Order this document
by AN1612/D

AN1612

Shock and Mute Pager Applications Using Accelerometer

Prepared 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 be quite annoying or embarrassing sometime 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 400Hz, 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 (Motorola P/N: MMA1000P) 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, which is 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.

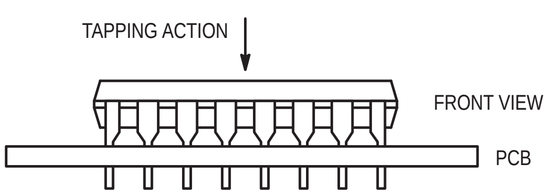

Figure 1. Tapping Action of Accelerometer

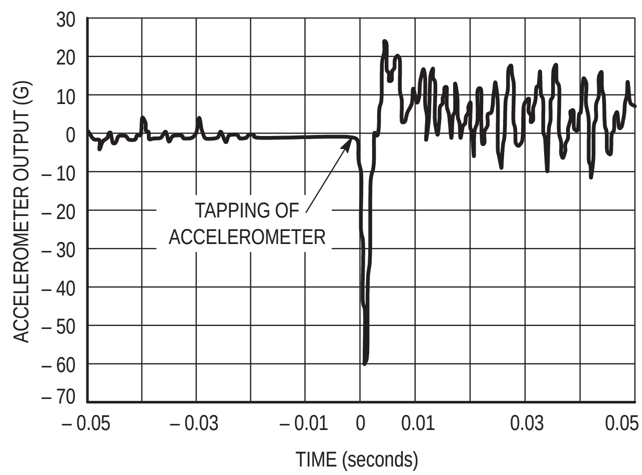

Figure 2. Typical Waveform of Accelerometer Under Tapping Action

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 MMA1000P 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" which is illustrated in Figure 4. To decrease the Vref voltage or increase the threshold impact in magnitude, turn the trimmer R2 anti–clockwise.

REV 2

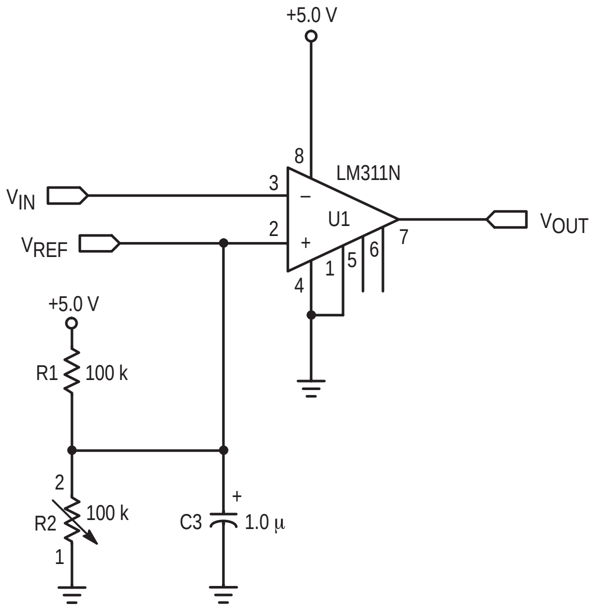

Figure 3. Comparator Circuitry

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

$$
\begin{aligned}
V_{REF} &= V_{OFFSET} + \left(\frac{\Delta V}{\Delta G} \times G_{THRESHOLD}\right) \\
&= 2.5 + (0.04 \times [-20]) \\
&= 1.7\ V
\end{aligned}
$$

Under normal condition, Vin (which is the output of the accelerometer) is at about 2.5V. Since Vin 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 Vref. 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.

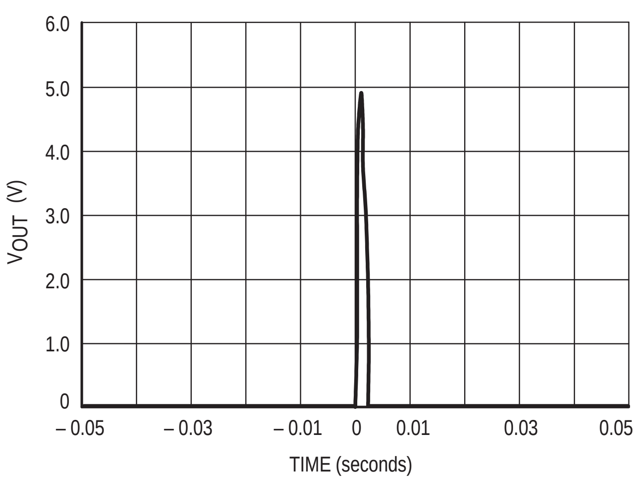

Figure 4. Comparator Output Waveform

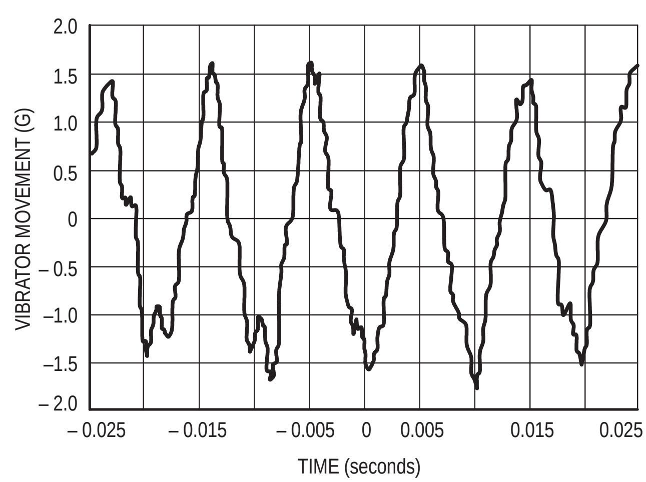

Figure 5. Vibrator Waveform

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

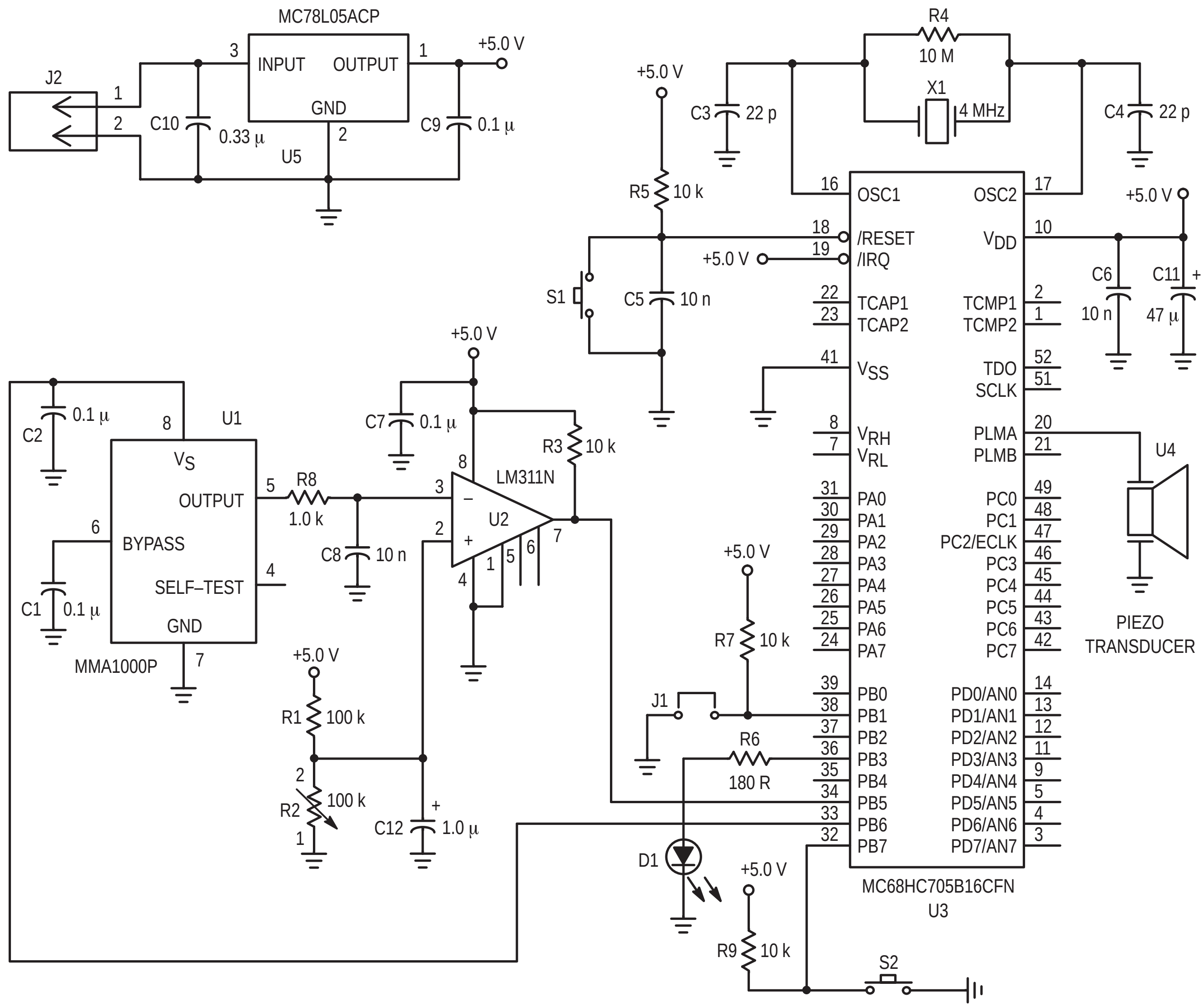

Figure 6. Overall Schematic Diagram of the Demo

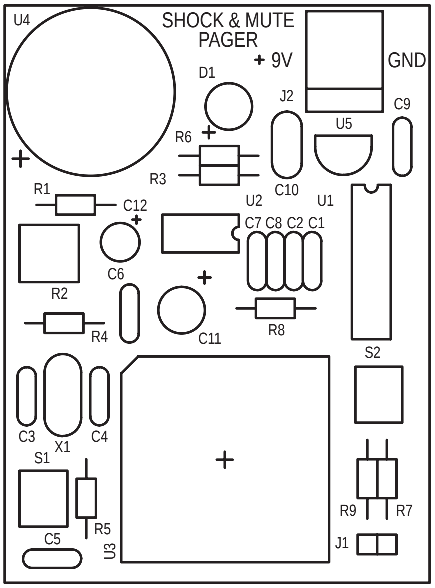

Figure 7. Silk Screen of the PCB

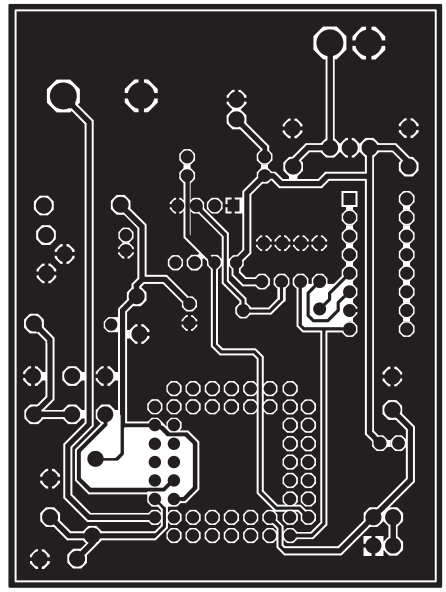

Figure 8. Solder Side of the PCB

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
MMA1000P	1	—	U1
LM311N	1	—	U2
MC68HC705B16CFN	1	—	U3
Piezo Transducer	1	—	U4
MC78L05ACP	1	—	U5
Crystal	1	4MHz	X1

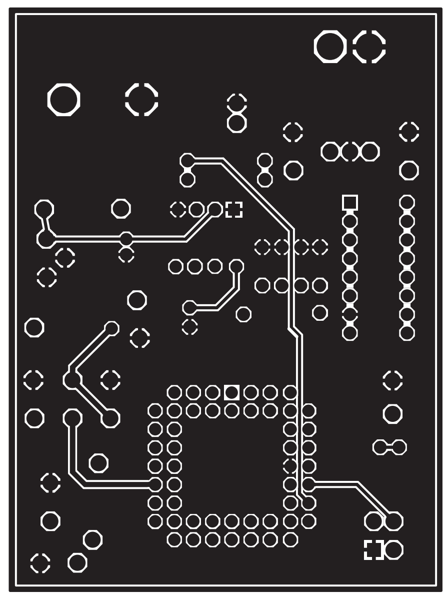

Figure 9. Component Side of the PCB

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), which is 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.

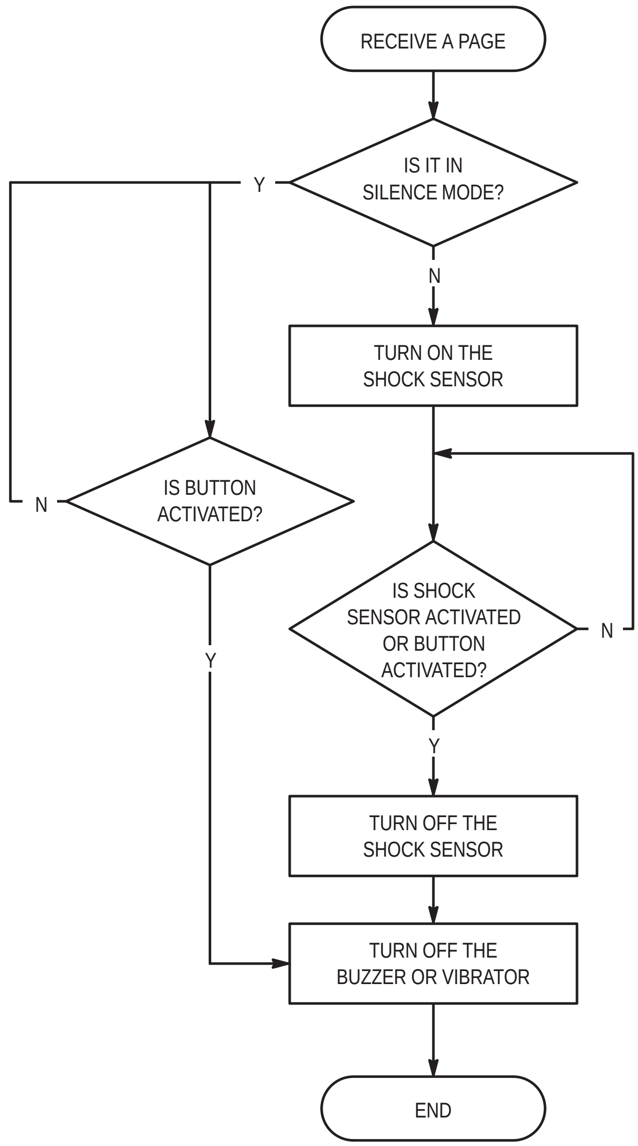

Figure 10. Main Program Flowchart

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:
USA / EUROPE / Locations Not Listed: Motorola Literature Distribution;
P.O. Box 5405, Denver, Colorado 80217. 1–303–675–2140 or 1–800–441–2447

JAPAN: Nippon Motorola Ltd.; SPD, Strategic Planning Office, 141,
4–32–1 Nishi–Gotanda, Shinagawa–ku, Tokyo, Japan. 81–3–5487–8488

Customer Focus Center: 1–800–521–6274

MfaxTM: RMFAX0@email.sps.mot.com – TOUCHTONE 1–602–244–6609
Motorola Fax Back System – US & Canada ONLY 1–800–774–1848
– http://sps.motorola.com/mfax/

ASIA / PACIFIC: Motorola Semiconductors H.K. Ltd.; 8B Tai Ping Industrial Park,
51 Ting Kok Road, Tai Po, N.T., Hong Kong. 852–26629298

HOME PAGE: http://motorola.com/sps/

MOTOROLA