

The PS/2 Mouse Interface

Source: <http://www.Computer-Engineering.org>

Author: Adam Chapweske

Last Updated: 04/01/03

Legal Information:

All information within this article is provided "as is" and without any express or implied warranties, including, without limitation, the implied warranties of merchantability and fitness for a particular purpose.

This article is protected under copyright law. This document may be copied only if the source, author, date, and legal information is included.

Abstract:

This article attempts to explain every aspect of the PS/2 mouse interface including the physical and electrical interface, low-level protocol, modes of operation, commands, and extensions. All code samples involving the mouse encoder are written in assembly for [Microchip's](#) PIC microcontrollers. All code samples related to the auxiliary device controller (keyboard controller) are written in x86 assembly

General Description:

There are many types of pointing devices available for modern PCs, including mice, trackballs, touchpads, electronic whiteboards, etc. Virtually all of these pointing devices communicate with a computer using one of two interfaces: Universal Serial Bus (USB) or the PS/2 Mouse Interface. See the following table for a comparison:

| | USB (v1.1) | PS/2 Mouse Interface |
|-----------------------------|---|--------------------------------------|
| Number of Devices Supported | Up to 127 | One |
| Maximum Data Rate | 12 Mbps | 40 kbps |
| Power (max) | 5V @ 500mA | 5V @ 100mA |
| Hot-Pluggable? | Yes | No |
| Documentation | Well-documented: http://www.usb.org | Out-of-print (IBM Tech Reference) |

Older pointing device interfaces include the Apple Desktop Bus (ADB), RS-232 serial port, and the bus mouse interface. These are obsolete and are not covered in this article.

The PS/2 mouse interface originally appeared in IBM's "Personal System/2" computers in the late 80's. It still remains a widely-supported interface for the sake of constantly maintaining backward compatibility. However, USB has quickly caught on these last few years and will eventually replace the PS/2 mouse interface entirely.

The PS/2 mouse interface uses a bidirectional serial protocol to transmit movement and button-position data to the computer's auxiliary device controller (keyboard controller). The computer, in turn, may send a number of commands to the mouse to set the report rate, resolution, reset the mouse, disable the mouse, etc. The computer also provides the mouse with an overload-protected 5V power supply.

Electrical Interface / Protocol:

The PS/2 mouse uses the same protocol as the PS/2 keyboard (aka AT keyboard). Click [here](#) for detailed information on this protocol.

Inputs, Resolution, and Scaling:

The standard PS/2 mouse interface supports the following inputs: X (right/left) movement, Y (up/down) movement, left button, middle button, and right button. The mouse reads these inputs at a regular frequency and updates various counters and flags to reflect movement and button states. There are many PS/2 pointing devices that have additional inputs and may report data differently than described in this document. One popular extension I cover later in this document is the Microsoft Intellimouse, which includes support for the standard inputs as well as a scrolling wheel and two additional buttons.

The standard mouse has two counters that keep track of movement: the X-movement counter and the Y-movement counter. These are 9-bit 2's complement values and each has an associated overflow flag. Their contents, along with the state of the three mouse buttons, are sent to the host in the form of a 3-byte movement data packet (as described in the next section.) The movement counters represent the amount of movement that has occurred since the last movement data packet was sent to the host (ie, they do not represent absolute positions.)

When the mouse reads its inputs, it records the current state of its buttons and checks for movement. If movement has occurred it increments (for +X or +Y movement) or decrements (for -X or -Y movement) its X and/or Y movement counters. If either of the counters has overflowed, it sets the appropriate overflow flag.

The parameter that determines the amount by which the movement counters are incremented/decremented is the *resolution*. The default resolution is 4 counts/mm and the host may change that value using the "Set Resolution" (0xE8) command.

There is a parameter that does not effect the movement counters, but does effect the reported [\(1\)](#) value of these counters. This parameter is *scaling*. By default, the mouse uses 1:1 scaling, which has no effect on the reported mouse movement. However, the host may select 2:1 scaling by sending the "Set Scaling 2:1" (0xE7) command. If 2:1 scaling is enabled, the mouse will apply the following algorithm to the counters before sending their contents to the host:

| Movement Counter | Reported Movement |
|------------------|-------------------|
| 0 | 0 |
| 1 | 1 |
| 2 | 1 |
| 3 | 3 |
| 4 | 6 |
| 5 | 9 |
| N > 5 | 2 * N |

Movement Data Packet:

The standard PS/2 mouse sends movement/button information to the host using the following 3-byte packet [\(4\)](#):

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------|------------|------------|------------|------------|----------|------------|-----------|----------|
| Byte 1 | Y overflow | X overflow | Y sign bit | X sign bit | Always 1 | Middle Btn | Right Btn | Left Btn |
| Byte 2 | X Movement | | | | | | | |
| Byte 3 | Y Movement | | | | | | | |

The movement counters are 9-bit 2's complement integers, where the most significant bit appears as a sign bit in Byte 1 of the movement data packet. These counters are updated when the mouse reads its input and finds movement has occurred. Their value is the amount of movement that has occurred since the last movement data packet was sent to the host (ie, after a packet is sent to the host, the movement counters are reset.) The range of values that can be expressed by the movement counters is -255 to +255. If this range is exceeded, the appropriate overflow bit is set.

As I mentioned earlier, the movement counters are reset whenever a movement data packet is successfully sent to the host. They are also reset after the mouse receives any command from the host other than the "Resend" (0xFE) command.

Modes of Operation:

Data reporting is handled according to the mode in which the mouse is operating. There are four standard modes of operation:

- *Reset* - The mouse enters Reset mode at power-up or after receiving the "Reset" (0xFF) command.
- *Stream* - This is the default mode (after Reset finishes executing) and is the mode in which most software uses the mouse. If the host has previously set the mouse to Remote mode, it may re-enter Stream mode by sending the "Set Stream Mode" (0xEA) command to the mouse.
- *Remote* - Remote mode is useful in some situations and may be entered by sending the "Set Remote Mode" (0xF0) command to the mouse.
- *Wrap* - This mode isn't particularly useful except for testing the connection between the mouse and its host. Wrap mode may be entered by sending the "Set Wrap Mode" (0xEE) command to the mouse. To exit Wrap mode, the host must issue the "Reset" (0xFF) command or "Reset Wrap Mode" (0xEC) command. If the "Reset" (0xFF) command is received, the mouse will enter Reset mode. If the "Reset Wrap Mode" (0xEC) command is received, the mouse will enter the mode it was in prior to Wrap Mode.

(Note: The mouse may also enter "extended" modes of operation, as described later in this document. However, this is not a feature of

the standard PS/2 mouse.)

Reset Mode:

The mouse enters reset mode at power-on or in response to the "Reset" (0xFF) command. After entering this mode, the mouse performs a diagnostic self-test referred to as BAT (Basic Assurance Test) and sets the following default values:

- Sample Rate = 100 samples/sec
- Resolution = 4 counts/mm
- Scaling = 1:1
- Data Reporting Disabled

It then sends a BAT completion code of either AAh (BAT successful) or FCh (Error). The host's response to a completion code other than AAh is undefined.

Following the BAT completion code (AAh or FCh), the mouse sends its device ID of 00h. This distinguishes it from a keyboard, or a mouse in an extended mode. I have read documents saying the host is not *supposed* to transmit any data until it receives a device ID. However I've found that some BIOS's will send the "Reset" command immediately following the 0xAA received after a power-on reset.

After the mouse has sent its device ID to the host, it will enter Stream Mode. Note that one of the default values set by the mouse is "Data Reporting Disabled". This means the mouse will not issue any movement data packets until it receives the "Enable Data Reporting" command.

Stream Mode:

In stream mode, the mouse sends movement data when it detects movement or a change in state of one or more mouse buttons. The maximum rate at which this data reporting may occur is known as the *sample rate*. This parameter ranges from 10 samples/sec to 200 samples/sec. Its default value is 100 samples/sec and the host may change that value by using the "Set Sample Rate" command. Stream mode is the default mode of operation following reset.

Remote Mode:

In this mode the mouse reads its inputs and updates its counters/flags at the current sample rate, but it does not automatically issue data packets when movement has occurred. Instead, the host must poll the mouse using the "Read Data" command. Upon receiving this command the mouse will send a single movement data packet and reset its movement counters.

Wrap Mode:

This is an "echoing" mode in which every byte received by the mouse is sent back to the host. Even if the byte represents a valid command, the mouse will not respond to that command--it will only echo that byte back to the host. There are two exceptions to this: the "Reset" command and "Reset Wrap Mode" command. The mouse treats these as valid commands and does not echo them back to the host.

Intellimouse Extensions:

A popular extension to the standard PS/2 mouse is the Microsoft Intellimouse. This includes support for a total of five mouse buttons and three axes of movement (right-left, up-down, and a scrolling wheel). These additional features require the use of a 4-byte movement data packet rather than the standard 3-byte packet. Since standard PS/2 mouse drivers cannot recognize this packet format, the Microsoft Intellimouse is required to operate exactly like a standard PS/2 mouse unless it knows the drivers support the extended packet format. This way, if a Microsoft Intellimouse is used on a computer which only supports the standard PS/2 mouse, the Microsoft Intellimouse will still function, except for its scrolling wheel and 4th and 5th buttons.

After power-on or reset the Microsoft Intellimouse operates just like a standard PS/2 mouse (ie, it uses a 3-byte movement data packet, responds to all commands in the same way as a standard PS/2 mouse, and reports a device ID of 00h.) To enter "scrolling wheel" mode, the host sends the following command sequence:

```
Set sample rate 200
Set sample rate 100
Set sample rate 80
```

The host then issues the "Get device ID" command and waits for a response. If a standard PS/2 mouse (ie, non-Intellimouse) is attached, it will respond with a device ID of 00h. In this case, the host will recognize the fact that the mouse does have a scrolling wheel and will continue to treat it as a standard PS/2 mouse. However, if a Microsoft Intellimouse is attached, it will respond with an ID of 03h. This tells the host that the attached pointing device has a scrolling wheel and the host will then expect the mouse to use the following 4-byte movement data packet:

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------|------------|------------|------------|------------|----------|------------|-----------|----------|
| Byte 1 | Y overflow | X overflow | Y sign bit | X sign bit | Always 1 | Middle Btn | Right Btn | Left Btn |
| Byte 2 | X Movement | | | | | | | |
| Byte 3 | Y Movement | | | | | | | |
| Byte 4 | Z Movement | | | | | | | |

Z Movement is a 2's complement number that represents the scrolling wheel's movement since the last data report. Valid values are in the range of -8 to +7. This means the number is actually represented only by the least significant four bits; the upper four bits act only as sign extension bits.

To enter "scrolling wheel + 5 button" mode, the host sends the following command sequence:

Set sample rate 200
Set sample rate 200
Set sample rate 80

The host then issues the "Get device ID" command and waits for a response. A Microsoft Intellimouse will respond with a device ID of 04h, then use the following 4-byte movement data packet:

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------|------------|------------|------------|------------|----------|------------|-----------|----------|
| Byte 1 | Y overflow | X overflow | Y sign bit | X sign bit | Always 1 | Middle Btn | Right Btn | Left Btn |
| Byte 2 | X Movement | | | | | | | |
| Byte 3 | Y Movement | | | | | | | |
| Byte 4 | Always 0 | Always 0 | 5th Btn | 4th Btn | Z3 | Z2 | Z1 | Z0 |

Z0-Z3 is a 2's complement number which represents the amount of movement that has occurred since the last data report. Valid values range from -8 to +7.

4th Btn: 1 = 4th mouse button is pressed; 0 = 4th mouse button is not pressed.

5th Btn: 1 = 5th mouse button is pressed; 0 = 5th mouse button is not pressed.

You may have seen mice with two scrolling wheels--one vertical and the other horizontal. These mice use the Microsoft Intellimouse data packet format as described above. If the vertical wheel is scrolled upward, the Z-counter is incremented by one and if that wheel is scrolled down, the Z-counter is decremented by one. This is normal operation for a scrolling wheel. However, if the *horizontal* wheel is scrolled right, the Z-counter is incremented by *two* and if it is scrolled left, the Z-counter is decremented by *two*. This seems like an odd way to implement the second scrolling wheel, but it works since the placement of the two wheels make it impossible to use both of them at the same time (and if you try to trick the software and use both at the same time, it will ignore the horizontal wheel.)

Command Set:

The following is the set of command accepted by the standard PS/2 mouse. If the mouse is in Stream mode, the host should disable data reporting (command F5h) before sending any other commands.

- FFh (Reset) - The mouse responds to this command with "acknowledge" (FAh) then enters Reset Mode.
- FEh (Resend) - The host sends this command whenever it receives invalid data from the mouse. The mouse responds by resending the last(2) packet(3) it sent to the host. If the mouse responds to the "Resend" command with another invalid packet, the host may either issue another "Resend" command, issue an "Error" command, cycle the mouse's power supply to reset the mouse, or it may inhibit communication (by bringing the Clock line low). The action taken depends on the host.
- F6h (Set Defaults) - The mouse responds with "acknowledge" (FAh) then loads the following values: Sampling rate = 100, Resolution = 4 counts/mm, Scaling = 1:1, Disable Data Reporting. The mouse then resets its movement counters and enters stream mode.
- F5h (Disable Data Reporting) - The mouse responds with "acknowledge" (FAh) then disables data reporting and resets its movement counters. This only effects data reporting in Stream mode and does not disable sampling. Disabled stream mode functions the same as remote mode.
- F4h (Enable Data Reporting) - The mouse responds with "acknowledge" (FAh) then enables data reporting and resets its movement counters. This command may be issued while the mouse is in Remote Mode (or Stream mode), but it will only effect data reporting in Stream mode.
- F3h (Set Sample Rate) - The mouse responds with "acknowledge" (FAh) then reads one more byte from the host. The mouse saves this byte as the new sample rate. After receiving the sample rate, the mouse again responds with "acknowledge" (0xFA) and resets its movement counters. Valid sample rates are 10, 20, 40, 60, 80, 100, and 200 samples/sec.
- F2h (Get Device ID) - The mouse responds with "acknowledge" (FAh) followed by its device ID (00h for the standard PS/2 mouse.) The mouse should also reset its movement counters.
- F0h (Set Remote Mode) - The mouse responds with "acknowledge" (FAh) then resets its movement counters and enters remote mode.
- EEh (Set Wrap Mode) - The mouse responds with "acknowledge" (FAh) then resets its movement counters and enters wrap mode.
- ECh (Reset Wrap Mode) - The mouse responds with "acknowledge" (FAh) then resets its movement counters and enters the mode

it was in prior to wrap mode (Stream Mode or Remote Mode.)

- EBh (Read Data) - The mouse responds with acknowledge (FAh) then sends a movement data packet. This is the only way to read data in Remote Mode. After the data packets has been successfully sent, it resets its movement counters.
- EAh (Set Stream Mode) - The mouse responds with "acknowledge" then resets its movement counters and enters stream mode.
- E9h (Status Request) - The mouse responds with "acknowledge" then sends the following 3-byte status packet (then resets its movement counters.):
 -

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------|-------------|-------|--------|---------|----------|----------|------------|-----------|
| Byte 1 | Always 0 | Mode | Enable | Scaling | Always 0 | Left Btn | Middle Btn | Right Btn |
| Byte 2 | Resolution | | | | | | | |
| Byte 3 | Sample Rate | | | | | | | |

Right, Middle, Left Btn = 1 if button pressed; 0 if button is not pressed.

Scaling = 1 if scaling is 2:1; 0 if scaling is 1:1. (See commands E7h and E6h)

Enable = 1 if data reporting is enabled; 0 if data reporting is disabled. (See commands F5h and F4h)

Mode = 1 if Remote Mode is enabled; 0 if Stream mode is enabled. (See commands F0h and EAh)

- E8h (Set Resolution) - The mouse responds with acknowledge (FAh) then reads one byte from the host and again responds with acknowledge (FAh) then resets its movement counters. The byte read from the host determines the resolution as follows:

| Byte Read from Host | Resolution |
|---------------------|------------|
| 0x00 | 1 count/mm |
| 0x01 | 2 count/mm |
| 0x02 | 4 count/mm |
| 0x03 | 8 count/mm |

- E7h (Set Scaling 2:1) - The mouse responds with acknowledge (FAh) then enables 2:1 scaling (discussed earlier in this document.)
- E6h (Set Scaling 1:1) - The mouse responds with acknowledge (FAh) then enables 1:1 scaling (discussed earlier in this document.)

The only commands the standard PS/2 mouse will send to the host are the "Resend" (FEh) and "Error" (FCh). They both work the same as they do as host-to-device commands.

Initialization:

The PS/2 mouse is normally detected/initialized only when the computer is booting up. That is, the mouse is not hot-pluggable and you must restart your computer whenever you add/remove a PS/2 mouse (furthermore, some motherboards may be damaged if you add/remove a PS/2 mouse while the computer is running.)

The initial detection of the PS/2 mouse occurs during POST. If a mouse is detected, the BIOS will allow the operating system to configure/enable the mouse. Otherwise, it will inhibit communication on the mouse's bus. If you boot the computer with a mouse attached, then detach/reattach the mouse while in Windows, the OS *may* be able to detect the mouse was reattached. Microsoft tried to support this, but it only works about 50% of the time.

The following is the communication between my computer (running Win98SE) and mouse when it boots up with a standard PS/2 mouse attached. It is fairly typical of how a PS/2 mouse is initialized and if you want to emulate a PS/2 mouse it must (at minimum) be able to support the following sequence of commands...

- Power-on Reset:

```

Mouse: AA Self-test passed
Mouse: 00 Mouse ID
Host: FF Reset command
Mouse: FA Acknowledge
Mouse: AA Self-test passed
Mouse: 00 Mouse ID
Host: FF Reset command
Mouse: FA Acknowledge
Mouse: AA Self-test passed
Mouse: 00 Mouse ID
Host: FF Reset command
Mouse: FA Acknowledge
Mouse: AA Self-test passed
Mouse: 00 Mouse ID
Host: F3 Set Sample Rate : Attempt to Enter Microsoft
Mouse: FA Acknowledge : Scrolling Mouse mode

```

```

Host: C8 decimal 200      :
Mouse: FA Acknowledge    :
Host: F3 Set Sample Rate :
Mouse: FA Acknowledge    :
Host: 64 decimal 100     :
Mouse: FA Acknowledge    :
Host: F3 Set Sample Rate :
Mouse: FA Acknowledge    :
Host: 50 decimal 80      :
Mouse: FA Acknowledge    :
Host: F2 Read Device Type:
Mouse: FA Acknowledge    :
Mouse: 00 Mouse ID       : Response 03 if microsoft scrolling mouse
Host: F3 Set Sample Rate :
Mouse: FA Acknowledge    :
Host: 0A decimal 10      :
Mouse: FA Acknowledge    :
Host: F2 Read Device Type:
Mouse: FA Acknowledge    :
Mouse: 00 Mouse ID       :
Host: E8 Set resolution  :
Mouse: FA Acknowledge    :
Host: 03 8 Counts/mm     :
Mouse: FA Acknowledge    :
Host: E6 Set Scaling 1:1 :
Mouse: FA Acknowledge    :
Host: F3 Set Sample Rate :
Mouse: FA Acknowledge    :
Host: 28 decimal 40      :
Mouse: FA Acknowledge    :
Host: F4 Enable          :
Mouse: FA Acknowledge    :
Initialization complete...

```

If I then press the Left Button...

```

Mouse: 09 1 1 00001001; bit0 = Left button state; bit3 = always 1
Mouse: 00 1 1 No X-movement
Mouse: 00 1 1 No Y-movement
... and release the Left Button:
Mouse: 08 0 1 00001000 bit0 = Left button state; bit3 = always 1
Mouse: 00 1 1 No X-movement
Mouse: 00 1 1 No Y-movement

```

The following is the communication between my computer (running Win98SE) and mouse when it boots up with an (emulated) Intellimouse...

- Power-on Reset:

```

Mouse: AA Self-test passed
Mouse: 00 Mouse ID
Host: FF Reset command
Mouse: FA Acknowledge
Mouse: AA Self-test passed
Mouse: 00 Mouse ID
Host: FF Reset command
Mouse: FA Acknowledge
Mouse: AA Self-test passed
Mouse: 00 Mouse ID
Host: FF Reset command
Mouse: FA Acknowledge
Mouse: AA Self-test passed
Mouse: 00 Mouse ID
Host: F3 Set Sample Rate : Attempt to Enter Microsoft
Mouse: FA Acknowledge    : Scrolling Mouse mode
Host: C8 decimal 200     :
Mouse: FA Acknowledge    :
Host: F3 Set Sample Rate :
Mouse: FA Acknowledge    :
Host: 64 decimal 100     :
Mouse: FA Acknowledge    :
Host: F3 Set Sample Rate :
Mouse: FA Acknowledge    :
Host: 50 decimal 80      :
Mouse: FA Acknowledge    :
Host: F2 Read Device Type:
Mouse: FA Acknowledge    :
Mouse: 03 Mouse ID       : Response 03 if microsoft scrolling mouse
Host: E8 Set Resolution  :
Mouse: FA Acknowledge    :
Host: 03 8 counts/mm     :
Mouse: FA Acknowledge

```

```

Host: E6 Set scaling 1:1
Dev: FA Acknowledge
Host: F3 Set Sample Rate
Mouse: FA Acknowledge
Host: 28 decimal 40
Mouse: FA Acknowledge
Host: F4 Enable device
Mouse: FA Acknowledge

```

If I then press the left mouse button:

```

Mouse: 09 00001001 bit0 = Left button state; bit3 = always 1
Mouse: 00 No X-movement
Mouse: 00 No Y-movement
Mouse: 00 No Z-movement

```

...and then release the left mouse button button:

```

Mouse: 08 00001000 bit0 = Left button state; bit3 = always 1
Mouse: 00 No X-movement
Mouse: 00 No Y-movement
Mouse: 00 No Z-movement

```

After I downloaded/installed the Microsoft's Intellimouse drivers with support for the 4th and 5th buttons, the following sequence was found:

```

... (starts same as before) ...
Host: F3 Set Sample Rate : Attempt to Enter Microsoft
Mouse: FA Acknowledge : Scrolling Mouse mode.
Host: C8 decimal 200 :
Mouse: FA Acknowledge :
Host: F3 Set Sample Rate :
Mouse: FA Acknowledge :
Host: 64 decimal 100 :
Mouse: FA Acknowledge :
Host: F3 Set Sample Rate :
Mouse: FA Acknowledge :
Host: 50 decimal 80 :
Mouse: FA Acknowledge :
Host: F2 Read Device Type :
Mouse: FA Acknowledge :
Mouse: 03 Mouse ID : Response 03 if microsoft scrolling mouse.
Host: F3 Set Sample Rate : Attempt to Enter Microsoft 5-button
Mouse: FA Acknowledge : Scrolling Mouse mode.
Host: C8 decimal 200 :
Mouse: FA Acknowledge :
Host: F3 Set Sample Rate :
Mouse: FA Acknowledge :
Host: C8 decimal 200 :
Mouse: FA Acknowledge :
Host: F3 Set Sample Rate :
Mouse: FA Acknowledge :
Host: 50 decimal 80 :
Mouse: FA Acknowledge :
Host: F2 Read Device Type :
Mouse: FA Acknowledge :
Mouse: 04 Mouse ID : Response 04 if 5-button scrolling mouse.
... rest of initialization same as before ...

```

Emulation/Interfacing:

- Click here for routines that emulate a PS/2 mouse or keyboard
- Click here for routines that emulate a PS/2 host (ie, interface a mouse/keyboard)
- Click [here](#) for a fully-functional PS/2 mouse written for the PIC16F

Footnotes:

1) 2:1 scaling only applies to the automatic data reporting in Stream mode. It does not effect the reported data sent in response to the "Read Data" command.

2) The mouse and host do not buffer "Resend" commands. This means "Resend" will never be sent in response to the

"Resend" command.

3) A "packet" may be a 3-byte movement data packet, a 4-byte movement data packet (for the Intellimouse), a 3-byte status packet (see "Status Request" command) a 2-byte completion-code-ID packet (AAh,00h or FCh,00h), or a 1-byte response to a command.