

Installing & Configuring *GNU/Linux* in an *iPAQ 3950*

Juan Gutiérrez Plaza

20th March 2003

Contents

1	Configuring <i>iPAQ 3950</i>	2
1.1	Installation of <i>GNU/Linux</i> operating system	2
1.2	Configure Wireless LAN	4
1.3	Configure <i>iPAQ</i> for access to the Internet	6
1.4	Install AODV protocol	7
1.5	Configure sound in <i>iPAQ</i>	9

1 Configuring *iPAQ 3950*

In order to integrate our Ad Hoc protocol into the testing platform, we need to configure GNU/Linux system in the iPAQ 3950 [5] nodes. The system has to be configured for supporting Wireless LAN, audio and our *AODV AD Hoc implementation*. The following sections present a detailed sequence of steps for a successfully configuration of the nodes.

Basic steps are as follows:

1. Install Linux.
2. Configure wireless LAN.
3. Configure iPAQ for access to the Internet.
4. Install AODV protocol.
5. Configure sound in iPAQ.

1.1 Installation of *GNU/Linux* operating system

First of all, installing an operating system depends directly from hardware supported. In our case, the hardware is very specific and it puts a lot of restrictions in order to choose a Linux distribution.

This is the iPAQ hardware specifications:

- **CPU.** Intel XScale-PXA250 400 MHz revision 4.
- **RAM.** 64 MB.
- **ROM.** 32 MB Flash ROM.
- **Sound.** Philips UDA1380.
- **Wireless card.** PCMCIA D-Link DCF-660W.

With this hardware configuration, we chose the *Familiar* Linux distribution. We found this distribution, applications and documentation in [1].

Once chosen a distribution, the steps to install Linux are:

1. Connect the iPAQ via the USB cradle to Win machine.
2. Use the ActiveSync application to connect the iPAQ from the PC.
3. Copy boot blaster program (we chose it in [2]) to the default folder on the iPAQ from the Windows machine using drag and drop or cut and paste. Ignore any messages that say it *may need to convert* file formats.

4. Copy boot loader program (we chose it in [3]) to the default folder on the iPAQ from the Windows machine. Again, ignore any messages that say it *may need to convert* file formats.
5. On the iPAQ, find *BootBlaster3900.exe* and then execute it.
6. From the *Flash* menu, select *Save*. This will save a copy of the current bootloader to DRAM on the iPAQ (under the name *saved_bootldr.bin*).
7. Copy the *saved_bootldr.bin* off of the iPAQ and put it in a safe place in order to be able to restore it.
8. From the *Flash menu*, select *Save Windows gz*. This will copy and compress all the flash ROM on your iPAQ into a *.gz file* along with a file containing the asset information from your iPAQ. This will take a while. After it is complete, copy these files to the PC to save them. Under normal circumstances, installing Linux will not touch the asset partition in flash, but it is safer to have a backup copy.
9. From the *Flash* menu on BootBlaster, select *Program*. A file dialog will open allowing you to select the bootloader to use. Select the bootloader from step 5). This step can take a while.
10. From the *Flash* menu on BootBlaster, select *Verify*. If it does not say that you have a valid bootloader, do **NOT** reset your iPAQ. Instead, try programming the flash again. If that doesn't work, program your flash with your saved bootloader.
11. Configure the terminal emulator properly with:
 - 115200 baud.
 - 8N1.
 - No flow control.
 - No hardware handshaking.
12. To get to the bootldr's command line prompt (*boot*) and avoid booting Windows CE: depress and hold the center of the joypad while pushing the recessed reset button.
13. Unlock flash with this command: `pflash 0x40000 0xffff 0` by typing it at the boot prompt.
14. Reboot iPAQ again type `reboot` for example.
15. Type `partition reset`.
16. Type `load root`

17. Send the file system image using *Xmodem* with the terminal emulator. We have used [4] file system image. This file contains *2.4.19-rmk4-pxa2-hh8* kernel which has all modules necessary for configure sound and wLAN (This step consumes 1 hour approximately).
18. Type `boot`.

Now, iPAQ has a Linux operating system running.

1.2 Configure Wireless LAN

Wireless configuration is not very hard following those steps:

1. Find out the pcmcia module with the command `cardctl ident`. In our case, we obtained the following results:

```
Socket 0:
product info: "D-Link", "DCF-660W", ""
manfid: 0xd601, 0x0005
function: 6 (network)
```

2. Add the following lines to `/etc/pcmcia/config`:

```
card "D-Link DCF-660W"
    manfid 0xd601, 0x0005
    bind "orinoco_cs"
```

3. Edit `/etc/pcmcia/network.opts` as the following example:

```
# Network adapter configuration
#
# The address format is "scheme,socket,instance,hwaddr".
#
# Note: the "network address" here is NOT the same as the IP
# address.
# See the Networking HOWTO. In short, the network address is
# the IP address masked by the netmask.
#
case "$ADDRESS" in
*,*,*,*)
INFO="Sample private network setup"
#$ Transceiver selection, for some cards -- see 'man ifport'
IF_PORT="10base2"
```

```

# Use BOOTP (via /sbin/bootpc, or /sbin/pump)? [y/n]
BOOTP="n"
#Use DHCP(via /sbin/dhcpd,/sbin/dhclient,or /sbin/pump)?[y/n]
#DHCP="y"
#PUMP='n'
#If you need to explicitly specify a hostname for DHCP requests
#DHCP_HOSTNAME=""
#Host's IP address, netmask, network address, broadcast address
IPADDR="10.0.0.3"
NETMASK="255.255.255.255"
NETWORK="10.0.0.0"
BROADCAST="255.255.255.255"
# Gateway address for static routing
GATEWAY="10.0.0.1"
# Things to add to /etc/resolv.conf for this interface
DOMAIN="netlab.org"
SEARCH=""
DNS_1=""
DNS_2=""
DNS_3=""
# NFS mounts, should be listed in /etc/fstab
MOUNTS=""
# If you need to override the interface's MTU...
MTU=""
# For IPX interfaces, the frame type and network number
IPX_FRAME=""
IPX_NETNUM=""
# Extra stuff to do after setting up the interface
start_fn () { return; }
# Extra stuff to do before shutting down the interface
stop_fn () { return; }
# Card eject policy options
NO_CHECK=n
NO_FUSER=n
;;
esac

```

4. Edit '/etc/pcmcia/wireless.opts' as following:

```

case "$ADDRESS" in

# $NOTE: Remove the following six lines to activate the samples below...
# ----- START SECTION TO REMOVE -----

```

```
*,*,*,*)
ESSID="netlab"
MODE="ad-hoc"
;;
# ----- END SECTION TO REMOVE -----
```

1.3 Configure iPAQ for access to the Internet

There are two different ways of configuring iPAQ for accessing to the Internet, by a *ppp* connection or by a *USB* connection.

iPAQ 3950 has many problems with USB connection so it is easier configuring *ppp* connection.

Steps for configuring *ppp* are as follows:

1. Make sure that `/etc/passwd` contains a line like.

```
ppp::101:101:PPP User:/home/ppp:/sbin/pppd
```

2. Create or modify `/etc/ppp/options` as follows.

```
-detach
defaultroute
noauth
local
nocrtscts
lock
lcp-echo-interval 5
lcp-echo-failure 3
/dev/tts/0
115200
```

3. Make sure `/etc/modules.conf` has the appropriate aliases.

```
alias /dev/ppp          ppp_generic
```

4. Make sure `/usr/sbin/pppd` is executable by user *ppp*.

```
chmod 4755 /usr/sbin/pppd
```

5. Load the following modules.

```
insmod slhc.o
insmod ppp_generic.o
insmod ppp_async.o
```

iPAQ is now configured. To access iPAQ from desktop just type this command:

```
pppd /dev/ttyS1 115200 192.168.0.1:192.168.0.2 debug nodetach
local noauth nocrtscts lock user ppp connect "/usr/sbin/chat -v
-t3 ogin--ogin: ppp"
```

Establish a connection with iPAQ can require several attempts.
To access to iNET is necessary copy `/etc/resolv.conf`.

```
scp /etc/resolv.conf root@192.168.0.2:/etc/resolv.conf
```

Finally, *masquerading* must be available for communicating iPAQ with outside world.

```
iptables --flush
iptables --table nat --flush
iptables --delete-chain
iptables --table nat --delete-chain
iptables --table nat --append POSTROUTING --out-interface eth0 -j
MAQUERADE
iptables --append FORWARD --in-interface ppp0 -j ACCEPT
```

Now iPAQ is able to surf in the Internet.

1.4 Install AODV protocol

For installing AODV we need a cross compiler and a compile kernel. We have used *ToolChain cross compiler* [6] in order to compile kernel and aodv code for iPAQ.

This version of ToolChain is the one (called *monmotha*) that can compile code for xscale processor. It must be installed in `/opt/arm`.

We have used *2.4.19-rmk4-pxa2-hh8* kernel source code. We have used CVS environment to download. These are the steps:

1. `export CVSROOT=:pserver:anoncvs@handhelds.org:/cvs`
2. `cvs login`. Use `anoncvs` when password was asked.
3. `cvs checkout -r K2-4-19-rmk4-pxa2-hh8`. Do this in at `/opt/src`.

When we have downloaded kernel source and installed cross compiler, we can compile kernel for iPAQ.

We are now at `/opt/src/linux/kernel` and we have to follow next steps:

1. Edit configure file at `./arch/arm/def-configs/h3900` and enable netfilter module by `NF_QUEUE=m`.
2. `make h3900_config`
3. `make oldconfig`
4. `make dep`
5. `make zImage`
6. `modules`
7. `mv System.map System.map.orig`
8. `mv scripts/ipkg-make-kernel-packages scripts/ipkg-make-kernel-packages.orig`
9. Download a new version of this script from [7].
10. Download a *ipkg-build* from [8].
11. `mkdir ipkgs; cd ipkgs`
12. `../scripts/ipkg-make-kernel-packages $PWD/..`
`2.4.19-rmk4-pxa2-hh8`
13. `ln -s /opt/src/linux/kernel/include/asm`
`/opt/arm/arm-linux/include/asm`
14. `ln -s /opt/src/linux/kernel/include/linux`
`/opt/arm/arm-linux/include/linux`

We had some problems with *i2c* devices, so, before `make dep` we have done `make xconfig` and we have disabled these devices.

Now we have a compiled kernel and its modules and we can compile other programs.

In order to install `aodv`, we need to install the new *ip_tables* module which we have compiled previously. One of the all possibles ways to install is as follows:

1. We guess that PC and iPAQ is already connected by a PPP conection.
2. `scp iptables-modules-2.4.19-rmk4-pxa2-hh8__arm.ipk`
`root@192.168.0.2:/`

3. `ssh root@192.168.0.2`
4. `cd /ipkg`
5. `install iptables-modules-2.4.19-rmk4-pxa2-hh8__arm.ipk
-force-depends`
6. `modprobe ip_filter`

Now we can compile and install properly aadv code. In this iPAQ we had to do some modifications in aadv original code.

1. Update this line at *Rules.make*:
`KINC_ARM=-I/opt/src/linux/kernel/include.`
2. Add this line at *Rules.make*:
`INCLUDE=-I/opt/arm/arm-linux/include.`
3. Comment structure *in_pktinfo* sited at file *aadv_socket.c*.
4. `make arm`

Aadv is already for using in iPAQ, the installation process is as follows:

1. We guess that PC and iPAQ is already connected by a PPP conection.
2. We guess that we are at main directory of aadv.
3. `scp kernel/ipq-arm.o root@192.168.0.2:/lib/modules
/2.4.19-rmk4-pxa2-hh8/kernel/net/ipv4/ipq.o`
4. `scp kernel/r1-arm.o root@192.168.0.2:/lib/modules/
2.4.19-rmk4-pxa2-hh8/kernel/net/ipv4/r1.o`
5. `scp uu-daemon/aadvd-arm root@192.168.0.2:~`
6. `scp uu-daemon/aadv.sh root@192.168.0.2:~`

Now aadv is already for using.

1.5 Configure sound in iPAQ

It is easy configure sound in iPAQ, although order in loading modules is very important because iPAQ could be frozen.

This is the correct order:

1. `insmod uda1380.o`
2. `insmod h3900-uda1380.o`

Sound is already enabled.

References

- [1] <http://www.handhelds.org>
- [2] <http://handhelds.org/feeds/BootBlaster3900/BootBlaster3900.exe>
- [3] <http://handhelds.org/feeds/bootldr/pxa/bootldr-2.19.57.bin>
- [4] <http://handhelds.org/pb/unstable/bootopie-pb6a-h3900.jffs2>
- [5] <http://h20022.www2.hp.com/busprod/overview/0,12512,series=96474%5Etype=64929%5Ecategory=215383,00.html?sidebarLayId=106&rsidebarLayId=63>
- [6] <ftp://ftp.handhelds.org/pub/linux/arm/toolchain/monmotha>
- [7] <http://www.tct.hut.fi/~xlei/ipaq/download/ipkg-make-kernel-packages>
- [8] <http://www.tct.hut.fi/~xlei/ipaq/download/ipkg-build>