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 with slots. 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 LED is 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.
Signals at phototransistors are similar for vertical and horizontal motion.
Track 1 leads track 0 by 90 degrees.
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 \(15.7/160 \approx 0.098\)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 npn 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.

<table>
<thead>
<tr>
<th>(TRK1, TRK0) _t</th>
<th>(TRK1, TRK0) _t-1</th>
<th>Binary Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CW</td>
<td>CCW</td>
<td></td>
</tr>
<tr>
<td>0 1 0 0 4</td>
<td>0 0 0 0 E</td>
<td></td>
</tr>
<tr>
<td>1 1 0 1 D</td>
<td>0 0 0 1 1</td>
<td></td>
</tr>
<tr>
<td>1 0 1 1 B</td>
<td>0 1 1 1 7</td>
<td></td>
</tr>
<tr>
<td>0 0 1 0 2</td>
<td>1 1 1 0 E</td>
<td></td>
</tr>
</tbody>
</table>

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 npn 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.
Cycles: ;**********************************************************************
; SAMPLE SENSOR INPUT
; INC OR DEC THE POSITION
;**********************************************************************
;
; SENSOR:
1    LD    B,#GTEMP
3    LD    A,PORT GP
1    RRC   A
2    AND   A,#03C ; G6,G5,G4,G3
1    X     A, [B] ; (GTEMP)

2    LD    A, [B+1] ; (GTEMP) X IN 3,2
1    RRC   A
1    RRC   A
2    OR    A, #03
1    OR    A, [B] ; (TRACKS)
2    OR    A, #080 ; X MOVEMENT TABLE
3    JID

NOISEX: JP YDIR
;
3    INCX: LD A,XINC
1    INC   A
3    JP COMX
;
DECX:  LD A,XINC
1    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, #000
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 SHIT RPT, [B]
1 LD $B, #GTEMP
ESENS:
2 LD A, [B+]; (GTEMP) INS, 4, 1, 0
1 X A, [B] ; (TRACKS) NEW TRACK STATUS
5 RET
;

.; =0BO
MOVEMX:
.ADDR NOISEX ; 0
.ADDR INCX ; 1
.ADDR DECY ; 2
.ADDR DECY ; 3
.ADDR NOISEX ; 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
;

.; =0CO
MOVEMY:
.ADDR NOISEY ; 0
.ADDR INCY ; 1
.ADDR DECY ; 2
.ADDR DECY ; 3
.ADDR NOISEY ; 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

5
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

<table>
<thead>
<tr>
<th>Bit</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>L</td>
<td>R</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>X1</td>
<td>X0</td>
</tr>
<tr>
<td>Y1</td>
<td>Y0</td>
</tr>
<tr>
<td>Y2</td>
<td>Y1</td>
</tr>
<tr>
<td>X2</td>
<td>X1</td>
</tr>
<tr>
<td>X7</td>
<td>X6</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Bit</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>X7</td>
<td>X6</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Y1</td>
<td>Y0</td>
</tr>
<tr>
<td>Y1</td>
<td>Y0</td>
</tr>
<tr>
<td>X2</td>
<td>X1</td>
</tr>
<tr>
<td>X7</td>
<td>X6</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Pin</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>G0</td>
<td>Interrupt Input (Monitoring RTS Toggle)</td>
</tr>
<tr>
<td>G1</td>
<td>Reserved for Input Data (TXD of Host)</td>
</tr>
<tr>
<td>G2</td>
<td>Output Data (RXD of Host)</td>
</tr>
<tr>
<td>G3–G6</td>
<td>LED Sensor Input</td>
</tr>
<tr>
<td>L0–L2</td>
<td>Button Input</td>
</tr>
<tr>
<td>L3</td>
<td>Jumper Input (for Default Mouse Mode)</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>SENDST</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>IDLE</td>
</tr>
<tr>
<td>1</td>
<td>START BIT</td>
</tr>
<tr>
<td>2–8</td>
<td>DATA (FOR MICROSOFT)</td>
</tr>
<tr>
<td>2–9</td>
<td>DATA (FOR MOUSE SYSTEMS)</td>
</tr>
<tr>
<td>9–10</td>
<td>STOP BIT (FOR MICROSOFT)</td>
</tr>
<tr>
<td>10–11</td>
<td>STOP BIT (FOR MOUSE SYSTEMS)</td>
</tr>
<tr>
<td>11</td>
<td>NEXT WORD (FOR MICROSOFT)</td>
</tr>
<tr>
<td>12</td>
<td>NEXT WORD (FOR MOUSE SYSTEMS)</td>
</tr>
</tbody>
</table>

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 records are stored 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 “SRPUTS” 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 ;NUMBER 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
SHIT 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
SHIT GIE,PSW ;ENABLE INTERRUPT
RET

```

7
CONCLUSION
The COP822C has been used as a mouse controller. The code presented is a minimum requirement for implementing a mouse system 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.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Advantage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port G</td>
<td>Schmitt Triggered Input for Photo-Transistors</td>
</tr>
<tr>
<td>G0</td>
<td>External Interrupt for RTS Toggling</td>
</tr>
<tr>
<td>Timer</td>
<td>For Baud Rate Generation</td>
</tr>
<tr>
<td>Low Power</td>
<td>4 mA at 5 MHz</td>
</tr>
<tr>
<td>Small Size</td>
<td>20-Pin DIP</td>
</tr>
</tbody>
</table>

REFERENCE

APPENDIX A—MEMORY UTILIZATION

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

APPENDIX B—SUBROUTINE SUMMARY

<table>
<thead>
<tr>
<th>Subroutine</th>
<th>Location</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLOOP</td>
<td>03D</td>
<td>Main Program Loop</td>
</tr>
<tr>
<td>SENSOR</td>
<td>077</td>
<td>Sample Photo-Transistor Input</td>
</tr>
<tr>
<td>INTRP</td>
<td>0FF</td>
<td>Interrupt Service Routines</td>
</tr>
<tr>
<td>SRPTUS</td>
<td>136</td>
<td>Set Up Report for Microsoft</td>
</tr>
<tr>
<td>SRPTMS</td>
<td>16C</td>
<td>Set Up 1st 3 Bytes Report for Mouse Systems</td>
</tr>
<tr>
<td>SDATA</td>
<td>191</td>
<td>Drive Data Transmission Pin According to Bit</td>
</tr>
<tr>
<td>SY2RPT</td>
<td>1D1</td>
<td>Set Up Last 2 Bytes Report for Mouse Systems</td>
</tr>
<tr>
<td>BUTUS</td>
<td>200</td>
<td>Sample Button Input for Microsoft</td>
</tr>
<tr>
<td>BUTMS</td>
<td>210</td>
<td>Sample Button Input for Mouse Systems</td>
</tr>
</tbody>
</table>
APPENDIX C—SYSTEM SCHEMATIC, SYSTEM
Flowchart, complete program listing.

Note 1: All diodes are 1N4148.
Note 2: All resistor values are in ohms, 5%, 1/8W.
Note: Unless otherwise specified

FIGURE 3. System Schematic
Flowchart for Mouse Systems and Microsoft Mouse

START

Initialization & Setup

Send data

Sense Movement

in the middle of sending report?

YES

Microsoft Mouse?

Sense button input for microsoft protocol

Set up report in microsoft format

NO

Set up report in mouse system format

NO

Mouse Systems

Sense button input for mouse systems protocol

Set up report in mouse systems format

TL/DD/10799-6
NATIONAL SEMICONDUCTOR CORPORATION

CVP800 CROSS ASSEMBLER, REV.D1.12 OCT 88

ANMUSE

; MICROSOFT AND MOUSE SYSTEM COMPATIBLE MOUSE
; 02/14/89
; NAME : ANMUSE.MAC

.TITLE ANMUSE
.CHIP #20

0000 FORTLD = 0000 ; PORT L DATA
0001 FORTLC = 0001 ; PORT L CONFIG
0002 FORTLP = 0002 ; PORT L PIN

0003 FORTGD = 0003 ; PORT G DATA
0004 FORTGC = 0004 ; PORT G CONFIG
0005 FORTGP = 0005 ; PORT G PIN

0006 TIMRO = 0006 ; TIMER LOW BYTE
0007 TIMHI = 0007 ; TIMER HIGH BYTE

0008 TAULO = 0008 ; TIMER REGISTER LOW BYTE
0009 TAUHI = 0009 ; TIMER REGISTER HIGH BYTE

000A CNTRL = 000A ; CONTROL REGISTER
000B PSW = 000B ; PSW REGISTER

; CONSTANT DECLARE

000C INTR = 0
000D TO = 3
000E SO = 4
000F SI = 5

0010 CRO = 7
0011 TSEL = 7
0012 CSEL = 6
0013 TEDG = 5
0014 TRUN = 4
0015 MSEL = 3
0016 IMEG = 2
0017 SI = 1
0018 SO = 0

0019 NCAR = 7
001A CARRY = 6
001B TPND = 5
001C ENTI = 4
001D IPND = 3
001E BUSY = 2
001F ENI = 1

0020 GIE = 0

TL/DD/10799-7
; TSTATUS BITS

; TAUB = 2 ; BAUD RATE TIMER BIT
0002
;
0000
;
RPT = 0 ; REPORT BIT OF CHANGE (CHANGE)
0001
;
STRT = 1 ; SET UP MOUSE SYSTEM LAST 2 WORDS (CHANGE)
0007
;
USOFI = 7 ; MICROSOFT (MTYPE)
0002
;
XMT = 2 ; G2 AS XMT BIT (PORT2)
0003

; SW = 3 ; SLIDE SWITCH (PORTLP, MTYPE)
64
;
REGISTER ASSIGNMENTS
65
;
COF0
;
RSVD = 0F0
06
;
COF1
;
TEMP = 0F1
68
;
COF3
;
TAUB = 0F3 ; BAUD RATE TIMER
69
;
COF4
;
ASAVE = 0F4 ; SAVE A
70
;
COF5
;
BSAVE = 0F5 ; SAVE B
71
;
COF6
;
PSAVE = 0F6 ; SAVE FSM
72

; VARIABLES
73
;
0000
;
WORDPT = 000 ; WORD POINTER
76
;
COO1
;
WORD1 = 001 ; BUFFER TO STORE REPORTS
77
;
COO2
;
WORD2 = 002
78
;
COO3
;
WORD3 = 003
79

; CHANGE = 004 ; MOVEMENT CHANGE OR BUTTON PRESSED
80
;
0005
;
XINC = 005 ; X DIRECTION COUNTER
82
;
0006
;
YINC = 006 ; Y DIRECTION COUNTER
83
;
0007
;
NUMWORD = 007 ; NUMBER OF BYTES TO SEND
84
;
0008
;
SENDST = 008 ; SERIAL PROTOCOL STATE
85

; TBAUR = 009 ; BAUD RATE TIMER RELOAD
86
;
000A
;
TSTATUS = 00A ; COUNTER STATUS
88
;
000B
;
MTYPE = 00B ; MOUSE TYPE
89

; GTEMP = 00C ; TRACK INPUT FROM G
90
;
000D
;
TRkX = 00D ; PREVIOUS TRACK STATUS
92
;
000E
;
BTEMP = 00E ; BUTTON INPUT
94
;
000F
;
BUTSTAT = 00F ; PREVIOUS BUTTON STATUS
95

; MOST POSITIVE = SPACE = HI = ON = 0 ; START BIT = RBIT
96
;
; MOST NEGATIVE = MARK = LO = OFF = 1 ; STOP BIT = SBIT
97
;
100
;
MICROSOFT FORMAT
101
;
1 L R Y7 Y6 X7 X6
102
; 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
; G0, 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
0002 BEE00
0005 CEEB00
0008 BCD504
000B BCD404
000E BCD130
0011 BCD00F
0014 5B
0015 9A00
0017 9A00
0019 9A00
001B 9A00
001E 9D6
0020 B0
0021 933C
0023 9CD
0025 3067

LD SF,#02F ;DISABLE INTR
LD PDM,#0
LD CNTRL,#080 ;10000000 - AUTORELOAD
;RISING EDGE EXT INT
LD PORTC,#004 ;G2 AS OUTPUT, OTHERS AS HI-Z
LD PORTD,#004 ;G2 DATA 1 "MARK"
LD PORTLC,#030 ;HI-Z INPUTS FOR L6-7, OUTPUT L4,5
LD PORTLD,#0F ;WEAK PULL UP FOR L0-3
LD B,#CHANGE
LD [B+],#0 ;(CHANGE)
LD [B+],#0 ;(XINC)
LD [B+],#0 ;(YINC)
LD TSTATUS,#0
LD A,PORTC
RRC A
AND A,#03C ;NOW IN 6,5,4,3
AND X,A,TRACKS ;GET INITIAL VALUE OF SENSORS
JSR SELECT ;SELECT MOUSE TYPE

TL/DD/10799-9
154 ;********************************************************************************
155 ;
156 ; CRYSTAL FREQ = 4.96 Mhz  2.016 US INST CYCLE
157 ; FOR 3200 BAUD - TIMER = 413 COUNT
158 ;
159 ;********************************************************************************
160 ;
161  STIMER:
162  0027 35E0           LD  @,PROMRD  ;FOR 2.016 US CYCLE
163  0029 35B0           LD  @$4890D
164  002A 35D0           LD  @$1,4890D
165  002B 35D0           LD  @$1,4890D
166  002C 35D0           LD  @$1,4890D
167 ;
168  0033 35D0           LD  SENST,80 ;SET TO IDLE STATE
169  0034 35D0           LD  A,59W
170  0035 35D0           OR  A,4033 ;ENABLE INTRS SET CIE
171  0036 35D0           X  A,PE
172  0036 35D0           X  @,TOKC;START TIMER
173 ;
174  MLOOP:
175  003D 35D0           LD  PORTD,DBP ;TURN ON LED (NOT USED)
176  0040 3160           JSR  SINTE
177  0042 3877           JSR  SENSE
178  0044 35D0           LD  A,SENSR ;IF SENSING REPORT
179  0046 35D0           JSR  A,80 ;JUST DO SENSOR
180  0048 35D0           JP  MLOOP
181 ;
182  0049 35D0           LD  A,PORTLP ;GET INPUT FROM BUTTONS (10,11,12)
183  004B 35D0           MOV  A,95 ;PUT IN CARRY FOR CHECKING
184  004D 35D0           LD  B,RTMP ;PREPARATION TO SEE WANT BUTTON IS PRESSED
185 ;
186  004E 35D7           JSR  START,MSYS
187  0050 60           JP  LPSR
188 ;
189  0051 3210           JSR  BUTMS ;MOUSE SYSTEMS
190  0053 3164           JSR  DINPMS
191 ;
192  0055 30273          JSR  SW,PORTLP
193  0056 35D0           JP  MLOOP ;CONTINUE IF NO CHANGE IN SWITCH
194  0059 304B           JSR  UCOM ;ELSE NEW SET UP
195  005B 40           JP  MLOOP
196 ;
197  005C 3100           JSR  BUTMS ;MICROSOFT
198  005E 3136           JSR  SWCOPY
199 ;
200  0060 30273          JSR  SW,PORTLP
201  0063 3171           JSR  SYM ;IF CHANGED IN SWITCH, NEW SET UP
202  0066 2130           JP  MLOOP
203 ;
204 ;********************************************************************************

TL/DD/10799-10
205 ; SELECT MOUSE TYPE
206
207
208 SELECT:
209 D067 BD3273 JBIT SW,PORTP ; CHECK JUMPER
210 D06A 06 JP """"""SIM"
211 ;
212 ;
213 D068 54 LD B,WMYP
214 D06C 1F SBIT U167, [0] ; (WMYP) IS NAVITEC MODEL
215 D06D BD3F07 LD REGH, [0] ; NO KEYPRESSED
216 0706 RE RET
217 ;
218 ;
219 D071 54 LD B,WMYP
220 0702 4F SBIT U167, [0] ; (WMYP) IS MOUSE SYSTEMS
221 D073 BD3F00 LD REGH, [0] ; NO KEYPRESSED
222 0706 RE RET
223 ;
224 ;
225 ; SAMPLE SENSOR INPUT
226 ; INC OR DEC THE POSITION
227 ; -127 IS USED INSTEAD OF -128 IN CHECKING
228 ; NEGATIVE-going POSITION SO THAT BOTH
229 ; NAVITEC AND MOUSE SYSTEMS FIT IN
230 ;
231 ;
232 SENSOR:
233 D077 53 LD B,RTMP
234 D07B 9016 LD A,PORTP
235 D07E BD3F0F LD PORTW, AS : (NOT USED) TURN OFF LED
236 D07F BD 00 NFC A
237 D07F 953C AND A, #3C ; C5, C4, C3, C2
238 1080 A6 X A, [0] ; (RTMP)
239 ;
240 ;
241 ;
242 ;
243 ;
244 ;
245 ;
246 ;
247 ;
248 ;
249 ;
250 ;
251 ;
252 D0BE AA LD A, [17] ; (RTMP) X IN 1, 2
253 D0BF BD NFC A
254 D0B0 BD NFC A
255 D0BE 9503 AND A, #83 ; SET X TRACKS
086 87 OR A,[B] ; OVERLAY WITH PREVIOUS (TRACKS)
087 9780 ON A, [B] ; MOVEMENT TABLE
088 089 A5 JID
089 ;
090 091 A5 NOISEX, JP YDIR
091 ;
092 ;
093 INCR:
094 095 B05 LD A,KINC
095 096 B0 A INC A
096 097 B3 JP COMK ; CHECK IF LIMIT IS REACHED
097 ;
098 DISK:
099 09B B05 LD A,KINC
100 09C B0 DIS A
101 ;
102 COMK:
103 092 025 LT EQ A, #80
104 093 B5 JP YDIR ; Y IS ZERO, DO NOTHING
105 095 B13 X A,KINC ; LEAVE NEW POSITION
106 097 3B LD B,#CHANGE
107 098 3B XOR A, BPT, [B] ; CHANGE
108 099 32 LD B,#TRACKS
109 ;
110 ;
111 YDIR:
112 09A 52 LD B,#TRACKS
113 09B 5D LD A,[B-] ; (TRACKS) Y IN 5,4
114 09C 6D SWAP A
115 09D 6D XRC A
116 09E 6D XRC A
117 09F 6D XRC A
118 0A0 6D AND A, #FOC
119 0A1 6D OR A,[B] ; [TIMES]
120 0A2 65 SWAP A
121 0A3 6D OR A, #FOC ; Y MOVEMENT TABLE
122 0A4 A5 JID
123 ;
124 ;
125 MOVEX:
126 0A5 8A ADDR NOISEX, #0
127 0A6 8F ADDR DECX, #1
128 0A7 8B ADDR INCX, #2
129 0A8 8A ADDR NOISEX, #3
130 0A9 8B ADDR INCX, #4
131 0AA 8A ADDR NOISEX, #5
132 0AB 8A ADDR NOISEX, #6
133 0AC 8F ADDR DECX, #7
134 0AD 8A ADDR NOISEX, #8
135 0AE 8A ADDR NOISEX, #9
136 0AF 8A ADDR NOISEX, #A
137 0B0 8B ADDR INCX, #B
138 0B1 8A ADDR NOISEX, #C
139 0B2 8B ADDR INCX, #D
140 0B3 8B ADDR DECX, #E

TL/DD/19799-12
30 DBY RA
  JADDR MOISEY ;P
31
32 DOC
  ;<DOC
33
34 MOVE:
35 1 ODC 00
  JADDR MOISEY ;0
36 2 ODC 01
  JADDR INCY ;1
37 3 ODC 02
  JADDR DECY ;2
38 4 ODC 00
  JADDR MOISEY ;3
39 5 ODC 05
  JADDR DECY ;4
40 6 ODC 00
  JADDR MOISEY ;5
41 7 ODC 01
  JADDR INCY ;6
42 8 ODC 01
  JADDR INCY ;7
43 9 ODC 00
  JADDR MOISEY ;9
44 10 ODC 00
  JADDR MOISEY ;A
45 11 ODC 05
  JADDR DECY ;B
46 12 ODC 00
  JADDR MOISEY ;C
47 13 ODC 05
  JADDR DECY ;D
48 14 ODC 00
  JADDR INCY ;E
49 15 ODC 00
  JADDR MOISEY ;F
50
51 DIS 00 OF
  MOISEY ;F IEENS
52
53 7 DOX 006
  INC A A,FINC
54 8 DOX 00A
  INC A
55 9 DOX 03
  JP COMY
56
57 DICT:
58 0 DOX 005
  LD A,FINC
59 1 DICT 88
  DEC A
60
61 COM:
62 2 ODD 920
  STFQ A,PF
63 3 ODD 05
  JP IEENS
64 4 ODD 070
  X A,FINC
65 5 ODD 58
  LD B,MOISEY
66 6 DOR 78
  BXH HPT,[B] ;(CHANGE)
67 7 ODF 53
  LD B,BTEMP
68
69 IEENS:
70 0 000 53
  LD B,BTEMP
71 1 001 8A
  LD A,[B] ;(BTEMP IN 9,1,0
72 2 002 8A
  X A,[B] ;(TRACKS NEW TRACK STATUS
73 3 003 BE
  RET
74
75
76 DOFF
  ;<DOFF
77
78 ;*****************************************************************************************************************************
79
80 ;*****************************************************************************************************************************
81
82
83 DOFF KRY1
  INTF A A,DSVE

TL/DD/10790-13
358 ;
359 010: ING; TS IFIT TPNO, PSW
360 0104 ST JP TINTH
361 0105 BEQX3 JS FP TPNO, PSW
362 0106 BA JP TINTH
363 ;
364 INTENT: ; INTERRUPT RETURN
365 0109 NEF4 LD A, A, A
366 0109 FF RETI
367 ;
368 ;******************************************************************************
369 ; TIMER INTERRUPT
370 ; UPDATE ALL THE COUNTERS
371 ;******************************************************************************
372 ;
373 TINTH: RETI TPNO, PSW
374 0107 NEF4A JR ST TINTH, STATUS ; SET BIT IN STATUS
375 0102 P6 JP INTENT
376 ;
377 ;******************************************************************************
378 ; RESPONSE TO RTS TOGGING
379 ;******************************************************************************
380 ; BY SENDING AN "M" CODE
381 ;******************************************************************************
382 ;
383 ;
384 011: XINTH: RETI TPNO, PSW
385 011: ING; TS IFIT RST, MTYPE ; ONLY IF MICROSOFT PROTOCOL
386 0119 01 JP XINTH1 ; CONTINUE
387 011A 01 JP INTENT ; ELSE DO NOTHING
388 XINTH1: LD A, 'H' ; ALL MARK
389 011B NEF2F JR LD 'M', 'W'
390 011C NEF2D JR LD 'W', 'M'
391 011D NEF2E JR LD 'W', 'W'
392 ;
393 0124 NEF5B JR LD A, 'SEND'
394 0126 NEF59 JR EQ A, '0' ; IF IDLE, SEND "W"
395 0128 05 JR RTS2
396 ;
397 0129 NEF50 JR LD 'H', 'SEND', 'W', 'H', 'H', 'D' ; FAKE CONTINUE LAST CHAR
398 012C 21 9A JR JP INTENT
399 ;
400 ;
401 NEF2: JR LD 'H', 'SEND', 'W', 'H', 'D' ; 'M' ONLY
402 0131 NEF2B JR LD 'SEND', '0' ;
403 0134 21 9A JR JP INTENT
404 ;
405 ;******************************************************************************
406 ; SUBMITTING SET OF REPORT "SHIFT" FOR MICROSOFT
407 ;******************************************************************************
408 ; CHANGE OF STATUS DETECTED

TL/CA/10799–14
19

; SET UP THE 3 WORDS FOR REPORTING IF IN IDLE STATE
10 ;
11 ;
12 depth:  
13 3136 5B    LD  B,CIANCE
14 3137 70    INBIT  RPT,[B]
15 3138 01    JP  sKuS
16 3139 BE    RET  ;EXIT IF NOT CHANCE
17 ;
18 sKuS:  
19 313A B0EF80    RBIT  GIL,PSW  ;DISABLE INTERRUPT
20 313B 5F    LD  B,WORDP
21 313C B830    LD  [B],(WORD)  ;(WORD)SET WORD POINTER
22 313D 8C05    LD  A,INC
23 313E 65    SWAP  A
24 313F BD    RRC  A
25 3140 B0    RRC  A
26 3141 5523    AND  A,#13  ;K7,FR
27 3142 D6    X  A,[B]  ;(WORD)
28 ;
29 3143 8C06    LD  A,INC
30 3144 65    SWAP  A
31 3145 852C    AND  A,RCC  ;17,FR
32 3146 BD    OR  A,[B]  ;(WORD)
33 3147 9730    OR  A,#150  ;SET BIT 8
34 3148 BD07    OR  A,STAT  ;SET BUTTON STATUS
35 3149 53A2    X  A,[B]  ;(WORD)
36 ;
37 314A 6505    LD  A,INC
38 314B 852F    AND  A,ROF  ;RO-FL
39 314C FA78    X  A,[B]  ;(WORD)
40 ;
41 314D B06F    LD  A,INC
42 314E 952F    AND  A,ROF  ;RO-FL
43 314F 50F2    X  A,[B]  ;(WORD)
44 3150 BD78    OR  A,STAT  ;RESET BUTTON STATUS
45 3151 B830    LD  [B],#8  ;(INC)
46 3152 BD80    LD  [B],#8  ;(INC)
47 3153 BD80    LD  [B],#8  ;(INC)
48 ;
49 3154 0103    LD  [B],#3  ;(MMWORD)SEND 3 BYTES
50 3155 CD01    LD  [B],#1  ;(SEND)SET TO START BIT STATE
51 ;
52 3156 B0EF87    RBIT  GIL,PSW  ;ENABLE INTERRUPT
53 3157 BE    RET  ;
54 ;
55 ;***********************************************************************
56 ; SUBROUTINE SET UP REPORT "SEND" FOR MOUSE SYSTEMS
57 ;
58 ; CHANGE OF STATUS DETECTED
59 ; SET UP THE FIRST 3 WORDS FOR REPORTING

TL/DD/10799--15
460  ; IF IN IDLE STATE
461  ;*******************************************************************************
462  ;
463  ; START:
464  ;
465  014C 5B   LD B, RCHG
466  014D 70   STRT RPT, [B]
467  014E 81   JR SHMT2
468  014F 8E   RET
469  ; EXIT IF NO CHANGE
470  ;
471  ; SHMT1:
472  0170 82EF40  BSET  G1, PW
473  0173 5F    LD B, RPM2
474  0174 9AD1   LD [B], RDATA
475  0176 9DFE   SET WRT, WPTR
476  0178 A2    X A, [B]=
477  0179 9D05   ; WRTDI
478  ;
479  017C A1    SC
480  017D 64    CLR A
481  017E 82DA71  SUBC A, [B]=
482  0181 A2    X A, [B]=
483  ;
484  0182 68    BRST RPT, [B]
485  0183 79    SETW WPTR, [B]
486  0184 AA    LD A, [B]=
487  0185 9A00   INC B
488  0188 9A00   LD [B]=, #0
489  0189 9A00   ; INC
490  0189 9A00   LD [B]=, #0
491  0199 9E01   ; GDIS
492  019D B278   SBIT G13, PW
493  019E 8E   RET
494  ;
495  ;
496  ;*******************************************************************************
497  ; SUBROUTINE TO SEND DATA 'DATA'
498  ; CHECK THE BIT TO SEND AND DRIVE THE OUTPUT TO THE
499  ; DESIRED VALUE
500  ;
501  ;
502  ; SENDOUT STATE
503  ; 0 IDLE
504  ; 1 START BIT
505  ; 2-8 DATA
506  ; 9-10 STOP BIT
507  ; 11 NEXT WORD
508  ; 12 NEXT WORD (FOR MODE SYSTEMS)
509  ;
510  ; TL/DD/10799--96
; ****************************************
; 031 0193 93       DATA:  LD  B, $TSTATUS
; 032 0193 72       SBIT  TM22H, [B]  ; $TSTATUS CHECK IF BAUD RATE TIMER ENDS
; 033 0193 01       JP  SDATA1
; 034 0193 8E       RET
; 035 0193 A4       ;
; 036 0193 40       IDLE:  RET  ; EXIT IF IDLE
; 037 0193 77       SBIT  UD20F, [B]  ; $MTYPE
; 038 0193 14       JP  STOP
; 039 0193 4D       DATAR:
; 040 0193 90       LD  A, $WORPD
; 041 0193 0F       MFE  $X  A, [H]  ; $B POINTS TO THE WORD
; 042 0193 A1       ;
; 043 0193 A5       RC
; 044 0193 0E       LD  A, [B]
; 045 0193 B0       RNC  A  ; EXIT LEAST SIG BIT
; 046 0193 A6       X  A, [B]
; 047 0193 D6       LD  B, $POINTC
; 048 0193 88       INC
; 049 0193 7A       SBIT  $MT, [B]
; 050 0193 99       INC
; 051 0193 6A       RBIT  $MT, [B]
; 052 0193 96       NEXT:  LD  A, $SENDAT
; 053 0193 04       INC  A
; 054 0193 0B       X  A, $SENDAT
; 055 0193 8E       RET  ; EXIT
; 056 0193 77       SBIT  UD20F, [B]  ; $MTYPE
; 057 0193 04       JP  WORPD
; 058 0193 41       ;
; 059 0193 04       MFE  $X  A, $WORPD
; 060 0193 03       SBIT  $NUMW20H  ; NUMBER OF WORDS TO SEND
; 061 0193 09       JP  ENDRT  ; END OF REPORT
; 062 0193 C0       X  A, $WORPD
; 063 0193 00       LD  $SENDST, [B]  ; SEND START BIT
; 064 0193 82       ;
544 END
545 ;
546 END
547 ;
548 END
549 ;
550 ;
551 ;
552 ;
553 ;
554 ;
555 ;
556 ;
557 ;
558 ;
559 ;
560 ;
561 ;
562 ;
563 ;
564 ;
565 ;
618  ;
619  ;----------------------------------------------------------------------
620  ;        SAMPLE BUTTON INPUT FOR MICROSOFT
621  ;----------------------------------------------------------------------
622  ;        INDICATE BUTTON STATUS
623  ;----------------------------------------------------------------------
624  ;
625  627 0200 0000  LD  [B],#0  ;(TEMP), (A-PORT8, CARRY ROTATED)
626  626 0202 00  IFNC  ;(MICROSOFT) 0-KEY DEPRESSED
627  627 0203 7D  SBIT  5,[B]  ;(TEMP)
628  628 0204 80  RNC  A
629  629 0205 80  RNC  A
630  630 0206 90  IFNC
631  631 0207 7C  SBIT  4,[B]  ;(TEMP)
632  632 0208 80  RNC  A
633  633 0209 90  IFNC
634  634 020A 80  RNC  A
635  635 020B 80  RNC  A
636  636 020C 0047  SBIT  RPT,CHANGE ;INDICATE TO SEND DATA
637  637 020F 80  RET
638  638  ;
639  639  ;----------------------------------------------------------------------
640  ;        SAMPLE BUTTON INPUT FOR MOUSE SYSTEMS
641  ;----------------------------------------------------------------------
642  ;        INDICATE BUTTON STATUS
643  ;----------------------------------------------------------------------
644  ;
645  646 0210 0007  LD  [B],#07  ;(TEMP)
646  647 0212 80  IFWC  ;(MICROSOFT) 0-KEY DEPRESSED
648  648 0213 80  RBIT  2,[B]  ;(TEMP)
649  649 0214 80  RNC  A
650  650 0215 80  IFNC
651  651 0216 80  RBIT  1,[B]  ;(TEMP)
652  652 0217 80  RNC  A
653  653 0218 80  IFNC
654  654 0219 80  RBIT  0,[B]  ;(TEMP)

TL/CD/10790–19
664    ;
665 ORGA AA    LD A, (B+)    ; (ROMF)
666 ORG 02H    INQ A, (B)    ; (INSTATE)
667 ORG 02C HE RET   ; NO CHANGE
668    ;
669 ORG 02D A6    X A, (B)    ; (INSTATE)
670 ORG 00H 00F71 SBIT RPT, CHANGE ; INDICATE TO SEND DATA
671 ORG 021 HE RET
672    ;
673 ;******************************************************************************
674    ;
675 EXD0    .EQXO
676 ORG 00H 00F    .BYTE " (C) 1990 NATIONAL SEMICONDUCTOR AMONGE VER 1.0"
<table>
<thead>
<tr>
<th>Address</th>
<th>Symbol</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>BAPE</td>
<td>SAVE</td>
</tr>
<tr>
<td>0001</td>
<td>BSAVE</td>
<td>TEMP</td>
</tr>
<tr>
<td>0002</td>
<td>RTI</td>
<td>RTST</td>
</tr>
<tr>
<td>0003</td>
<td>CKS</td>
<td>CKST</td>
</tr>
<tr>
<td>0004</td>
<td>COM</td>
<td>CSST</td>
</tr>
<tr>
<td>0005</td>
<td>HEC</td>
<td>RDXST</td>
</tr>
<tr>
<td>0006</td>
<td>XMDR</td>
<td>RMDR</td>
</tr>
<tr>
<td>0007</td>
<td>XGDR</td>
<td>XGDR</td>
</tr>
<tr>
<td>0008</td>
<td>XGDR1</td>
<td>XGDR1</td>
</tr>
<tr>
<td>0009</td>
<td>XGDR2</td>
<td>XGDR2</td>
</tr>
</tbody>
</table>

**NOTES**

- No Warning Lines
- No Error Lines
- Source Checksum = 078A
- Object Checksum = 0A9F

**INPUT FILE:** DIMOUSE.WC
**ASSEMBLING FILE:** DIMOUSE.PMN
**OBJECT FILE:** DIMOUSE.LA

TL/DD/10799–22

TL/DD/10799–23

25
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 does not assume any responsibility for use of any circuitry described, no circuit patent licenses are implied and National reserves the right at any time without notice to change said circuitry and specifications.
IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI’s terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are neither designed nor intended for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

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