AN519

Implementing a Simple Serial Mouse Controller

INTRODUCTION

The mouse is becoming increasingly popular as a standard pointing data entry device. There is no doubt that the demand for the mouse is increasing. Various kinds of mice can be found in the market, including optical mice, opto-mechanical mice, and their close relative, trackballs. The mouse interfaces to the host via a dedicated interface card or an RS-232 port. Their mechanisms are very similar. The major electrical components of a mouse are:

- Microcontroller
- Photo-transistors
- Infrared emitting diode
- Voltage conversion circuit

The intelligence of the mouse is provided by the microcontroller, therefore the features and performance of a mouse is greatly related to the microcontroller used.

This application note describes the implementation of a serial mouse using the PIC16C54. The PIC16C54 is a high speed 8-bit CMOS microcontroller offered by Microchip Technology Inc. It is an ideal candidate for a mouse controller.

THEORY OF OPERATION

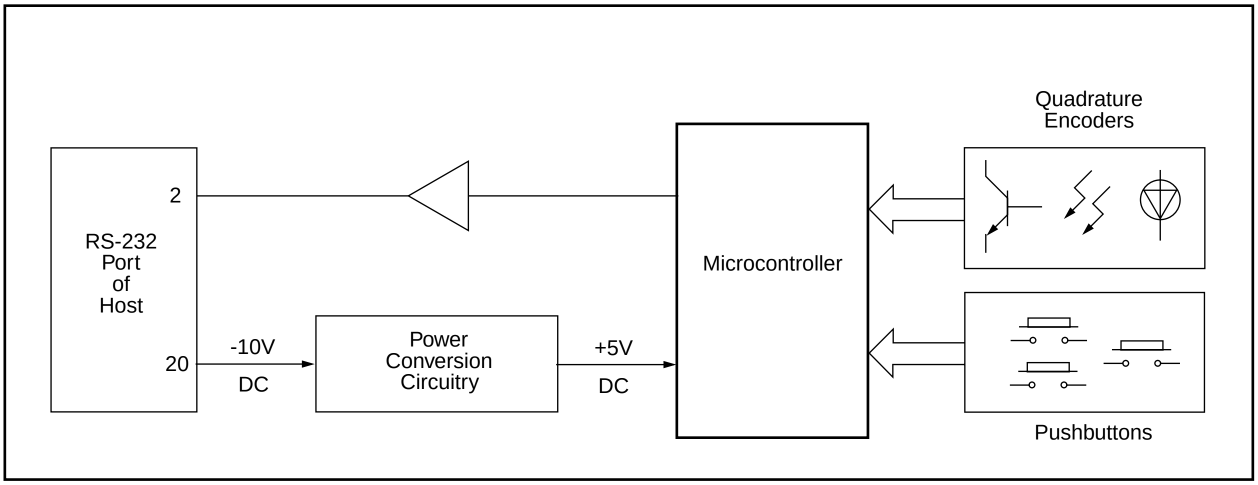
A mouse can be divided into several functional blocks:

- Microcontroller
- Button detection
- Motion detection
- RS-232 signal generation
- 5V DC power supply unit

A typical functional block diagram is shown in Figure 1.

In Figure 2, three pushbuttons are connected to the input ports of the PIC16C54. When a switch opening or closure is detected, a message is formatted and sent to the host. The X and Y movements are measured by counting the pulses generated by the photo-couplers. In the case of an opto-mechanical mouse, the infrared light emitted by the infrared diode is blocked by the rotating wheel, so that the pulses are generated on the photo-transistor side. In case of an optical mouse, the infrared light emitted by the infrared diode is reflected off the reflective pad patterned with vertical and horizontal grid lines. It is then received by the photo-transistor in the mouse. When any X or Y movement is detected, a message is formatted and sent to the host.

FIGURE 1: FUNCTIONAL BLOCKS OF A SERIAL MOUSE



The Microsoft® Mouse System and the Mouse Systems® device both use serial input techniques. The Mouse System protocol format contains five bytes of data. One byte describes the status of three push buttons, two bytes for the relative X movements and two bytes for the relative Y movements. The Microsoft protocol format contains three bytes of data describing the status of two push buttons and the relative X and Y movements. The details of these protocols are given in Table 1.

Three lines are connected to the host via the RS-232 port:

- Signal Ground
- Received Data
- Request to Send

"Received Data" carries the message sent by the mouse. While "Request to Send" provides a –10 VDC for voltage conversion circuitry. A voltage of +5 VDC is required for electronic components inside the mouse, however, +5 VDC is not part of an RS-232 port, so voltage conversion circuitry is required. This circuit is typically composed of a 555 timer, Zener diodes, and capacitors.

An example circuit is shown in Figure 3. Since the current supplied through the RS-232 port is limited to 10 mA, the mouse cannot be designed to consume more than 10 mA current unless an external power supply is provided. The PIC16C54, running at 4 MHz (1 µs instruction cycle) can provide a very high tracking speed. An 8 MHz version of PIC16C54 is also available if higher performance is desired.

FIGURE 2: PIC16C54 PIN ASSIGNMENTS

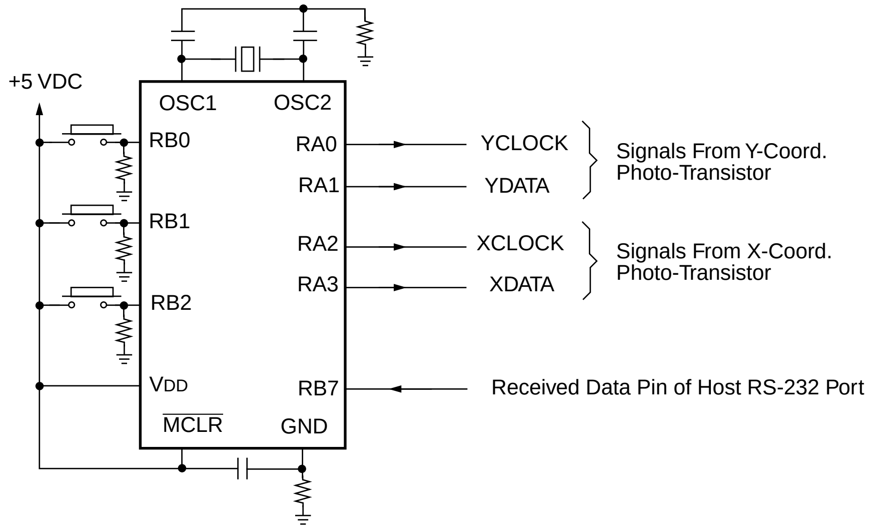
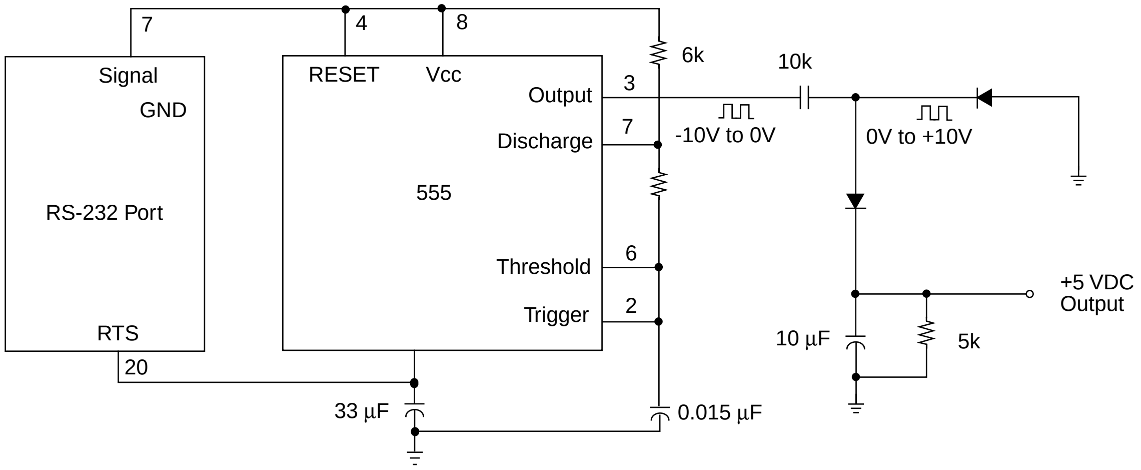

+5 VDC
OSC1
OSC2
RB0
RB1
RB2
VDD
MCLR
RA0
RA1
RA2
RA3
RB7
GND
YCLOCK
YDATA
XCLOCK
XDATA
Signals From Y-Coord. Photo-Transistor
Signals From X-Coord. Photo-Transistor
Received Data Pin of Host RS-232 Port

FIGURE 3: VOLTAGE CONVERSION CIRCUITRY



ABOUT THE SOFTWARE

The major tasks performed by the software are button scanning, X and Y motion scanning, formatting and sending serial data to the host. These tasks need to be performed in parallel in order to gain better tracking speed. The pulses generated by the photo-couplers are counted while transmitting the serial signals to the RS-232 port. The number of pulses reflects the speed of the movement. The more pulses, the faster the movement.

The directions of movement are determined by the last states and the present states of the outputs of the photo-transistors. In Figure 4, XCLOCK and XDATA are outputs from the photo-transistors corresponding to the X-axis movement. XDATA is read when a rising or a falling edge of XCLOCK is detected. For right movement, XDATA is either LOW at the rising edge of XCLOCK or HIGH at the falling edge of XCLOCK. The up and down movement detections follow the same logic. In Table 1, X7:X0 are data for relative movement. If X is positive, it implies that the mouse is moving to the right. If X is negative, it implies a movement to the left. Similarly, if Y is positive, it indicates that the mouse is moving down and if Y is negative, it indicates that the mouse is moving up. The pulses generated by the photo-couplers are checked before every bit is sent. A bit takes 1/1200 second to send, if the distance between the grid lines is 1 mm, the tracking speed will be up to 1200 mm/second.

FIGURE 4: VOLTAGE CONVERSION CIRCUITRY

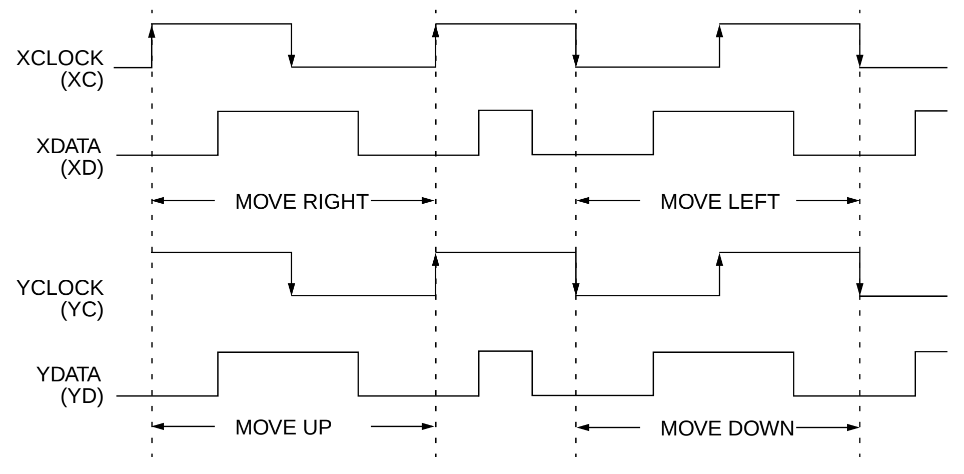


TABLE 1: MOUSE SYSTEM AND MICROSOFT PROTOCOLS

	Mouse System Format*								Microsoft Format*							
Bit Position	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Byte 1	1	0	0	0	0	L	M	R	1	1	L	R	Y7	Y6	X7	X6
Byte 2	X7	X6	X5	X4	X3	X2	X1	X0	0	0	X5	X4	X3	X2	X1	X0
Byte 3	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0	0	0	Y5	Y4	Y3	Y2	Y1	Y0
Byte 4	X7	X6	X5	X4	X3	X2	X1	X0								
Byte 5	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0								

* L = Left Key Status
M = Middle Key Status
R = Right Key Status
1 = Pressed
0 = Released
X7-X0 = X-Axis Movement Data
Y7-Y0 = Y-Axis Movement Data

The buttons are scanned after a message is sent and the time used to send the message is used as the debouncing time. The message is in an RS-232 format with 1200 baud, eight data bits, no parity, and two stop bits.

The flowcharts of the main program, subroutine `BYTE` and subroutine `BIT` are shown in Figure 5, Figure 6, and Figure 7. Figure 5 shows that the trigger flag is set when any change in button status or X/Y movement is detected. Subroutine `BYTE` is called in the main program five times to send five bytes of information. Subroutine `BYTE` controls the status of the “Received Data” (RD) pin. If Trigger Flag is clear, RD will always be HIGH. Hence, no message will be sent even when subroutine `BYTE` is called. Figure 7 shows that subroutine `BIT` counts the number of pulses from the outputs of the photo-transistors, determines the directions, and generates a 1/1200 second delay to get 1200 baud timing.

The mouse has been tested in Mouse System Mode and functions properly. The setup and software have been tested and function within the given guidlines. A listing of the source program is given in Appendix A.

SUMMARY

The PIC16C54 from Microchip Technology Inc. provides a very cost-effective, high performance mouse implementation. Its low power consumption (typically< 2 mA at 1 µs instruction cycle), small package (18-pin) and high reliability (on-chip watchdog timer to prevent software hang-ups) are some of the many reasons why the PIC16C54 is uniquely suitable for mouse applications.

Note: This application note provides the user with a simple, fully functional serial mouse implementation. The user may use this as a starting point for a more comprehensive design.

FIGURE 5: FLOWCHART OF THE MAIN PROGRAM

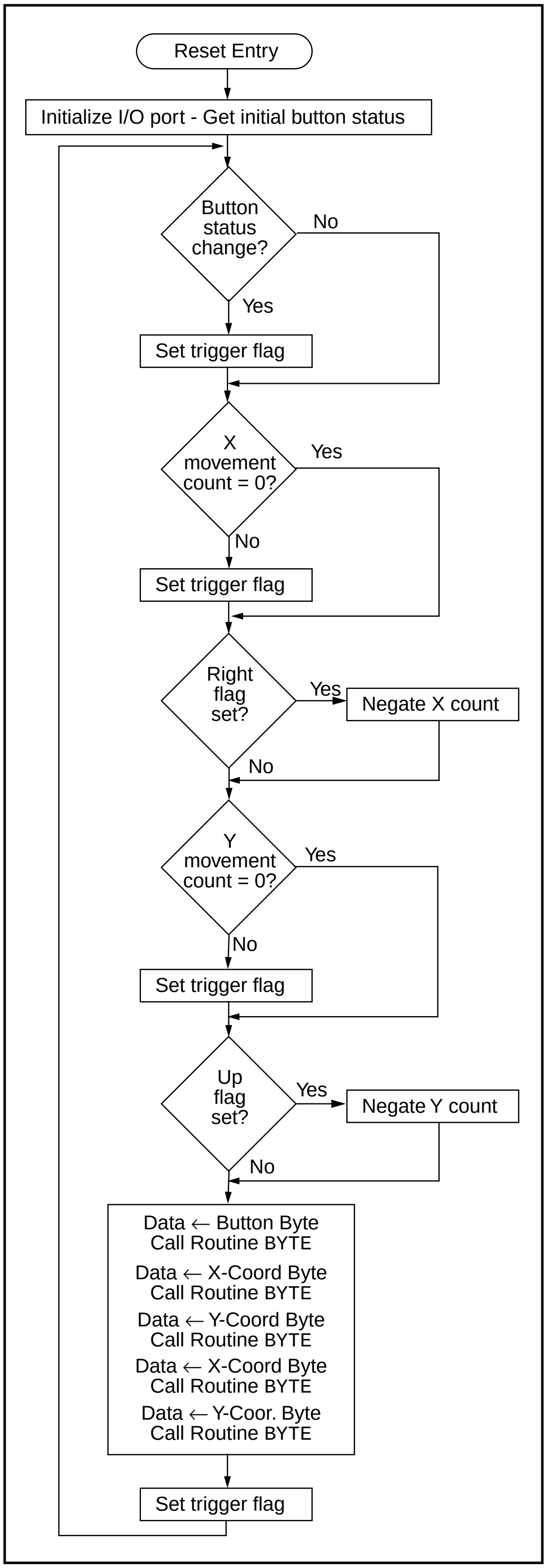


FIGURE 6: FLOWCHART OF ROUTINE BYTE

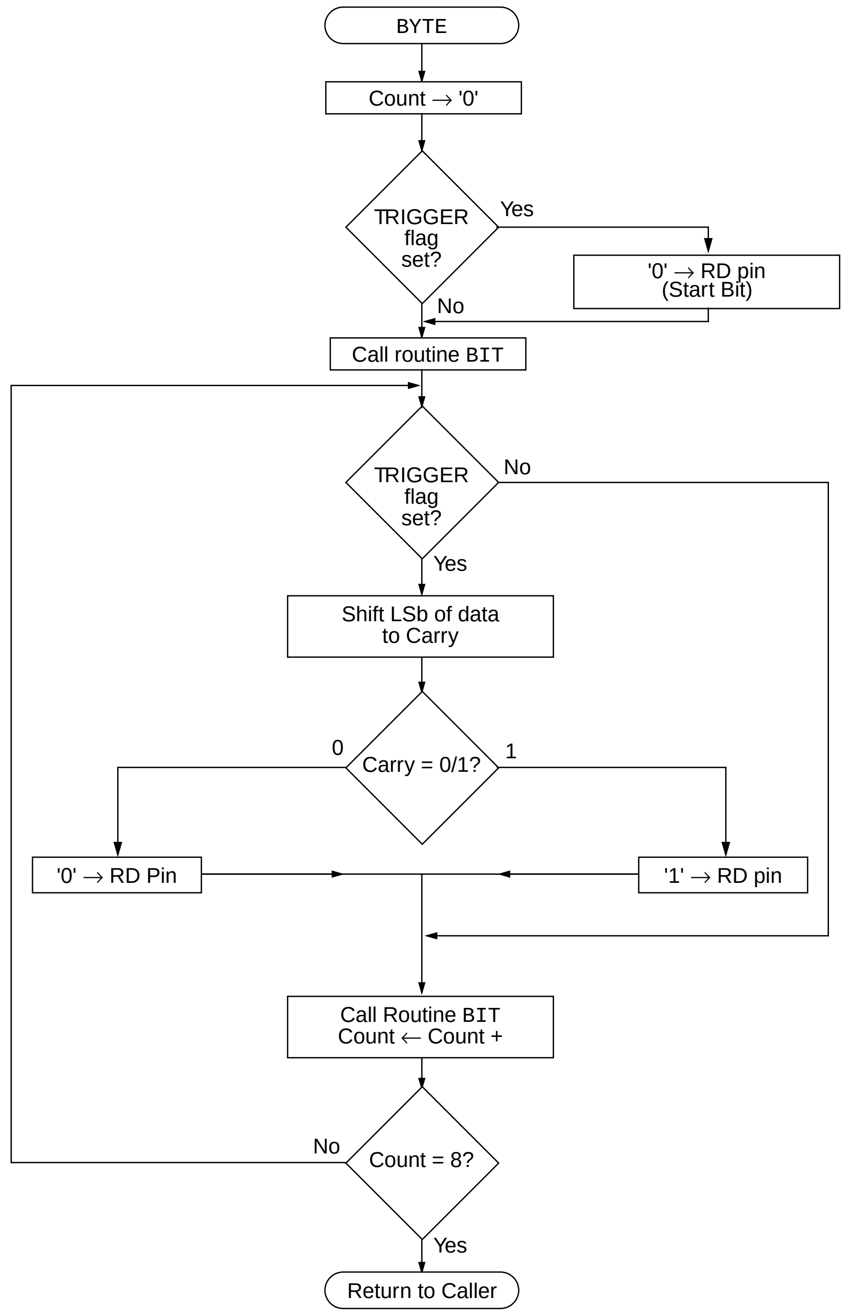
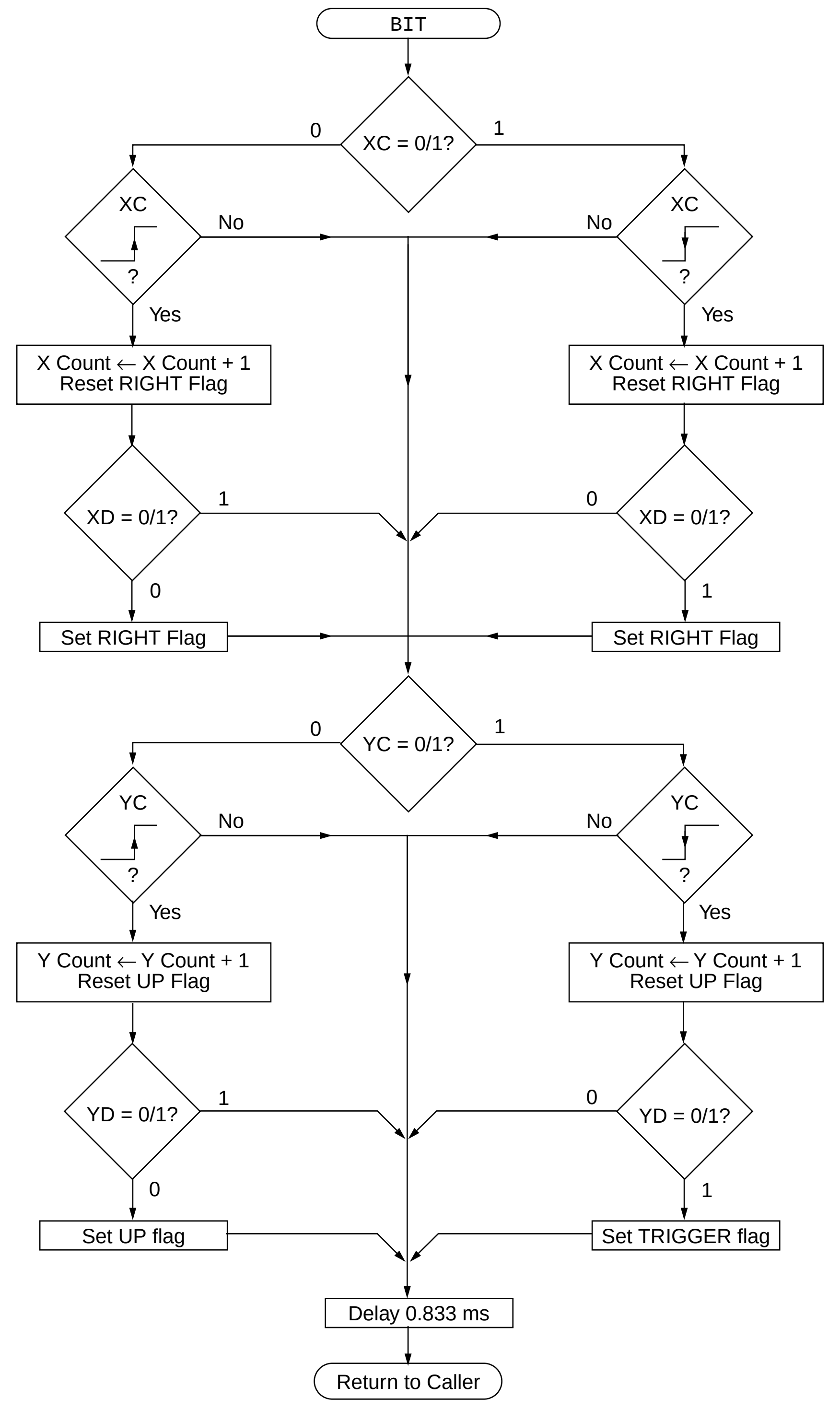


FIGURE 7: FLOWCHART OF ROUTINE BIT



Please check the Microchip BBS for the latest version of the source code. Microchip’s Worldwide Web Address: www.microchip.com; Bulletin Board Support: MCHIPBBS using CompuServe® (CompuServe membership not required).

APPENDIX A: MOUSE.ASM

```
MPASM 01.40 Released            MOUSE.ASM   1-16-1997  12:44:22         PAGE  1


LOC  OBJECT CODE     LINE SOURCE TEXT
  VALUE

                    00001  TITLE “ MOUSE “
                    00002  LIST   P=16C54,R=O
                    00003 ;
                    00004 ;
                    00005 ;********************************************************************
                    00006 ;
                    00007 ;    MOUSE CONTROLLER
                    00008 ;    PIC MODE = PIC16C54XT   CLK=4.0MHZ
                    00009 ;
                    00010 ;       Program:          MOUSE.ASM
                    00011 ;       Revision Date:    25 APRIL, 1990
                    00012 ;                         1-13-97      Compatibility with MPASMWIN 1.40
                    00013 ;
                    00014 ;********************************************************************
                    00015 ;
                    00016 ;-------------------------------------------
                    00017 ;       FILES ASSIGNMENT
                    00018 ;-------------------------------------------
                    00019 ;
  00000003          00020 STATUS     EQU 3            ;STATUS REGISTER
  00000005          00021 RA         EQU 5            ;I/O PORT A
  00000006          00022 RB         EQU 6            ;I/O PORT B
  00000008          00023 TIMER1     EQU 10           ;COUNTER FOR DELAY
  0000000C          00024 CSTAT      EQU 14           ;CO-ORDINATE STATUS
  0000000D          00025 BSTAT      EQU 15           ;BUTTON STATUS
  0000000E          00026 DATA0      EQU 16           ;
  0000000F          00027 DATA1      EQU 17           ;
  00000010          00028 DATA2      EQU 20           ;5 BYTE RS232 DATA
  00000011          00029 DATA3      EQU 21           ;
  00000012          00030 DATA4      EQU 22           ;
  00000013          00031 FLAGA      EQU 23           ;GENERAL PURPOSE FLAG
  00000014          00032 XCOUNT     EQU 24           ;X-MOVEMENT COUNTER
  00000015          00033 YCOUNT     EQU 25           ;Y-MOVEMENT COUNTER
  00000016          00034 FLAGB      EQU 26           ;GENERAL PURPOSE FLAG
  00000018          00035 COUNT      EQU 30           ;GENERAL PURPOSE COUNTER
  00000019          00036 DATA_AREA  EQU 31           ;FOR TEMP. STORAGE
                    00037 ;
                    00038 ;------------------------------------------
                    00039 ;       BIT ASSIGNMENT
                    00040 ;------------------------------------------
                    00041 ;
  00000000          00042 YC         EQU 0                ;Y-CLOCK PIN
  00000001          00043 YD         EQU 1                ;Y-DATA PIN
  00000001          00044 UP         EQU 1                ;MOVING UP FLAG
  00000002          00045 XC         EQU 2                ;X-CLOCK PIN
  00000003          00046 XD         EQU 3                ;X-DATA PIN
  00000003          00047 RI         EQU 3                ;MOVING RIGHT FLAG
  00000000          00048 BU1        EQU 0                ;BUTTON #1 PIN
  00000002          00049 BU2        EQU 2                ;BUTTON #2 PIN
  00000000          00050 CA         EQU 0                ;CARRY FLAG
  00000007          00051 RD         EQU 7                ;RECEIVED DATA PIN TO RS232
  00000002          00052 ZERO_AREA  EQU 2                ;ZERO FLAG
  00000002          00053 TR         EQU 2                ;TIGGER FLAG
                    00054 ;
  00000001          00055 F          EQU 1
                    00056 ;
                    00057 ;===========================================
                    00058 ;       SUBROUTINES
                    00059 ;===========================================
                    00060 ;
                    00061 ;*******************************************
0000                00062  ORG 0
                    00063 ;*******************************************
                    00064 ;
                    00065 ;===========================================
                    00066 ; DELAY A BIT TIME AND CHECK XC & YC STATUS
                    00067 ;===========================================
0000                00068 BIT
0000 0745           00069  BTFSS   RA,XC            ;XC = 1 ?
0001 0A0A           00070  GOTO    BIT0
0002 064C           00071  BTFSC   CSTAT,XC         ;(XC=1)
0003 0A11           00072  GOTO    BITY             ;(XC ALWAYS = 1)
0004 02B4           00073  INCF    XCOUNT, F        ;(XC --|__)
0005 0476           00074  BCF     FLAGB,RI         ;DEFAULT LEFT
0006 0765           00075  BTFSS   RA,XD            ;LEFT / RIGHT ?
0007 0A11           00076  GOTO    BITY
0008 0576           00077  BSF     FLAGB,RI
0009 0A11           00078  GOTO    BITY
000A                00079 BIT0
000A 074C           00080  BTFSS   CSTAT,XC          ;(XC=0)
000B 0A11           00081  GOTO    BITY              ;(XC ALWAYS = 0)
000C 02B4           00082  INCF    XCOUNT, F         ;(XC __|--)
000D 0476           00083  BCF     FLAGB,RI          ;DEFAULT LEFT
000E 0665           00084  BTFSC   RA,XD             ;LEFT / RIGHT ?
000F 0A11           00085  GOTO    BITY
0010 0576           00086  BSF     FLAGB,RI
0011                00087 BITY
0011 0705           00088  BTFSS   RA,YC             ;YC = 1 ?
0012 0A1B           00089  GOTO    BITY0
0013 060C           00090  BTFSC   CSTAT,YC          ;(YC=1)
0014 0A22           00091  GOTO    BITDY             ;(YC ALWAYS = 1)
0015 02B5           00092  INCF    YCOUNT, F         ;(YC --|__)
0016 0436           00093  BCF     FLAGB,UP          ;DEFAULT DOWN
0017 0725           00094  BTFSS   RA,YD             ;DOWN / UP ?
0018 0A22           00095  GOTO    BITDY
0019 0536           00096  BSF     FLAGB,UP
001A 0A22           00097  GOTO    BITDY
001B                00098 BITY0
001B 070C           00099  BTFSS   CSTAT,YC          ;(YC=0)
001C 0A22           00100  GOTO    BITDY             ;(YC ALWAYS = 0)
001D 02B5           00101  INCF    YCOUNT, F         ;(YC __|--)
001E 0436           00102  BCF     FLAGB,UP          ;DEFAULT DOWN
001F 0625           00103  BTFSC   RA,YD             ;DOWN / UP ?
0020 0A22           00104  GOTO    BITDY
0021 0536           00105  BSF     FLAGB,UP
0022                00106  BITDY
0022 0205           00107  MOVF    RA,W              ;SAVE COOR. STATUS
0023 002C           00108  MOVWF   CSTAT
0024 0CC1           00109  MOVLW   193D              ;0.833 MS DELAY
0025 0028           00110  MOVWF   TIMER1
0026                00111 BITD0
0026 0000           00112  NOP
0027 02E8           00113  DECFSZ  TIMER1, F
0028 0A26           00114  GOTO    BITD0
0029 0800           00115  RETLW 0
                    00116 ;
                    00117 ;=================================================
                    00118 ;
                    00119 ;************************************************
                    00120 ;*      SUBROUTINE TO SEND A BYTE               *
                    00121 ;*      AS RS232C FORMAT 8,N,1                  *
                    00122 ;************************************************
                    00123 ;
002A                00124 BYTE
002A 0078           00125  CLRF    COUNT             ;RESET 8 BIT COUNT
002B 0753           00126  BTFSS   FLAGA,TR          ;ANY TRIGGER
002C 0A2E           00127  GOTO    BYTE0
002D 04E6           00128  BCF     RB,RD             ;LOW RD FOR START BIT
002E                00129 BYTE0
002E 0900           00130  CALL    BIT
002F                00131 BYTE1
002F 0753           00132  BTFSS   FLAGA,TR          ;ANY TRIGGER ?
0030 0A37           00133  GOTO BYTE3
0031 0339           00134  RRF     DATA_AREA, F      ;SHIFT DATA TO CARRY
0032 0703           00135  BTFSS   STATUS,CA         ;0 / 1 ?
0033 0A36           00136  GOTO    BYTE2
0034 05E6           00137  BSF     RB,RD             ;SEND A 1
0035 0A37           00138  GOTO    BYTE3
0036                00139 BYTE2
0036 04E6           00140  BCF     RB,RD             ;SEND A 0
0037                00141 BYTE3
0037 0900           00142  CALL    BIT
0038 02B8           00143  INCF    COUNT, F
0039 0778           00144  BTFSS   COUNT,3           ;COUNT = 8 ?
003A 0A2F           00145  GOTO    BYTE1
003B 0753           00146  BTFSS   FLAGA,TR          ;ANY TRIGGER ?
003C 0A42           00147  GOTO    BYTE4
003D 04E6           00148  BCF     RB,RD             ;SEND SENT BIT
003E 0900           00149  CALL    BIT
003F 05E6           00150  BSF     RB,RD
0040 0900           00151  CALL    BIT
0041 0A44           00152  GOTO    BYTE5
0042                00153 BYTE4
0042 0900           00154  CALL    BIT
0043 0900           00155  CALL    BIT
0044                00156 BYTE5
0044 0800           00157  RETLW   0
                    00158 ;
                    00159 ;============================================
                    00160 ;       RESET ENTRY
                    00161 ;============================================
                    00162 ;
0045                00163 INIT
0045 0CC1           00164  MOVLW   B’11000001’       ;DISABLE WATCHDOG
0046 0002           00165  OPTION
0047 0C0F           00166  MOVLW   B’00001111’       ;INIT RB0~3 BE INPUTS
0048 0006           00167  TRIS    RB                ;RB4~7 BE OUTPUTS
0049 0CFF           00168  MOVLW   B’11111111’       ;INIT RA0~3 BE INPUTS
004A 0005           00169  TRIS    RA
004B 05E6           00170  BSF     RB,RD             ;HIGH RD PIN
004C 0246           00171  COMF    RB,W              ;GET INIT BUTTON INPUTS
004D 0E05           00172  ANDLW   B’00000101’
004E 0D80           00173  IORLW   B’10000000’
004F 002D           00174  MOVWF   BSTAT
0050 002E           00175  MOVWF   DATA0
0051 0205           00176  MOVF    RA,W
0052 002C           00177  MOVWF   CSTAT
0053 0073           00178  CLRF    FLAGA             ;CLEAR TR FLAG
0054 0074           00179  CLRF    XCOUNT            ;RESET XCOUNT & YCOUNT
0055 0075           00180  CLRF    YCOUNT
0056                00181 SCAN
0056 006F           00182  CLRF    DATA1             ;UPDATE X,Y MOVEMENT DATA
0057 0070           00183  CLRF    DATA2
0058 0071           00184  CLRF    DATA3
0059 0072           00185  CLRF    DATA4
005A 0214           00186  MOVF    XCOUNT,W          ;XCOUNT = 0 ?
```

```
005B 0743           00187  BTFSS   STATUS,ZERO_AREA
005C 0A80           00188  GOTO    WRITX
005D                00189 SCANA
005D 0215           00190  MOVF    YCOUNT,W          ;YCOUNT = 0 ?
005E 0743           00191  BTFSS   STATUS,ZERO_AREA
005F 0A92           00192  GOTO    WRITY
0060                00193 SCANB
0060 0246           00194  COMF    RB,W              ;BUTTON STATUS CHANGE ?
0061 0E05           00195  ANDLW   B’00000101’
0062 0D80           00196  IORLW   B’10000000’
0063 00AD           00197  SUBWF   BSTAT, F
0064 0643           00198  BTFSC   STATUS,ZERO_AREA  ;IF CHANGE THEN TRIGGER
0065 0A6B           00199  GOTO    SCANC             ;(NO CHANGE)
0066 0553           00200  BSF     FLAGA,TR          ;(CHANGE) SET TRIGGER FLAG
0067 0246           00201  COMF    RB,W              ;FORMAT BUTTON STATUS DATA
0068 0E05           00202  ANDLW   B’00000101’
0069 0D80           00203  IORLW   B’10000000’
006A 002E           00204  MOVWF   DATA0
006B                00205 SCANC
006B 0246           00206  COMF    RB,W
006C 0E05           00207  ANDLW   B’00000101’
006D 0D80           00208  IORLW   B’10000000’
006E 002D           00209  MOVWF   BSTAT
006F 020E           00210  MOVF    DATA0,W           ;SEND DATA0,1,2,3,4 TO HOST
0070 0039           00211  MOVWF   DATA_AREA
0071 092A           00212  CALL    BYTE
0072 020F           00213  MOVF    DATA1,W
0073 0039           00214  MOVWF   DATA_AREA
0074 092A           00215  CALL    BYTE
0075 0210           00216  MOVF    DATA2,W
0076 0039           00217  MOVWF   DATA_AREA
0077 092A           00218  CALL    BYTE
0078 0211           00219  MOVF    DATA3,W
0079 0039           00220  MOVWF   DATA_AREA
007A 092A           00221  CALL    BYTE
007B 0212           00222  MOVF    DATA4,W
007C 0039           00223  MOVWF   DATA_AREA
007D 092A           00224  CALL    BYTE
007E 0453           00225  BCF     FLAGA,TR          ;CLEAR TRIGGER FLAG
007F 0A56           00226  GOTO    SCAN
                    00227 ;
0080                00228 WRITX
0080 0553           00229  BSF     FLAGA,TR          ;SET TRIGGER FLAG
0081 0C40           00230  MOVLW   40H               ;IF XCOUNT > 64 THEN XCOUNT <-64
0082 0094           00231  SUBWF   XCOUNT,W
0083 0603           00232  BTFSC   STATUS,CA
0084 0A8D           00233  GOTO    WRITR
0085                00234 WRITS
0085 0776           00235  BTFSS   FLAGB,RI         ;LEFT / RIGHT ?
0086 0A90           00236  GOTO    WRITL
0087 0274           00237  COMF    XCOUNT, F        ;(RIGHT) NEG XCOUNT
0088 0294           00238  INCF    XCOUNT,W
0089                00239 WRITA
0089 002F           00240  MOVWF   DATA1
008A 0031           00241  MOVWF   DATA3
008B 0074           00242  CLRF    XCOUNT           ;RESET XCOUNT
008C 0A5D           00243  GOTO    SCANA
                    00244 ;
008D                00245 WRITR
008D 0C40           00246  MOVLW  40H               ;XCOUNT <- 64
008E 0034           00247  MOVWF  XCOUNT
008F 0A85           00248  GOTO   WRITS
                    00249 ;
0090                00250 WRITL
0090 0214           00251  MOVF   XCOUNT,W          ;(LEFT)
0091 0A89           00252  GOTO   WRITA
```

```
                    00253 ;
0092                00254 WRITY
0092 0553           00255  BSF    FLAGA,TR          ;SET TRIGGER FLAG
0093 0C40           00256  MOVLW  40H               ;IF YCOUNT > 64 THEN YCOUNT <-64
0094 0095           00257  SUBWF  YCOUNT,W
0095 0603           00258  BTFSC  STATUS,CA
0096 0A9F           00259  GOTO   WRITV
0097                00260 WRITW
0097 0736           00261  BTFSS  FLAGB,UP          ;DOWN / UP ?
0098 0AA2           00262  GOTO   WRITD
0099 0275           00263  COMF   YCOUNT, F         ;(UP) NEG YCOUNT
009A 0295           00264  INCF   YCOUNT,W
009B                00265 WRITB
009B 0030           00266  MOVWF  DATA2
009C 0032           00267  MOVWF  DATA4
009D 0075           00268  CLRF   YCOUNT            ;RESET YCOUNT
009E 0A60           00269  GOTO   SCANB
                    00270 ;
009F                00271 WRITV
009F 0C40           00272  MOVLW  40H               ;YCOUNT <- 64
00A0 0035           00273  MOVWF  YCOUNT
00A1 0A97           00274  GOTO   WRITW
                    00275 ;
00A2                00276 WRITD
00A2 0215           00277  MOVF   YCOUNT,W          ;(DOWN)
00A3 0A9B           00278  GOTO   WRITB
                    00279 ;
                    00280 ;===========================================
                    00281 ;       RESET ENTRY
                    00282 ;===========================================
                    00283 ;
01FF                00284  ORG    777
01FF 0A45           00285  GOTO   INIT              ;JUMP TO PROGRAM STARTING
                    00286 ;
                    00287  END

0000 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX
0040 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX
0080 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXX------------ ----------------
01C0 : ---------------- ---------------- ---------------- ---------------X

All other memory blocks unused.

Program Memory Words Used:   165
Program Memory Words Free:   347


Errors   :     0
Warnings :     0 reported,     0 suppressed
Messages :     0 reported,     0 suppressed
```

WORLDWIDE SALES & SERVICE

AMERICAS

Corporate Office

Microchip Technology Inc.
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 602-786-7200 Fax: 602-786-7277
Technical Support: 602 786-7627
Web: http://www.microchip.com

Atlanta

Microchip Technology Inc.
500 Sugar Mill Road, Suite 200B
Atlanta, GA 30350
Tel: 770-640-0034 Fax: 770-640-0307

Boston

Microchip Technology Inc.
5 Mount Royal Avenue
Marlborough, MA 01752
Tel: 508-480-9990 Fax: 508-480-8575

Chicago

Microchip Technology Inc.
333 Pierce Road, Suite 180
Itasca, IL 60143
Tel: 630-285-0071 Fax: 630-285-0075

Dallas

Microchip Technology Inc.
14651 Dallas Parkway, Suite 816
Dallas, TX 75240-8809
Tel: 972-991-7177 Fax: 972-991-8588

Dayton

Microchip Technology Inc.
Two Prestige Place, Suite 150
Miamisburg, OH 45342
Tel: 937-291-1654 Fax: 937-291-9175

Los Angeles

Microchip Technology Inc.
18201 Von Karman, Suite 1090
Irvine, CA 92612
Tel: 714-263-1888 Fax: 714-263-1338

New York

Microchip Technology Inc.
150 Motor Parkway, Suite 416
Hauppauge, NY 11788
Tel: 516-273-5305 Fax: 516-273-5335

San Jose

Microchip Technology Inc.
2107 North First Street, Suite 590
San Jose, CA 95131
Tel: 408-436-7950 Fax: 408-436-7955

Toronto

Microchip Technology Inc.
5925 Airport Road, Suite 200
Mississauga, Ontario L4V 1W1, Canada
Tel: 905-405-6279 Fax: 905-405-6253

ASIA/PACIFIC

Hong Kong

Microchip Asia Pacific
RM 3801B, Tower Two
Metroplaza
223 Hing Fong Road
Kwai Fong, N.T., Hong Kong
Tel: 852-2-401-1200 Fax: 852-2-401-3431

India

Microchip Technology India
No. 6, Legacy, Convent Road
Bangalore 560 025, India
Tel: 91-80-229-0061 Fax: 91-80-229-0062

Korea

Microchip Technology Korea
168-1, Youngbo Bldg. 3 Floor
Samsung-Dong, Kangnam-Ku
Seoul, Korea
Tel: 82-2-554-7200 Fax: 82-2-558-5934

Shanghai

Microchip Technology
RM 406 Shanghai Golden Bridge Bldg.
2077 Yan'an Road West, Hongiao District
Shanghai, PRC 200335
Tel: 86-21-6275-5700
Fax: 86 21-6275-5060

Singapore

Microchip Technology Taiwan
Singapore Branch
200 Middle Road
#10-03 Prime Centre
Singapore 188980
Tel: 65-334-8870 Fax: 65-334-8850

Taiwan, R.O.C

Microchip Technology Taiwan
10F-1C 207
Tung Hua North Road
Taipei, Taiwan, ROC
Tel: 886 2-717-7175 Fax: 886-2-545-0139

EUROPE

United Kingdom

Arizona Microchip Technology Ltd.
Unit 6, The Courtyard
Meadow Bank, Furlong Road
Bourne End, Buckinghamshire SL8 5AJ
Tel: 44-1628-851077 Fax: 44-1628-850259

France

Arizona Microchip Technology SARL
Zone Industrielle de la Bonde
2 Rue du Buisson aux Fraises
91300 Massy, France
Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79

Germany

Arizona Microchip Technology GmbH
Gustav-Heinemann-Ring 125
D-81739 Müchen, Germany
Tel: 49-89-627-144 0 Fax: 49-89-627-144-44

Italy

Arizona Microchip Technology SRL
Centro Direzionale Colleone
Palazzo Taurus 1 V. Le Colleoni 1
20041 Agrate Brianza
Milan, Italy
Tel: 39-39-6899939 Fax: 39-39-6899883

JAPAN

Microchip Technology Intl. Inc.
Benex S-1 6F
3-18-20, Shin Yokohama
Kohoku-Ku, Yokohama
Kanagawa 222 Japan
Tel: 81-4-5471- 6166 Fax: 81-4-5471-6122

5/8/97

MICROCHIP