The Linux Busmouse HOWTO

Table of Contents

The Linux Busmouse HOWTO....1
Chris Bagwell, cbagwell@sprynet.com....1
1.Introduction....1
2.Determining your mouse type....1
3.Getting your mouse working....1
4.Using your mouse....1
5.Still can't get your mouse going?....2
1.Introduction....2
1.1 Copyright and Disclaimer....2
1.2 Feedback....2
1.3 Acknowledgements....2
2.Determining your mouse type....3
2.1 Hardware interfaces....3
Inport mice....3
Logitech mice....4
PS/2 mice....4
ATI combo video/mice....4
IBM PC110 palmtop digitizer....4
Apple Desktop Mouse....4
Hybrid Mice....5
2.2 Mouse protocols....5
3.Getting your mouse working....5
3.1 Setting the mouse interrupt....5
Common IRQ usage.In most cases IRQ4 is used for the first serial port (/dev/ttyS0),IRQ3 for the second (/
Inport and Logitech mice....6
ATI−XL mice....6
PS/2 mice....6
3.2 Configuring the kernel....7
Compiling the kernel....7
Changing interrupts with newer kernels....9
3.3 The mouse devices....9
4.Using your mouse....10
4.1 Configuring Applications....10
Redhat....11
Other configurations....11
4.2 gpm....11
4.3 XFree86....11
4.4 XFree86 and gpm....12
5.Still can't get your mouse going?....13
5.1 Other resources....13
3−Button Mouse HOWTO....13
Laptop−HOWTO....13

The Linux Busmouse HOWTO

Chris Bagwell, cbagwell@sprynet.com

v2.0, 14 Feb 2000

This document describes how to install, configure and use a busmouse under Linux. It lists the supported busmice and attempts to answer the most frequently asked questions with regards to busmice. It also contains some pointers for serial mice as well.

1.Introduction.

- 1.1 Copyright and Disclaimer.
- 1.2 Feedback.
- 1.3 Acknowledgements.

2.Determining your mouse type.

- 2.1 Hardware interfaces.
- 2.2 Mouse protocols.

3.Getting your mouse working.

- 3.1 Setting the mouse interrupt.
- 3.2 Configuring the kernel.
- 3.3 The mouse devices.

4.Using your mouse.

- 4.1 Configuring Applications
- 4.2 gpm.
- 4.3 XFree86.
- 4.4 XFree86 and gpm.

5.Still can't get your mouse going?

- 5.1 Other resources

1.Introduction.

This document is a guide to getting your busmouse working with Linux. With the more advanced distributions available today setting up your busmouse is generally easy but when you do run into problems this document will help give you a better understanding of how to manually setup your busmouse hardware and software.

Busmouse support has been in the kernel for as long as I can remember, and hasn't changed much in a long time, so this document should be relevant to any version of Linux you're likely to have.

1.1 Copyright and Disclaimer.

This document is Copyright (c) 2000 by Chris Bagwell. This document may be distributed under the terms set forth in the Linux Documentation Project License at http://linuxdoc.org/copyright.html. Please contact the author if you are unable to get the license.

The author disclaims all warranties with regard to this document, including all implied warranties of merchantability and fitness for a certain purpose; in no event shall the author be liable for any special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortious action, arising out of or in connection with the use of this document.

1.2 Feedback.

If you find any mistakes in this document, have any comments about its contents or an update or addition, send them to me at the address listed at the top of this howto.

1.3 Acknowledgements.

This howto has been, in the spirit of Linux, a community effort. Thanks goes out to Mike Battersby, `mib@post.com` as he started this FAQ. Any errors are most likely added by me.

Many thanks go to Johan Myreen for the sections on the PS/2 mice, Robert T. Harris for help on the ATI−XL sections and Reuben Sumner for miscellaneous info and constructive criticism.

Thanks also to the multitudes of people who have sent me mouse information, fixes or words of encouragement.

2.Determining your mouse type.

There are two separate but important characteristics you will need to know about your mouse before you go on: what hardware interface it uses and what mouse protocol it uses.

The hardware interface is the hardware aspect of the mouse, taking into account things like which i/o ports it uses and how to check if it is installed. This is the part which the kernel is concerned with so that it knows how to read data from the mouse. For serial mice users, this part of the story is easy since their interface is always the serial port device drivers.

The mouse protocol is the software aspect of the mouse. Applications need to know the protocol to interpret the raw mouse data they receive from the kernel device driver.

2.1 Hardware interfaces.

The Linux kernel up to the 2.2 series supports four different kinds of busmouse hardware interface : Inport (Microsoft), Logitech, PS/2 and ATI−XL.

The 2.4 series of kernel also includes support for several new busmice including an IBM PC110 digitizer pad and Apple Desktop mouse. It also contains drivers for USB mice which are sometimes discussed with busmice since they fall outside the more common serial driver interface.

There is no sure−fire way of determining your hardware interface. For those that have Windows 9x install, it sometimes helps to go to the Systems screen under the control panel and look for devices controlling your mouse. This screen will often tell you the interface type and what I/O ports and interrupts this hardware uses.

Inport mice.

Inport mice include most of the old style Microsoft mice which are shaped like a bar of dove soap. Inport mice generally connect to an interface card which plugs into the bus on your motherboard. If the plug which connects your mouse cord to the interface card is round, has 9 pins, and a notch in one side you likely have an Inport mouse.

ATI currently claims that all ATI VGA + Busmouse combo cards used the Microsoft Inport hardware and thus owners of these cards should first attempt using the Microsoft Busmouse driver.

The ATI−XL VGA+Busmouse card technically is compatible with the Microsoft Inport mouse driver but has slightly different provisioning for interrupts and therefore has a specific driver for it. Avoid this driver if possible since its not supported very well.

Logitech mice.

Logitech mice in general appear almost exactly the same as Inport mice. They too connect to an interface card via a 9 pin mini−din connector. Hopefully, it will have come in a Logitech box or have ``Logitech'' printed on the connector card so that you can tell it actually is a Logitech mouse.

PS/2 mice.

The PS/2 mouse interface is not on an expansion card, the mouse is connected to the PS/2 port on the keyboard controller. This is sometimes located on the keyboard or more often as a extra port somehow connected directly to the computer case.

A PS/2 port uses a 6−pin mini DIN connector, similar to the keyboard connector. Many laptops also use this kind of interface to their trackballs or touchpads; in this case the mouse is internally connected to the PS/2 port and needs no connector.

ATI combo video/mice.

ATI−XL mice are a variant of Inport mice, with some slight differences in interrupt setup. They come on the ATI−XL combined video adaptor/mouse card. Unless you know you have an ATI−XL card (and thus an ATI−XL mouse), you probably don't have one of these. It is possible for ATI−XL mice to use either the ATI−XL or Inport kernel drivers, although the ATI−XL driver should give better results.

There is also an older ATI video adaptor/mouse card called either ATI VGA1024 or ATI VGA Wonder. These cards are setup the same as the ATI−XL but use the Logitech mouse protocol. For these mice the hardware drivers are setup the same as for the ATI−XL but you setup the software the same as the Logitech mice.

IBM PC110 palmtop digitizer.

The IBM PC110 palmtop contains a digitizer pad that can be used to emulate a mouse using the PS/2 mouse protocol. In this case, you set up the hardware interface using the IBM PC110 device driver and setup software as you would a PS/2 mouse.

Apple Desktop Mouse

This bus mouse is common on Macintoshes and uses a 4 pin connector. The hardware is controled by the kernel but application software will need to be aware of its unique mouse protocol. I am currently unaware of what protocols this mouse uses.

Hybrid Mice

The period of time that busmouse were popular, the hardware companies were experementing alot and cross−licensing each others designs. Therefore, you could have a ATI mouse that uses the Logitech hardware interface or a Logitech that uses the Inport interface. If you know you have a busmouse but can't get it working with the standard interface/protocol setups, try experementing by using different interface device drivers with different mouse protocols.

2.2 Mouse protocols.

The PC world is full of different and conflicting mouse protocols. Fortunately, the choice for bus mice is considerable smaller than that for serial mice. Most Inport, Logitech and ATI−XL mice use the ``BusMouse'' protocol, although there are some ancient Logitech mice which use the ``MouseSystems'' protocol, and some even older Microsoft mice which use the "Logitech" protocol. PS/2 mice will always use the ``PS/2'' protocol.

3.Getting your mouse working.

Once you have figured out your mouse interface and protocol types, you're ready to proceed.

3.1 Setting the mouse interrupt.

Now that you've found out what hardware interface your mouse uses you'll need to know which interrupt number your mouse is using, and make sure it doesn't conflict with any other peripherals you have installed. That last part deserves to be repeated! **Make sure that it does not conflict with any other peripherals you have installed!**

You should **make sure** that your mouse is not trying to use the same interrupt as any of your other devices −−− it is not possible for the mouse to share an interrupt under Linux, even though it may work fine under other operating systems. Check the documentation for all your peripherals to see which interrupt they use.

Under Linux, busmice don't register which IRQ they are using until after they have been opened by an application that makes use of them. Plug−n−Play hardware often registers their interrupts during bootup. This creates a possibility for a Plug−n−Play peice of hardware to steal the IRQ away from your mouse. Bear in mind that other operating systems may be initialzing the P−n−P cards to an IRQ that is not in conflict with your busmouse but things may not work out as nicely under Linux. It is up to you to make sure there are no IRQ conflicts between all of your equipment.

Common IRQ usage.In most cases IRQ4 is used for the first serial port (`/dev/ttyS0`),IRQ3 for the second (`/dev/ttyS1`) (these are assuming youactually have such devices −−− if you don't you can happily usetheir IRQ's), IRQ5 for some SCSI adaptors, and IRQ12 for some network

cards.Having a other card use IRQ12 is a big problem for machines with PS/2 ports asyou are forced to use IRQ12 only for the PS/2 port.For ATI–XL, Inport and Logitech mice the kerneldefault is to use IRQ5, so if you are stuck with a pre–compiledkernel (eg, CD–ROM users) you will have to use that. If you areusing an Inport or Logitech mice with a newer kernel you may be able topass a command line option to the kernel to tell it what interruptto use without recompiling.

Inport and Logitech mice.

If you open up your computer's case and look at the card which your mouse plugs into, you should notice a block of jumpers on the card (hopefully labeled ``INTERRUPT'') with positions for interrupt (otherwise known as IRQ) numbers 2,3,4 and 5. To change the interrupt simply move the jumper from its current position onto the correct pair of pins.

```
***     MAKE SURE YOUR COMPUTER IS TURNED OFF   ***
***     BEFORE CHANGING THE JUMPERS AROUND!!    ***
```

ATI–XL mice.

ATI–XL and a few other ATI busmice have a software selectable IRQ – you should have received with your mouse a MS–DOS program (`VSETUP.EXE`) to set the IRQ. In order to do so you must (temporarily) boot MS–DOS and run this program. Note that the VSETUP program takes an optional parameter ``/70'' to increase the vertical refresh rate (which results in less flicker). The VSETUP program also allows you to select either the primary or secondary mouse address – you should set this to the primary address or the kernel will not be able to detect your mouse.

Once VSETUP has been run you must perform a hard reset for the new configuration to take effect.

PS/2 mice.

The PS/2 mouse always uses IRQ12 –– there is no way of changing this (except with a soldering gun.) In the rare case that some other device is using IRQ12, you'll have to rejumper that peripheral to use another IRQ number.

3.2 Configuring the kernel.

In order for your busmouse to operate correctly you will need to configure your kernel to compile in busmouse support. If you are using a pre−compiled kernel then it often comes with support for all three busmouse included. This may still not be enough. The kernel could be trying to use the wrong interrupt or the detection can get confused and treat your mouse as the wrong type.

In pre−2.4 series kernel there is no support for auto−detection of the IRQ that a busmouse is using. Therefore, if your card is set to anything but the kernel default value of IRQ 5 then you will need to let the kernel know what IRQ to use instead. This can be done in one of two ways. The easiest is to pass command line options to the kernel during the bootup process. Please read thru the "Compiling the kernel" section for reference but attempt the directions in "Changing interrupts with newer kernels" first. Also of great help in this area is the Bootprompt−HOWTO, avaliable at most sites that carry this HOWTO.

Compiling the kernel.

Change to your kernel directory (here assumed to be (`/usr/src/linux`) and do a

```
make config
```

If you are unsure as to your mouse type, the first time you recompile the kernel you may wish to enable all of the busmouse options in the hope that the kernel will autodetect your mouse properly. People have mixed success with this: it doesn't always work, but on the other hand it might save you any further compiles.

Answer ``y'' or "m" to the question pertaining to your type of busmouse interface and ``n'' to all the other busmouse questions. Use the "m" option if you have your system setup to support loading kernel modules if you do not or do not know what that means then it will be safe to always answer "y" to have the support directly compiled into your kernel.

As an example, if you have an Inport mouse you should answer ``y'' to

```
Microsoft busmouse support
```

and ``n'' to all other busmouse questions. Answer the non−mouse related questions as you usually would.

To compile the kernel with PS/2 mouse support answer ``y'' to the question.

```
PS/2 mouse (aka "auxiliary device") support
```

The PS/2 mouse driver actually supports two kinds of devices: the standard PS/2 Auxiliary Device controller and a special PS/2 mouse interface chip from Chips & Technologies which is used in the Texas Instruments Travelmate and Gateway Nomad laptops. To compile in support for the trackballs on these computers,

answer ``y'' to the

```
C&T 82C710 mouse port support (as on TI Travelmate)
```

question. Note that you will still have to answer ``y'' to the question about the standard PS/2 driver to even get a chance to answer this question, since the 82C710 driver is actually an add−on to the standard PS/2 mouse driver.

When configured both for a standard PS/2 mouse device and the 82C710 device, the driver first tries to locate a 82C710 chip at boot time. Failing this, the standard driver is used instead, so using a kernel configured for both types of interface on a machine with a standard PS/2 mouse port should work too. However, there has been one report of a falsely detected 82C710 chip, so to be on the safe side do not configure in support for the 82C710 if you don't need it.

You will now need to tell the kernel what interrupt your mouse uses. You can skip this step if your using a PS/2 mouse as it always uses IRQ 12.

If you have a Logitech, Inport mouse, or an ATI mouse that uses the Logitech protocol, edit the file `/usr/src/linux/include/linux/busmouse.h` and change the line which says

```
#define MOUSE_IRQ 5
```

to reflect the interrupt number for your mouse (see the section Setting the mouse interrupt for details on finding your interrupt number).

If you have an ATI−XL mouse, edit the file `/usr/src/linux/drivers/char/atixlmouse.c` and change the line which says

```
#define ATIXL_MOUSE_IRQ 5
```

to reflect your mouse's interrupt number.

Due to the vagaries of the PC architecture, if you have set your mouse to use interrupt 2, you must set the #define to use interrupt 9.

Examples

For a mouse on interrupt 3, you should change the line to read

```
#define MOUSE_IRQ 3
```

For a mouse on interrupt 2, you should change the line to read

```
#define MOUSE_IRQ 9
```

Next, compile your kernel as per the instructions which come with it, and boot from the new kernel. You should now have the busmouse support correctly compiled in.

Changing interrupts with newer kernels.

The steps to compile into the kernel what interrupt it uses works with any version of the kernel to date. Newer kernels (starting somewhere in the 2.x.x's) allow you to pass arguments to the kernel during load time using something like LILO or LOADLIN to specify the interrupt number for Logitech and Microsoft Inport mice. This can be a real time saver as you do not need to recompiler your kernel (or know how to). If you've configured your kernel to load the mouse drivers as modules then you will need to pass this information when loading the module.

You can add the following options to your boot line in LILO to change interrupt:

```
bmouse=3  (Logitech Busmice)
msmouse=3 (for Microsoft Inport mice)
```

Substitute the 3 with your mouse's actual interrupt. An example of using this with lilo is:

```
LILO:linux msmouse=3
```

You can consult your LILO or LOADLIN docs to see how to add this type information to their configuration files so that you do not need to type it.

If your system uses kerneld to auto load modules, you can edit your /etc/conf.modules or /etc/modules.conf file, which ever your system uses, and add one of the following lines.

```
options msbusmouse mouse_irq=3
options busmouse mouse_irq=3
```

3.3 The mouse devices.

Mice under Linux are accessed via the devices in the `/dev` directory. The following table gives a list of interface types and which device you should use.

```
INTERFACE        DEVICE        MAJOR    MINOR
```

```
-----------------------------------------------
Logitech        /dev/logibm      10      0
PS/2            /dev/psaux       10      1
Inport          /dev/inportbm    10      2
ATI-XL          /dev/atibm       10      3
```

Note: If you are using your ATI−XL mouse with the Inport driver, you should use the `inportbm` device, not the `/dev/atibm` device.

The major and minor entries are the device numbers for that particular device.

If you find that you do not have these devices, you should create them first. To do so, execute the following as root.

```
mknod /dev/logimm    c 10 0
mknod /dev/psaux     c 10 1
mknod /dev/inportbm  c 10 2
mknod /dev/atibm     c 10 3
```

Note: Some time in the (progressively less) recent history of Linux the names for the busmouse devices have changed. The following device names have been superceded by those above and should be removed: `bmousems, bmouseps2, bmouseatixl, and bmouselogitech.`.

Many people like to create a symbolic link from their mouse device to `/dev/mouse` so that they don't have to remember which device they need to be using. If you have one of the current Linux distributions you will almost certainly find that you have such a link. If you have such a link, or create one, you should make sure that it is pointing to the correct device for your mouse.

4.Using your mouse.

This section deals with the general use of your mouse with various applications.

4.1 Configuring Applications

Most Linux Distributions will prompt you for what type of mouse you have during installation and then setup both gpm and X windows for you. If you do not select the correct mouse protocol or install new mouse hardware you can usually run a command line program that will reconfigure both gpm and X windows to use the new protocol.

Redhat

Under RedHat you can run /usr/sbin/mouseconfig.

Other configurations

Consult your installation documents for what to run under other distributions.

4.2 gpm.

Gpm is a program which allows you to do mouse based 'cut– and–paste' between Linux virtual consoles, much like you can under X, and is a good way of testing your mouse out. The most current version of gpm can be found at ftp://ftp.prosa.it/pub/gpm. Most Linux distributions come with a precompiled gpm binary.

When invoking gpm, use the –t switch to indicate which protocol your mouse is using and the –m option to indicate which mouse device you are using. Three protocols useful for most busmice are logi, bm, and ps2. The default for mouse device is to use `/dev/mouse`, so you can omit the –m option if you have the appropriate symbolic link. An example for a Microsoft Inport mouse is:

```
gpm -t bm
```

or if you use the PS/2 protocol:

```
gpm -t ps2
```

You should then be able move your mouse and see a block move around the screen and also be able to cut and paste text between virtual consoles using the mouse buttons. Read the documentation with gpm, or do a ``man gpm'' for more information on how to operate it.

4.3 XFree86.

To use your busmouse under XFree86, you will need to set your mouse protocol type in your Xconfig file. If you have a BusMouse protocol mouse, your Xconfig should contain (including the quotes)

```
Section "Pointer"
  Protocol "Busmouse"
  Device   "/dev/mouse"

  # Any other options such as Emulate3Buttons
EndSection
```

For PS/2 mice change the protocol line to:

```
Protocol        "PS/2"
```

If you have a two button mouse, it should also contain the line

```
Emulate3Buttons
```

which will allow you to emulate the use of the middle mouse button by pressing both mouse buttons simultaneously. All other mouse related lines, such as ``BaudRate" and ``SampleRate" should be commented out, as these have no effect on bus mice.

4.4 XFree86 and gpm.

For a long period of the kernel developement, it was not possible to share busmice between processes. Because of this it was hard to run both XFree86 and gpm at the same time. If you try to run X with gpm running and you get errors like the following then you know you are using one of these older kernels.

```
Fatal server error:
Cannot open mouse (Device or resource busy)
```

There are two meathods of getting gpm working with XFree86 with these kernels. The first is to kill any copy of gpm you have running before you start up XFree86. The second is to use gpm's "repeater" option (it takes mouse data and repeats the information to multiple applications).

I would recommend upgrading your kernel if possible so that you can share busmice between processes. For this document, I will only explain the simplest meathod of using XFree86 and gpm together with older kernels. Please see gpm's documentation if you would like to use the repeater meathod.

Gpm allows you to terminate running copies of itself by executing:

```
gpm -k
```

This should be done before starting up X11. Take whatever script you use to start up your X session, such as `startx`, and add the above command to the top of the script so that gpm is shut down automatically. You may wish to also put a command that restarts gpm at the bottom of the script so that it restarts upon exiting your X session.

5.Still can't get your mouse going?

So you've read through this howto a dozen times, done everything exactly as you think you should have, and your mouse still doesn't work? The best advice I can give you is this: experiment. Sure, it's a pain in the posterior, but in the end the only way to find out what is going to work with your mouse is to try all of the alternatives until you have success.

As always, if there is something you don't understand, try reading the manual page first and see if that helps. If you have a specific question, or a problem you think I might be able to help with, feel free to contact me at the address listed at the top of this howto, and I'll see if I can help you out or point you to someone who can.

The `comp.os.linux.setup` newsgroup or `comp.os.linux.hardware` is the appropriate forum for discussion and/or questions regarding setup −−− please don't post questions to other groups, and especially don't crosspost questions to two or more of the Linux groups, they are more than cluttered enough as it is! When posting, you will get a much better response (and much fewer flames) if you use appropriate Subject: and Keywords: lines. For example:

```
Subject: BUSMICE - Gateway 2000 mouse wont work.
Keywords: mouse busmouse gateway
```

5.1 Other resources

The following are useful resources for information regarding Linux and Mice.

3−Button Mouse HOWTO.

The 3−Button Mouse HOWTO is of great use if you have a combination 3−Button Serial/PS2 mouse. It gives pointers on how to get the middle button working.

Laptop−HOWTO.

The Linux Laptop−HOWTO gives pointers on getting external mice working along with your built in mouse.

Wacom Tablet HOWTO gives pointers on using the Wacom Tabel as a mouse.