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 merchantibility 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.

General Description

There are many types of pointing devices available for the modern PC including mice, trackballs, touchpads, electronic whiteboards, etc. Virtually all of these devices communicate on one of two interfaces: Universal Serial Bus (USB) or the PS/2 mouse interface. 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 and it remains a widely-supported interface. However, USB has quickly caught on these last few years and will eventually replace the PS/2 interface entirely.

The PS/2 mouse interface utilizes a bidirectional serial protocol to transmit movement and button-state data to the computer's auxiliary device controller (part of the keyboard controller). The controller, 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 host provides the mouse with a 5V ~100 mA power supply.

Electrical Interface / Protocol

The PS/2 mouse uses the same protocol as the PS/2 keyboard (a.k.a. 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 periodically reads these inputs 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 covered 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. The movement counters represent the mouse's offset relative to its position when the previous movement data packet was issued, or when the last non-"Resend" (0xFE) command was successfully sent to the host.

When the mouse reads its inputs it records the current state of its buttons and increments/decrements the movement counters according to the amount of movement that has occurred since the last input sample. If either of the counters has overflowed, the appropriate overflow flag is set. Futher modification of the counter is disabled until it the counters are reset (due to a packet being sent).

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 affect the movement counters, but does affect the reported 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 issuing the "Set Scaling 2:1" (0xE7) command. If 2:1 scaling is enabled, the mouse will apply the following algorithm to the movement 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

2:1 scaling only applies to the automatic data reporting in stream mode. It does not affect the reported data sent in response to the "Read Data" (0xEB) command.

Movement Data Packet

The standard PS/2 mouse sends movement/button information to the host using the following 3-

byte 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							

The movement values 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. Their value represents the mouse's offset relative to its position when the previous packet was sent, in units determined by the current resolution. The range of values that can be expressed is -255 to +255. If this range is exceeded, the appropriate overflow bit is set.

Modes of Operation

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

- Reset - The initial mode, in which the mouse performs initialization and self-diagnostics.
- Stream - The default operating mode, in which the mouse issues movement data packets when movement occurs or button state changes.
- Remote - The host must poll for movement data packets.
- Wrap - A purely diagnostic mode where the mouse echoes every received packet back to the host.

Reset Mode

The mouse enters reset mode at power-on or in response to the "Reset" (0xFF) command. Upon entering this mode, the mouse performs a diagnostic self-test called *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

The mouse then sends a BAT completion code of either 0xAA (BAT successful) or 0xFC (Error). The host's response to a completion code other than 0xAA is undefined.

Following the BAT completion code (0xAA or 0xFC), the mouse sends its device ID of 0x00. This distinguishes it from a keyboard or nonstandard mouse. I have read documents indicating the host should not transmit any data until it receives a device ID. However I've found some BIOS's will send the "Reset" (0xFF) command immediately following the 0xAA received after a power-on reset.

Once the mouse has sent its device ID to the host, it enters stream mode.

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 may be reported is known as the *sample rate*. This parameter ranges from 10-200 samples/sec, with a default value of 100 samples/sec. The host may set this value using the "Set Sample Rate" (0xF3) command.

Note that reporting is **disabled** by default. The mouse will not actually issue any movement data packets until it receives the "Enable Data Reporting" (0xF4) command.

Stream mode is the default operating mode, and is otherwise set using the "Set Stream Mode" (0xEA) command.

Remote Mode

In remote 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 occured. Instead, the host polls the mouse using the "Read Data" (0xEB) command. Upon receiving this command the mouse will issue a single movement data packet and reset its movement counters.

The mouse enters remote mode upon receiving the "Set Remote Mode" (0xF0) command.

Remote mode is rarely used.

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" (0xFF) and "Reset Wrap Mode" (0xEC) commands. The mouse treats these as valid commands and does not echo them back to the host.

Wrap mode is rarely used.

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 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 an Intellimouse is used on a computer which only supports the standard PS/2 mouse, it will still function except its scrolling wheel and 4th and 5th buttons will be disabled.

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 0x00.) To enable the scrolling wheel, the host sends the following command sequence:

1. Set sample rate 200
2. Set sample rate 100
3. Set sample rate 80

The host then issues the "Get device ID" (0xF2) command and waits for a response. If a standard PS/2 mouse (i.e., non-Intellimouse) is attached, it will respond with a device ID of 0x00. 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 0x03. 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 value 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.

To enable the scrolling wheel AND the 4th and 5th buttons, 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" (0xF2) command and waits for a response. A Microsoft Intellimouse will respond with a device ID of 0x04, then uses 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	Z movement

"4th Btn" = 1 iff the 4th mouse button is pressed, and "5th Btn" = 1 iff the 5th mouse button is pressed. "Z movement" is a 2's complement value which represents the amount of movement that has occurred since the last data report. Valid values range from -8 to +7.

There are 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 virtually 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

Following is the set of commands accepted by the standard PS/2 mouse. If the mouse is in stream mode, the host should disable data reporting (command 0xF5) before sending any other commands.

- 0xFF (Reset) - The mouse responds to this command with "acknowledge" (0xFA) then enters reset mode.
- 0xFE (Resend) - The host sends this command whenever it receives invalid data from the mouse. The mouse responds by resending the last packet 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" (0xFC) command, cycle the mouse's power supply to reset the mouse, or it may inhibit communication (by bringing the clock line low). This command is not buffered, which means "Resend" will never be sent in response to the "Resend" command.
- 0xF6 (Set Defaults) - The mouse responds with "acknowledge" (0xFA) then loads the following values: Sampling rate = 100, resolution = 4 counts/mm, Scaling = 1:1, data reporting = disabled. The mouse then resets its movement counters and enters stream mode.
- 0xF5 (Disable Data Reporting) - The mouse responds with "acknowledge" (0xFA) then disables data reporting and resets its movement counters. This only affects data reporting in stream mode and does not disable sampling. Disabled stream mode functions the same as remote mode.
- 0xF4 (Enable Data Reporting) - The mouse responds with "acknowledge" (0xFA) then enables data reporting and resets its movement counters. This command may be issued while the mouse is in remote mode, but it will only affect data reporting in stream mode.
- 0xF3 (Set Sample Rate) - The mouse responds with "acknowledge" (0xFA) 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.
- 0xF2 (Get Device ID) - The mouse responds with "acknowledge" (0xFA) followed by its device ID (0x00 for the standard PS/2 mouse). The mouse should also reset its movement

counters.

- 0xF0 (Set Remote Mode) - The mouse responds with "acknowledge" (0xFA) then resets its movement counters and enters remote mode.
- 0xEE (Set Wrap Mode) - The mouse responds with "acknowledge" (0xFA) then resets its movement counters and enters wrap mode.
- 0xEC (Reset Wrap Mode) - The mouse responds with "acknowledge" (0xFA) then resets its movement counters and enters the mode it was in prior to wrap mode (stream mode or remote mode).
- 0xEB (Read Data) - The mouse responds with "acknowledge" (0xFA) then sends a movement data packet. This is the only way to read data in remote mode. After the data packet has successfully been sent, the mouse resets its movement counters.
- 0xEA (Set Stream Mode) - The mouse responds with "acknowledge" (0xFA) then resets its movement counters and enters stream mode.
- 0xE9 (Status Request) - The mouse responds with "acknowledge" (0xFA) 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 0xE7 and 0xE6).
Enable = 1 if data reporting is enabled; 0 if data reporting is disabled (see commands 0xF5 and 0xF4).
Mode = 1 if remote mode is enabled; 0 if stream mode is enabled (see commands 0xF0 and 0xEA).

- 0xE8 (Set Resolution) - The mouse responds with "acknowledge" (0xFA) then reads one byte from the host and again responds with "acknowledge" (0xFA) then resets its movement counters. The byte read from the host determines the resolution as follows:

Byte Read from Host	**Resolution**
00	1 count/mm
01	2 count/mm
02	4 count/mm
03	8 count/mm

- 0xE7 (Set Scaling 2:1) - The mouse responds with "acknowledge" (0xFA) then enables 2:1 scaling.
- 0xE6 (Set Scaling 1:1) - The mouse responds with "acknowledge" (0xFA) then enables 1:1 scaling.

The only commands the standard PS/2 mouse will send to the host are "Resend" (FEh) and "Error" (FCh).

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. Adding/removing the PS/2 mouse while the computer is running may physically damage some motherboards.

The initial detection of the PS/2 mouse occurrs 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. Testing this on Win98 SE, it seems to work about 50% of the time.

The following is the communication between my computer (running Win98 SE) and a standard PS/2 mouse during the boot process. 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 ...
```