

OPTICAL MOUSE

**USER
MANUAL**

CONTENTS

CHAPTER 1	GENERAL INFORMATION.....	1
CHAPTER 2	FEATURES.....	2
CHAPTER 3	OPTICAL MOUSE DRIVER.....	4
CHAPTER 4	PROGRAMMING INTERFACE.....	7

CHAPTER 1

GENERAL INFORMATION

The Optical Mouse is an advanced optical mouse with an ultra high 1020 DPI resolution. When compared with ordinary mice whose resolutions range from 120 to 1020 the Optical Mouse clearly surpasses them in resolution and nearly cuts in half the amount of desk space needed to cover an equivalent amount of area on the CRT screen. The result is a mouse that saves you a great deal of effort, desk space and time.

Since your new mouse is optical, there is no mechanical tracking ball which typically wears out with time. There is no mechanical maintenance or extra power required for a special interface. These features increase the mouse's utility while decreasing its cost.

The Optical Mouse is fully compatible with the entire family of IBM PC/XT/AT computers along with all the compatibles which use the standard RS-232C output port. The Optical Mouse supports dual software compatibility by being compatible with both the Microsoft mouse and Mouse System mouse drivers. The Optical Mouse can be easily switched between these two modes by means of installing the appropriate drivers and selecting the correct position of the mode switch on the mouse. In addition to the standard three buttons, the Optical Mouse sports a special Turbo/Slow slide switch which allow you to fine tune your mouse's performance or reassign the cursor for even greater control. All of these features, combined with the special mouse pad, enable you to achieve exceptionally smooth mouse operation and years of good service from your Optical Mouse.

Naturally the Optical mouse supports all the popular applications packages on today's market such as AutoCAD, PC Paintbrush and PC Paint.

CHAPTER 2

FEATURES

2.1 SYSTEM REQUIREMENTS

In order to operate your Optical Mouse you will need the following:

1. An IBM PC, PC/XT, PC/AT or compatible computer.
2. IBM PC DOS (Disk Operating system) or Microsoft DOS Version 2.00 or later.
3. One RS-232C serial port.
4. A minimum of 128 kilobytes of RAM (Random Access Memory).

2.2 TECHNICAL SPECIFICATIONS

1. Optical mouse
2. Standard RS-232C interface
3. Compatible with Microsoft mouse and Mouse System mouse
4. Three buttons plus a Turbo/Slow slide switch to control speed
5. A MS/PC slide switch to change from Microsoft mouse mode or Mouse System mouse mode
6. Size: 4.5"L x 2.7"W x 1.6"H (115x68x40mm)
7. Cable: 6 feet
8. Resolution: 120-1020 DPI (dot per inch)
9. Weight: 2.00 ounces (85g)

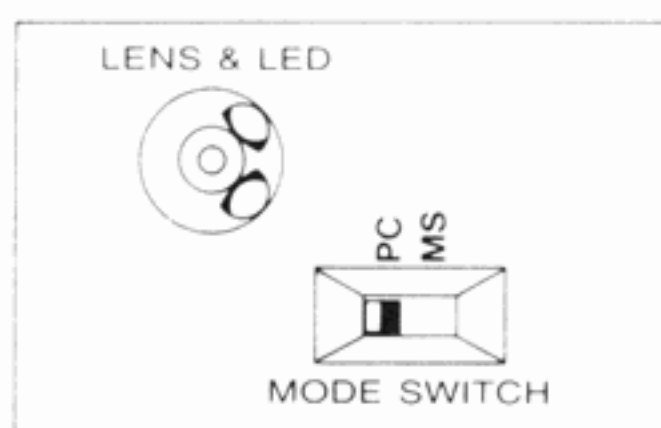
2.3 PACKAGE CONTENTS

1. Optical Mouse with cable and DB9 connector
2. User's Manual
3. Dr. HALO Paint & Graph User's Manual
4. Optical Mouse pad
5. DB9 to DB25 adapter
6. Optical Mouse software package

2.4 HARDWARE INSTALLATION

The following steps tell you how to install your Optical Mouse in your computer.

1. Turn off your computer and unplug all peripheral equipment.
2. Locate the RS-232C serial port, either COM1 or COM2, on the back of your computer.
3. Plug the Optical Mouse connector into the COM port.
4. Your Optical Mouse has a DB9 connector for IBM AT compatibles, and a DB9 to DB25 adapter for connection to IBM PC/XT compatibles.
5. You can select the Mouse system Mouse mode or the Microsoft Mouse mode, depending on your application package, by selecting the proper switch position located on the underside of the Mouse.



PC. PC Mouse Mode
MS. Microsoft Mouse Mode

6. Place the Optical Mouse on its special pad.

You are now ready to turn on your computer, install your mouse driver software, and put your Optical Mouse to work.

Note: To change the mouse mode, you need to first change the switch position, then reboot your computer.

2.5 CHANGING THE CURSOR MOVEMENT SPEED

You can change the Optical Mouse resolution at any time by changing the speed switch position between turbo, normal, and slow modes. The switch is located on the left side of the mouse.

In turbo mode, the Optical Mouse resolution is Dynamic. The faster the mouse is moved, the higher resolution the mouse is.

CHAPTER 3

OPTICAL MOUSE DRIVER

The mouse driver must be installed in your computer before you can use the Optical Mouse.

There are a number of ways that a disk can be damaged. For your protection, we suggest that you use the following steps to make a back up copy of the Mouse Driver Disk.

3.1 BACK UP YOUR MOUSE DRIVER DISK With A Two-Floppy Disk Drive System

1. Insert your DOS disk in drive A and turn your computer on.
2. Insert a blank disk in drive B.
3. At the A>, type

```
FORMAT B: /S <enter>
```

This will create a formatted disk with the operating system so that it can be booted.

4. Remove your DOS disk from drive A, and insert Mouse Driver Disk, at the A>, type

```
COPY*,*B:<enter>
```

5. To create an autoexec.bat file, at the A>, type

```
COPY CON B:AUTOEXEC.BAT <enter>
```

```
LMOUSE/xn <enter>
```

```
<F6> <enter>
```

This instructs the computer to install the mouse driver automatically every time you turn your computer on.

Note:

/	Start of the parameters
x	Mouse type (M: Microsoft mode, P: Mouse system Mode, please refer to chapter 2 for proper selection)
n	Communication port (1: COM1,2: COM2)

With A Hard Disk System

1. Start your computer on the hard disk drive.
2. At the C>, type

```
MD/LMOUSE <enter>
```

This will create a subdirectory named LMOUSE.
3. At the C>, type

```
CD/LMOUSE <enter>
```
4. Insert the Mouse Driver disk in drive a, at the C>, type

```
COPY A:*. * <enter>
```
5. To install the mouse driver every time you boot your computer, you should use a word processor to place the following lines at the beginning of the AUTOEXEC.BAT.

```
CD\LMOUSE  
LMOUSE/xn  
CD\
```

Note:

/	Start of the parameters
x	Mouse type (M: Microsoft mode, P: Mouse System Mode, please refer to Chapter 2 for proper selection)
n	Communication port (1: COM1,2: COM2)

6. If you do not wish to install the mouse driver automatically every time you boot your computer, skip Step 5 and put the subdirectory \LMOUSE in the PATH command of your AUTOEXEC.BAT so you can install the mouse driver from any directory at any time.

3.2 MOUSE DRIVER PARAMETERS

In your software package, you will find the Optical Mouse driver LMOUSE.COM which is used for both the Mouse System Mouse and the Microsoft Mouse modes. Type the following command at the DOS prompt to install the mouse driver.

```
LMOUSE /xn
```

LMOUSE-	Invoke the driver LMOUSE.COM
/-	Start of parameter
x-	Mouse type (M: Microsoft, P: Mouse System)
n-	Communication port (1: COM1,2: COM2)

Example- Insert the diskette into driver A and type:

```
LMOUSE /M1
```

This installs the Microsoft Mouse driver on COM1.

LMOUSE /M2

This installs the Microsoft Mouse driver on COM2.

LMOUSE /P1

This installs the Mouse System Mouse Driver on COM1.

LMOUSE /P2

This installs the Mouse System Mouse driver on COM2.

3.3 CREATING A CONFIG. SYS FILE

Creata CONFIG.SYS file using an editor or DOS copy command. There are three step:

1. Copy CON := CONFIG. SYS and Enter
2. DEVICE = LMOUSE.SYS /xn
3. Press the F6 and then Enter

Note: The LMOUSE.SYS and CONFIG.SYS files must be at the same root directory of the disk which is use to start DOS. The xn parameters were add to choose the LMOUSE driver's mode and communication port which were described on the above.

CHAPTER 4

PROGRAMMING INTERFACE

This section will define the Mouse System Mouse driver programming language interface. This will be of greatest interest for assembly language programmers and users interested in developing their own mouse applications.

4.1 Assembly Language Interface

To make mouse system calls you must:

1. Load AX,BX,CX and DX registers with the parameter values.
2. Execute software interrupt 51(33H). Values returned by the mouse functions will be placed in the register.

Example: The following instructions are used to set the cursor to position 100 (horizontal) and 100 (vertical).

```
;Set cursor to location(100,100)
MOV  AX,4           ;Function Call 4
MOV  CX,100        ;Set Horizontal To 100
MOV  DX,100        ;Set Vertical To 100
INT  51            ;(33H) Interrupt To Mouse
```

4.2 BASIC Interpreter Interface

In order to make mouse system calls inside the BASIC interpreter follow these procedures:

1. Insert an initialization sequence such as:
10 DEF SEG=0
11 MSEG=256xPEEK(51x4+3)+PEEK(51x4+2)
12 MOUSE=256xPEEK(51x4+1)+PEEK(51x4)+2
13 DEF SEG=MSEG
2. Make sure that these statements appear before any calls to mouse functions. The CALL statement is then used as follows:
CALL MOUSE (M1%,M2%,M3%,M4%)

MOUSE is the variable containing the entry offset of the mouse software; and M1%,M2%,M3%, and M4% are the names of the integer variables chosen for the parameters in this call. These variables correspond to the values for AX,BX,CX and DX in the section above.

Example:

```

100      'Set minimum and maximum vertical position
110      M1%=8 'function 8
120      M3%=0 'minimum coordinate
130      M4%=200 'maximum coordinate

140      CALL MOUSE (M1%,M2%,M3%,M4%)

```

4.3 List of Mouse Driver Functions

The following functions are available for the Optical Mouse. These functions will be described in greater detail in subsequent sections.

FUNCTIONS	FUNCTION NUMBER
Mouse Initalization	0
Show Cursor	1
Hide Cursor	2
Get Mouse Position & Button Status	3
Set Mouse Cursor Position	4
Get Button Press Information	5
Get Button Release Information	6
Set Min/Max Horizontal Position	7
Set Min/Max Vertical Position	8
Define Graphics Cursor Block	9
Define Text Cursor	10
Read Mouse Motion Counters	11
Define Event Handler	12
Light Pen Emulation Mode On	13
Light Pen Emulation Mode Off	14
Set Mouse Mickey/Pixel Ratio	15
Conditional Hide Cursor	16
Set Double-Speed Threshold	19

4.4 Description of The Mouse Driver Functions

This section gives detailed explanations for each of the mouse functions listed in the previous section. Each function description specifies the input parameters needed, the expected output values, and any special measures to be taken.

Function 0: Mouse Initialization

This function reports the status of the mouse hardware and software. The mouse status is 0 if the mouse hardware and software are not installed and -1 if the hardware and software are installed.

Input AX = 0

Output AX = mouse status (0:not installed, -1: installed)
 BX = number of buttons (3:Mouse System Mouse,2: Microsoft Mouse)

Basic:

Input: M1%=0

Output: M1%=mouse status(0: not installed, -1: installed)
 M2%=number of buttons(3: Mouse System Mouse, 2: Microsoft Mouse)

Example: Use the at the beginning of a program to determine if the Optical MOUSE driver is loaded and to initialize the Optical Mouse.

```
100        ;Check if mouse is present
10        M1%=0
120        CALL MOUSE (M1%,M2%,M3%,M4%)
130        If not M1% Then PRINT "Mouse not installed":END
```

Function 1: Show Cursor

This function increments the internal cursor level counter. If the counter is zero the cursor is displayed on the screen.

Input AX = 1
Output None

Function 2: Hide Cursor

This function decrements the internal cursor level counter. If the cursor was on the screen it will now be hidden. Even though the cursor is hidden it still tracks the motion of the mouse, changing as the mouse changes position.

Input	AX = 2
Output	None

Note: The internal cursor counter value is always 0 or less. When the counter is 0 you can see the cursor on the screen. In this mode the cursor tracks the motion of the mouse changing position as the mouse changes position. As the counter value is changing in value as the mouse changes position you can call function 1 and function 2 any number of times. You must, however, always call function 1 at least once after each function 2 call in order to restore the previous value of cursor counter.

Function 3: Get Mouse Position & Button Status

This function reports the status of mouse buttons and the vertical & horizontal cursor coordinates. Button status is a single integer value as follows:

- Bit 0 = left button (Msmouse, Microsoft)
- Bit 1 = right button (Msmouse, Microsoft)
- Bit 2 = middle button (Msmouse only)

These bits are 1 when the corresponding button is depressed and 0 when it is released.

Input	AX = 3
Output	BX = button status CX = horizontal cursor position DX = vertical cursor position

Function 4: Set Mouse Cursor Position

This function assigns the cursor to the specified position on the screen. The new values must be within the specified ranges of the virtual screen. If you are not using a high resolution monitor the values are rounded to the nearest values permitted by the screen for horizontal and vertical positions.

Function 7: Set Minimum & Maximum Horizontal Cursor Positions

Minimum and maximum horizontal cursor positions on the screen are set by this function. All cursor movement is restricted to this area once it has been defined. The virtual screen determines the maximum and minimum values for the cursor. If the cursor is outside the defined area when the call is made it will move just inside the predefined area. If the minimum value is greater than the maximum value, the two values will be exchange.

Input	AX = 7 CX = minimum horizontal cursor position DX = maximum vertical cursor position
Output	None

Function 8: Set Minimum & Maximum Vertical cursor position

This function determines minimum and maximum vertical cursor positions on the screen. All cursor movement is restricted to this area once it has been defined. The virtual screen determines the maximum and minimum values for the cursor. If the cursor is outside the defined area when the call is made it will moves just inside the predefined area. If the minimum value is greater than the maximum the two values will be exchanged.

Input	AX = 8 CX = minimum vertical cursor position DX = maximum vertical cursor position
Output	None

Function 9: Define Graphics Cursor Block

The shape and color of the cursor when it is in the graphics mode are determined by this function. Your program must call function 1 in order to display the graphics cursor. The graphics cursor is defined by two 16bit arrays which are the screen mask array and the cursor mask array. The cursor hot spot values must be within the range of -16 to 16. These values define one pixel within and one pixel outside the cursor block. The upper left corner is (0,0).

Input	AX = 9 BX = horizontal cursor hot spot CX = vertical cursor hot spot DX = pointer to screen & cursor mask
Output	None

Function 10: Define Text Cursor

Use this function to select the software or hardware text cursor. Parameter BX selects the cursor type. When this value is 0 the software is selected. When the value is 1 the hardware text cursor is selected.

With the software text cursor selected the values of parameters CX and DX define the screen and cursor masks. With the hardware cursor selected the parameters CX and DX must contain the line numbers of the first and last scan line of the cursor to be shown on the screen.

Input	AX = 10 BX = select cursor (0 for software text: 1 for hardware text) CX = screen mask value/scan line start DX = cursor mask value/scan line stop
-------	---

Function 11: Read Mouse Motion Counters

Use this function to return the horizontal and vertical motion counts since the last call to this function. The values returned are relative cursor movements with positive horizontal and vertical counts indicated to the right and down, and negative horizontal/vertical counts indicated to the left and up. The movement count is the distance that the mouse moves in 1/250 inch increments. The movement count is always within the range -32769 to 32767 and set to 0 after the call is completed.

Input	AX = 11
Output	CX = horizontal count DX = vertical count

Function 12: Define event handler

Use this routine to set the call mask and subroutine address for the mouse software interrupts. Software interrupts pause execution of the program and call the

specified subroutine whenever one or more of the conditions defined by the call mask occurs. The call mask is a single integer value which defines the conditions necessary to cause an interrupt. Each bit in the call mask corresponds to specific condition:

Mask Bit	Condition
0	change cursor position
1	press left button
2	release left button
3	press right button
4	release right button
5-15	not used
Input	AX = 12 CX = call mask ES:DX = pointer to user subroutine
Output	None

Function 13: Light Pen Emulation Mode On

This function enables the mouse to emulate a light pen. When the mouse emulates a light pen, calls to the pen function will return the cursor position from the last pen down. "Pen down" is controlled by the mouse buttons. The pen is down or on the screen when both buttons are down. The pen is off the screen when both buttons are up.

Input	AX = 13
Output	None

Function 14: Light Pen Emulation Mode Off

To disable the light pen emulation mode use this function. When light pen emulation is disabled, calls to the pen function are for information about the pen only.

Input	AX = 14
Output	None

Function 15: Set Mouse Mickey/Pixel Ratio

This function sets the mouse motion to screen pixel ratio. The horizontal and vertical ratios specify the motion count of the mouse at 8 pixels with the values ranging from 1 to 32767. The default values are 8 motion counts for each 8 horizontal pixels and 16 motion counts for each 8 vertical pixels. This is the equivalent of 3.2 inches of horizontal mouse movement and 2.0 inches of vertical mouse movement.

Input	AX = 15 CX = horizontal step to pixel ratio DX = vertical step to pixel ratio
Output	None

Function 16: Conditional Hide Cursor

This function allows the user to define a region on the screen to be updated. If the mouse cursor is in the specified region it will be hidden while the region is being updated. After this function is called (function 16), function 1 must be called in order to display the cursor again.

Input	AX = 16 ES : DX = pointer to the region array
Output	None

Function 19: Set Double-Speed Threshold

Use this function to set the threshold speed which will trigger the doubling of the cursor's speed of motion on the screen. This function makes it easier to point at images which are far apart on the screen. The fourth parameter M4% defines the mouse's threshold speed. If there is no value given or if the mouse is reset by a call to function 0, a default value of 64 mickeys per second is assumed. If the mouse moves faster than the value of M4% the cursor motion speed will double. The threshold speed will remain fixed until function 19 is called again or until the mouse is reset by a function 0 call. Once the program turns on this speeddoubling feature it is always on. However, the program can effectively turn off this feature by setting M4% to a speed as high as 10,000, which is faster than the mouse can physically move, and then calling function 19.

Input	AX = 19 DX = threshold speed in mickeys/second
Output	None

OPTICAL MOUSE

1989.08-002
2241040030000

