Shock and Mute Pager Applications Using Accelerometers

by: C.S. Chua
Sensor Application Engineering, Singapore, A/P

INTRODUCTION

In the current design, whenever there is an incoming page, the buzzer will “beep” until any of the buttons is depressed. It can sometimes be quite annoying or embarrassing when the button is not within your reach. This application note describes the concept of muting the “beeping” sound by tapping the pager lightly, which could be located in your pocket or handbag. This demo board uses an accelerometer, microcontroller hardware/software and a piezo audio transducer. Due to the wide frequency response of the accelerometer from d.c. to 400 Hz, the device is able to measure both the static acceleration from the Earth's gravity and the shock or vibration from an impact. This design uses a 40g accelerometer, which yields a minimum acceleration range of -40g to +40g.

CONCEPT OF TAP DETECTION

To measure the tapping of a pager, the accelerometer must be able to respond in the range of hundreds of hertz. During the tapping of a pager at the top surface, illustrated in Figure 1, the accelerometer will detect a negative shock level between -15g to -50g of force depending on the intensity. Similarly, if the tapping action comes from the bottom of the accelerometer, the output will be a positive value. Normally, the peak impact pulse is in the order of a few milliseconds. Figure 2 shows a typical waveform of the accelerometer under shock.

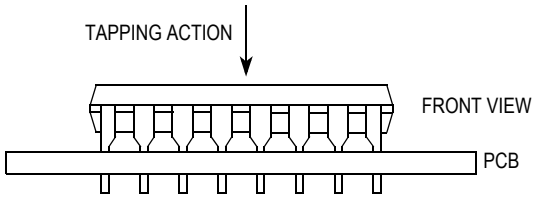

Figure 1. Tapping Action of Accelerometer

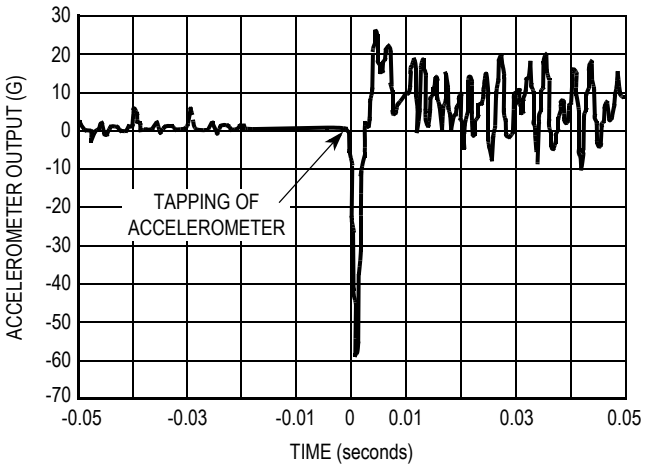

Figure 2. Typical Waveform of Accelerometer Under Tapping Action

Therefore, we could set a threshold level, either by hardware circuitry or software algorithm, to determine the tapping action and mute the “beeping.” In this design, a hardware solution is used because there will be minimal code added to the existing pager software. However, if a software solution is used, the user will be able to program the desire shock level.

HARDWARE DESCRIPTION AND OPERATION

Since MMA1201P is fully signal-conditioned by its internal op-amp and temperature compensation, the output of the accelerometer can be directly interfaced with a comparator. To simplify the hardware, only one direction (tapping on top of the sensor) is monitored. The comparator is configured in such a way that when the output voltage of the accelerometer is less than the threshold voltage or Vref (refer to Figure 3), the output of the comparator will give a logic 1, illustrated in Figure 4. To decrease the V_{REF} voltage or increase the threshold impact in magnitude, turn the trimmer R2 anti-clockwise.

© Freescale Semiconductor, Inc., 2006. All rights reserved.

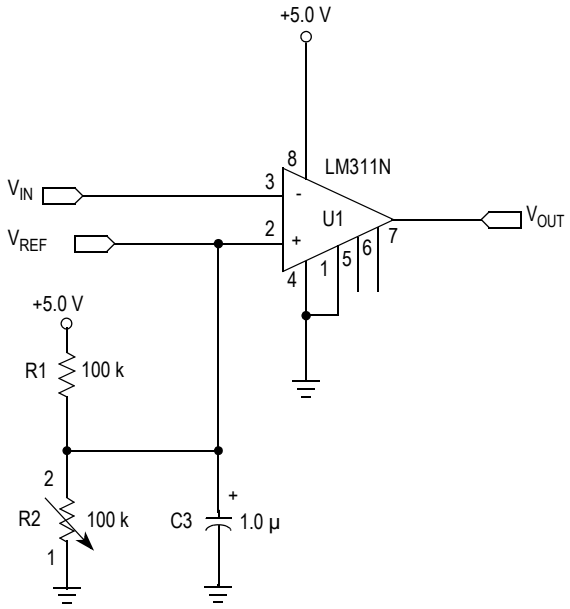

Figure 3. Comparator Circuitry

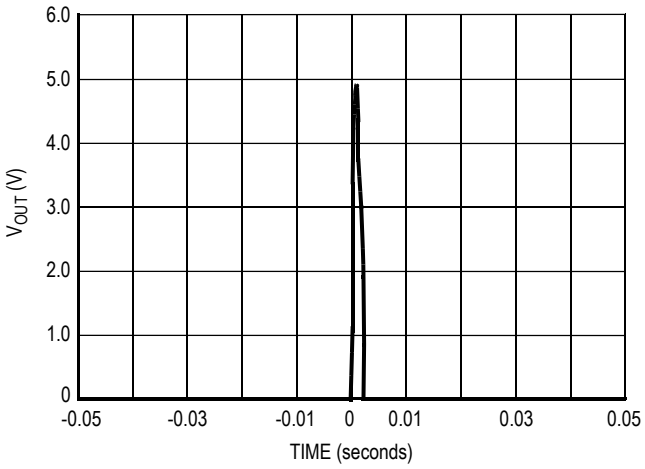

Figure 4. Comparator Output Waveform

For instance, if the threshold level is to be set to -20g, this will correspond to a Vref voltage of 1.7 V.

$$V_{REF} = V_{OFFSET} + \left(\frac{\Delta V}{\Delta G} \times G_{THRESHOLD}\right)$$

$$= 2.5 + (0.04 \times [-20])$$
$$= 1.7\ V$$

Under normal condition, V_{IN} (which is the output of the accelerometer) is at about 2.5 V. Since V_{IN} is higher than Vref, the output of the comparator is at logic 0. During any shock or impact which is greater than -20g in magnitude, the output voltage of the accelerometer will go below V_{REF}. In this case, the output logic of the comparator changes from 0 to 1.

When the pager is in silence mode, the vibrator produces an output of about ±2g. This will not trigger the comparator. Therefore, even in silence mode, the user can also tap the pager to stop the alert. Refer to Figure 5 for the vibrator waveform.

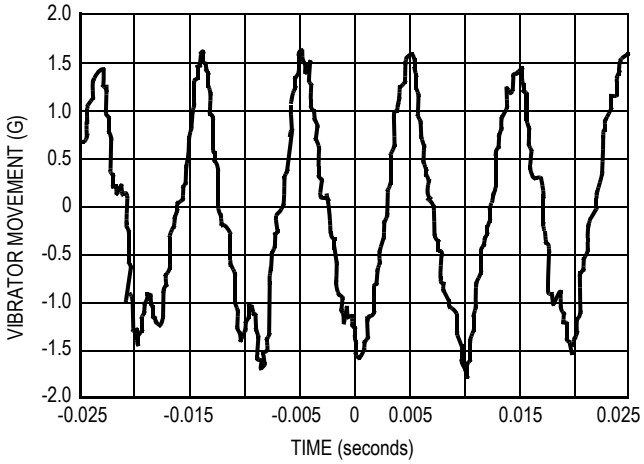

Figure 5. Vibrator Waveform

Figure 6 is a schematic drawing of the whole demo and Figure 7, Figure 8, and Figure 9 show the printed circuit board and component layout for the shock and mute pager. Table 1 is the corresponding part list.

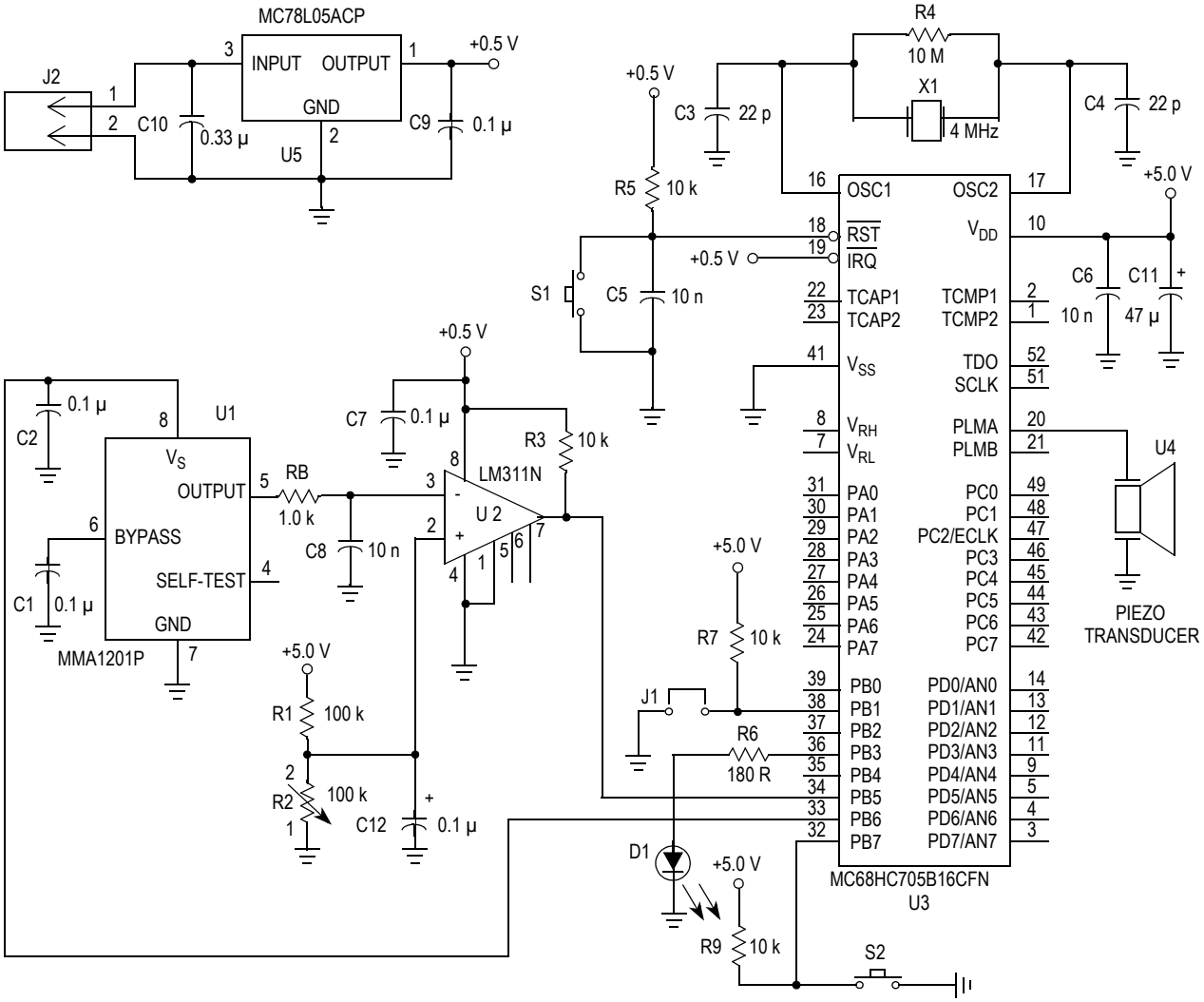

Figure 6. Overall Schematic Diagram of the Demo

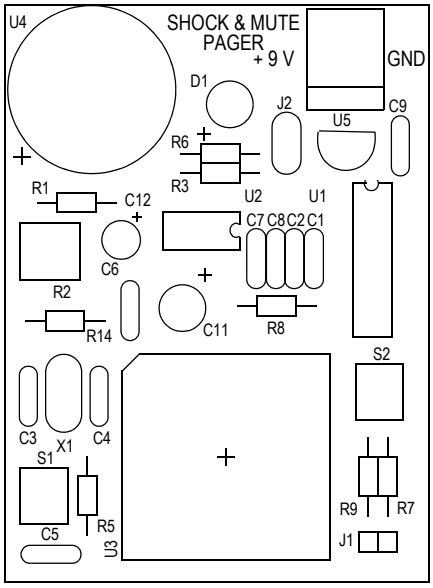

Figure 7. Silk Screen of the PCB

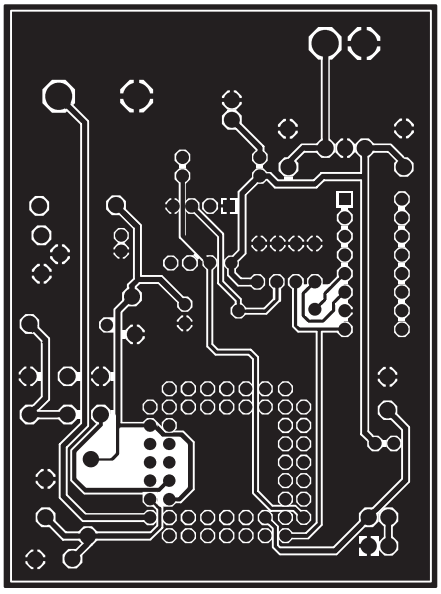

Figure 8. Solder Side of the PCB

Table 1. Bill of Material for the Shock and Mute Pager

Device Type	Qty.	Value	References
Ceramic Capacitor	4	0.1µ	C1, C2, C7, C9
Ceramic Capacitor	2	22p	C3, C4
Ceramic Capacitor	3	10n	C5, C6, C8
Solid Tantalum	1	0.33µ	C10
Electrolytic Capacitor	1	47µ	C11
Electrolytic Capacitor	1	1µ	C12
LED	1	5mm	D1
Header	1	2 way	J1
PCB Terminal Block	1	2 way	J2
Resistor ±5% 0.25W	1	100k	R1
Single Turn Trimmer	1	100k	R2
Resistor ±5% 0.25W	4	10k	R3, R5, R7, R9
Resistor ±5% 0.25W	1	10M	R4
Resistor ±5% 0.25W	1	180R	R6
Resistor ±5% 0.25W	1	1k	R8
Push Button	2	6mm	S1, S2
MMA1201P	1	—	U1
LM311N	1	—	U2
MC68HC705B16CFN	1	—	U3
Piezo Transducer	1	—	U4
MC78L05ACP	1	—	U5
Crystal	1	4MHz	X1

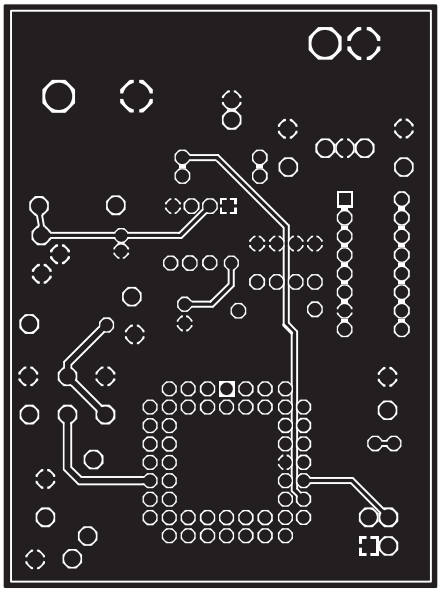

Figure 9. Component Side of the PCB

SOFTWARE DESCRIPTION

Upon powering up the system, the piezo audio transducer is activated simulating an incoming page, if the pager is in sound mode (jumper J1 in ON). Then, the accelerometer is powered up and the output of the comparator is sampled to obtain the logic level. The “beeping” will continue until the accelerometer senses an impact greater than the threshold level. Only then the alert is muted. However, when the pager is in silence mode (jumper J1 is OFF), indicated by the blinking red LED, the accelerometer is not activated. To stop the alert, press the push-button S2.

To repeat the whole process, simply push the reset switch S1.

Figure 10 is a flowchart for the program that controls the system.

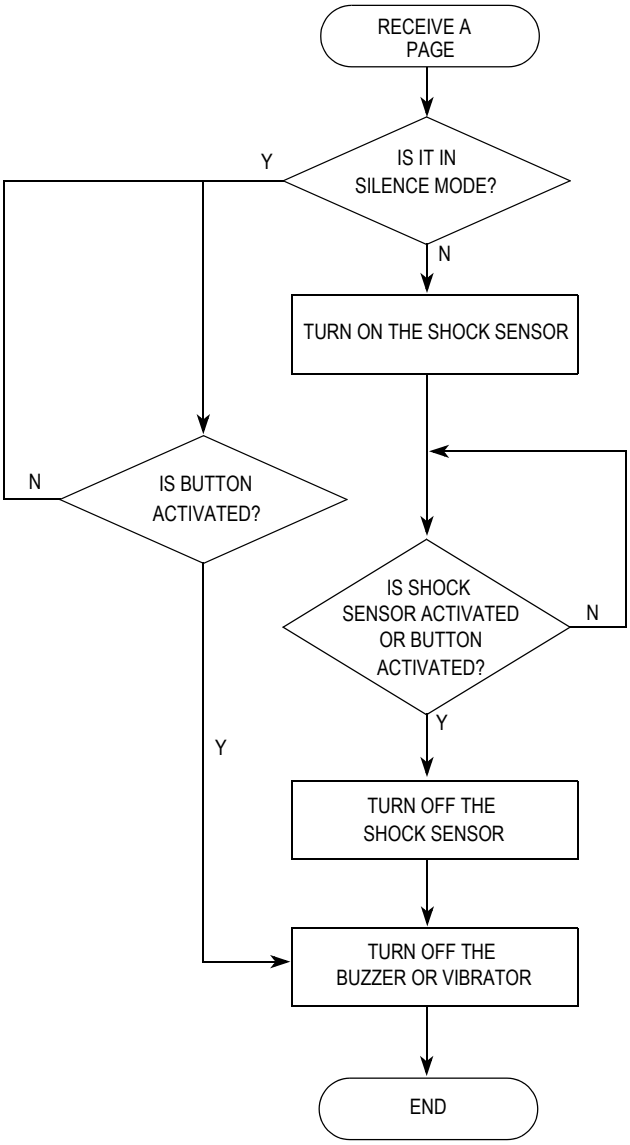

Figure 10. Main Program Flowchart

CONCLUSION

The shock and mute pager design uses a comparator to create a logic level output by comparing the accelerometer output voltage and a user-defined reference voltage. The flexibility of this minimal component, high performance design makes it compatible with many different applications, e.g. hard disk drive knock sensing, etc. The design presented here uses a comparator which yields excellent logic-level outputs and output transition speeds for many applications.

SOFTWARE SOURCE/ASSEMBLY PROGRAM CODE

```
*****************************************************************************
*                                                                           *
*****************************************************************************
*                                                                           *
*                  Pager Shock & Mute Detection Version 1.0                 *
*                                                                           *
*   The following code is written for MC68HC705B16 using MMDS05 software    *
*   Version 1.01                                                            *
*   CASM05 - Command line assembler Version 3.04                            *
*   P & E Microcomputer Systems, Inc.                                       *
*                                                                           *
*                         Written by : C.S. Chua                            *
*                             9th January 1997                              *
*                                                                           *
*                           Software Description                            *
*                                                                           *
*   J1 ON - Sound mode                                                      *
*   Buzzer will turn off if the accelerometer is tapped or switch S2 is     *
*   depressed.                                                              *
*                                                                           *
*   J1 OFF - Silence mode                                                   *
*   LED will turn off if and only if S2 is depressed                        *
*                                                                           *
*****************************************************************************
****************************************
*                                      *
*            I/O Declaration           *
*                                      *
****************************************
PORTB         EQU     $01           ; Port B
PLMA          EQU     $0A           ; D/A to control buzzer
TCONTROL      EQU     $12           ; Timer control register
TSTATUS       EQU     $13           ; Timer Status Register
OCMPHI1       EQU     $16           ; Output Compare Register 1 High Byte
OCMPLO1       EQU     $17           ; Output Compare Register 1 Low Byte
TCNTHI        EQU     $18           ; Timer Count Register High Byte
TCNTLO        EQU     $19           ; Timer Count Register Low Byte
OCMPHI2       EQU     $1E           ; Output Compare Register 2 High Byte
OCMPLO2       EQU     $1F           ; Output Compare Register 2 Low Byte
****************************************
*                                      *
*        RAM Area ($0050 - $0100)      *
*                                      *
****************************************
              ORG     $50
STACK         RMB     4             ; Stack segment
TEMPTCNTLO    RMB     1             ; Temp. storage of timer result (LSB)
TEMPTCNTHI    RMB     1             ; Temp. storage of timer result (MSB)
****************************************
*                                      *
*        ROM Area ($0300 - $3DFD)      *
*                                      *
****************************************
              ORG     $300
****************************************
*                                      *
*   Program starts here upon hard reset*
*                                      *
****************************************
RESET         CLR     PORTB             ; Initialise Ports
              LDA     #%01001000        ; Configure Port B
              STA     $05
              LDA     TSTATUS           ; Dummy read the timer status register so as to clear the OCF
              CLR     OCMPHI2
              CLR     OCMPHI1
              LDA     OCMPLO2
              JSR     COMPRGT
              LDA     #$40              ; Enable the output compare interrupt
              STA     TCONTROL
              LDA     #10               ; Idle for a while before "beeping"
IDLE          JSR     DLY20
              DECA
              BNE     IDLE
              CLI                       ; Interrupt begins here
              BRSET   1,PORTB,SILENCE   ; Branch if J1 is off
              BSET    6,PORTB           ; Turn on accelerometer
              JSR     DLY20             ; Wait till the supply is stable
TEST          BRSET   5,PORTB,MUTE      ; Sample shock sensor for tapping
              BRCLR   7,PORTB,MUTE      ; Sample switch S2 for muting
              JMP     TEST
MUTE          BCLR    6,PORTB           ; Turn off accelerometer
              SEI
              CLR     PLMA              ; Turn off buzzer
```

```
DONE          JMP      DONE              ; End
SILENCE       BRSET    7,PORTB,SILENCE   ; Sample switch S2 for stopping LED
              SEI
              BCLR     3,PORTB           ; Turn off LED
              JMP      DONE              ; End
*****************************************
*                                       *
*       Timer service interrupt         *
*       Alternates the PLMA data        *
*       and bit 3 of Port B             *
*                                       *
*****************************************
TIMERCMP      BSR      COMPRGT           ; Branch to subroutine compare register
              BRSET    1,PORTB,SKIPBUZZER  ; Branch if J1 is OFF
              LDA      PLMA
              EOR      #$80              ; Alternate the buzzer
              STA      PLMA
              RTI
SKIPBUZZER    BRSET    3,PORTB,OFF_LED     ; Alternate LED supply
              BSET     3,PORTB
              RTI
OFF_LED       BCLR     3,PORTB
              RTI
*****************************************
*                                       *
*           Subroutine reset            *
*      the timer compare register       *
*                                       *
*****************************************
COMPRGT       LDA      TCNTHI            ; Read Timer count register
              STA      TEMPTCNTHI        ; and store it in the RAM
              LDA      TCNTLO
              STA      TEMPTCNTLO
              ADD      #$50              ; Add C350 H = 50,000 periods
              STA      TEMPTCNTLO        ; with the current timer count
              LDA      TEMPTCNTHI        ; 1 period = 2 us
              ADC      #$C3
              STA      TEMPTCNTHI        ; Save the next count to the register
              STA      OCMPHI1
              LDA      TSTATUS           ; Clear the output compare flag
              LDA      TEMPTCNTLO        ; by access the timer status register
              STA      OCMPLO1           ; and then access the output compare register
              RTS
*****************************************
*                                       *
*       Delay Subroutine for 0.20 sec   *
*                                       *
*   Input: None                         *
*   Output: None                        *
*                                       *
*****************************************
DLY20         STA      STACK+2
              STX      STACK+3
              LDA      #!40           ; 1 unit = 0.7725 mS
OUTLP         CLRX
INNRLP        DECX
              BNE      INNRLP
              DECA
              BNE      OUTLP
              LDX      STACK+3
              LDA      STACK+2
              RTS
*****************************************
*                                       *
*   This subroutine provides services   *
*    for those unintended interrupts    *
*                                       *
*****************************************
SWI           RTI                     ; Software interrupt return
IRQ           RTI                     ; Hardware interrupt
TIMERCAP      RTI                     ; Timer input capture
TIMERROV      RTI                     ; Timer overflow interrupt
SCI           RTI                     ; Serial communication Interface Interrupt
              ORG      $3FF2          ; For 68HC05B16, the vector location
              FDB      SCI            ; starts at 3FF2
              FDB      TIMERROV       ; For 68HC05B5, the address starts at 1FF2
              FDB      TIMERCMP
              FDB      TIMERCAP
              FDB      IRQ
              FDB      SWI
              FDB      RESET
```

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including “Typicals”, must be validated for each customer application by customer’s technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.

© Freescale Semiconductor, Inc. 2006. All rights reserved.

AN1612
Rev 5
10/2006