

## LBOX – Internet Connectivity, nfs, etc.

This describes my experiences on a quick first attempt, without checking out alternative possibilities.

### 1. Configure the ethernet interface, eth0

I'll assume here that you are connected from your workstation to the LBOX via minicom or the like through the serial port. The bootup messages suggest how to establish internet connectivity to the LBOX. So, with a live ethernet cable connected to the ethernet connector closest to the LBOX board edge, I checked out the network interface from my minicom terminal window via command line entry

```
ifconfig
```

and found that eth0 was not yet configured. I am using dhcp so, following the suggested command in the bootup messages, I entered

```
dhcpcd -p -a eth0 &
```

A subsequent check with ifconfig showed that eth0 was now configured and gave the IP address of the LBOX. We'll use 'LBOX\_IP\_address' to represent this number in the sections that follow.

### 2. Telnet into the LBOX

The LBOX provides telnet, but not ssh. So I activated telnet on my workstation, an exercise not described here, being somewhat distribution specific. I could then telnet into the LBOX from my workstation using a terminal window other than that provided by minicom, e.g.

```
telnet LBOX_IP_address
```

So I now have two ways to talk to the LBOX from my workstation:

- through the serial port using the minicom terminal
- through ethernet using the telnet program from any other workstation terminal window

### 3. The LBOX's resident web site

I can also access the LBOX's resident web site with a browser by pointing the browser at

```
http://LBOX_IP_address
```

The web site is pretty cute and allows you to turn a led on the LBOX on or off.

#### 4. Mounting a workstation directory on the LBOX via nfs

Storage space is quite limited on the LBOX, so nfs mounting a directory from the workstation onto the LBOX is very attractive. For example, with the cross development tools on the workstation, we can develop LBOX applications in that directory i.e. not only compiling, but also executing/testing them.

The LBOX is setup to act as an nfs client, so I configured the Linux workstation to act as an nfs server. Before describing the details I'll mention that an attempt to nfs mount the workstation directory on the LBOX file system failed when the workstation and LBOX were on different subnets, but succeeded when they were on the same subnet. Presumably the failure case didn't succeed because there was an intervening firewall which forbade nfs traffic. I am not the sys admin for this setup, so I didn't investigate further. Working on the same subnet is probably the typical situation during development.

To prepare the workstation for this nfs capability, you must have configured the kernel to have nfs server capabilities. Reconfiguration and recompilation of the kernel is beyond the scope of this note, but is widely available information – Google is your friend, as needed.

Let's say that we have a directory on our workstation that is to be nfs mounted on the LBOX. Let's say it is /home/my\_user\_name/lbox\_stuff. We identify this to the server by adding an entry to the workstation file '/etc/exports' of form:

```
/home/my_user_name/lbox_stuff (ro,insecure)
```

Note that spaces in the parenthetical options are not tolerated. We must restart the nfs server to make it aware of this. On my debian based workstation, I do this as root by entering:

```
/etc/init.d/nfs-kernel-server restart
```

Now the workstation is set to go.

Next we need to use the mount command from within the LBOX system. We can use the workstation's minicom terminal for this or telnet to the LBOX. Before I use the mount command I might make a mount point e.g.

```
mkdir /var/nfs_dir
```

Next, enter the mount command:

```
mount -t nfs -o nolock workstation_IP:/home/my_user_name/lbox_stuff /var/nfs_dir
```

where 'workstation\_IP' is the workstation's IP address.