LMC555

AN-681 PC Mouse Implementation Using COP800

Literature Number: SNAA005

PC® MOUSE Implementation Using COP800

National Semiconductor
Application Note 681
Alvin Chan
June 1990

ABSTRACT

The mouse is a very convenient and popular device used in data entry in desktop computers and workstations. For desktop publishing, CAD, paint or drawing programs, using the mouse is inevitable. This application note will describe how to use the COP822C microcontroller to implement a mouse controller.

INTRODUCTION

Mouse Systems was the first company to introduce a mouse for PCs. Together with Microsoft and Logitech, they are the most popular vendors in the PC mouse market. Most mainstream PC programs that use pointing devices are able to support the communication protocols laid down by Mouse Systems and Microsoft.

A typical mouse consists of a microcontroller and its associated circuitry, which are a few capacitors, resistors and transistors. Accompanying the electronics are the mechanical parts, consisting of buttons, roller ball and two disks with slots. Together they perform several major functions: motion detection, host communication, power supply, and button status detection.

MOTION DETECTION

Motion detection with a mouse consists of four commonly known mechanisms. They are the mechanical mouse, the opto-mechanical mouse, the optical mouse and the wheel mouse.

The optical mouse differs from the rest as it requires no mechanical parts. It uses a special pad with a reflective surface and grid lines. Light emitted from the LEDs at the bottom of the mouse is reflected by the surface and movement is detected with photo-transistors.

The mechanical and the opto-mechanical mouse use a roller ball. The ball presses against two rollers which are connected to two disks for the encoding of horizontal and vertical motion. The mechanical mouse has contact points on the disks. As the disks move they touch the contact bars, which in turn generates signals to the microcontroller. The opto-mechanical mouse uses disks that contain evenly spaced slots. Each disk has a pair of LEDs on one side and a pair of photo-transistors on the other side.

The wheel mouse has the same operation as the mechanical mouse except that the ball is eliminated and the rollers are rotated against the outside surface on which the mouse is placed.

HOST COMMUNICATION

Besides having different operating mechanisms, the mouse also has different modes of communication with the host. It can be done through the system bus, the serial port or a special connector. The bus mouse takes up an expansion slot in the PC. The serial mouse uses one of the COM ports.

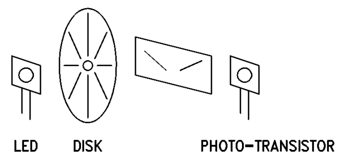
Although the rest of this report will be based on the opto-mechanical mouse using the serial port connection, the same principle applies to the mechanical and the wheel mouse.

MOTION DETECTION FOR THE OPTO-MECHANICAL MOUSE

The mechanical parts of the opto-mechanical mouse actually consist of one roller ball, two rollers connected to the disks and two pieces of plastic with two slots on each one for LED light to pass through. The two slots are cut so that they form a 90 degree phase difference. The LEDs and the photo-transistors are separated by the disks and the plastic. As the disks move, light pulses are received by the photo-transistors. The microcontroller can then use these quadrature signals to decode the movement of the mouse.

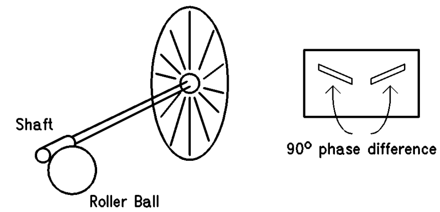
Figure 1a shows the arrangement of the LEDs, disks, plastic and photo-transistors. The shaft connecting the disk and the ball is shown separately on *Figure 1b*. *Figure 2* shows the signals obtained from the photo-transistors when the mouse moves. The signals will not be exactly square waves because of unstable hand movements.

PC® is a registered trademark of International Business Machines Corporation.



TL/DD/10799–1

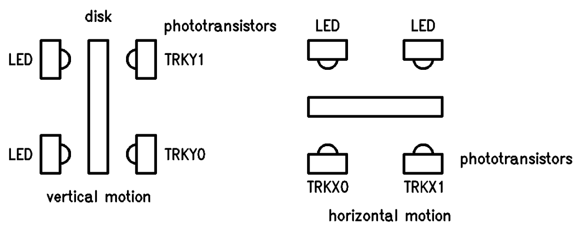
a



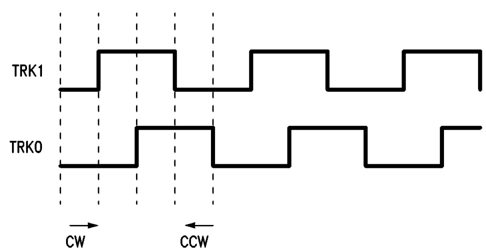
TL/DD/10799–2

b

FIGURE 1



TL/DD/10799–3



TL/DD/10799–4

Signals at phototransistors are similar for vertical and horizontal motion.
Track 1 leads track 0 by 90 degrees

FIGURE 2

RESOLUTION, TRACKING SPEED AND BAUD RATE

The resolution of the mouse is defined as the number of movement counts the mouse can provide for each fixed distance travelled. It is dependent on the physical dimension of the ball and the rollers. It can be calculated by measuring the sizes of the mechanical parts.

An example for the calculation can be shown by making the following assumptions:

- The disks have 40 slots and 40 spokes
- Each spoke has two data counts
  (This will be explained in the section "An Algorithm for Detecting Movements")
- Each slot also has two data counts
- The roller has a diameter of 5mm

For each revolution of the roller, there will be $40 \times 2 \times 2 = 160$ counts of data movement. At the same time, the mouse would have travelled a distance of $\pi \times 5 = 15.7$mm. Therefore the resolution of the mouse is $15.7/160 = 0.098$mm per count. This is equivalent to 259 counts or dots per inch (dpi).

The tracking speed is defined as the fastest speed that the mouse can move without the microcontroller losing track of the movement. This depends on how fast the microcontroller can sample the pulses from the photo-transistors. The effect of a slow tracking speed will contribute to jerking movements of the cursor on the screen.

The baud rate is fixed by the software and the protocol of the mouse type that is being emulated. For mouse systems and microsoft mouse, they are both 1200. Baud rate will affect both the resolution and the tracking speed. The internal movement counter may overflow while the mouse is still sending the last report with a slow baud rate. With a fast baud rate, more reports can be sent for a certain distance moved and the cursor should appear to be smoother.

POWER SUPPLY FOR THE SERIAL MOUSE

Since the serial port of the PC has no power supply lines, the RTS, CTS, DTR and DSR RS232 interface lines are utilized. Therefore the microcontroller and the mouse hardware should have very little power consumption. National Semiconductor's COP822C fits into this category perfectly. The voltage level in the RS232 lines can be either positive or negative. When they are positive, the power supply can be obtained by clamping down with diodes. When they are negative, a 555 timer is used as an oscillator to transform the voltage level to positive. The 1988 National Semiconductor Linear 3 Databook has an example of how to generate a variable duty cycle oscillator using the LMC555 in page 5-282.

While the RTS and DTR lines are used to provide the voltage for the mouse hardware, the TXD line of the host is utilized as the source for the communication signals. When idle, the TXD line is in the mark state, which is the most negative voltage. A pnp transistor can be used to drive the voltage of the RXD pin to a voltage level that is compatible with the RS232 interface standard.

AN ALGORITHM FOR DETECTING MOVEMENTS

The input signal of the photo-transistors is similar to that shown in *Figure 2*. Track 1 leads track 0 by 90 degrees. Movement is recorded as either of the tracks changes state. State tables can be generated for clockwise and counter-clockwise motions.

With the two tracks being 90 degrees out of phase, there could be a total of four possible track states. It can be observed that the binary values formed by combining the present and previous states are unique for clockwise and counter-clockwise motion. A sixteen entry jump table can be formed to increment or decrement the position of the cursor. If the value obtained does not correspond to either the clockwise or counter-clockwise movement, it could be treated as noise. In that case either there is noise on the microcontroller input pins or the microcontroller is tracking motions faster than the movement of the mouse. A possible algorithm can be generated as follows. The number of instruction cycles for some instructions are shown on the left.

CCW

$(TRK1, TRK0)_t$		$(TRK1, TRK0)_{t-1}$		Binary Value
0	1	0	0	4
1	1	0	1	D
1	0	1	1	B
0	0	1	0	2

CW

$(TRK1, TRK0)_t$		$(TRK1, TRK0)_{t-1}$		Binary Value
1	0	0	0	8
0	0	0	1	1
0	1	1	1	7
1	1	1	0	E

```
CYCLES     ;***************************************************
           ;        SAMPLE SENSOR INPUT
           ;        INC OR DEC THE POSITION
           ;***************************************************
           ;
           SENSOR:
1                   LD        B,#GTEMP
3                   LD        A,PORTGP
1                   RRC       A
2                   AND       A,#03C          ; G6,G5,G4,G3
1                   X         A, [B]          ; (GTEMP)
           ;
2                   LD        A, [B+]         ; (GTEMP) X IN 3,2
1                   RRC       A
1                   RRC       A
2                   AND       A, #03
1                   OR        A, [B]          ; (TRACKS)
2                   OR        A, #0B0         ; X MOVEMENT TABLE
3                   JID
           ;
           NOISEX:  JP        YDIR
           ;
3          INCX:    LD        A,XINC
1                   INC       A
3                   JP        COMX
           ;
           DECX:    LD        A,XINC
                    DEC       A
           COMX:
2                   IFEQ      A, #080
1                   JP        YDIR
3                   X         A, XINC
1                   LD        B, #CHANGE
1                   SBIT      RPT, [B]
1                   LD        B, #TRACKS
           ;
           YDIR:
2                   LD        A, [B-]         ; (TRACKS) Y IN 5, 4
1                   SWAP      A
1                   RRC       A
1                   RRC       A
1                   RRC       A
2                   AND       A, #0C0
1                   OR        A, [B]          ; (GTEMP)
```

```
1                 SWAP     A
2                 OR       A, #0C0          ; Y MOVEMENT TABLE
3                 JID
         ;
         NOISEY:  JP       ESENS
         ;
3        INCY:    LD       A, YINC
1                 INC      A
3                 JP       COMY
         DECY:
                  LD       A, YINC
                  DEC      A
         COMY:
2                 IFEQ     A, #080
1                 JP       ESENS
3                 X        A, YINC
1                 LD       B, #CHANGE
1                 SBIT     RPT, [B]
1                 LD       B, #GTEMP
         ESENS:
2                 LD       A, [B+]          ; (GTEMP) IN5, 4, 1, 0
1                 X        A, [B]           ; (TRACKS) NEW TRACK STATUS
5                 RET
         ;
                  .=0B0
         MOVEMX:
                  .ADDR    NOISEX           ; 0
                  .ADDR    INCX             ; 1
                  .ADDR    DECX             ; 2
                  .ADDR    NOISEX           ; 3
                  .ADDR    DECX             ; 4
                  .ADDR    NOISEX           ; 5
                  .ADDR    NOISEX           ; 6
                  .ADDR    INCX             ; 7
                  .ADDR    INCX             ; 8
                  .ADDR    NOISEX           ; 9
                  .ADDR    NOISEX           ; A
                  .ADDR    DECX             ; B
                  .ADDR    NOISEX           ; C
                  .ADDR    DECX             ; D
                  .ADDR    INCX             ; E
                  .ADDR    NOISEX           ; F
         ;
                  .=0C0
         MOVEMY:
                  .ADDR    NOISEY           ; 0
                  .ADDR    INCY             ; 1
                  .ADDR    DECY             ; 2
                  .ADDR    NOISEY           ; 3
                  .ADDR    DECY             ; 4
                  .ADDR    NOISEY           ; 5
                  .ADDR    NOISEY           ; 6
                  .ADDR    INCY             ; 7
                  .ADDR    INCY             ; 8
                  .ADDR    NOISEY           ; 9
                  .ADDR    NOISEY           ; A
                  .ADDR    DECY             ; B
                  .ADDR    NOISEY           ; C
                  .ADDR    DECY             ; D
                  .ADDR    INCY             ; E
                  .ADDR    NOISEY           ; F
```

Going through the longest route in the sensor routine takes 75 instruction cycles. So at 5 MHz the microcontroller can track movement changes within 150 μs by using this algorithm.

MOUSE PROTOCOLS

Since most programs in the PC support the mouse systems and microsoft mouse, these two protocols will be discussed here. The protocols are byte-oriented and each byte is framed by one start-bit and two stop-bits. The most commonly used reporting mode is that a report will be sent if there is any change in the status of the position or of the buttons.

MICROSOFT COMPATIBLE DATA FORMAT

6	5	4	3	2	1	0	Bit Number
1	L	R	Y7	Y6	X7	X6	Byte 1
0	X5	X4	X3	X2	X1	X0	Byte 2
0	Y5	Y4	Y3	Y2	Y1	Y0	Byte 3

L, R = Key data (Left, Right key) 1 = key depressed

X0–X7 = X distance 8-bit two's complement value −128 to +127

Y0–Y7 = Y distance 8-bit two's complement value −128 to +127

Positive = South

In the Microsoft Compatible Format, data is transferred in the form of seven-bit bytes. Y movement is positive to the south and negative to the north.

FIVE BYTE PACKED BINARY FORMAT (MOUSE SYSTEMS CORP)

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

L*, M*, R* = Key data (Left, Middle, Right key), 0 = key depressed

X0–X7 = X distance 8-bit two's complement value −127 to +127

Y0–Y7 = Y distance 8-bit two's complement value −127 to +127

In the Five Byte Packed Binary Format data is transferred in the form of eight-bit bytes (eight data bits without parity). Bytes 4 and 5 are the movement of the mouse during the transmission of the first report.

THE COP822C MICROCONTROLLER

The COP822C is an 8-bit microcontroller with 20 pins, of which 16 are I/O pins. The I/O pins are separated into two ports, port L and port G. Port G has built-in Schmitt-triggered inputs. There is 1k of ROM and 64 bytes of RAM. In the mouse application, the COP822C's features used can be summarized below. Port G is used for the photo-transistor's input. Pin G0 is used as the external interrupt input to monitor the RTS signal for the microsoft compatible protocol. The internal timer can be used for baud rate timing and interrupt generation. The COP822C draws only 4 mA at a crystal frequency of 5 MHz. The instruction cycle time when operating at this frequency is 2 μs.

A MOUSE EXAMPLE

The I/O pins for the COP822C are assigned as follows:

Pin	Function
G0	Interrupt Input (Monitoring RTS Toggle)
G1	Reserved for Input Data (TXD of Host)
G2	Output Data (RXD of Host)
G3–G6	LED Sensor Input
L0–L2	Button Input
L3	Jumper Input (for Default Mouse Mode)

The timer is assigned for baud rate generation. It is configured in the PWM auto-reload mode (with no G3 toggle output) with a value of 1A0 hex in both the timer and the auto-reload register. When operating at 5 MHz, it is equivalent to 833 μs or 1200 baud. When the timer counts down, an interrupt is generated and the service routine will indicate in a timer status byte that it is time for the next bit. The subroutine that handles the transmission will look at this status byte to send the data.

The other interrupt comes from the G0 pin. This is implemented to satisfy the microsoft mouse requirement. As the RTS line toggles, it causes the microcontroller to be interrupted. The response to the toggling is the transmission of the character "M" to indicate the presence of the mouse.

The main program starts by doing some initializations. Then it loops through four subroutines that send the report, sense the movement, sense the buttons, and set up the report format.

Subroutine "SDATA" uses a state table to determine what is to be transmitted. There are 11 or 12 states because microsoft has only 7 data bits and mouse systems has 8. The state table is shown below:

SENDST	State
0	IDLE
1	START BIT
2–8	DATA (FOR MICROSOFT)
2–9	DATA (FOR MOUSE SYSTEMS)
9–10	STOP BIT (FOR MICROSOFT)
10–11	STOP BIT (FOR MOUSE SYSTEMS)
11	NEXT WORD (FOR MICROSOFT)
12	NEXT WORD (FOR MOUSE SYSTEMS)

The G2 pin is set to the level according to the state and the data bit that is transmitted.

Subroutine "SENSOR" checks the input pins connected to the LEDs. The horizontal direction is checked first followed by the vertical direction. Two jump tables are needed to decode the binary value formed by combining the present and previous status of the wheels. The movements are recorded in two counters.

Subroutines "BUTUS" and "BUTMS" are used for polling the button input. They compare the button input with the value polled last time and set up a flag if the value changes. Two subroutines are used for the ease of setting up reports for different mice. The same applies for subroutines "SRPTMS" and "SRPTUS" which set up the report format for transmission. The status change flag is checked and the report is formatted according to the mouse protocol. The movement counters are then cleared. Since the sign of the vertical movement of mouse systems and microsoft is reversed, the counter value in subroutine "SRPTMS" is complemented to form the right value.

There is an extra subroutine "SY2RPT" which sets up the last two bytes in the mouse systems' report. It is called after the first three bytes of the report are sent.

The efficiency of the mouse depends solely on the effectiveness of the software to loop through sensing and transmission subroutines. For the COP822C, one of the most effective addressing modes is the B register indirect mode. It uses only one byte and one instruction cycle. With autoincrement or autodecrement, it uses one byte and two instruction cycles. In order to utilize this addressing mode more often, the organization of the RAM data has to be carefully thought out. In the mouse example, it can be seen that by placing related variables next to each other, the saving of code and execution time is significant. Also, if the RAM data can fit in the first 16 bytes, the load B immediate instruction is also more efficient. The subroutine "SRPTMS" is shown below and it can be seen that more than half the instructions are B register indirect which are efficient and compact.

```
;
;     VARIABLES
;
      WORDPT   =       000      ;WORD POINTER
      WORD1    =       001      ;BUFFER TO STORE REPORTS
      WORD2    =       002
      WORD3    =       003
      CHANGE   =       004      ;MOVEMENT CHANGE OR BUTTON PRESSED
      XINC     =       005      ;X DIRECTION COUNTER
      YINC     =       006      ;Y DIRECTION COUNTER
      NUMWORD  =       007      ;NUMER OF BYTES TO SEND
      SENDST   =       008      ;SERIAL PROTOCOL STATE
;
;*******************************************************
;     SUBROUTINE SET UP REPORT 'SRPT' FOR MOUSE SYSTEMS
;     CHANGE OF STATUS DETECTED
;     SET UP THE FIRST 3 WORDS FOR REPORTING
;     IF IN IDLE STATE
;*******************************************************
;
SRPTMS:
      LD       A,CHANGE
      IFEQ     A, #0            ; EXIT IF NO CHANGE
      RET
;
      RBIT     GIE, PSW         ; DISABLE INTERRUPT
      LD       B, #WORDPT
      LD       [B+], #01        ; (WORDPT) SET WORD POINTER
      LD       A, BUTSTAT
      X        A, [B+]          ; (WORD1)
;
      LD       A, XINC
      X        A, [B+]          ; (WORD2)
;
      SC
      CLR      A
      SUBC     A, YINC          ; FOR MOUSE SYSTEM NEG Y
      X        A, [B+]          ; (WORD3)
;
      RBIT     RPT, [B]         ; (CHANGE) RESET CHANGE OF STATUS
      SBIT     SYRPT, [B]                  ; (CHANGE)
      LD       A, [B+]          ; INC B
      LD       [B+], #0         ; (XINC)
      LD       [B+], #0         ; (YINC)
;
      LD       [B+], #03        ; (NUMWORD) SEND 3 BYTES
      LD       [B], #01         ; (SENDST) SET TO START BIT STATE
;     SBIT     GIE, PSW         ; ENABLE INTERRUPT
      RET
;
```

CONCLUSION

The COP822C has been used as a mouse controller. The code presented is a minimum requirement for implementing a mouse systems and microsoft compatible mouse. About 550 bytes of ROM code has been used. The remaining ROM area can be used for internal diagnostics and for communicating with the host's mouse driver program. The unused I/O pins can be used to turn the LED's on only when necessary to save extra power. This report demonstrated the use of the efficient instruction set of the COP800 family. It can be seen that the architecture of the COP822C is most suitable for implementing a mouse controller. The table below summarizes the advantages of the COP822C.

Feature	Advantage
Port G	Schmitt Triggered Input for Photo-Transistors
G0	External Interrupt for RTS Toggling
Timer	For Baud Rate Generation
Low Power	4 mA at 5 MHz
Small Size	20-Pin DIP

REFERENCE

The mouse still reigns over data entry—Electronic Engineering Times, October 1988.

MICE for mainstream applications—PC Magazine, August 1987.

Logimouse C7 Technical Reference Manual—Logitech, January 1986.

APPENDIX A—MEMORY UTILIZATION

RAM Variables

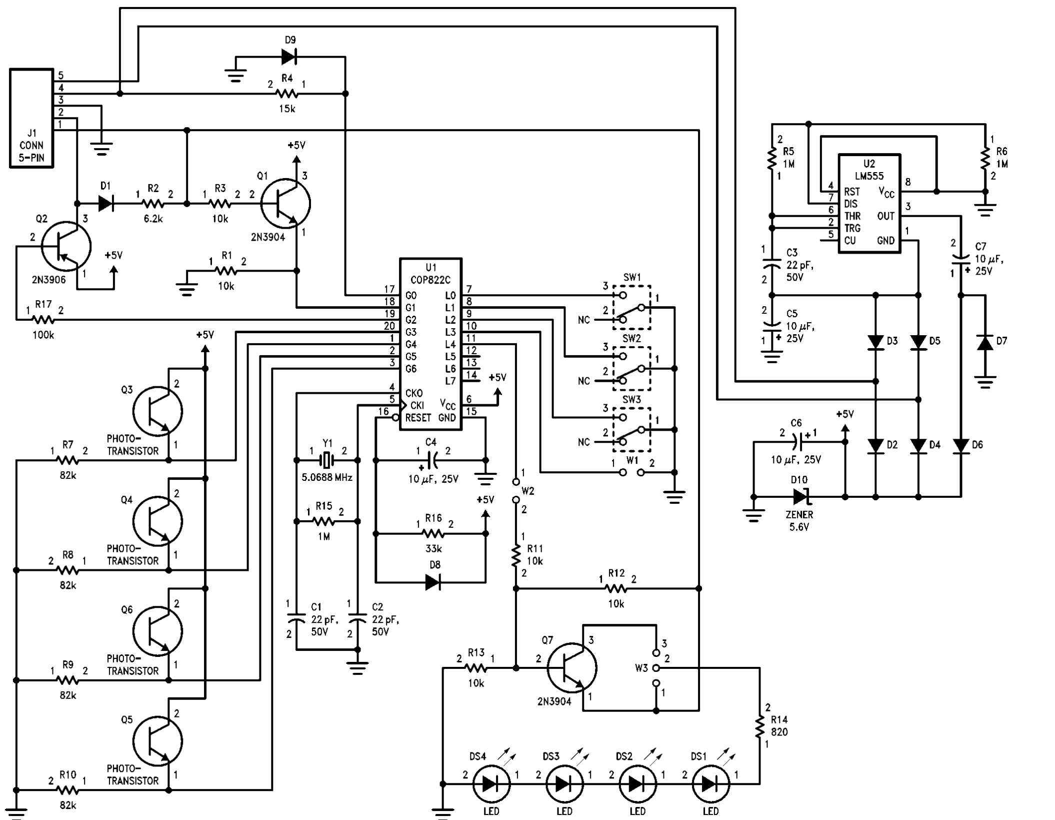
TEMP	=	0F1	Work Space
ASAVE	=	0F4	Save A Register
PSSAVE	=	0F6	Save PSW Register
WORDPT	=	000	Word Pointer
WORD1	=	001	Buffer to Store Report
WORD2	=	002	Buffer
WORD3	=	003	Buffer
CHANGE	=	004	Movement or Button Change
XINC	=	005	X Direction Counter
YINC	=	006	Y Direction Counter
NUMWORD	=	007	Number of Bytes to Send
SENDST	=	008	Serial Protocol State
TSTATUS	=	00A	Counter Status
MTYPE	=	00B	Mouse Type
GTEMP	=	00C	Track Input from G Port
TRACKS	=	00D	Previous Track Status
BTEMP	=	00E	Button Input from L Port
BUTSTAT	=	00F	Previous Button Status

APPENDIX B—SUBROUTINE SUMMARY

Subroutine	Location	Function
MLOOP	03D	Main Program Loop
SENSOR	077	Sample Photo-Transistor Input
INTRP	0FF	Interrupt Service Routines
SRPTUS	136	Set Up Report for Microsoft
SRPTMS	16C	Set Up 1st 3 Bytes Report for Mouse Systems
SDATA	191	Drive Data Transmission Pin According to Bit Value of Report
SY2RPT	1D1	Set Up Last 2 Bytes Report for Mouse Systems
BUTUS	200	Sample Button Input for Microsoft
BUTMS	210	Sample Button Input for Mouse Systems

APPENDIX C—SYSTEM SCHEMATIC, SYSTEM

Flowchart, complete program listing.



TL/DD/10799–5

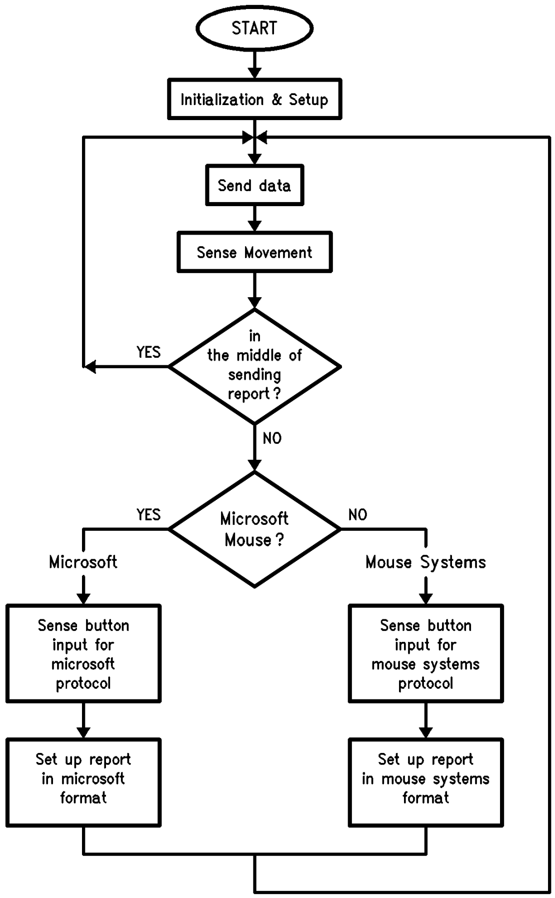
Note 1: All diodes are 1N4148.

Note 2: All resistor values are in ohms, 5%, 1/8W.

Note: Unless otherwised specified

FIGURE 3. System Schematic

Flowchart for Mouse Systems and Microsoft Mouse



TL/DD/10799–6

NATIONAL SEMICONDUCTOR CORPORATION
COP800 CROSS ASSEMBLER,REV:D1,12 OCT 88
AMOUSE

```
                    ;
                    ;       MICROSOFT AND MOUSE SYSTEM COMPATIBLE MOUSE
                    ;       02/14/89
                    ;       NAME : AMOUSE.MAC
                    ;
                            .TITLE AMOUSE
                            .CHIP 820
                    ;
                    ;
      00D0          PORTLD  =       0D0     ; PORT L DATA
      00D1          PORTLC  =       0D1     ; PORT L CONFIG
      00D2          PORTLP  =       0D2     ; PORT L PIN
                    ;
      00D4          PORTGD  =       0D4     ; PORT G DATA
      00D5          PORTGC  =       0D5     ; PORT G CONFIG
      00D6          PORTGP  =       0D6     ; PORT G PIN
                    ;
      00EA          TMRLO   =       0EA     ; TIMER LOW BYTE
      00EB          TMRHI   =       0EB     ; TIMER HIGH BYTE
      00EC          TAULO   =       0EC     ; TIMER REGISTER LOW BYTE
      00ED          TAUHI   =       0ED     ; TIMER REGISTER HIGH BYTE
                    ;
      00EE          CNTRL   =       0EE     ; CONTROL REGISTER
      00EF          PSW     =       0EF     ; PSW REGISTER
                    ;
                    ;       CONSTANT DECLARE
                    ;
      0000          INTR    =       0
      0003          TIO     =       3
      0004          SO      =       4
      0005          SK      =       5
      0006          SI      =       6
      0007          CKO     =       7
                    ;
      0007          TSEL    =       7
      0006          CSEL    =       6
      0005          TEDG    =       5
      0004          TRUN    =       4
      0003          MSEL    =       3
      0002          IEDG    =       2
      0001          S1      =       1
      0000          S0      =       0
                    ;
      0007          HCARRY  =       7
      0006          CARRY   =       6
      0005          TPND    =       5
      0004          ENTI    =       4
      0003          IPND    =       3
      0002          BUSY    =       2
      0001          ENI     =       1
      0000          GIE     =       0
```

TL/DD/10799–7

```
                    ;
                    ;       TSTATUS BITS
                    ;
        0002        TBAUB   =       2       ;BAUD RATE TIMER BIT
                    ;
        0000        RPT     =       0       ;REPORT BIT OF CHANGE (CHANGE)
        0001        SYRPT   =       1       ;SET UP MOUSE SYSTEM LAST 2 WORDS (CHANGE)
        0007        USOFT   =       7       ;MICROSOFT (MTYPE)
        0002        XMT     =       2       ;G2 AS XMT BIT (PORTGD)
                    ;
        0003        SW      =       3       ;SLIDE SWITCH (PORTLP,MTYPE)
                    ;
                    ;       REGISTER ASSIGNMENTS
                    ;
        00F0                RSVD    =       0F0
        00F1                TEMP    =       0F1
        00F3                TBAU    =       0F3     ;BAUD RATE TIMER
        00F4                ASAVE   =       0F4     ;SAVE A
        00F5                BSAVE   =       0F5     ;SAVE B
        00F6                PSSAVE  =       0F6     ;SAVE PSW
                    ;
                    ;       VARIABLES
                    ;
        0000                WORDPT  =       000     ;WORD POINTER
        0001                WORD1   =       001     ;BUFFER TO STORE REPORTS
        0002                WORD2   =       002
        0003                WORD3   =       003
                    ;
        0004                CHANGE  =       004     ;MOVEMENT CHANGE OR BUTTON PRESSED
        0005                XINC    =       005     ;X DIRECTION COUNTER
        0006                YINC    =       006     ;Y DIRECTION COUNTER
        0007                NUMWORD =       007     ;NUMBER OF BYTES TO SEND
        0008                SENDST  =       008     ;SERIAL PROTOCOL STATE
                    ;
        0009                TBAUR   =       009     ;BAUD RATE TIMER RELOAD
        000A                TSTATUS =       00A     ;COUNTER STATUS
        000B                MTYPE   =       00B     ;MOUSE TYPE
                    ;
        000C                GTEMP   =       00C     ;TRACK INPUT FROM G
        000D                TRACKS  =       00D     ;PREVIOUS TRACK STATUS
                    ;
        000E                BTEMP   =       00E     ;BUTTON INPUT
        000F                BUTSTAT =       00F     ;PREVIOUS BUTTON STATUS
                    ;
                    ;
                    ; MOST POSITIVE = SPACE = HI = ON = 0 = START BIT = RBIT
                    ; MOST NEGATIVE = MARK = LO = OFF = 1 = STOP BIT = SBIT
                    ;
                    ;       MICROSOFT FORMAT
                    ;
                    ;       1  L  R  Y7  Y6  X7  X6
```

TL/DD/10799–8

```
                  ;      0  X5 .............  X0
                  ;      0  Y5 .............  Y0
                  ;
                  ;      1200 BAUD 7 BIT NO PARITY 2 STOP BITS
                  ;
                  ;      MOUSE SYSTEMS FORMAT (FIVE BYTE PACKED BINARY)
                  ;
                  ;      1  0   0   0   0   L*  M*  R*
                  ;      X7 ......................  X0
                  ;      Y7 ......................  Y0
                  ;      X7 ......................  X0
                  ;      Y7 ......................  Y0
                  ;
                  ;      1200 BAUD 7 BIT NO PARITY 2 STOP BITS
                  ;
                  ;      G6,G5,G4,G3 ARE SENSOR INPUTS
                  ;
                  ;      L0, L1 AND L2 ARE BUTTON INPUTS
                  ;
                  ;      G0 IS INTERRUPT INPUT FOR DETECTING RTS TOGGLE
                  ;
                  ;      USE G2 AS TRANSMIT
                  ;
                  ;      G1 USED FOR RECEIVING COMMANDS FROM HOST (RESERVED)
                  ;
                  START:
0000 DD2F                LD      SP,#02F
0002 BCEF00              LD      PSW,#0          ;DISABLE INTR
0005 BCEE80              LD      CNTRL,#080      ;10000000 - AUTORELOAD
                                                 ;RISING EDGE EXT INT
0008 BCD504              LD      PORTGC,#004     ;G2 AS OUTPUT, OTHERS AS HI-Z
000B BCD404              LD      PORTGD,#004     ;G2 DATA 1 "MARK"
                  ;
000E BCD130              LD      PORTLC,#030     ;HI-Z INPUTS FOR L6-7,OUTPUT L4,5
0011 BCD00F              LD      PORTLD,#0F      ;WEAK PULL UP FOR L0-3
                  ;
                  ;      INIT RAM
                  ;
0014 5B                  LD      B,#CHANGE
0015 9A00                LD      [B+],#0         ;(CHANGE)
0017 9A00                LD      [B+],#0         ;(XINC)
0019 9A00                LD      [B+],#0         ;(YINC)
001B BC0A00              LD      TSTATUS,#0
                  ;
001E 9DD6                LD      A,PORTGP
0020 B0                  RRC     A
0021 953C                AND     A,#03C          ;NOW IN 6,5,4,3
0023 9C0D                X       A,TRACKS        ;GET INITIAL VALUE OF SENSORS
                  ;
0025 3067                JSR     SELECT          ;SELECT MOUSE TYPE
                  ;
```

TL/DD/10799–9

```
                    ;*****************************************************
                    ;
                    ;       CRYSTAL FREQ = 4.96 MHZ  2.016 US INST CYCLE
                    ;       FOR 1200 BAUD - TIMER = 413 COUNT
                    ;
                    ;*****************************************************
                    ;
                    LTIMER:
0027 DEEA                   LD      B,#TMRLO
0029 9A9D                   LD      [B+],#09D       ;FOR 2.016 US CYCLE
002B 9A01                   LD      [B+],#01
002D 9A9D                   LD      [B+],#09D
002F 9E01                   LD      [B],#01
                    ;
0031 BC0800                 LD      SENDST,#0       ;SET TO IDLE STATE
0034 9DEF                   LD      A,PSW
0036 9713                   OR      A,#013          ;ENABLE INTRS SET GIE
0038 9CEF                   X       A,PSW
003A BDEE7C                 SBIT    TRUN,CNTRL      ;START TIMER
                    ;
                    MLOOP:
003D BCD03F                 LD      PORTLD,#03F     ;TURN ON LED (NOT USED)
0040 3191                   JSR     SDATA
0042 3077                   JSR     SENSOR
0044 9D08                   LD      A,SENDST        ;IF SENDING REPORT
0046 9300                   IFGT    A,#0            ;JUST DO SENSOR
0048 F4                     JP      MLOOP
                    ;
0049 9DD2                   LD      A,PORTLP        ;GET INPUT FROM BUTTONS (L0,L1,L2)
004B B0                     RRC     A               ;PUT IN CARRY FOR CHECKING
004C 51                     LD      B,#BTEMP        ;PREPARATION TO SEE WHAT BUTTON IS PRESSED
                    ;
004D BD0B77                 IFBIT   USOFT,MTYPE
0050 0B                     JP      LPUS
                    ;
0051 3210                   JSR     BUTMS           ;MOUSE SYSTEMS
0053 316C                   JSR     SRPTMS
                    ;
0055 BDD273                 IFBIT   SW,PORTLP
0058 E4                     JP      MLOOP           ;CONTINUE IF NO CHANGE IN SWITCH
0059 306B                   JSR     USM             ;ELSE NEW SET UP
005B E1                     JP      MLOOP
                    LPUS:
005C 3200                   JSR     BUTUS           ;MICROSOFT
005E 3136                   JSR     SRPTUS
                    ;
0060 BDD273                 IFBIT   SW,PORTLP
0063 3071                   JSR     SYM             ;IF CHANGED IN SWITCH, NEW SET UP
0065 203D           +       JP      MLOOP
                    ;
                    ;**************************************************
```

TL/DD/10799–10

```
                          ;       SELECT MOUSE TYPE
                          ;***************************************************
                          ;
                          SELECT:
0067 BDD273                       IFBIT   SW,PORTLP        ;CHECK JUMPER
006A 06                           JP      SYM
                          ;
                          USM:
006B 54                           LD      B,#MTYPE
006C 7F                           SBIT    USOFT,[B]        ;(MTYPE) IS MICROSOFT MOUSE
006D BC0F87                       LD      BUTSTAT,#087     ;NO KEY PRESSED
0070 8E                           RET
                          ;
                          SYM:
0071 54                           LD      B,#MTYPE
0072 6F                           RBIT    USOFT,[B]        ;(MTYPE) IS MOUSE SYSTEMS
0073 BC0F00                       LD      BUTSTAT,#0       ;NO KEY PRESSED
0076 8E                           RET
                          ;
                          ;***************************************************
                          ;       SAMPLE SENSOR INPUT
                          ;       INC OR DEC THE POSITION
                          ;       -127 IS USED INSTEAD OF -128 IN CHECKING
                          ;       NEGATIVE GOING POSITION SO THAT BOTH
                          ;       MICROSOFT AND MOUSE SYSTEMS FIT IN
                          ;***************************************************
                          ;
                          SENSOR:
0077 53                           LD      B,#GTEMP
0078 9DD6                         LD      A,PORTGP
007A BCD00F                       LD      PORTLD,#0F       ;(NOT USED) TURN OFF LED
007D B0                           RRC     A
007E 953C                         AND     A,#03C           ;G5,G4,G3,G2
0080 A6                           X       A,[B]            ;(GTEMP)
                          ;
                          ;
                          ;               (TRK1,TRK0)t-1  (TRK1,TRK0)t
                          ;       CCW       0    1          0    0       4
                          ;                 1    1          0    1       D
                          ;                 1    0          1    1       B
                          ;                 0    0          1    0       2
                          ;
                          ;       CW        1    0          0    0       8
                          ;                 0    0          0    1       1
                          ;                 0    1          1    1       7
                          ;                 1    1          1    0       E
                          ;
0081 AA                           LD      A,[B+]           ;(GTEMP) X IN 3,2
0082 B0                           RRC     A
0083 B0                           RRC     A
0084 9503                         AND     A,#03            ;GET X TRACKS
```

TL/DD/10799–11

```
0086 87                OR     A,[B]          ;OVERLAY WITH PREVIOUS (TRACKS)
0087 97B0              OR     A,#0B0         ;X MOVEMENT TABLE
0089 A5                JID
                ;
008A 0F         NOISEX: JP     YDIR
                ;
                INCX:
008B 9D05              LD     A,XINC
008D 8A                INC    A
008E 03                JP     COMX           ;CHECK IF LIMIT IS REACHED
                DECX:
008F 9D05              LD     A,XINC
0091 8B                DEC    A
                COMX:                        ;CHECK FOR LIMIT
0092 9250              IFEQ   A,#80
0094 05                JP     YDIR           ;YES DO NOTHING
0095 9C05              X      A,XINC         ;ELSE NEW POSITION
0097 5B                LD     B,#CHANGE
0098 78                SBIT   RPT,[B]        ;(CHANGE)
0099 52                LD     B,#TRACKS
                ;
                YDIR:
009A 52                LD     B,#TRACKS
009B AB                LD     A,[B-]         ;(TRACKS) Y IN 5,4
009C 65                SWAP   A
009D B0                RRC    A
009E B0                RRC    A
009F B0                RRC    A
00A0 95C0              AND    A,#0C0
00A2 87                OR     A,[B]          ;(GTEMP)
00A3 65                SWAP   A
00A4 97C0              OR     A,#0C0         ;Y MOVEMENT TABLE
00A6 A5                JID
                ;
     00B0              .=0B0
                MOVEMX:
00B0 8A                .ADDR  NOISEX         ;0
00B1 8F                .ADDR  DECX           ;1
00B2 8B                .ADDR  INCX           ;2
00B3 8A                .ADDR  NOISEX         ;3
00B4 8B                .ADDR  INCX           ;4
00B5 8A                .ADDR  NOISEX         ;5
00B6 8A                .ADDR  NOISEX         ;6
00B7 8F                .ADDR  DECX           ;7
00B8 8F                .ADDR  DECX           ;8
00B9 8A                .ADDR  NOISEX         ;9
00BA 8A                .ADDR  NOISEX         ;A
00BB 8B                .ADDR  INCX           ;B
00BC 8A                .ADDR  NOISEX         ;C
00BD 8B                .ADDR  INCX           ;D
00BE 8F                .ADDR  DECX           ;E
```

TL/DD/10799–12

```
00BF 8A                  .ADDR   NOISEX        ;F
                 ;
     00C0                .=0C0
                 MOVEMY:
00C0 D0                  .ADDR   NOISEY        ;0
00C1 D1                  .ADDR   INCY          ;1
00C2 D5                  .ADDR   DECY          ;2
00C3 D0                  .ADDR   NOISEY        ;3
00C4 D5                  .ADDR   DECY          ;4
00C5 D0                  .ADDR   NOISEY        ;5
00C6 D0                  .ADDR   NOISEY        ;6
00C7 D1                  .ADDR   INCY          ;7
00C8 D1                  .ADDR   INCY          ;8
00C9 D0                  .ADDR   NOISEY        ;9
00CA D0                  .ADDR   NOISEY        ;A
00CB D5                  .ADDR   DECY          ;B
00CC D0                  .ADDR   NOISEY        ;C
00CD D5                  .ADDR   DECY          ;D
00CE D1                  .ADDR   INCY          ;E
00CF D0                  .ADDR   NOISEY        ;F
                 ;
00D0 0F          NOISEY: JP      ESENS
                 ;
00D1 9D06        INCY:   LD      A,YINC
00D3 8A                  INC     A
00D4 03                  JP      COMY
                 DECY:
00D5 9D06                LD      A,YINC
00D7 8B                  DEC     A
                 COMY:
00D8 9280                IFEQ    A,#080
00DA 05                  JP      ESENS
00DB 9C06                X       A,YINC
00DD 5B                  LD      B,#CHANGE
00DE 78                  SBIT    RPT,[B]       ;(CHANGE)
00DF 53                  LD      B,#GTEMP
                 ;
                 ESENS:
00E0 53                  LD      B,#GTEMP
00E1 AA                  LD      A,[B+]        ;(GTEMP) IN 5,4,1,0
00E2 A6                  X       A,[B]         ;(TRACKS)NEW TRACK STATUS
00E3 8E                  RET
                 ;
                 ;
     00FF                .=0FF
                 ;
                 ;***************************************************
                 ;       INTERRUPT ROUTINES
                 ;***************************************************
                 ;
00FF 9CF4        INTRP:  X       A,ASAVE
```

TL/DD/10799–13

```
                        ;
0101 BDEF75                     IFBIT   TPND,PSW
0104 07                         JP      TINTR
0105 BDEF73                     IFBIT   IPND,PSW
0108 0A                         JP      XINTR
                        ;
                        INTRET:                         ;INTERRUPT RETURN
0109 9DF4                       LD      A,ASAVE
010B 8F                         RETI
                        ;
                        ;**************************************************
                        ;       TIMER INTERRUPT
                        ;       UPDATE ALL THE COUNTERS
                        ;**************************************************
                        ;
                        TINTR:
010C BDEF6D                     RBIT    TPND,PSW
010F BD0A7A                     SBIT    TBAUB,TSTATUS   ;SET BIT IN TSTATUS
0112 F6                         JP      INTRET
                        ;
                        ;********************************************************
                        ;       EXTERNAL INTERRUPT
                        ;       RESPONSE TO RTS TOGGLING
                        ;       BY SENDING AN 'M' 4DH
                        ;********************************************************
                        ;
0113 BDEF6B             XINTR:  RBIT    IPND,PSW
0116 BD0B77                     IFBIT   USOFT,MTYPE     ;ONLY IF MICROSOFT PROTOCOL
0119 01                         JP      XINTR1          ;CONTINUE
011A EE                         JP      INTRET          ;ELSE DO NOTHING
                        XINTR1:
011B BC01FF                     LD      WORD1,#0FF      ;ALL MARK
011E BC024D                     LD      WORD2,#'M'
0121 BC0702                     LD      NUMWORD,#02
                        ;
0124 9D08                       LD      A,SENDST
0126 9200                       IFEQ    A,#0            ;IF IDLE, SEND 'M'
0128 05                         JP      RTSR2
                        ;
0129 BC0001                     LD      WORDPT,#WORD1   ;FAKE CONTINUE LAST CHAR
012C 2109         +             JP      INTRET
                        ;
                        RTSR2:
012E BC0002                     LD      WORDPT,#WORD2   ;'M' ONLY
0131 BC0801                     LD      SENDST,#01
0134 2109         +             JP      INTRET
                        ;
                        ;********************************************************
                        ;       SUBROUTINE SET UP REPORT 'SRPT' FOR MICROSOFT
                        ;                                       -------------
                        ;       CHANGE OF STATUS DETECTED
```

TL/DD/10799–14

```
                    ;       SET UP THE 3 WORDS FOR REPORTING IF IN IDLE STATE
                    ;*********************************************************
                    ;
                    SRPTUS:
0136 5B                     LD      B,#CHANGE
0137 70                     IFBIT   RPT,[B]
0138 01                     JP      SRUS1
0139 8E                     RET                     ;EXIT IF NOT CHANGE
                    ;
                    SRUS1:
013A BDEF68                 RBIT    GIE,PSW         ;DISABLE INTERRUPT
013D 5F                     LD      B,#WORDPT
013E 9A01                   LD      [B+],#WORD1     ;(WORDPT)SET WORD POINTER
0140 9D05                   LD      A,XINC
0142 65                     SWAP    A
0143 B0                     RRC     A
0144 B0                     RRC     A
0145 9503                   AND     A,#03           ;X7,X6
0147 A6                     X       A,[B]           ;(WORD1)
                    ;
0148 9D06                   LD      A,YINC
014A 65                     SWAP    A
014B 950C                   AND     A,#0C           ;Y7,Y6
014D 87                     OR      A,[B]           ;(WORD1)
014E 9740                   OR      A,#040          ;SET BIT 6
0150 BD0F87                 OR      A,BUTSTAT       ;GET BUTTON STATUS
0153 A2                     X       A,[B+]          ;(WORD1)
                    ;
0154 9D05                   LD      A,XINC
0156 953F                   AND     A,#03F          ;X0-X5
0158 A2                     X       A,[B+]          ;(WORD2)
                    ;
0159 9D06                   LD      A,YINC
015B 953F                   AND     A,#03F          ;Y0-Y5
015D A2                     X       A,[B+]          ;(WORD3)
015E 68                     RBIT    RPT,[B]         ;(CHANGE)RESET CHANGE OF STATUS
015F AA                     LD      A,[B+]          ;INC B
0160 9A00                   LD      [B+],#0         ;(XINC)
0162 9A00                   LD      [B+],#0         ;(YINC)
                    ;
0164 9A03                   LD      [B+],#03        ;(NUMWORD)SEND 3 BYTES
0166 9E01                   LD      [B],#01         ;(SENDST)SET TO START BIT STATE
                    ;
0168 BDEF78                 SBIT    GIE,PSW         ;ENABLE INTERRUPT
016B 8E                     RET
                    ;
                    ;*********************************************************
                    ;       SUBROUTINE SET UP REPORT 'SRPT' FOR MOUSE SYSTEMS
                    ;                                   -----------------
                    ;       CHANGE OF STATUS DETECTED
                    ;       SET UP THE FIRST 3 WORDS FOR REPORTING
```

TL/DD/10799–15

```
                    ;       IF IN IDLE STATE
                    ;**********************************************************
                    ;
                    SRPTMS:
016C 5B                     LD      B,#CHANGE
016D 70                     IFBIT   RPT,[B]
016E 01                     JP      SRMS1
016F 8E                     RET                     ;EXIT IF NO CHANGE
                    ;
                    SRMS1:
0170 BDEF68                 RBIT    GIE,PSW         ;DISABLE INTERRUPT
0173 5F                     LD      B,#WORDPT
0174 9A01                   LD      [B+],#WORD1     ;(WORDPT)SET WORD POINTER
0176 9D0F                   LD      A,BUTSTAT
0178 A2                     X       A,[B+]          ;(WORD1)
                    ;
0179 9D05                   LD      A,XINC
017B A2                     X       A,[B+]          ;(WORD2)
                    ;
017C A1                     SC
017D 64                     CLR     A
017E BD0681                 SUBC    A,YINC          ;FOR MOUSE SYSTEM NEG Y
0181 A2                     X       A,[B+]          ;(WORD3)
                    ;
0182 68                     RBIT    RPT,[B]         ;(CHANGE)RESET CHANGE OF STATUS
0183 79                     SBIT    SYRPT,[B]       ;(CHANGE)
0184 AA                     LD      A,[B+]          ;INC B
0185 9A00                   LD      [B+],#0         ;(XINC)
0187 9A00                   LD      [B+],#0         ;(YINC)
                    ;
0189 9A03                   LD      [B+],#03        ;(NUMWORD)SEND 3 BYTES
018B 9E01                   LD      [B],#01         ;(SENDST)SET TO START BIT STATE
                    ;
018D BDEF78                 SBIT    GIE,PSW         ;ENABLE INTERRUPT
0190 8E                     RET
                    ;
                    ;
                    ;**********************************************************
                    ;       SUBROUTINE TO SEND DATA 'SDATA'
                    ;       CHECK THE BIT TO SEND AND DRIVE THE OUTPUT TO THE
                    ;       DESIRED VALUE
                    ;
                    ;       SENDST          STATE
                    ;         0             IDLE
                    ;         1             START BIT
                    ;         2-8           DATA
                    ;         2-9           DATA    (FOR MOUSE SYSTEMS)
                    ;         9-10          STOP BIT
                    ;         10-11         STOP BIT   (FOR MOUSE SYSTEMS)
                    ;         11            NEXT WORD
                    ;         12            NEXT WORD  (FOR MOUSE SYSTEMS)
```

TL/DD/10799-16

```
                        ;
                        ;**********************************************************
                        ;
0191 55                 SDATA:  LD      B,#TSTATUS
0192 72                         IFBIT   TBAUB,[B]       ;(TSTATUS)CHECK IF BAUD RATE TIMER ENDS
0193 01                         JP      SDATA1
0194 8E                         RET
                        ;
                        SDATA1:
0195 6A                         RBIT    TBAUB,[B]       ;(TSTATUS)
0196 AA                         LD      A,[B+]          ;INC B TO (MTYPE)
0197 9D08                       LD      A,SENDST
0199 97F0                       OR      A,#0F0
019B A5                         JID
                        ;
019C 8E                 IDLE:   RET                     ;EXIT IF IDLE
                        ;
019D 77                 STAT9:  IFBIT   USOFT,[B]       ;(MTYPE)
019E 16                         JP      STOPB
                        DATAB:
019F 9D00                       LD      A,WORDPT
01A1 9CFE                       X       A,B             ;B POINTS TO THE WORD
                        ;
01A3 A0                         RC
01A4 AE                         LD      A,[B]
01A5 B0                         RRC     A               ;XMIT LEAST SIG BIT
01A6 A6                         X       A,[B]
01A7 DED4                       LD      B,#PORTGD
01A9 88                         IFC
01AA 7A                         SBIT    XMT,[B]
01AB 89                         IFNC
01AC 6A                         RBIT    XMT,[B]
                        ;
01AD 9D08               NEXT:   LD      A,SENDST
01AF 8A                         INC     A
01B0 9C08                       X       A,SENDST
01B2 8E                         RET                     ;EXIT
                        ;
01B3 77                 STAT11: IFBIT   USOFT,[B]       ;(MTYPE)
01B4 04                         JP      NXWORD
                        ;
01B5 BDD47A             STOPB:  SBIT    XMT,PORTGD
01B8 F4                         JP      NEXT
                        ;
01B9 9D00               NXWORD: LD      A,WORDPT
01BB 8A                         INC     A
01BC BD0783                     IFGT    A,NUMWORD       ;NUMBER OF WORDS TO SEND
01BF 09                         JP      ENDRPT          ;END OF REPORT
01C0 9C00                       X       A,WORDPT
01C2 BC0801                     LD      SENDST,#01      ;SEND START BIT
                        ;
```

TL/DD/10799–17

```
01C5 BDD46A     STARTB: RBIT   XMT,PORTGD      ;SEND START BIT
01C8 E4                 JP     NEXT
                ;
01C9 BD0471     ENDRPT: IFBIT  SYRPT,CHANGE
01CC 04                 JP     SY2RPT
                ;
01CD BC0800             LD     SENDST,#0
01D0 8E                 RET
                ;
                ;**************************************************
                ;       SET UP LAST 2 WORDS IN MOUSE SYSTEM FORMAT
                ;**************************************************
                ;
                SY2RPT:
01D1 BDEF68             RBIT   GIE,PSW         ;DISABLE INTERRUPT
                ;
01D4 5F                 LD     B,#WORDPT
01D5 9A01               LD     [B+],#WORD1     ;(WORDPT)SET WORD POINTER
01D7 9D05               LD     A,XINC
01D9 A2                 X      A,[B+]          ;(WORD1)
                ;
01DA A1                 SC
01DB 64                 CLR    A
01DC BD0681             SUBC   A,YINC          ;FOR MOUSE SYSTEM NEG Y
01DF A2                 X      A,[B+]          ;(WORD2)
                ;
01E0 AA                 LD     A,[B+]          ;INC B
01E1 69                 RBIT   SYRPT,[B]       ;(CHANGE)RESET CHANGE OF STATUS
01E2 AA                 LD     A,[B+]          ;INC B
01E3 9A00               LD     [B+],#0         ;XINC
01E5 9A00               LD     [B+],#0         ;YINC
                ;
01E7 9A02               LD     [B+],#02        ;(NUMWORD)SEND 2 BYTES
01E9 9E01               LD     [B],#01         ;(SENDST)SET TO START BIT STATE
                ;
01EB BDEF78             SBIT   GIE,PSW         ;ENABLE INTERRUPT
01EE 21C5     +         JP     STARTB
                ;
     01F0               .=01F0
                ;
01F0 9C                 .ADDR  IDLE            ;0
01F1 C5                 .ADDR  STARTB          ;1
01F2 9F                 .ADDR  DATAB           ;2
01F3 9F                 .ADDR  DATAB           ;3
01F4 9F                 .ADDR  DATAB           ;4
01F5 9F                 .ADDR  DATAB           ;5
01F6 9F                 .ADDR  DATAB           ;6
01F7 9F                 .ADDR  DATAB           ;7
01F8 9F                 .ADDR  DATAB           ;8
01F9 9D                 .ADDR  STAT9           ;9
01FA B5                 .ADDR  STOPB           ;10
```

TL/DD/10799–18

```
01FB B3              .ADDR   STAT11          ;11
01FC B9              .ADDR   NXWORD          ;12
01FD 9C              .ADDR   IDLE            ;13
01FE 9C              .ADDR   IDLE            ;14
01FF 9C              .ADDR   IDLE            ;15
             ;
             ;
             ;**************************************************
             ;       SAMPLE BUTTON INPUT     FOR MICROSOFT
             ;                               -------------
             ;       INDICATE BUTTON STATUS
             ;**************************************************
             ;
             BUTUS:
0200 9E00            LD      [B],#0          ;(BTEMP), (A=PORTLP, CARRY ROTATED)
0202 89              IFNC                    ;MICROSOFT: 1=KEY DEPRESSED
0203 7D              SBIT    5,[B]           ;(BTEMP)
             ;
0204 B0              RRC     A
0205 B0              RRC     A
0206 89              IFNC
0207 7C              SBIT    4,[B]           ;(BTEMP)
             ;
0208 AA              LD      A,[B+]          ;(BTEMP)
0209 82              IFEQ    A,[B]           ;(BUTSTAT)
020A 8E              RET                     ;NO CHANGE
             ;
020B A6              X       A,[B]           ;(BUTSTAT)
020C BD0478          SBIT    RPT,CHANGE      ;INDICATE TO SEND DATA
020F 8E              RET
             ;
             ;
             ;**************************************************
             ;       SAMPLE BUTTON INPUT     FOR MOUSE SYSTEMS
             ;                               -----------------
             ;       INDICATE BUTTON STATUS
             ;**************************************************
             ;
             BUTMS:
0210 9E87            LD      [B],#087        ;(BTEMP)
             ;
0212 89              IFNC                    ;MOUSE SYSTEM: 0=KEY DEPRESSED
0213 6A              RBIT    2,[B]           ;(BTEMP)
             ;
0214 B0              RRC     A
0215 89              IFNC
0216 69              RBIT    1,[B]           ;(BTEMP)
             ;
0217 B0              RRC     A
0218 89              IFNC
0219 68              RBIT    0,[B]           ;(BTEMP)
```

TL/DD/10799–19

```
                         ;
021A AA                        LD      A,[B+]          ;(BTEMP)
021B 82                        IFEQ    A,[B]           ;(BUTSTAT)
021C 8E                        RET                     ;NO CHANGE
                         ;
021D A6                        X       A,[B]           ;(BUTSTAT)
021E BD0478                    SBIT    RPT,CHANGE      ;INDICATE TO SEND DATA
0221 8E                        RET
                         ;
                         ;************************************************************
                         ;
      03D0                     .=03D0
03D0 28                        .BYTE   '(C) 1990 NATIONAL SEMICONDUCTOR AMOUSE VER 1.0'
03D1 43
03D2 29
03D3 20
03D4 31
03D5 39
03D6 39
03D7 30
03D8 20
03D9 4E
03DA 41
03DB 54
03DC 49
03DD 4F
03DE 4E
03DF 41
03E0 4C
03E1 20
03E2 53
03E3 45
03E4 4D
03E5 49
03E6 43
03E7 4F
03E8 4E
03E9 44
03EA 55
03EB 43
03EC 54
03ED 4F
03EE 52
03EF 20
03F0 41
03F1 4D
03F2 4F
03F3 55
03F4 53
03F5 45
03F6 20
```

TL/DD/10799-20

```
03F7 56
03F8 45
03F9 52
03FA 20
03FB 31
03FC 2E
03FD 30
                         ;
                               .END
```

TL/DD/10799-21

```
NATIONAL SEMICONDUCTOR CORPORATION
COP800 CROSS ASSEMBLER,REV:D1,12 OCT 88
AMOUSE
SYMBOL TABLE


ASAVE  00F4     B      00FE     BSAVE  00F5  *  BTEMP  000E
BUSY   0002  *  BUTMS  0210     BUTSTA 000F     BUTUS  0200
CARRY  0006  *  CHANGE 0004     CKO    0007  *  CNTRL  00EE
COMX   0092     COMY   00D8     CSEL   0006  *  DATAB  019F
DECX   008F     DECY   00D5     ENDRPT 01C9     ENI    0001  *
ENTI   0004  *  ESENS  00E0     GIE    0000     GTEMP  000C
HCARRY 0007  *  IDLE   019C     IEDG   0002  *  INCX   008B
INCY   00D1     INTR   0000  *  INTRET 0109     INTRP  00FF  *
IPND   0003     LPUS   005C     LTIMER 0027  *  MLOOP  003D
MOVEMX 00B0  *  MOVEMY 00C0  *  MSEL   0003  *  MTYPE  000B
NEXT   01AD     NOISEX 008A     NOISEY 00D0     NUMWOR 0007
NXWORD 01B9     PORTGC 00D5     PORTGD 00D4     PORTGP 00D6
PORTLC 00D1     PORTLD 00D0     PORTLP 00D2     PSSAVE 00F6  *
PSW    00EF     RPT    0000     RSVD   00F0  *  RTSR2  012E
S0     0000  *  S1     0001  *  SDATA  0191     SDATA1 0195
SELECT 0067     SENDST 0008     SENSOR 0077     SI     0006  *
SK     0005  *  SO     0004  *  SP     00FD     SRMS1  0170
SRPTMS 016C     SRPTUS 0136     SRUS1  013A     START  0000  *
STARTB 01C5     STAT11 01B3     STAT9  019D     STOPB  01B5
SW     0003     SY2RPT 01D1     SYM    0071     SYRPT  0001
TAUHI  00ED  *  TAULO  00EC  *  TBAU   00F3  *  TBAUB  0002
TBAUR  0009  *  TEDG   0005  *  TEMP   00F1  *  TINTR  010C
TIO    0003  *  TMRHI  00EB  *  TMRLO  00EA     TPND   0005
TRACKS 000D     TRUN   0004     TSEL   0007  *  TSTATU 000A
USM    006B     USOFT  0007     WORD1  0001     WORD2  0002
WORD3  0003  *  WORDPT 0000     X      00FC     XINC   0005
XINTR  0113     XINTR1 011B     XMT    0002     YDIR   009A
YINC   0006
```

TL/DD/10799–22

```
NATIONAL SEMICONDUCTOR CORPORATION
COP800 CROSS ASSEMBLER,REV:D1,12 OCT 88
AMOUSE
MACRO TABLE


   NO WARNING LINES

   NO ERROR LINES

  556  ROM BYTES USED

SOURCE CHECKSUM = 987A
OBJECT CHECKSUM = 0A39

INPUT   FILE D:BMOUSE.MAC
LISTING FILE D:BMOUSE.PRN
OBJECT  FILE D:BMOUSE.LM
```

TL/DD/10799–23

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation
2900 Semiconductor Drive
P.O. Box 58090
Santa Clara, CA 95052-8090
Tel: 1(800) 272-9959
TWX: (910) 339-9240

National Semiconductor GmbH
Livry-Gargan-Str. 10
D-82256 Fürstenfeldbruck
Germany
Tel: (81-41) 35-0
Telex: 527649
Fax: (81-41) 35-1

National Semiconductor Japan Ltd.
Sumitomo Chemical Engineering Center
Bldg. 7F
1-7-1, Nakase, Mihama-Ku
Chiba-City,
Ciba Prefecture 261
Tel: (043) 299-2300
Fax: (043) 299-2500

National Semiconductor Hong Kong Ltd.
13th Floor, Straight Block,
Ocean Centre, 5 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 2737-1600
Fax: (852) 2736-9960

National Semiconductores Do Brazil Ltda.
Rue Deputado Lacorda Franco
120-3A
Sao Paulo-SP
Brazil 05418-000
Tel: (55-11) 212-5066
Telex: 391-1131931 NSBR BR
Fax: (55-11) 212-1181

National Semiconductor (Australia) Pty, Ltd.
Building 16
Business Park Drive
Monash Business Park
Nottinghill, Melbourne
Victoria 3168 Australia
Tel: (3) 558-9999
Fax: (3) 558-9998

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Audio	www.ti.com/audio	Communications and Telecom	www.ti.com/communications
Amplifiers	amplifier.ti.com	Computers and Peripherals	www.ti.com/computers
Data Converters	dataconverter.ti.com	Consumer Electronics	www.ti.com/consumer-apps
DLP® Products	www.dlp.com	Energy and Lighting	www.ti.com/energy
DSP	dsp.ti.com	Industrial	www.ti.com/industrial
Clocks and Timers	www.ti.com/clocks	Medical	www.ti.com/medical
Interface	interface.ti.com	Security	www.ti.com/security
Logic	logic.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Power Mgmt	power.ti.com	Transportation and Automotive	www.ti.com/automotive
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Mobile Processors	www.ti.com/omap		
Wireless Connectivity	www.ti.com/wirelessconnectivity		
		TI E2E Community Home Page	e2e.ti.com

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265