PS/2 mouse tutorial:

Index:

Initializing

Commands

Scroll and extended buttons * not finsih

USB mouse and other mousses

IRQ handler

Some on resolution, sample rate, and scaling.

Links

Initializing the mouse:

To initialize the mouse, you will need to “open” the AUX port. By default bios disable this port.
To enable you have to send “0xA8” to the keyboard controller (port 0x64). But before, be sure that the controller is ready to receive a command (by pulling port 0x60 until bit 1 = 0). After you can send commands to the mouse, this is done by writing “0xD4” to the keyboard controller. Then after you send the “command” you want to send to the mouse. But we are not finish with the Initializing, we need to force the mouse to send the data to us, via the IRQ12 (the interrupt number varies, but on bochs is it 0x74) The best thing to do, is to reset the mouse, with the command “0xff”. After you send the commands you want (if you want another packet rate, or what ever.) when you are ready to receive packets, you have to do the fowling:

Send the command 0x20(get status byte), and then wait until data is ready (pulling port 64 until bit 0 is 0), and then read from port 60. The input is a byte, and you have to change it, in order to enable IRQ12. The only modifications you have to do, is to set bit 1, (enable irq12). Then you will have to send the command “0x60” (set status byte), followed by the modificated status byte, with has to go though port 60 (remember to wait until the controller is ready).
Now we are close, to have initialized the mouse, the only thing you need, is to send the command “0xF4” (send packets) to the mouse.

Commands:

There are quite a few commands, but not so many.

These are ONLY mouse commands, if you want to know something about the other commands, dealing with the mouse, you will have to se a keyboard specification. A great note is: all commands are supposed to send “ACK (FA)” in return to a command.

All numbers is in HEX

FF: reset the mouse, and chose default settings, and disabled packet reporting

FE: resend, resends the last packed. If you get “FC, then there is a error”

F6: set default settings, sets the default settings (resolution: 4 ; scaling 1:1 ; data reporting: disabled ; sampling rate: 100)

F5: Disable packed reporting.

F4: enable packed reporting (it only sends, when the mouse change state, see IRQ handler for more info)

F3:set packed rate “sample rate” (after sending this command, you will have to send the new rate to port 0x60, remember to wait)

F2: get mouse id (after sending this command, you will have to read port 0x60, remember to wait)

EE: set warp mode, warp mode is where the mouse response to all the commands you were sending (the result is the command you send)

EC: reset warp mode, the mouse sends first “ACK (0xFA)”, and then it reset the movement counters, and enters the last abscessed mode (except from warp mode)

EB: request mouse packet, the mouse sends “ACK”, then it sends the number of bytes it is supposed to (normally 3 bytes), and reset its counter movement. (A little note from me; this does not work, when you are in stream mode, and in remote mode, bios may caught the data, so be aware of unusable data)

EA: set steam mode, enters a mode where the mouse only send movements data, when it is been moved.

E9: status request, the mouse sends “ACK”, and then it sends 3 bytes (port 0x60), b

E8: set resolution (pixels / mm) (the only usable values are: 0,1,2,3. The result is: 1,2,4,8 pixels /mm)

E7: set scaling (2:1), the mouse sends “ACK”, and then enables 2:1 scaling

E6: set scaling (1:1), the mouse sends the “ACK”, and then enables 1:1 scaling

A little note: The data send back, can also be “error (0xFC)”, or “Resend (0xFE)”.

Scroll and extended buttons:

Lol

USB mouse and other mousses:

The USB mouse appears as a ps/2 mouse, but the only different, is that the IRQ happens at the USB’s IRQ, not the AUX. A touchpad looks and appear as a ps/2 mouse, meaning that it will work with normal ps/2 configurations.

IRQ handler:

Handling the IRQ is quite simple, but you have to keep 1 thing in mind: when the IRQ fires, it only has 1 byte ready. That means that you have to wait until the next IRQ fires, to get the next byte, and so on.
The mouse sends the number of packets it has been programmed to, but if you have not madden the scrolling wheel working, or similarly, it is only 3 packets. The way you get the packed, is by reading port 0x60
remembered NOT to wait, or else you will “erase” the input. When we are talking about X, and Y it means: X is vertical movement (if you move to left it is negative). Y is horizontal movement (if you move down, it is negative; keep in mind that if you not reverse the y input, it will be like inverting the mouse on windows) the packets are described as the fowling:

1 byte:
bit 7: Y overflow, if it is set there is an overflow in the “Y” value.

Bit 6: X overflow, if it is set there is an overflow in the “X” value.

Bit 5: Y sign, if this bit is set, Y is negative, witch is described later.

Bit 4: X sign, if this bit is set, X is negative, witch is described later

Bit 3: Always 1. So if this bit is not set, it is not a “valid” packed.

Bit 2: middle button, set if the middle button is down

Bit 1: right button, set if the right button is down

Bit 0: left button, set if the left button is down

2 byte:

This is the “X” movement.

3 byte:

This is the “Y” movement.

If there is a 4 packed: (and it is only wheel activated)

4 byte: “Z” movement, the mouse wheel.

If the fifth button (also the 4.th button) has been activated, then the 4.th bit is like this:

Byte 4:

Bit 7: always 0

Bit 6: always 0

Bit 5: button 5, if it is set, the button is down

Bit 4: button 4, if it is set, the button is down

Bit 3→ bit 0: Z movement.

A good advice to the IRQ handler: store the packets at a place the handling program can get the data.

Another tip is keeping track on how many packets you have, and how many you are supposed to have.

Some on resolution, sample rate, and scaling:

Resolution:

This is very easy, because there are only 4 different values for this, and all it does is to determinate how sensitive the mouse is:

00 = 1 count / mm

01= 2 counts/mm

02 = 4 counts /mm

03= 8 counts / mm

Sample rate:

Sample rate is numbers of samples / second (how many packets it can send in one second), there is only 7 possible values for samples, those are: (NOT hex numbers)

10,20,40,60,80,100,200.

Scaling:

This is not fairly complex, even if you might think,
if it is 1:1, there is nothing to think about, it reports all movement.

However if it is 2:1, then it is not like 1:1
in 2:1 the formula is like this:
N > 5 ; movement counter

N*2; reported movement

This means that you will get the twice as many reports, if the movement counter is above 5. Else it is close to be the same:

Movement counter	**Reported movement**
0	0
1	1
2	1
3	3
4	6
5	9

Links:

Here is some good links, and articles I have used. (Date: the 28 October 2008)

http://www.computer-engineering.org/index.php?title=PS/2_Mouse_Interface

This is not a copyright document, but I would like you to ask for my permission, if you would like it to go comical.
Send a email to :
kasper@webmasteren.eu

Or ask me at:
osdev.org (at the forum)