

RASCAL/LINUXBOT SLACKWARE v3.1 INSTALLATION MANUAL

Version 1.0

Intelligent Autonomous Systems Laboratory
University of the West of England, Bristol
Coldharbour Lane
BRISTOL

Copyright © 2000, University of the West of England, Bristol

Preface

This manual describes, in detail, the process of Linux installation for the Rascal/LinuxBot/Moorebot series of miniature-wheeled mobile robots. To be precise, this manual covers the installation of Slackware Linux version 3.1 (1996). This older version of Slackware has the merit that it is compact and can be installed, complete with C and C++ compilers and a web server, comfortably within a 70Mbyte partition. This manual describes installation of both single boot Linux and (in an appendix) dual boot MSDOS/Linux configurations.

All Slackware distributions may be downloaded from:

<ftp://ftp.freeware.com/pub/linux/slackware/>

How to use this manual. This manual describes a series of steps that need to be undertaken, as described and in sequence, in order to complete successfully the Rascal/LinuxBot installation from scratch. The procedure described is the result of the author's experience from January 1999 to date. Any departure from the sequence described here may result in an incomplete or unpredictable Linux installation.

In this manual user input is shown underlined in 12 pt courier, ie

```
darkstar:~# cd /etc
```

means, following the prompt `darkstar:~#` type `cd /etc` and hit return.

Please report any errors, omissions or suggestions for improvements – together with the version number of this manual - to the author,

Alan Winfield

email: Alan.Winfield@uwe.ac.uk

tel: +44 (0)117 344 3498

Intelligent Autonomous Systems Laboratory

University of the West of England, Bristol

Coldharbour Lane

BRISTOL

BS16 1QY

<http://www.ias.uwe.ac.uk/>

Version 1.0

August 2000

Contents

1. Introduction and Prerequisites
 - 1.1 Hardware Set up
 - 1.2 Preparation of BOOT and ROOT Diskettes
2. Linux Installation
 - 2.1 Step 1: CMOS Set up
 - 2.2 Step 2: BOOT Linux from Floppy Disks
 - 2.3 Step 3: Partition the solid-state disk drive
 - 2.4 Step 4: Create and activate the linux swap partition
 - 2.5 Step 5: Install Linux from the CDROM
 - 2.6 Step 6: Check swap partition is enabled
 - 2.7 Step 7: Wireless network installation and configuration
 - 2.8 Step 8: Create a new user

Appendix A Installation of dual MSDOS/Linux system

1. Introduction and Prerequisites

This manual describes the process of Linux installation and configuration from scratch. The main body of this manual describes installation of a single boot Linux system, i.e. one which after reset boots directly into Linux using the Linux boot Loader (LILO). This is the simplest and preferred approach, which makes maximum use of the solid-state disk space available and minimises the bootup time.

However, in some circumstances a dual boot MSDOS/Linux system may be preferable (or indeed the only option if LILO and the CPU BIOS are incompatible). In a dual boot system we first boot into a small MSDOS partition, and then load Linux from within MSDOS using LOADLIN. Appendix A details the slightly different approach which is needed to install a dual boot MSDOS/Linux system.

Although not essential it is strongly recommended that you gain some familiarity with Linux before attempting the installation and configuration process described in this manual.

A good starting point is the **DOS-to-Linux HOWTO**, which (along with other HOWTOs) is available from the Linux Documentation Project web site, at <http://www.linuxdoc.org/>

Before proceeding you will need the following:

- PC/104 Processor: ie Ampro 3SXi fitted with 8Mbytes RAM;
- PC/104 PCMCIA adapter;
- PC/104 VGA adapter (unless built into development system);
- PCMCIA Wireless LAN card, ie DEC Roamabout/DS or Lucent Wavelan/IEEE;
- Solid-state Disk Drive, ie SanDisk or Lexar Compact Flash;
- 3.5" Floppy Disk Drive unit and cable;
- IDE/ATAPI CDROM drive;
- Slackware 3.1 distribution on CD;
- Two 3.5" diskettes, prepared as Linux BOOT and ROOT disks (as described below) ;
- 3.5" diskette with precompiled Roamabout/DS and Wavelan/IEEE drivers (UWE supplied);
- (optional) 3.5" MSDOS 6.22 system diskette

1.1 Hardware Setup

Linux installation and configuration may be performed with or without a PC/104 development system. If no development system is available then the PC/104 stack – comprising the processor card, VGA card and PCMCIA adapter – can be mounted onto the LinuxBot motherboard. In this case, the motherboard does no more than provide 5V power for the PC/104 stack. External power sources will however need to be arranged for the Floppy Disk and CDROM drives.

The overall PC/104 setup is shown below in block diagram, in figure 1 below.

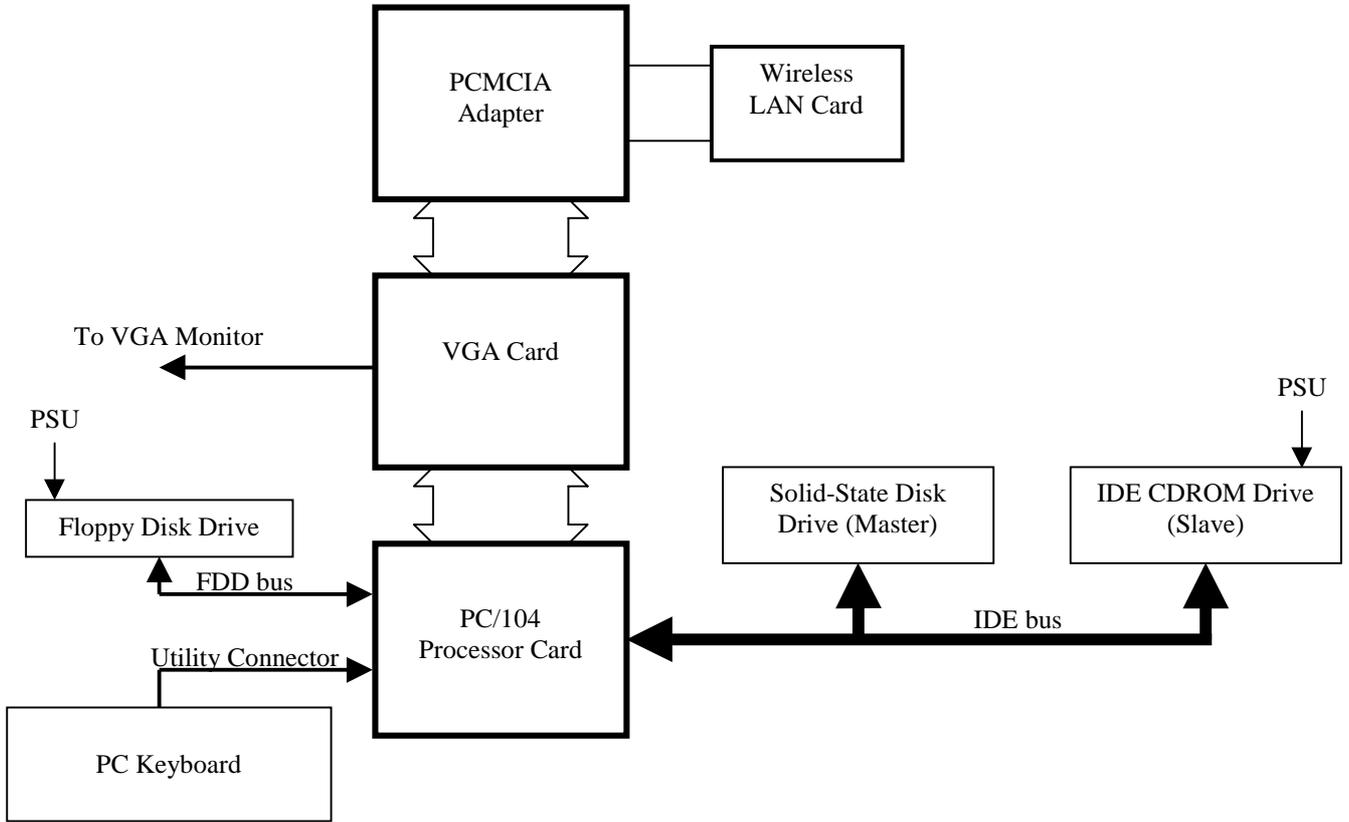


Figure 1 Hardware Installation Setup Block Diagram

1.2 Preparation of BOOT and ROOT Diskettes

On a desktop PC mount the Slackware CD. From the 'bootdsk.144' sub-directory on the CDROM, copy the following files to a temporary directory on the C: drive:

```
rawrite.exe  
bare.i
```

From the 'rootdsk' sub-directory copy the following files:

```
gzip.exe  
color.gz
```

Then exit MS-Windows into MSDOS mode (not an MSDOS window). With a blank diskette in the A: drive, create the BOOT diskette by typing:

```
>rawrite bare.i a:
```

Then with the second diskette, create (an uncompressed) ROOT disk by typing:

```
>gzip -d color.gz  
>rawrite color a:
```

3. Linux Installation

2.1 Step1: CMOS Setup

Power up the system. Enter the CMOS configuration utility and set up the following:

- Date and time (this is so that installed Linux files are correctly date and time stamped)
- A: floppy disk drive
- C: hard disk drive (auto detect drive parameters if possible)

Note that, depending on the BIOS, the CDROM may not be detected at all. This is not a problem since the Linux BOOT process will normally detect all drives, including the CDROM.

2.2 Step 2: BOOT Linux from Floppy Disks

Without powering down (otherwise date and time may be lost), insert the BOOT diskette (from section 1.2 above) into the A: drive and press reset (or just exit the CMOS set up). Linux should now boot directly from the disk.

In the unlikely event that Linux should fail to boot from the BOOT disk – typically indicated by the Linux Loader (LILO) hanging, after printing 'LI' or 'LIL' (each character is printed after a different stage in the load process) – then your BIOS is probably incompatible with LILO. In this case you may need to build a dual boot system, which first boots into MSDOS and then loads Linux from MSDOS. Refer to appendix A for details of installing a dual boot system.

Assuming that LILO booted successfully from the BOOT disk, then at the first command prompt type

```
mount root=/dev/fd0
```

and hit enter. Note do not change disks at this stage.

Then, when prompted, remove the BOOT disk and replace with the ROOT disk. Then hit enter and the floppy disk (/dev/fd0) will be mounted as the root file system. (Sometimes there will be an 'image too big' error. This can safely be ignored by hitting return again.)

At the login prompt '#', type 'root' to login as the root user. You are now running a single-user Linux shell with a limited set of commands.

2.3 Step 3: Partition the solid-state disk drive

At the Linux shell prompt run the Linux disk-partitioning tool as follows.

```
#fdisk /dev/hda
```

At this stage you may be experiencing drive_cmd errors. These can safely be ignored (unless they persist after completion of installation). Type 'p' to display the current partition information. This will almost certainly show a DOS partition. Use the 'd' command to delete this partition.

With any existing partitions deleted, now create a new partition using the 'n' command. This must be a primary ('p') partition.

```
Partition no: 1  
First cylinder: 1  
Last cylinder: 499
```

We can type either the last cylinder number "499", or the required partition size in megabytes "+70M". For the SanDisk SDIB-80 this will create a 'root' partition of approximately 70Mbytes.

Then use the 'n' command again to create a second 'swap' partition, of approximately 10Mbytes. Again this will be a primary ('p') partition.

```
Partition no: 2  
First cylinder: 500  
Last cylinder: 568
```

Now use the 't' command to change the type of partition 2, to type '82'. This identifies the partition as a Linux swap partition.

The 'p' command should now show something like this:

	Boot	Begin	Start	End	Blocks	Id	System
/dev/hda1		1	1	499	71840	83	linux native
/dev/hda2		500	500	568	9936	82	linux swap

If you are happy with this use the 'w' command to write the partition table to the disk drive, and exit fdisk. To abort and start again use 'q'. Make a note of the swap partition device name (/dev/hdxx) and the number of blocks shown in the partition table. You will need these values to activate the swap partition.

Note that the start and end cylinder numbers shown above are for an 80Mbyte SanDisk device (SDIB-80). Other disk drives will require different cylinder numbers. The partition table shown above assumes that we are building a Linux only system, which will use the Linux Loader (LILO) to boot Linux directly from /dev/hda. An alternative arrangement would be to include a small MSDOS partition; boot into MSDOS, then use the linux loader (loadlin) to boot Linux, from MSDOS.

Here is an example of a SanDisk SDIB-80 partition table, with a 10Mbyte MSDOS partition, 60Mbyte linux root partition, and 10Mbyte linux swap partition:

	Boot	Begin	Start	End	Blocks	Id	System
/dev/hda1	*	1	1	72	10352	1	DOS
/dev/hda2		73	73	499	61488	83	linux native
/dev/hda3		500	500	567	9792	82	linux swap

2.4 Step 4: Create and activate the linux swap partition

NOTE: for systems with 16MB RAM or more, then step 4 may be safely skipped.

To manually create the linux swap partition type, at the shell prompt:

```
#mkswap -c /dev/hda2 9936
```

Where /dev/hda2 is the swap partition, and 9936 is the number of blocks in the swap partition (insert your own values, noted from the fdisk partition table shown above). After this command completes, activate the swap partition as follows:

```
#swapon /dev/hda2
```

2.5 Step 5: Install Linux from the CDROM

To run the menu driven Slackware installation tool type, at the shell prompt:

```
#setup
```

From the main setup menu screen select the:

KEYMAP menu

1. select the uk.map from the keyboard map selection menu;
2. then test the keyboard, following the onscreen instructions;

3. then, you will be asked if you wish to proceed to setup the 'swap' partition. IF you skipped step 4 (i.e. you have 16MB RAM or more) then choose 'yes'. Otherwise, if you setup the swap space manually then you must skip the SWAP menu.

SWAP space menu (select ONLY IF STEP 4 WAS SKIPPED)

1. the system should first indicate that a swap space has been detected;
2. then, answer 'yes' to 'install /dev/hdax as swap space' (/dev/hda3 or /dev/hda2 depending on whether you have a DOS partition or not);
3. accept the mkswap warning message;
4. choose 'yes' to use 'mkswap';
5. choose 'yes' to activate the swap space (ignore any warning messages);
6. choose 'ok' to the swap space configured message;
7. then, choose 'yes' to continue with installation and proceed to the...

TARGET menu

1. select the root linux partition (normally /dev/hda1) as the installation target;
2. select the 'check' formatting option, with 1 inode per 1024 bytes, and bad-block checking;
3. when formatting is complete (formatting requires typically about 3 minutes) continue to the:

SOURCE menu

1. choose installation from CDROM;
2. at the next menu choose option 1 'works with most IDE/ATAPI drives';
3. at the next menu choose /dev/hdb
4. at the next menu pick the installation method: choose option 2 'slackware Normal installation to hard drive';
5. ignore the 'iso9660 message' and continue to the:

DISK SETS menu

1. first choose the A, AP, D and N disk sets to install (use the spacebar to select or de-select disk sets);
2. continue to installation, and select prompting mode 'normal'.

Each disk set will install in turn. Some packages are not optional and will be installed without a user prompt. Other packages are optional, and you will be asked to indicate whether or not they should be installed.

The following tables indicate which, of the optional packages, should be installed and which can safely be skipped in a LinuxBot install.

Disk Set A: Install	Do not install	Comments
comms		serial comms and file transfer
	getty	alternative to 'agetty'
	gpm	mouse support

	ibcs2	intel binary compatibility support
ide		kernel without SCSI
keytbls		keyboard mappings
	loadlin*	MSDOS linux loader
pcmcia		pcmcia support package: essential
	scsi	kernel with SCSI
	tcsch	alternative shell

*if Linux is to be booted from a DOS partition, then loadlin may be needed.

Disk Set AP: Install	Do not install	Comments
	ash	command shell, alternate to bash
	bc	arbitrary precision calculator lang
diff		shows differences between files
	ghostscr	postscript file viewer
gp9600		to change default baud rates
	gsfonts1	fonts for ghostscript
joe		nice wordstar-like text editor
	mt_st	magnetic tape drive support
	quota	multi-user disk quota management
	sudo	allows limited root access
	ispell	spell checker
	jpeg	jpeg package
	workbone	audio CD player
	gsfonts2	more fonts for ghostscript
	jove	another text editor
	jed	and another text editor
manpgs		manual pages: essential
	mc	midnight commander (file mngr)
	sc	simple spreadsheet program
	texinfo	on-line documentation system
	vim	another version of 'vi'
	zsh	another shell

Disk Set D: Install	Do not install	Comments
	bison	parser generator in style of 'yacc'
	byacc	yet another compiler compiler
	g77	gnu Fortran 77 compiler
man2		manual pages: essential
	gccaout	legacy a.out from gcc C compiler
strace		traces system calls and signals
	gcl	gnu Lisp
man3		manual pages: essential
gdb		gnu debugger
gxx272		gnu C++ compiler**
	p2c	pascal to C converter
	gxxaout	legacy a.out from gcc C++ compiler
	objc272	objective C compiler
	libaout	library for a.out format

rcs		revision control system
	terminfo	terminal info database
libcinfo		information files on gnu C
	objcaout	legacy a.out from objective C
libgxx		library and includes for C++**
	ncurses	terminal screen output library
	svgalib	svga graphics library
	perl1	Perl
	tools	tools to generate DLLs
	perl2	Perl

**This is the C++ compiler: install only if required.

Note that installation of the standard C compiler 'gcc272' and its libraries 'binutils', 'linuxinc' and 'libc' from disk series D are not optional.

Disk set N: Install	Do not install	Comments
bind		name server utilities
	cnews	news spooler
	dip	dialup IP connection handler
elm		standard text based email client
	mailx	intelligent mail processing system
	metamail	mime implementation
	procmail	mail processing system
	inn	news transport system
lynx		useful text web browser
	nn-nntp	newsreader
	rdist	remote file distribution program
	nn-spool	newsreader (local spool)
	pine	mail program
	ppp	support for dial-up networking
	smaillcfg	config files for sendmail
	tin	another newsreader
netpipes		enables tcpip calls in shell scripts
	sendmail	mail transport agent
	trn-nntp	threaded newsreader
	trn	another threaded newsreader
tcpip		core TCP/IP support: essential
apache		the web server
	uucp	unix to unix copy

Note that once Linux installation has been completed then individual packages may be removed, or installed from CDROM, using the Slackware 'pkgtool' utility.

Now we proceed to the:

INSTALL LINUX KERNEL menu

1. choose the 'skip this menu and use the default /vmlinuz' option.
2. when asked 'configure your system' choose yes, then proceed to the:

MAKE BOOTDISK menu

1. choose 'continue'. (We cannot make a boot diskette at this stage since root is currently mounted to /dev/fd0.)

CONFIGURATION menus

Answer NO to each of:

1. modem configuration;
2. mouse configuration;
3. screen font configuration;
4. set modem speed.

LILIO INSTALLATION menu

If you are building a system with an MSDOS partition, and you intend to boot Linux from DOS, using loadlin, then you can safely skip this section. Otherwise install LILO (the Linux Boot Loader) as follows.

From the LILIO installation menu,

1. select BEGIN to start LILO config with a new header,
there is no need for the optional append line;
select MBR (Master Boot Record) as the LILO target location;
choose NONE for LILO delay
2. select LINUX to add a linux partition to the LILO config file,
ignore the warning;
type "/dev/hda1" for a linux root partition;
type "linux" for the partition name
3. select VIEW LILO config file

The file should appear as follows:

```
#LILO configuration file
#generated by `liloconfig'
#
#start LILO global section
boot=/dev/had
#compact          #faster but won't work on all systems
#delay=5
vga=normal        #force sane state
#ramdisk=0        #paranoia setting
#end global LILO section
#Linux bootable partition config begins
image=/vmlinuz
  root=/dev/hda1
  label=linux
```

```
    read-only      #non-umsdos partitions should be
mounted read-only
#linux bootable partition config ends
```

4. select INSTALL to install LILO

LILO installation is now complete. Later we will make a bootable rescue diskette.

Setup now proceeds to further configuration menus

Answer NO to each of:

1. configure network (we shall do this later);
2. configure timezone

Setup is now complete, so choose EXIT from the main setup menu screen.

At the shell prompt, type

```
    #sync
then
    #shutdown -r now
```

After the Linux shutdown process has finished and the processor resets, remove the ROOT diskette from the floppy disk drive. The system should now boot directly into Linux on the solid-state disk drive.

Do not worry if a number of error messages appear during the boot sequence, such as pcmcia/cardmgr error messages – the pcmcia system hasn't yet been configured. There may also be an httpd error message.

Providing we reach the following login prompt, then all is well:

```
Welcome to Linux 2.0.27.
darkstar login: root
Last login: Wed Jul 12 16:44:45 on tty1.
Linux 2.0.27.
you have mail.
darkstar:~#_
```

Now login as root (there is no password yet). Then proceed to the next step in configuring the new system.

2.6 Step 6: Check swap partition is enabled

NOTE: if step 4 was skipped (e.g. if you have 16MB RAM or more), then step 6 may also be skipped.

At the shell prompt type:

```
darkstar:~# cd /etc
```

To move to the /etc directory. Then list the file system table file 'fstab' by typing

```
darkstar:/etc# cat fstab
```

There should be at least two lines, showing

```
/dev/hda1  /      ext2  defaults    1    1
none      /proc  proc  defaults    1    1
```

The first line shows that device /dev/hda1 is where the root file system is mounted (or /dev/hda2 if that is where your root file system resides).

If the file shows a third line, something like this:

```
/dev/hda2  swap  swap  defaults
```

Then the swap partition already has an entry and we don't need to create one. If however, this entry does not exist, then you must now edit 'fstab' to add this line. Either use the basic linux editor 'vi' if you are familiar with it, or use the Wordstar like editor 'joe', by typing

```
darkstar:/etc# joe fstab
```

In joe just use the arrow keys to move the cursor, and type to insert text. A help screen is available by typing ctrl-k then h. To save the updated file and exit joe type ctrl-k x, or to abort without saving just type ctrl-c.

Note that the order of the lines in fstab is not critical, but I normally place the swap line in between the two existing lines, so the file appears as follows:

```
/dev/hda1  /      ext2  defaults    1    1
/dev/hda2  swap  swap  defaults
none      /proc  proc  defaults    1    1
```

Now sync the filesystem (sync flushes any buffers to disk), and shutdown as before, with:

```
#sync
#shutdown -r now
```

When Linux reboots you may now see the 'adding swap' message appear in among the start up screen. Login as before, as root, and we can proceed to set up the networking.

2.7 Step 7: Wireless network installation and configuration

The 'pcmcia' package installed as part of disk set A includes drivers for a number of wireless LAN cards; however at the time of writing the Lucent Wavelan/IEEE card is not one of these. This step in the installation process replaces the pcmcia package supplied with the Slackware distribution with a more recent version, into which the

author has integrated the Lucent supplied Wavelan/IEEE drivers. Even if you do not intend to use the Lucent Wavelan/IEEE wireless LAN card you would still be advised to proceed with step 7 as detailed below, in order to utilise the more recent version of the pcmcia package.

First change to the directory containing the startup script files, as follows:

```
darkstar:~#cd /etc/rc.d
```

Now, as a precaution, shut down the pcmcia services, by typing:

```
darkstar:/etc/rc.d#rc.pcmcia stop
```

Now change directory back to root, by typing:

```
darkstar:/etc/rc.d#cd /
```

To check we're at root, use the ls command to show a directory listing, as follows:

```
darkstar:/#ls  
bin/  dev/  home/ mnt/  sbin/ var/  
boot/ dosc/ lib/  proc/ tmp/  vmlinuz  
cdrom/ etc/  lost+found/ root/  usr/
```

Now insert the 3.5" diskette containing the IAS lab supplied precompiled wireless LAN drivers file "pcmcia.gz", into the floppy disk drive. Mount the disk by typing:

```
darkstar:/#mount -t msdos /dev/fd0 /mnt
```

The files on the diskette now appear in sub-directory /mnt. You may list these by typing:

```
darkstar:/#ls /mnt
```

Copy the required file, to root, by typing:

```
darkstar:/#cp /mnt/pcmcia.gz pcmcia.tar.gz
```

To check it copied ok, list the root directory again, with:

```
darkstar:/#ls -l
```

The file listing should show pcmcia.tar.gz, with file size of 257448 bytes (or something very similar). Now unpack the pcmcia file by carefully typing:

```
darkstar:/#tar xzvfw pcmcia.tar.gz
```

Tar will now unpack and place all files in the correct target directories. Tar will ask for confirmation before copying each file. Carefully answer 'n' to all directories

(anything ending with '/', like ./ lib/ or lib/modules/), answer 'y' to all files (anything not ending with '/').

After unpacking is complete, umount the floppy disk by typing:

```
darkstar: /#umount /dev/fd0
```

The diskette can now be removed from the drive. Next you must edit a number of configuration files in order to set up the networking.

First change directory to /etc, by typing:

```
darkstar: /#cd /etc
```

Now use vi or joe to edit file 'HOSTNAME' and, yes the filename is in capitals! The file only contains a single line, something like "darkstar.stanford.edu". Replace the whole of this line with your chosen hostname for this robot. "linuxbot1", for instance.

Now change to sub-directory /etc/pcmcia, by typing:

```
darkstar: /etc#cd pcmcia
```

Now edit file 'config.opts'. There are four changes to be made to this file, as follows:

1. modify line 6 to read include port 0x100-0x3ff
2. modify line 7 to read include memory 0xc0000-0xdffff
3. insert a single # character at the front of line 11 to comment this line out
4. add new lines to the end of the file, as follows¹:

```
# Module options for Lucent Wavelan/IEEE  
module "wavelan2_cs" opts "port_type=3"
```

Next edit file 'network.opts'. Here you will need this robot's IP address. Edit lines 16-21 as follows (inserting your own IP address fields, using the following example):

```
IPADDR="164.11.186.196"  
NETMASK="255.255.255.0"  
NETWORK="164.11.186.0"  
BROADCAST="164.11.186.255"  
GATEWAY=""  
DOMAIN=""
```

Now change directory to /etc/rc.d, which is where slackware keeps the startup script files, by typing:

```
darkstar: /etc/pcmcia#cd /etc/rc.d
```

Edit file 'rc.pcmcia' so that line 37 reads as follows:

¹ Note that port_type = 3 configures the Lucent Wavelan/IEEE to operate in 'ad-hoc' mode (i.e. without a fixed Access Point). To configure the Lucent Wavelan/IEEE to operate with an Access Point then set port_type = 1.

```
PCIC=tcic
```

Then edit file 'rc.modules'. Insert # characters at the front of lines 57 and 59, so that the slip and ppp modules are not loaded on startup.

Next edit file 'rc.inet2'. Insert # characters at the start of lines 32-35 and lines 72-92, since we do not require SUN RPC or PORTMAPPER services.

Finally, change directory to /var/lib/httpd/conf by typing:

```
darkstar:/etc/pcmcia#cd /var/lib/httpd/conf
```

This is where the apache web server configuration files are kept. Edit file 'httpd.conf' so that line 67 defines the Servername. Use the same name you earlier defined in /etc/HOSTNAME. Ie. Servername linuxbot1. You may need to uncomment this line by removing a # character at the front of the line.

2.8 Step 8: Create a new user

Now move to the home directory:

```
darkstar:/var/lib/httpd/conf#cd /home
```

Then add a new user by typing:

```
darkstar:/home#adduser
```

This command will prompt for a login name, as follows

```
Enter login name: robot  
Full name: i robot
```

For the remaining information requested simply accept the defaults offered (including the password 'robot'). When the adduser command is complete, request a directory listing:

```
darkstar:/home#ls  
ftp/  robot/
```

and the new user's directory will appear. Now enable full 'write' access permission to this new directory, in order that we can later transfer files into /home/robot using 'ftp' from a remote machine, as follows:

```
darkstar:/home#chmod a+w robot
```

Installation is complete, so now sync and reboot as follows:

```
darkstar:/home #sync  
darkstar:/home #shutdown -r now
```

Observe the system start up sequence. You should now see that the pcmcia services correctly identify the pcmcia adapter card and, if you have plugged in a wireless LAN card, this should also be correctly identified and installed. The error messages seen earlier should now have disappeared.

Congratulations! You have now completed the installation process. To test that the networking is operating correctly you will need access to another machine on the same wireless network. After logging in (as root) you can then test the networking by firstly 'pinging' another linux machine, as follows:

```
linuxbot1:/#ping 164.11.186.190
```

You may also login to the other machine, using 'telnet', as follows:

```
linuxbot1:/#telnet 164.11.186.190
```

Or, to check the status of our own ethernet interface, type:

```
linuxbot1:/#ifconfig eth0
```

If the wireless networking does not appear to be working correctly, first check that the remote machine is up and that you have its correct IP address. Then re-check that you have correctly completed the network installation steps above. If the wireless networking is still not working, then either refer to the author of this manual (contact details are given at the front), or refer to the excellent fault finding information in the **PCMCIA-HOWTO** (downloadable from <http://www.linuxdoc.org/>)

You can check how much hard disk space you have remaining with the 'df' (disk free) command, as follows:

```
linuxbot1:/#df
Filesystem    1024-blocks  Used      Available  Capacity
/dev/hda1     62821       42280     16949     71%
```

(Your system will of course show different values to those given above).

Finally, you may need to modify the CMOS setup parameters prior to transferring the hardware from the development system to the actual robot. In particular change the default boot device from the FDD to the hard drive, remove the CDROM, and so on.

Appendix A Installation of dual MSDOS/Linux system

In some circumstances the Linux boot Loader (LILO) will not boot properly. Some old BIOSs have for instances been reported as incompatible with LILO. A strong indication of this problem is if the Linux Loader hangs after printing 'LI' or 'LIL' (LILO prints each of the four characters after distinct phases of the boot process). Thus, step 2 of the installation process described in this manual (in section 2.2) will fail.

Alternatively, there may be circumstances where a dual MSDOS/Linux system is preferred for the robot.

This appendix describes the modified process of Linux installation under these circumstances.

Prerequisites

The prerequisites listed in section 1 above apply, except that we no longer require the Slackware Linux BOOT diskette, but must have available a bootable MSDOS system diskette. You will need the Slackware Linux ROOT diskette (see section 1.2 for how to create this.)

You will also need a copy of the Linux Loader 'loadlin' on a floppy disk, together with a Linux kernel image (filename zimage). These two files are located on the Slackware 3.1 CD in the following directories:

```
\slackware\kernels\loadlin.exe
\slackware\kernels\bare.i\zimage
```

A.1 Step 1 Create and format the MSDOS partition

After CMOS setup, use the MSDOS system disk to boot into MSDOS. Then run the MSDOS 'fdisk' utility.

Using the MSDOS fdisk tool, first delete any existing MSDOS partition on the solid-state hard disk drive (most are preformatted for MSDOS by the manufacturer). Then, again using fdisk, create a new MSDOS partition, and make this active. A reasonable size for this partition would be 4Mbytes. The fdisk tool will then show the following partition information for the drive (for an 80Mbyte device):

Partition	Status	Type	Volume	Label	Mbytes	System	Usage
C: 1	A	PRI DOS	MSDOS		4	FAT12	5%

Next use the MSDOS 'format' utility to format this 4Mbyte partition.

Then use 'SYS' to make this MSDOS partition bootable, and copy any required MSDOS utilities from the diskette to the C: drive.

Then, create a sub-directory called, say, 'linux' on the C: drive, and copy into this from diskette, the two files 'loadlin.exe' and 'zimage'.

Finally, use the MSDOS editor to create a one line batch file called, say, 'linux.bat' containing the line:

```
loadlin zimage mount root=/dev/hda2
```

After Linux has been installed onto the hard drive (/dev/hda), this batch file will be used to load Linux.

A.2 Step 2 Mount the Linux ROOT diskette

From within the 'linux' directory you have just created on the C: drive, type:

```
>loadlin zimage mount root=/dev/fd0 rw
```

This command will load the Linux kernel 'zimage' and then proceed to mount the root file system on the diskette. When the prompt appears "VFS: Insert root floppy and press ENTER" put the ROOT diskette into the disk drive and press enter. Then at the Linux login prompt '#' login as root. You can now proceed to set up the Linux partitions as follows.

A.3 Step 3 Create the Linux Partitions

Now proceed as described in section 2.3 above, except that we shall not now be deleting the DOS partition. Run the Linux fdisk command by typing:

```
#fdisk /dev/hda
```

Use the 'p' command to show the existing partition. This should appear similar to the following:

	Boot	Begin	Start	End	Blocks	Id	System
/dev/hda1	*	1	1	13	4144	1	DOS 12-bit FAT

Now use the 'n' command to create two new Linux partitions (partitions 2 and 3), then the 't' command to change the type partition 3 to '82' (Linux swap). For an 80Mbyte device, specify e.g. +65M to create a 65M Linux native partition. On completion the 'p' command would (for a Lexar Compact Flash device) show:

	Boot	Begin	Start	End	Blocks	Id	System
/dev/hda1	*	1	1	13	4144	1	DOS 12-bit FAT
/dev/hda2		14	14	222	66880	83	Linux native
/dev/hda3		223	223	252	9600	82	Linux swap

Don't forget to use the 'w' command to write the new partition table to the disk, assuming you are happy with it.

A.4 Step 4: Create and activate the linux swap partition

NOTE: with 16MB RAM or more, this step may be skipped.

To create the linux swap partition type, at the shell prompt:

```
#mkswap -c /dev/hda3 9600
```

Where /dev/hda3 is the swap partition, and 9600 is the number of blocks in the swap partition (insert your own values, noted from the fdisk partition table shown above). After this command completes, activate the swap partition as follows:

```
#swapon /dev/hda3
```

A.5 Step 5: Install Linux from the CDROM

To run the menu driven Slackware installation tool type, at the shell prompt:

```
#setup
```

Now proceed as described in section 2.5 of this manual.

There will be two differences:

- Firstly, in the TARGET menu you will be asked if you would like to make the DOS partition accessible from Linux. Answer 'yes' to this, then specify /dev/hda1 as the name of the DOS partition, /dos_c as the 'mount point' within the Linux file structure, and then 'q' to complete this part of the setup.
- You must skip the LILO INSTALLATION menu, since you have elected to boot into DOS, and then load Linux from within DOS.

After shutting down Linux at the end of step 5, you will reboot back into the MSDOS C: partition. Then change directory into 'linux' and type 'linux' to run the batch file 'linux.bat' you created earlier in step A.1.

Providing we reach the following login prompt, then all is well:

```
Welcome to Linux 2.0.27.  
darkstar login: root  
Last login: Wed Jul 12 16:44:45 on tty1.  
Linux 2.0.27.  
you have mail.  
darkstar:~#_
```

Now login as root (there is no password yet). Then proceed to the next step in configuring the new system.

A.6 Step 6: Check swap partition is enabled

Proceed exactly as shown in section 2.6 (note, if you skipped step 4, then you can also skip this step).

Again, after shutting down Linux and rebooting the system you will need to manually load linux from the MSDOS C: partition by running the batch file 'linux.bat'.

A.7 Step 7: Wireless network installation and configuration

Proceed exactly as shown in section 2.7.

A.8 Step 8: Create a new user

Proceed exactly as shown in section 2.8.

Once again, after shutting down Linux and rebooting the system you will need to manually load linux from the MSDOS C: partition by running the batch file 'linux.bat'.

A.9 Step 9: Modify AUTOEXEC.BAT for dual boot

At this stage Linux installation and configuration is complete. You could now simply insert a call to 'linux.bat' as the final line in 'autoexec.bat', so that Linux will load automatically. However, a more elegant solution is to read the robot's status switch and then selectively boot into either MSDOS or Linux, depending on the state of the switch.

Hint: If you find yourself unable to edit autoexec.bat from within MSDOS, don't panic. You can edit autoexec.bat from within Linux. Simply change directory to /dos, which is where the MSDOS C: partition is mounted within the Linux directory tree. Then use the Linux editor 'vi' to make any changes to autoexec.bat.

First, we require a simple program to read and mask the status switches, then exit with the resulting value in an environment variable accessible to MSDOS. Here is an example of such a program, 'status.c'.

```
/* Amazingly simple program to read the RASCAL status switches, and
   return the value (0, 1, 2 or 3) to DOS, for processing by
   AUTOEXEC.BAT

   A.Winfield 18 May 1999
*/

#include <stdlib.h>
#include <dos.h>

#define STATUSPORT 0x224 /* status switches, bump sensors etc */

/* Set DOS ERRORLEVEL to 0,1,2 or 3 depending on status switches */
int main(void)
{
    exit( inportb(STATUSPORT) & 3 );
}
```

Compile this program (using, for instance, the Borland command line C compiler 'bcc'). Then call the compiled program ('status.exe') from with autoexec.bat, as follows.

```
echo Autoexec.bat for DOS/Linux Boot
echo reading RASCAL status switches...
status
```

```
if ERRORLEVEL 2 goto :linux
goto :dos

:linux
echo Starting Linux...
cd linux
linux.bat
goto :end

:dos
echo Starting DOS...

:end
```