Zip Drive Mini−HOWTO

Table of Contents

Zip Drive Mini−HOWTO

Kyle Dansie, `dansie@ibm.net`

v2.4, 10 January 1999

This Document provides a quick reference quide on setting up and using the Iomega ZIP drive with Linux.

6.Performance

7.Frequently asked questions

8.Getting more current information

1.Introduction

The Iomega ZIP drive is a popular, removable media disk drive. It is avaliable in several versions.

- Parallel Port
- SCSI
- IDE − ATAPI
- Plus

The most popular seems to be the Parallel port version. It has been around a long time and it is low cost. The SCSI version has also been around for a while but it needs a scsi card that many people do not have already, so it it not quite as popular. The ATAPI version now seems to be more available. The Plus is the newer version of the ZIP drive. It has the capability to detect what port its plugged into, scsi or parallel. This document will focus on the Parallel port version, but also offer some tips for other versions.

This document incorporates information collected and published by others, in particular:

- Grant Guenther, grant@torque.net
- Scot Wilcoxon, sewilco@fielday.mn.org
- Joe Mack, mack@ncifcrf.gov
- Byron Jeff, byron@cc.gatech.edu

I thank them for their important contributions, and accept responsibility for any errors that I have introduced.

A special thanks to Grant Guenther who wrote the 0.26 version of the ppa program and also wrote the original version of this HOWTO. If you find any mistakes or oversights in this document, please let me know. dansie@ibm.net

1.1 Home of this document

The most current version of this document will reside at http://www.njtcom.com/dansie/zip−drive.html

---

2.Quick Start

OK so you have purchased a new zip drive and now you want to use it. Some distributions have the modules already built. You can try insmod ppa for starters. (Make sure you have it hooked up and a disk inserted). If the ppa program is not there, then configure the kernel. My machine has 2 parallel ports, one for the zip and one for the printer. This makes things easy, no cable switching.

Configure the kernel:

- cd /usr/src/linux
- make xconfig
- scsi support = Y
- scsi disk support = Y
- Iomega zip support as a module
- printer support also as a module
- save it and exit
- make dep
- make clean
- make zImage or zlilo or zdisk
- make modules
- make modules_install

Now to use the drive:

- load the module `insmod ppa`
- build a mounting point. `mkdir /zip`
- insert a preformatted windoze type disk into the drive.
- mount the disk. `mount −t vfat /dev/sda4 /zip`
- use any standard file commands as in `l /zip, ls /zip, df, cp,`
- when you are finished `umount /zip`

If you had trouble with any of the previous steps, see the following sections for more detailed instructions.

3.Configuring a kernel for the ZIP drive

To use the ZIP drive with Linux, you must have a kernel configured with support for the SCSI system, support for SCSI disks, and support for the host adapter you are using. If you are not familiar with building a kernel, you should read up on the Linux Kernel HOWTO for information.

You must begin the process of building a kernel with the configuration step. Here, you identify the specific kernel components that you need. First step `cd /usr/src/linux`. There are several ways to actually do the configuration. Under X windows I use **make xconfig**. There is also `make menuconfig` or `make config` for command line prompts. The easiest way is with xconfig.

In the section **SCSI Support** set **SCSI support = Y**. Also set **SCSI disk support = Y**.

In the section **SCSI low−level drivers** you want to set **IOMEGA Parallel Port ZIP drive SCSI support = M**. The M stands for modules.

In the section **Character Devices** find and set **Parallell Printer support = M**

If you are a bit unsure about any of this, use zdisk for the make step. This will build and install the kernel to floppy. If you screw it up somehow, you still have a good bootable system on the hard drive.

Now build the kernel with these steps:

- make dep
- make clean
- make zImage or zlilo or zdisk
- make modules
- make mdoules_install

hint if you want to create an output log of the make zImage step you can use

```
make zImage 2>&1 | tee zImage.out
```

This is really a good thing to do, if you have compile errors or other conflicts in the configuration, now you have a file with the error messages or compile errors. This makes asking for help much easier.

You also need to load the modules somewhere. On my system I just added a couple of lines to the boot.local file.

- insmod ppa
- insmod lp

My system is Suse 5.3 and this file is in /etc/rc.d. The location of this file will vary by distribution, but there will be a file somewhere, for local changes to be made at start up time. You should use the file that is applicable for your distribution.

Now `shutdown and reboot.`

You can also build all or part of the SCSI system as modules. If you do this, be sure to load scsi.o, then sd.o and finally the driver for your host adapter, before you try to access the ZIP drive.

3.1 SCSI version

If you already have a SCSI disk in your system, and you are connecting the ZIP drive to the same controller, there is no additional kernel configuration required. Otherwise, you will most likely have to build a new kernel.

If you are building a kernel to support the SCSI version of the ZIP drive, you should select SCSI support and SCSI disk support. You must also select a driver for the interface card you will use. If you have a ZIP Zoom, select the aha152x driver.

Be sure to read the documentation for your adapter in the SCSI HOWTO and any README files in the drivers/scsi subdirectory of the Linux source tree. Pay attention to command line parameters that you might have to use to help the kernel initialise your adapter.

For instance, if you are using the ZIP Zoom card, you will have to add something like

```
aha152x=0x340,11,7,1
```

to the boot command (or include it in your /etc/lilo.conf file in an append clause). This tells the driver the port address and IRQ of your ZIP Zoom card − be sure to use the numbers that correspond to the way your jumpers are set.

You should also read Paul Gortmaker's BOOTPROMPT HOWTO for information about configuring your kernel with LILO or LOADLIN.

4.The ZIP drive

There are at least five or six versions of the Iomega ZIP 100 drive. The Iomega page for these drives is at http://www.iomega.com/product/zip/ They all accept special cartridges resembling a 3.5" floppy disk that hold 100 megabytes of data. The disks actually hold 96 cylinders of 2048 sectors each holding 512 bytes. This would normally be called 96 Megabytes.

The external drive comes in these types:

- Parallel port

- SCSI
- Plus

The internal drive comes in these types:

- SCSI
- IDE
- ATAPI

The internal versions can be mounted in 3.5 or 5.25 drive slots. The external versions are in a small blue plastic case enclosure, powered by an external wall brick.

All the drives have a large pushbutton on the front of the drive. This is used to eject the disk. Linux locks the door while using the drive, but if the button is pressed while the door is locked, the ZIP drive will remember and eject the disk as soon as the software unlocks it.

4.1 Parallel port version Zip100

This drive is described at http://www.iomega.com/product/zip/zippp.html

The Linux driver comes in a couple of different versions. The 0.26 version of the ppa.c program ships standard with the 2.0.x kernels. If your kernel is older than 2.0.35 then you have the **OLD** program. Check out

David Campbell's page for the more current version of this program. At the time of this writing the current version is 1.42.

The auto configuration features of the 1.4x program are very nice. You do NOT have to tell the progam what the base address of the parallel port is. The program will prob the standard address's for the parallel ports and attach the drive when it finds it.

The most common problem is to load the lp printer module before the ppa program. The ppa program must be loaded before the lp module. The lp module will tie up the port that the zip drive needs, this is why the ppa progam must be loaded first.

Also be sure that all cables are firmly attached.

If you did everything right :) you will see messages like this at boot time.

```
VFS: Mounted root (ext2 filesystem) readonly.
Adding Swap: 102528k swap-space (priority -1)
ppa: Version 1.42
ppa: Probing port 03bc
ppa: Probing port 0378
ppa:     SPP port present
ppa:     EPP 1.9 with hardware direction protocol
ppa: Found device at ID 6, Attempting to use EPP 32 bit
ppa: Communication established with ID 6 using EPP 32 bit
ppa: Probing port 0278
ppa:     SPP port present
```

```
scsi0 : Iomega parport ZIP drive
scsi : 1 host.
  Vendor: IOMEGA    Model: ZIP 100           Rev: D.09
  Type:   Direct-Access                      ANSI SCSI revision: 02
Detected scsi removable disk sda at scsi0, channel 0, id 6, lun 0
SCSI device sda: hdwr sector= 512 bytes. Sectors= 196608 [96 MB] [0.1 GB]
sda: Write Protect is off
  sda: sda4
lp2 at 0x0278, (polling)
```

You can view these messages with dmesg | more. You can save the messages to a file with dmesg > dmesg.txt. This is often usefull if you are trying to debug the installation.

Also see section Getting more information

4.2 SCSI external version

This drive is described at http://www.iomega.com/product/zip/zipscsi.html

The external SCSI version of the ZIP drive has two DB25F connectors, and two configuration switches. One switch selects the drive's target address. The choice is limited to target 5 or 6. The other enables an internal terminator, in case the drive is the last one on a chain. The 25 pin SCSI connectors use the familiar Macintosh style wiring. The drive is shipped with a Macintosh type cable, but standard cables and converters are easily obtained if you are using a host adapter with a Centronics or high−density connector.

I have not seen an internal SCSI drive, but I would expect it to have a standard 50 pin DIP header SCSI connector and the same two switches.

Make sure that the target address you choose does not conflict with any other SCSI devices you may have on the same bus. Also be sure that the physically last drive in a chain has termination enabled, or an external terminator installed.

If you have an internal SCSI disk or CD−rom, and you connect your ZIP drive to the existing adapter, you should check to see if there are any terminators on the card that must be removed. Only the two extreme ends of the SCSI bus should be terminated. If your bus is partly internal and partly external, there should be one terminator on the last external device and one on the last internal device, but no terminators on the adapter card itself.

Be sure that all cables are firmly attached.

The kernel configuration is much the same as the internel version. See the next section SCSI Internal version for an example configuration.

The ZIP ZOOM host adapter

Iomega markets a SCSI host adapter under the name ZIP Zoom. This is actually based on the design of the **Adaptec AHA1520** family of adapters. It has an external Macintosh type DB25F connector, compatible with the cable that comes with the ZIP drive.

Linux supports this adapter with the **aha152x** driver.

4.3 The SCSI Internal version

Install hardware as described in the "Installation and Reference Guide" noting which SCSI ID, IRQ and I/O Port Address are being used. (You'll need this info later.) Things will go smother if the drive and adapter card use different SCSI ID's.

Beware The zip zoom card is hardwired for scsi ID number 7. The scsi id in the command line below is for the card, not the drive. The card and the drive must use different scsi id numbers. For example if the card uses 7 then use 5 or some other number for the drive.

Recompile the kernel after configuring it to include 'SCSI', 'SCSI disk' and 'AHA152X/2825' support.
INSTALL NEW KERNEL :−(

Determine what your kernel command line is:

`aha152x=[I/O Port][,IRQ][,SCSI ID][,RECONNECT][,PARITY]`

For example :

`aha152x=0x140,10,7,1`

If your using LILO add your kernel command line to your lilo.conf file using the append command.

(ie. append = "aha152x=0x140,10,7,1")

If your using LOADLIN add your kernel command line to the command you use to initiate loadlin.

```
  (ie. loadlin c:\vmlinuz root=/dev/hda3 aha152x=0x140,10,5,1 ro)
```

4.4 The ZIP Plus − IMM driver

This drive is described at http://www.iomega.com/product/zip/zipplus.html

This is a **newer** version of the external Zip drive. The **Zip Plus** does not use the ppa driver, instead it uses the **imm** driver. Philippe Andersson sent in this excellent section on the imm driver for the Zip Plus. Thanks

Philippe.

First, rebuild your kernel to include the following items:

- modules support
- SCSI support
- SCSI Disk support
- lp as a module (if you need it)

You'll notice we didn't select any SCSI low−level driver. That's ok −− we'll build it separately later.

Then, you need to get the driver source code from Dave Campbell's home page (http://www.torque.net/~campbell/imm.tar.gz). Version 0.18 is the current one at the time of writing (Aug. 98). Unzip it and untar it somewhere (under `/usr/src`, for instance). Then just run `make`. You'll get the module you need (`imm.o`). Copy it to

```
/lib/modules/$(uname -r)/scsi
```

.

If your lp module was loaded (check with **lsmod**(1)), unload it (`rmmod lp`), then load `imm.o` (`insmod imm`) and you're all set basically.

If the ZIP drive was not connected and powered on at the time, you'll get the message "init_module: Device or resource busy", and the module won't load. (By the way, it looks like you don't need to switch your PC off to connect the ZIP −− just make sure the module is not loaded and the drive is turned off, then plug it in, turn the drive on, and load the module.)

If the ZIP was connected and powered on, but there was no disk in, the module will load all right, but you'll get the message that it can't read the disk partition table. This is ok, as the partition table will be automatically read when you insert a ZIP disk.

If there was a ZIP disk in, you'll get the full information displayed, including a list of partitions defined on the disk and its Write Protect status. Here is a sample load−time message:

```
vger:~# insmod imm
imm: Version 0.18
imm: Probing port 03bc
imm: Probing port 0378
imm:     SPP port present
imm:     ECP with a 16 byte FIFO present
imm:     PS/2 bidirectional port present
imm:     Passed Intel bug check.
imm: Probing port 0278
scsi0 : Iomega ZIP Plus drive
scsi : 1 host.
  Vendor: IOMEGA    Model: ZIP 100 PLUS      Rev: J.66
  Type:   Direct-Access                      ANSI SCSI revision: 02
Detected scsi removable disk sda at scsi0, channel 0, id 6, lun 0
SCSI device sda: hdwr sector= 512 bytes. Sectors= 196608 [96 MB] [0.1 GB]
sda: Write Protect is off
 sda: sda1
vger:~#
```

One last remark to mention that this version of the driver also locks the drive door while the disk is mounted. If the eject button is pressed in this situation, nothing happens, but the drive "remembers" and ejects the disk as soon as it is dismounted.

4.5 ATAPI version

This drive is described at http://www.iomega.com/product/zip/zipatapi.html

There was an IDE version of the drive produced for a while. I think that for the most part this has been replaced by the ATAPI version.

Donald Stidwell sent in these comments on the ATAPI version. Thanks Don.

I use an ATAPI Zip drive and it works with both 2.0.32 and 2.0.33 kernels. I've used it under both RH 5.0 and OpenLinux 1.2 (my current used distribution). To get it to work under OpenLinux, I just enabled **ATAPI floppy support** in the kernel. OpenLinux does not have this support compiled in by default.

No other drivers are needed. It will mount as an extended partition on partition 4. I.E, mine mounts on HDB4. I mount it under /mnt/zip as noauto, although I don't suppose there would be any real problem with automounting. I just wonder about ejecting disks. I always dismount the drive before ejecting a cartridge.

There are more detailed instructions for the ATAPI install in the Linux Gazette May 1998 issue. See the 2 cent tip section.

```
<url url="http://www.linuxgazette.com/issue28/lg_tips28.html#atapi">
```

4.6 IDE version

I have not used the IDE version. Eric Backus sent in these comments. Thanks Eric.

I have one of these. It came with my Gateway 2000 computer a year ago. I think most of these were shipped by large OEM companies like this, before the ATAPI version of the ZIP drive was available.

The good news about this drive: no kernel modules or modifications are needed to support it. It looks to the kernel like an IDE hard drive. It worked for me with no effort with kernel 2.0.31 and 2.0.32.

The bad news about this drive: because it doesn't use ATAPI, you can't use the SCSI−to−ATAPI translation, which means you can't use mtools to write−protect disks (or to eject them, for that matter).

4.7 USB version

This drive is described at http://www.iomega.com/product/zip/imac.html

These drives look very cool and the USB interface also looks good. The only problem is that the USB interface for Linux is still under development at this time. So for now, this drive is not usable under Linux. Check this address for updates.

http://peloncho.fis.ucm.es/~inaky/USB/news.html

4.8 ZIP250 version

This drive is desribed at http://www.iomega.com/product/zip/zip250.html

This drive is new as of January 1999. I just bought one of these and it works very well in Linux. Just use the IMM driver from http://www.torque.net/~campbell/imm.tar.gz Set up the driver the same way as the zip plus. Mainly just download the driver, unpack it, and run make. Then copy the imm.o executable to the modules directory. For example /lib/modules/2.0.35/scsi (The 2.0.35 is your kernel version). If you are not sure of the version do uname −r.

5.Using the ZIP drive

If you have built in all the required components, the kernel should recognize your adapter and drive at boot time. If you are using a loadable module for your driver, the following discussion applies once that module is loaded.

For the most part, the SCSI and parallel versions of the drive behave identically, except that the parallel version is somewhat slower.

5.1 Fdisk, mke2fs, mount, etc.

Once you know the drive name for your ZIP drive, you are set. You can manipulate the drive with the normal Linux disk management commands. `fdisk` (or perhaps `cfdisk`) is used to manipulate the partition tables on the disk. `mke2fs` can be used to format a partition with the ext2 filesystem − the one most commonly used in Linux. `mount` is used to connect a formatted partition into your directory hierarchy.

You should study the manual pages for these tools if you are not familiar with them. Be warned that there are now several quite different versions of the `fdisk` program − be careful.

I'll describe two common scenarios.

5.2 An existing DOS formatted disk

If you have a ZIP disk with a DOS file structure that was originally created by Iomega's tools, the partition scan should say that the disk has one partition, `/dev/sda4`.

You should make a place to mount the disk, lets say `/zip`, and then mount it as an MS−DOS filesystem:

```
mkdir /zip
mount −t vfat /dev/sda4 /zip
```

You could also use msdos instead of vfat. vfat supports long filenames where msdos does not. Now, the files on the disk should appear in /zip. While the disk is mounted, you will not be able to remove it. When you are finished with the disk you can umount it to release it and detach it from your directory hierarchy.

```
umount /zip
```

Once you've made the /zip mount point − you don't need to do it again, so you could come back later and mount something else there.

5.3 Re−format as a native Linux disk

If you want to erase a ZIP disk and make a Linux native file system on it. You should use fdisk on the entire disk:

```
fdisk /dev/sda
```

and delete any existing partitions (with the `d` command). Then create a new partition with the `n` command, make it primary partition number 1, use `w` to write the partition table to disk, and quit with `q`.

Format the partition

```
mke2fs /dev/sda1
```

(The 1 is the number that you gave this partition in `fdisk`). Now you can mount the disk:

```
mount −t ext2 /dev/sda1 /zip
```

(re−using that mount point we created before).

5.4 fstab entries

The /etc/fstab allows you to configure the mount command. I like to be able to mount and write to the zip disk from any login id on the system. I added 2 lines to the end of the fstab file that look like this.

```
/dev/sda1    /zip          ext2          noauto,rw,user,nosuid,sync
/dev/sda4    /zipdos       vfat          noauto,rw,user,nosuid,sync,mode=0777
```

These entries assume that you have a /zip and a /zipdos directory. If you want all users ids to read and write, make sure you set the permissions. For exmaple **chmod 666 /zip** . You must do the chmod as root. The fstab entries also assume that your dos disks are partitioned as 4 (sda4 in my case), and the ext2(linux) disks are partitoned as 1 (sda1). You can read more about the fstab options with **man mount**

With the fstab entries as above you can mount a dos disk with **mount /zipdos** If you are using a linux ext2 disk then use the command **mount /zip**

5.5 The ZIP Tools disk

There is some extra work to be done if you want to use the disk that comes with the ZIP drive. As shipped, the software controlled write protection is enabled. Most people have unlocked the disk under DOS before ever trying to use it with Linux. Linux cannot access a locked disk, and it must be unlocked with Iomega's tools.

A native Linux program to manage the write protection feature, among other things, is expected to be available soon.

6.Performance

With the standard version of the 2.0.x kernels you get the 0.26 version of the ppa.c program. This is a very reliable simple program. It has gotten a lot of use and abuse and held up rather well. The only bad thing about this program is that it's not very fast. On my machine I have version 2.0.30 kernel, and 2 parallel ports. This is also a low−buck machine pentium 133 − 16 meg memory, 5 gig maxtor eide hard drive. The machine cost me 500. U.S. Dollars and the drive was 250. I am cheap but so what.

I had the ZIP drive on the second port and the printer on the first port. The second port was an old add in card, no epp mode. This made the driver run in 4 bit mode.

To base line the performance I copied the netscape tar file from the hard drive to the zip drive. This file is about 10.6 meg. Time 7 minutes 15 seconds. Going from the zip drive to the hard drive was about 7 minutes even. Not exactly setting the world on fire.

I then made 2 changes. First I switched the ZIP drive to the first port and went into the bios and set EPP mode on for the first port. At the same time I hooked up the printer to the 2nd port.

Second change I went out and downloaded the new version of the program from David Campbells page Then I installed this program. Check out the readme file that is included.

Then I ran the same test again of copying the Netscape tar file to and from the ZiP drive. The copy to the C drive was 1:55 seconds. From the hard drive to the Zip drive was 1 minute flat. This compares to 7:15 and 7:00 with the old program. This was a big improvement over the 0.26 program. This was also using the zip drive as a vfat file type.

Next I formated a zip drive in ext2 format. Ran the same test again. The copy going to the hard drive was only 50 seconds. The copy from the hard drive to the Zip drive was only 40 seconds.

Performance Summary

For this test case of moving a 10.6 meg file, the 4 bit standard port mode and program version 0.26 is the slowest 7:00 and 7:15

The new program (1.41) and EPP mode with ext2 file types is the fastest at 50 seconds and 40 seconds.

7.Frequently asked questions

These are some of the questions that we get asked a lot. Please read the answers here before asking them again !

7.1 Can I plug a printer into the parallel ZIP drive ?

Yes, but. Currently, you cannot have both the lp and ppa drivers active on the same parallel port. A resource sharing protocol has been designed and is in the early stages of implementation, but will not be available before Linux 2.0 is released.

If you like living on the bleeding edge of things, get the 2.1.x kernel and you can have port sharing. See http://www.cyberelk.demon.co.uk/parport.html

The easy way to do this with 2.0.x kernel is to have 2 parallel ports. I bought my second one for about 15 dollars and it works great. You can use one for printing and one for your ZIP drive. Just build a kernel with both the lp and ppa drivers as modules. I use the newer (1.41) version of ppa, so I just load ppa first (insmod ppa). It then finds the zip drive on the first port. Then I load lp (insmod lp) and it assigns the printer to the second port.

If you use the 0.26 version of the program then upgrade to 1.41 or newer. The only answer you will get on qustions about this driver is to upgrade the program.

7.2 Do you plan to support EPP/ECP ports in PPA ?

Version 0.26 of the program does not support EPP. Version 1.41 **does** support the EPP. This program is also much faster. See the performance section for more details.

7.3 Can I run Linux from a ZIP drive ?

People have done this. It would seem this might make a great rescue disk or a way to test an install.

http://metalab.unc.edu/LDP/HOWTO/mini/ZIP−Install.html

7.4 Can I boot from the ZIP drive ?

This depends on what you are using for a host adapter. If your host adapter has a BIOS ROM with code that can boot from target 5 or 6, then yes, you can boot from the SCSI ZIP drive. You cannot boot from a parallel port drive.

If you have DOS on your main disk and want to boot Linux from a ZIP disk, you can do this using the LOADLIN boot loader. Your kernel image must be somewhere on the DOS disk, but the rest of your Linux system could be on the ZIP. Make sure that the kernel you use has the correct drivers built in.

As far as I am aware, there are no installation boot floppies for any of the popular distributions that contain the parallel ZIP driver.

7.5 Why does Iomega use partition number 4 ?

This is one of the most popular questions, but I don't think anyone has a definite answer to this one. Could be they had their head where the sun don't shine. Maybe there is no reason.

7.6 How can I have the disk mounted at boot time ?

All you need to do is to add a line to your `/etc/fstab` file. For instance, if you will always have a DOS disk in the drive when you boot, you could put

```
/dev/sda4   /zip  vfat  defaults  0 0
```

in the `fstab`. Depending on your distribution, the initialization scripts might try to run fsck on partitions listed in your fstab. Be aware that this could cause problems if you forget to put the disk in the drive when you boot, or have the wrong disk there.

7.7 What happens if there is no disk inserted when I boot ?

The kernel will try to read the partition table, but the operation will time out. There is nothing to worry about. Once you have inserted a disk, and try to do something with it the kernel will notice that there is a disk there now and rescan the partition table.

HINT: when you change disks, it is a good idea always to use fdisk to check the partition structure on the new disk.

The BIOS on some SCSI host adapters will attempt to read the partition table on your disk during the system boot. If you cannot disable this check, you may be forced always to boot with a disk in the drive.

7.8 Can I use the parallel drive as a real SCSI disk ?

The PPA−3 parallel−to−SCSI adapter is implemented as a single ASIC chip that Iomega calls a VPI0. It is embedded on the ZIP drive's controller card. There is an actual SCSI bus present, but not in a useful form.

Although I haven't tried to compare the two cards, it seems logical that the VPI0 replaces the conventional electrical buffering circuits that would be required if the SCSI bus were extended outside the package.

7.9 Can PPA be used with Iomega's parallel port tape drives ?

No. Those drives are floppy−tape drives. There is no SCSI involved.

Several people have indicated an interest in applying my methods to try to determine the protocol and develop a driver for these tapes. The first step in that process is to get the DOS driver working under DOSemu. I have not yet heard that anyone has been successful in doing it.

7.10 Will PPA work with the parallel port SyQuest EZ135 ?

No. The EZ135 is an IDE drive with a ShuttlePort parallel to IDE converter embedded in it.

However there has been some work done on this driver. Check out

http://www.torque.net/linux−pp.html

8.Getting more current information

More current versions of the parallel port zip drive (ppa.c) program and current technical information of the driver can be found at

http://www.torque.net/~campbell/

Information of many parallel port external device drivers can be found at

http://www.torque.net/linux−pp.html

Parallel Port sharing or through−port information is found at

http://www.cyberelk.demon.co.uk/parport.html

Installing Linux from a zip drive (for machines without a CD)

http://metalab.unc.edu/mdw/HOWTO/mini/Install−From−ZIP.html

An X frontend to the ziptool program can be found at

http://www.scripps.edu/~jsmith/jazip/

Information on configuration of Lilo can be found at

http://metalab.unc.edu/mdw/HOWTO/mini/LILO.html

Information on using bootprompts can be found at

http://metalab.unc.edu/mdw/HOWTO/BootPrompt−HOWTO.html

If you want to install the Linux OS onto a Zip Drive and run the OS from the Zip drive

http://metalab.unc.edu/LDP/HOWTO/mini/ZIP−Install.html

Iomega's web pages are at

http://www.iomega.com/.

And if you are looking for general information about parallel port programming for the PC, you might want to visit

http://www.lvr.com/parport.htm
