

LOC

Contents

Building development environment through evaluation board

Practice Test

Transmitting executable file created in Host PC to LOC

Flash file system

Converting to monitor mode

Adjusting characteristics of root directory

Console/ shell enable/disable methods

IP setting methods

Directory structure

Precautions for network programming

Configuration Diagram

Building development environment through evaluation board

- Connect RS-232C port in the host PC where the terminal emulate program has been installed with RS-232C port in LOC.
- An emulator program called “minicom” is that you can easily find for Linux. It is provided with most distributed versions.
- When you first activate minicom, enter **#minicom -s**, a dialog box will appear.
- In the Serial port setup menu, set as follows:

serial device :/dev/ttyS0
Bps/par/bits :9600 8N1
Hardware flow control :No
Software flow control :No

For 9600 bps,8 bit data,1 stop bit, and flow control, set as “not used” .

In case of using COM1 in the host PC, set as /dev/ttyS1 for /dev/ttyS0,COM2

- To change the setting in minicom, press ctrl+a and z to present a dialog box.
- After the minicom setting is completed, supply the power for the evaluation board. Booting procedure will be seen on the minicom consol and then log in will be prompted.
- The Log in ID is **root** and the password is **uClinux**.
- Now you enter into LOC. There are built-in basic Linux commands, so try them to check some functions.

Practice Test

Let me explain the way to print "Hello!LinuxOnChip." on the console screen. See how make a directory, "sample" under the /home/LOC, a working directory in the host PC and create code as a trial.

```
#pwd
/home/LOC
#mkdir sample
#cd sample
#buildenv
```

On completion, make a file, hello.c as follows:

```
main(int argc,char *argv[ ])
{
puts("Hello!LinuxOnChip.");
}
```

You need to compile. Activate it with a compile option as below:

```
#m68k-pic-coff-gcc hello.c -o hello
m68k-pic-coff-gcc :compiler executable file
hello.c :test program source
-o :compiler option
hello : executable file created
```

Transmitting executable file created in Host PC to LOC

- Transmit the executable file to an LOC target board using Network File System (NFS).
- You need to mount the working directory of host PC on LOC and set a certain directory in host PC to be used like local directory by LOC.
- To set the NFS, you need to have /etc/exports file in host PC, adjust the content of /etc/hosts file and stop/ start nfs daemon.
- As we assumed that /home/LOC directory in host PC is a working directory, set as below:

- Using an editor such as vi to add **/home/LOC (rw,insecure)** within **/etc/exports** file in host PC.

```
#cd /etc  
#vi exports  
/home/LOC (rw,insecure)  
#
```

- Then, adjust **/etc/hosts** file in host pc. Add **192.168.1.200 LOC** to **/etc/hosts** file.

```
#cd /etc  
#vi hosts  
192.168.1.200 LOC  
#
```

The IP of LOC has a factory setting, 192.168.1.200. Please refer to 'IP setting methods' to change it.

- If you have changed the **/etc/exports,/etc/hosts** file, activate as follows to make the change is in effect.

```
#/etc/rc.d/init.d/nfs stop  
#/etc/rc.d/init.d/nfs start
```

- Now, you need to mount a working directory in host PC by operating LOC and also bring an executable file into LOC. From here you can give command to LOC through the console connected to minicom and other equipment.

```
#mount -t nfs 192.168.1.7:/home/LOC /mnt  
#mount  
/dev/root on / type umsdos (rw)
```

```
/dev/ram0 on /var type ext2 (rw)  
proc on /proc type proc (rw)  
192.168.1.7:/home/LOC on /mnt type nfs (rw,addr=192.168.1.7)  
/dev/vsbb on /usr type umsdos (rw)
```

On completion the above process, the LOC can use host PC as its local directory. The IP address for host PC is 192.168.1.7.

- Now, activate the executable file 'test' created just before by copying to LOC.

```
#cd /mnt  
#cp hello /usr/hello  
#sync
```

The executable file 'hello' is stored under LOC's /usr directory. If you enter as follows, the results will be presented as expected:

```
#pwd  
/usr  
#./hello  
Hello!LinuxOnChip.  
#
```

To activate the process whenever LOC restarts, added the below to the /usr/rc file

```
/usr/hello &
```