CODE'
        JMP MAIN_START
        ORG 004H                ;;no used
        NOP
        NOP
        RETI
        ORG 008H                ;;no used
        NOP
        NOP

```

```

RETI
ORG 00CH                ;;no used
NOP
NOP
RETI
ORG 010H                ;;TIMER1 中断
NOP
NOP
JMP TIMER1_ISR

;;=====
====
MAIN_START:
CALL INITIAL_SUB        ;; 初始化程序
CALL SCAN_KEY_INITIAL_SUB ;; 参考频率下 (Touch Switch off) 的 TIMERB COUNT
                           ;; 值存入 BANK0 [50H~6FH]
CALL DELAY_300MS        ;; 延时 300ms
CLR SCAN_RCOSC_FLAG
CALL FIRST_TIME_SETUP   ;; 实时扫描的 TIMERB COUNT 值存入 BANK1
                           ;; [40H~5FH]

SCAN_ALL_KEY:
MOV A, 00H
MOV RF_DX, A
CLR F_KEY_ON
CHECK_SUB:
CLR INT1.0
CALLFIRST_CHECK_SUB     ;; 比较 BANK0 [50H~6FH] 与 BANK1 [40H~5FH] 的值
SNZ C                   ;; BANK0 [50H~6FH] > BANK1 [40H~5FH]?
JMP SCAN_NEXT          ;; NO
MOV A, BUF2            ;; YES
SUB A, X_X             ;; 差值与设置的 Sensitivity 值进行比较
MOV A, BUF3
SBC A, ZERO
SNZ C                   ;; 是否有 Touch Switch 按下?
JMP SCAN_NEXT          ;; NO
SET F_KEY_ON           ;; YES 设置有 Touch Switch 按下旗标
MOV A, RF_DX
MOV SAVE_RF_DX, A      ;; 存储被按下的 Touch Switch 相应得信道值 (ASON)
SCAN_NEXT:             ;; 继续扫描下一个 Touch Switch
SET INT1.0
SNZ TIMER1_ISR_FLAG
JMP $-1
CLR TIMER1_ISR_FLAG
INC RF_DX
MOV A, RF_DX
XOR A, 010H
SNZ Z                   ;; 是否 16 个 Touch Switch 已扫描完一轮?
JMP CHECK_SUB          ;; NO, 继续扫描下一个 Touch Switch
SNZ F_KEY_ON           ;; YES, 是否有键按下?
JMP NO_KEY             ;; NO

```

```

MOV A, SAVE_RF_DX                ;;YES
INC ACC
MOV KEY_NUMBER, A                ;;将 Touch Switch 的编号保存到 KEY_NUMBER
MMOVRF_EX, INITIAL_COUNT
JMP SCAN_ALL_KEY
NO_KEY:                          ;;16 个 Touch Switch 都没有被按下
SET KEY_NUMBER
SDZ RF_EX
JMP SCAN_ALL_KEY
CALLL_SAVE_INITIAL              ;;经过 50 个循环扫描 16 个 Touch Switch 都没有被按
                                ;;下过,即对参考频率的 TIMERB COUNT 值进行更新

MMOVRF_EX, INITIAL_COUNT
JMP SCAN_ALL_KEY
;;=====
INITIAL_SUB:                    ;;初始化
CLR_BANK0:                      ;;BANK0 清除为 0
    MOV A, 28H
    MOV MP0, A
    MOV A, 88
CLR_BANK0_LP:
    CLR R0
    INC MP0
    SDZ ACC
    JMP CLR_BANK0_LP
CLR_BANK1:                      ;;BANK1 清除为 0
    MMOV BP, 01H
    MOV A, 40H
    MOV MP1, A
    MOV A, 32
CLR_BANK1_LP:
    CLR R1
    INC MP1
    SDZ ACC
    JMP CLR_BANK1_LP
    SET KEY_NUMBER
    MMOV RF_EX, INITIAL_COUNT
    CLR PB                        ;;设置 I/O
    CLR PBC
    CLR PC
    CLR PCC
    CLR INTC0
    CLR INTC1
    CLR TMR0C
    CLR TMR1C
    MOV A, 01H
    MOV INTC0, A                ;;始能全局中断
    MOV A, 01H
    MOV INTC1, A                ;;始能 TIMER1 中断
    MOV A, 020H

```

```

MOV     TMR1C, A                ;;TIMER1:TIMER MODE(SYSTEM CLOCK/4)
MOV     A, 02H
MOV     RCOCR, A                ;;RC OSCILLATION CONVERTER MODE
CLR     TMRBL
CLR     TMRBH
MOV     A, RF_CX
MOV     ASCR, A
RET

;;=====
SCAN_KEY_INITIAL_SUB:          ;; 参考频率下(Touch Switch off)的TIMERB COUNT
值存入                          ;; BANK0 [50H~6FH]
    CLR     MODE_CX
    CLR     RF_CX
    CLR     ASCR                ;; 选择信道 1 打开,对 Touch Switch 1 进行扫描
    CLR     TMRBL
    CLR     TMRBH
    MMOV    TMRAL, LOW (65536-D_CHECK)    ;;TIMER A 定时时间为 16.667ms
    MMOV    TMR AH, HIGH (65536-D_CHECK)
    SET     TMR1C.4            ;;开 TIMER1
    CLR     INITIAL_RCOSC_FLAG
    SNZ     INITIAL_RCOSC_FLAG
    JMP     $-1                ;;等待 16 个 Touch Switch 扫描完毕
    RET

;;=====
FIRST_TIME_SETUP:              ;;实时扫描的 TIMERB COUNT 值存入 BANK1
                                ;; [40H~5FH]
    MMOV    MODE_CX, 01H
    MOV     A, 00H
    MOV     RF_CX, A
    MOV     ASCR, A            ;;选择信道 1 打开,对 Touch Switch 1 进行扫描
    CLR     TMR1C.4
    CLR     TMRBL
    CLR     TMRBH
    MMOV    TMRAL, LOW (65536-LXXXX)    ;;TIMER A 定时时间为 4.167ms
    MMOV    TMR AH, HIGH (65536-LXXXX)
    SET     TMR1C.4            ;;开 TIMER1
    SNZ     SCAN_RCOSC_FLAG
    JMP     $-1                ;;等待 16 个 Touch Switch 扫描一轮完毕
    RET

;;=====
FIRST_CHECK_SUB:                ;; BANK0 [50H~6FH] - BANK1 [40H~5FH]
    MOV     A, 40H
    ADD     A, RF_DX
    ADD     A, RF_DX
    MOV     MP1, A
    MOV     A, R1
    MOV     BUF0, A
    INC     MP1
    MOV     A, R1

```

```

MOV     BUF1, A
MOV     A, 50H
ADD     A, RF_DX
ADD     A, RF_DX
MOV     MP0, A
MOV     A, R0
SUB     A, BUF0           ;;对应的 TIMERB COUNT 值低位相减
MOV     BUF2, A
INC     MP0
MOV     A, R0
SBC     A, BUF1           ;;对应的 TIMERB COUNT 值高位相减
MOV     BUF3, A
RET

;;=====
L_SAVE_INITIAL:           ;;更新参考频率下(Touch Switch off)的 TIMERB
COUNT 值
    MMOV  MP0_STATE, INTC0
    CLR   INTC0.0
    MMOV  MP0, 050H
    MMOV  MP1, 040H
    MMOV  DELO, 32
L_SAVE_INITIAL_LOOP:
    MMOV  R0, R1
    INC   MP0
    INC   MP1
    SDZ   DELO
    JMP   L_SAVE_INITIAL_LOOP
    MMOV  INTC0, MP0_STATE
    RET

;;=====
TIMER1_ISR:               ;; TIMER1 中断服务程序
    PUSH
    SET   TIMER1_ISR_FLAG
    MOV   A, MODE_CX
    ADDM  A, PCL
    JMP   MODE_CX_0       ;; SET INITIAL_RCOSC_FLAG & TIMERB COUNT 值
                                ;; 存入 BANK0 [50H~6FH]
    JMP   MODE_CX_1       ;; SET SCAN_RCOSC_FLAG & (TIMERB COUNT 值*4)
                                ;; 存入 [40H~5FH] OF BANK1
    POP

;;-----
MODE_CX_0:
    MOV   A, TMRBH
    MOV   BUF1, A
    MOV   A, TMRBL
    MOV   BUF0, A
    MOV   A, 50H
    ADD   A, RF_CX
    ADD   A, RF_CX

```

```

MOV     MP0,    A           ;; 将 TIMERB COUNT 值存入 BANK0 [50H~6FH]
MOV     A,     BUF0
MOV     R0,    A
INC     MP0
MOV     A,     BUF1
MOV     R0,    A
INC     RF_CX
MOV     A,     RF_CX
MOV     ASCR,  A           ;; 轮流打开 1~16 个信道,对 16 个 Touch Switch 进
                           ;; 行扫描

XOR     A,     010H
SNZ     Z           ;; 16 个 Touch Switch 是否扫描完毕?
JMP     SCAN_LP_0     ;; NO
SET     INITIAL_RCOSC_FLAG   ;; YES, 设置相应旗标
JMP     EXIT_TIMER1_ISR_0

SCAN_LP_0:
CLR     TMRBL
CLR     TMRBH
MMOV    TMRAL, LOW (65536-D_CHECK)
MMOV    TMRAH, HIGH (65536-D_CHECK)
SET     TMRIC.4
JMP     EXIT_TIMER1_ISR_0

EXIT_TIMER1_ISR_0:
POP

;;-----
MODE_CX_1:
MOV     A,     MP1
MOV     MP1_STATE, A
MOV     A,     TMRBH
MOV     BUF10, A
MOV     A,     TMRBL
MOV     BUF00, A
CLR     C
RLC     BUF00           ;; TIMERB COUNT 值*2
RLC     BUF10
CLR     C
RLC     BUF00           ;; TIMERB COUNT 值*2
RLC     BUF10
MOV     A,     40H
ADD     A,     RF_CX
ADD     A,     RF_CX
MOV     MP1,    A           ;; 将 TIMERB COUNT 值*4 存入 BANK1 [40H~5FH]
MOV     A,     BUF00
MOV     R1,    A
INC     MP1
MOV     A,     BUF10
MOV     R1,    A
INC     RF_CX
MOV     A,     RF_CX

```

```

MOV    ASCR, A                ;; 轮流打开 1~16 个信道,对 16 个 Touch Switch
                                   ;; 进行扫描
XOR    A, 010H
SNZ    Z                      ;; 16 个 Touch Switch 是否扫描完毕?
JMP    SCAN_LP_1             ;; NO
SET    SCAN_RCOSC_FLAG       ;; YES,设置相应旗标
CLR    RF_CX
MOV    A, RF_CX
MOV    ASCR, A
SCAN_LP_1:
CLR    TMRBL
CLR    TMRBH
MMOV   TMRAL, LOW (65536-LXXXX)
MMOV   TMRAL, HIGH (65536-LXXXX)
SET    TMRIC.4
JMP    EXIT_TIMER1_ISR_1
EXIT_TIMER1_ISR_1:           ;; 退出 TIMER1 中断服务程序
MOV    A, MP1_STATE
MOV    MP1, A
POP
;;=====
DELAY_300MS:                  ;; 延时 300ms
CLR    DEL1
CLR    DEL0
SDZ    DEL0
JMP    $-1
SDZ    DEL1
JMP    $-3
RET
;;=====

```

程序说明：RC1 ~ RC16 pin 分别对应 1 ~ 16 Touch Switch，KEY_NUMBER 中存储扫描结果，没有按键按下时 KEY_NUMBER 中为 FFH，有按键被按下时 KEY_NUMBER 中为 Touch Switch 编号。另外系统上电的 0.3ms 内请勿碰触 Touch Switch，这段时间系统要扫描参考频率下 (Touch Switch off) 的 TIMERB COUNT 值。