LOSA

[Contact]

- Research Institute, Linudix Co., Ltd.
- E- mail: linudix@linudix.com
- Telephone: 82-2-3482-0991~2
- FAX: 82-2-3482-0993
- Amendment Date:2001.7.24.3

[LOSA (Linux On Strong Arm)]

Component	Description
Operating system	Linux kernel 2.4.0-test4-np1
File system	Journaling flash file system
Protocol	TCP/IP, HTTP, FTP, TELNET, SNMP SMTP, PPP, DHCP, NFS, etc
GUI	X-window based GUI (optional 1)
Browser	(Option 1)
Web server	Boa web server
Ethernet	10M ethernet, direct to LAN ADSL(DHCP) ADSL(PPPoE) Cable modem (DHCP)
PPP	PPP for wired or wireless modem
Development env.	GCC arm compiler NFS GDB debugger

Option 1: It is available only with 16Mbyte Flash ROM installed.

Component	Description
CPU	Intel Strong ARM processor SA1110-206Mhz operation
ROM	8, 16Mbyte flash ROM
RAM	8, 16, 32, 64Mbyte SDRAM, PC100
LAN	10base-T
Serial	3 UART
Bus	Full address bus(A0 ~ A25) 32bit Data bus(D0 ~ D31)
LCD controller &	Direct operation of Mono & color LCD
GPIO	Total 39 GPIO pin.4

1. CPU

It has a built-in Intel' s Strong ARM operating in maximum 200MIPS. In order to configure an application utilizing LOSA, you need to see the " Strong ARM manual" .
You can find relevant materials at http://developer.intel.com/design/strong/.

2. Flash ROM

It stores Linux kernel and user programs. Kernel is provided as being mounted on Flash ROM when shipped. There are two kinds of Flash ROMs - 8 Mbytes and 16 Mbytes. With embedded file system, user program can be stored in.

3. DRAM

In the area, kernel images in the flash ROM are to be copied and operated after LOSA booting. With PC-100 SDRAM installed, it can process in high speed. With 32-bit memory bus, it can utilize fast SDRAM.
It is available in 8 Mbytes, 16 Mbytes, 32 Mbytes, and 64 Mbytes.

4. Ethernet controller

There is a built-in 10base-T Ethernet controller. A 10base-T port can be secured by installing external transformer and connector only. For the installing methods and recommendations, see the " Evaluation board'. The 10/100 Mbps enabled products are to be provided as an option.

5. Serial port

The three UART ports are provided. LOSA itself does not have a level converter. A RS-232C port can be secured by installing external level converter chip (e.g. MAX232, etc.) and connector only outside LOSA. For the installing methods, see the "Evaluation board" .
An UART can be used as a console port. If consol function is excluded, it can be used as a normal serial port. For further information, see "Console, shell enable/disable".
There is an UART port that can be used along with IrDA.

6. LCD Controller & 8 General I/O Port

There are 27 general I/O ports provided. These ports are to be used as a basic general I/O. Each pin has a dual function.

LCD controller related 12 pins are provided. When LCD controller is not being used, you can use it as normal I/O pins. When a 16-bit LCD is used, it shares pins of general I/O port.

For further information, see the chapter 11 of Intel Strong ARM advanced developer's manual.

---

7. 3.3V Operating Voltage

Most users may feel difficulties in interfacing with application circuit operated in 5V. In brief, the followings are connection methods between circuits of 3.3V and 5V.

- Output: The output is generated in TTL level, you can use TTL IC (74LSxx) at the destination 5V side.
- Input: Input pin is rated as 3.3V, so avoid to directly input 5V to it. An easy way is to use LVC or LVT series buffer, but if the cost is mattered, you can configure a voltage divider to solve the problem.
- Bi-directional port : It is relatively easy to handle either the input or the output only, compared with connecting a bi-directional pin such as data bus between power of 3.3V and 5V. It is recommendable to use LVT245 or equivalent products. For further information on 3.3V to 5V interfacing products and detailed explanation, you can find application note or product manual through website of Motorola or TI.

[Constructing Development Environment]

Users may do programming and compiling on a PC, developing equipment, and implement desirable functions by storing programs developed in LOSA, the target board. Therefore, it is required to have a CPU, the compiler of Strong ARM used in LOSA on the development PC. Also the PC should have Linux installed. It is enough with recently distributed Linux version (since June 2001).

1. Compiler Installation

You need to mount the CD provided on the Linux system in the development PC (or Host PC).
#mount -t iso9660 /dev/cdrom / mnt

Then create a working directory in the Host PC. If the working directory is /home/LOSA, the followings are the process required to install the compiler from the CD.
#cd /home
#mkdir LOSA
#cd / mnt/cdrom/tools
#./install

2. Compiler Manual

Linudix does not provide compiler related documents. You can find them at http:/www.xxx.org. For an introductory guide, enter **#man xxxx** at Host PC.

3. Building Development Environment through Evaluation Board

- Connect an RS-232C port of the host PC with installed terminal emulated program to an RS-232C port of LOSA. The serial port being uses as consol at LOSA is J2 connector of LOSA-EB.

- The emulator you can easily find for Linux is a program named minicom. It will be provided with most distributions.

- When you activated the minicom for the first time, enter **#minicom –s**, a dialog box will be prompted.
  Then, in the item of Serial port setup, set as follows:

  serial device : /dev/ttyS0
  Bps/par/bits : 115200 8N1
  Hardware flow control : No
  Software flow control : No

Set 115200 bps, 8 bit data, 1 stop bit, and flow control as disabled.
In case of using COM 1 of the host PC set as /dev/ttyS0, or as /dev/ttyS1 for COM 2.

- To adjust environment setting using minicom, press ctrl+a and z to present a dialog box.

- On completion of minicom setting, connect the power to the evaluation board. Then, you can see the booting procedure on the minicom consol and finally a log in prompt will be shown.

- Log in ID is **root** and the password is **linuxonchip** .

- Now, you are logged in LOSA. There are basic Linux commands built-in, test it while using some of its functions.

4. Practice Test

Let us print "Hello! LinuxOnChip." on the console screen. First, create a directory called "sample" under the host PC working directory, /home/LOSA to write a trial code as follows:

#pwd

/home/LOSA

#mkdir sample

#cd sample

#buildenv

On completing the above, create a file named hello.c as below.

```
#include <stdio.h>
main(int argc, char *argv[])
{
puts("Hello! LinuxOnChip.");
}
```

You need to compile by assigning a compile option as follows:

#arm-linux-gcc hello.c -o hello

arm-linux-gcc : compiler executable file

hello.c : test program source

-o : compiler option

hello : binary file executable in the LOSA created.

5. Transmitting Executable File Created in the Host PC to LOSA

- Transmit the executable file to LOSA target board through Network File System (NFS).
- Mount the working directory of the host PC on LOSA, then, LOSA can use a certain host PC directory as its local directory.
- For NFS setting, you need to make adjustments for /etc/exports and /etc/hosts files of the host PC as well as process stop and start operation of nfs daemon.

- As the home/LOSA directory of the host PC is assumed as a working directory, proceed to the followings:
- Using an editor such as vi, add **home/LOSA (rw,insecure)** within **/etc/exports** file.

  #cd /etc

  #vi exports

  /home/LOSA (rw,insecure)

  #
- And then, adjust **/etc/hosts** file of the host PC. Add **192.168.1.200 LOSA**to the /etc/hosts file.

  #cd /etc

  #vi hosts

  192.168.1.200 LOSA

  #

The IP of LOSA is set as 192.168.1.200 on shipment. For the change of IP, see the 'netcofig'.

- When the adjustment of /etc/exports, /etc/hosts file is completed, activate the followings to turn the new setting be in effect.

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

  #/etc/rc.d/ init.d/nfs start
- Then, you need to mount the working directory of the host PC on LOSA to bring an executable file. You need to give orders for LOSA using the console having connection with Minicom, hereafter.

  #mount -t nfs 192.168.1.2:/home2/LOSA / mnt.

#mount
/dev/ram on / type ext2 (rw)
none on /proc type proc (rw)
192.168.1.2:/home2/LOSA on / mnt type nfs (rw,addr=192.168.1.2)

On completion as above, LOSA now can use the host PC as its local directory. The IP address of the host PC is 192.168.1.2.

- Now, activate the executable file ' hello' created by copying it into LOSA.

This way is using a command named ' cp' , so it is not possible to simply write on the Flash. Therefore, you need to add the file you want to RAM disk image, which is provided as a standard package in the CD for LOSA in order to make a new RAM disk image.

* See appendix [Creating RAM Disk Image]

Then, the executable file ' hello' is stored under the /usr/bin directory of LOSA.
By entering as follows, you can get the expected results printed:

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

Add /usr/bin/hello & to /etc/rc.d/rc.local file in order to arrange it to be run whenever the LOSA starts.

[DC characteristics]

- Supply voltage: min. 3.0V, max. 3.6V
- Input voltage: min. 3.0V, max. 3.6V
- Operating temp.: min. 0°C, max. 70°C
- Power rating: max. 0.5W

Power voltage supplies must maintain 3.3V. Otherwise the results are not predictable as the input voltage for the chip used is 3.3V.

[Memory map]

Area	Address range
60 | nCS1	0xd4000000 (16 MB)
61 | nCS2	0xd5000000 (16 MB)
62 | nCS3	0xd6000000 (16 MB)
63 | nCS4	0xd7000000 (8 MB)

[Flash file system]

You can create, delete, move, copy and adjust files as the same was as using a hard disk because LOSA has a file system embedded in the Flash memory.
By keying in the command, "mount", you can see the following results appeared.

 # mount
/dev/ram on / type ext2 (rw)
none on /proc type proc (rw)

If a working directory of development PC is mounted using the Network File System (NFS), the followings will present and you can see the /home2 directory of the 192.168.1.2 host is mounted.

 # mount
/dev/ram on / type ext2 (rw)
none on /proc type proc (rw)
192.168.1.2:/home2 on / mnt type nfs (rw,addr=192.168.1.2)

LOSA now can use the 192.168.1.2:/home2 directory as if it is own local directory.
When the programs developed with the development PC are mounted immediately after compiled, the activation and review can be faster and more efficiently done at LOSA.
If you want to store completed files in the Flash disk, undo the current RAM disk image at /dev/ram of the PC using Linux OS and mount /dev/ram on /mnt in order to add or change.
After completion, you can create a new RAM disk image and apply it to LOSA.
For further information, see [(Appendix) Creating RAM Disk Image].

[Converting into Monitor mode]

It is to make LOSA enter into monitor mode. While booting of LOSA is being progressed, if it sends ESC code to the console port 3 times or more, stop booting and enter into the monitor mode.

1. Press and hold ESC on the minicom, and
2. press the reset button on the LOSA.
3. If the prompt 'mon>' appears, then you are in the monitor mode.

Enter 'help' at the prompt; a command usage guide for the monitor mode will be presented as

follows:

Another Linux loader for SA-1110 (c) 2000 LinuxOnChip.com

press <esc> for entering monitor mode (1 seconds)

unprotect sector (kernel/ramdisk)

.....done.

mon>

mon>help

rx - enter xmodem recv mode

md - memory dispmay

mm - enter memory modify mode

in the mm mode

1) just enter - goto next addr

2) enter whith . - exit mm mode

3) enter whith data - modify data

env - set environment variable

ex) env a b (set environ a as b)

speed - change serial speed

ex) speed 38400 (avail speed: 9600/19200/38400/115200)

program kernel - program loaded kernel image to flash

program ramdisk - program loaded ramdisk image to flash

help - show this message

load - exit monitor mode, load Linux

If the ' unprotect sector' does not appear, the boot loader is not a current version. You need to update it to write kernel and RAM disk image in the monitor mode.

IP Setting Methods

To change IP environment allocated to the LOSA, you can use a netconfig command installed in for easy-to-use, rather than change initializing file (rc file), which is also available.

To complete IP setting, connect LOSA using minicom emulator or telnet to run netconfig on the prompt and set an accurate value.

bash-2.01# netconfig
network configuration
IPaddrs: 192.168.1.200
Netmask: 255.255.255.0
Gateway: 192.168.1.1
NameSer: 192.168.1.200
hostname: LOSA
ip=192.168.1.200
netmask=255.255.255.0
gateway=192.168.1.1
nameserver=192.168.1.200
hostname=LOSA
correct? y
update network environmnt
update boot parameters
unprotect sector 3
protect sector 3
done
bash-2.01#

For **Ipaddrs**, enter the IP address assigned by the network administrator. **netmask** and **network** are related each other. Set according to your network configuration. For further information for setting, you can refer to most of Linux manuals.

For **gateway** and **nameserver**, enter the assigned values.

For **host name**, you can enter whatever you want.

Evaluation Board Specifications

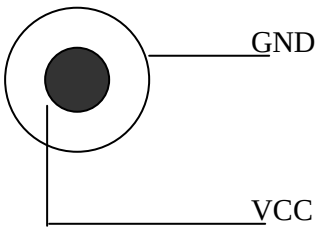
This board can be used for testing basic functions of LOSA. Its standard functions provided are as follows:

Component Description

Serial port	An RS-232C port (RX, TX only) A modem port
Ethernet 1	10base-T Ethernet port
LCD port	5(R),6(G),5(B) color LCD port
	LCD Inverter port
Audio	Optional
TV out	Optional
GPIO	8 LED out
	(can be used as general output)
	8 S/W input
	(can be used as general input)
Touch panel interface	Analog type interface, optional

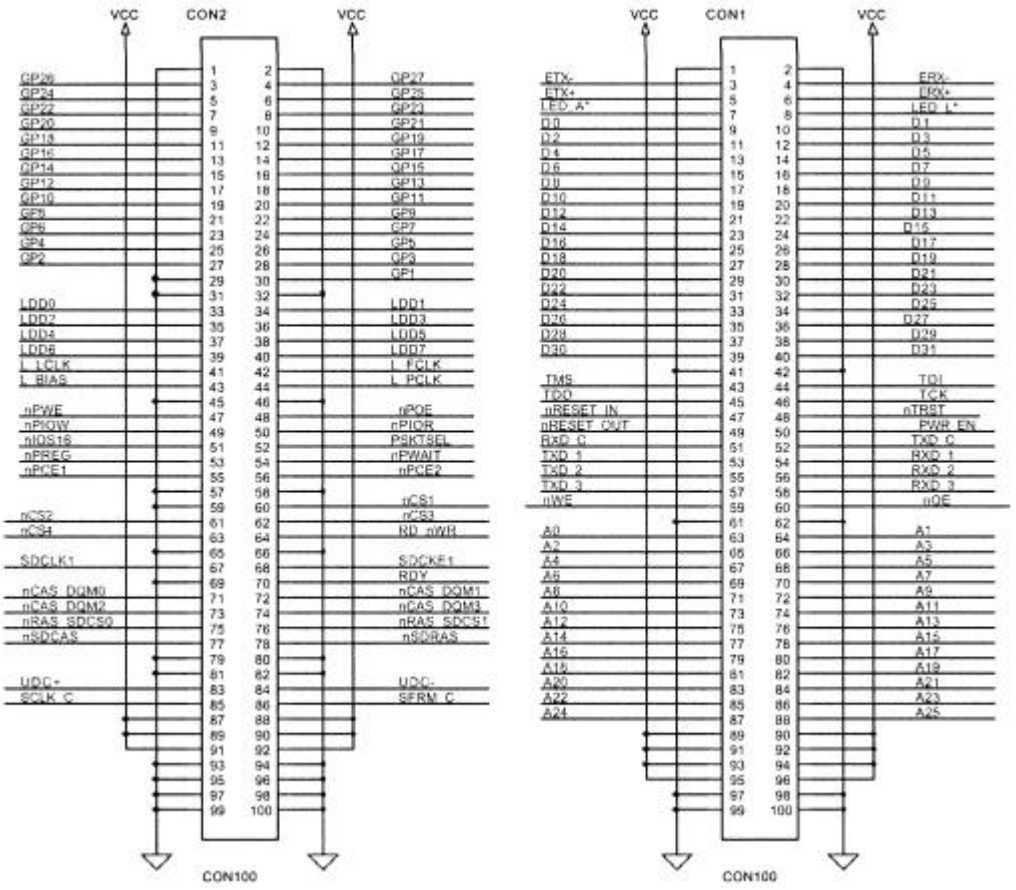
[Figure]

- Input Voltage: DC 9 ~ 12V



- For making enquiries on purchasing information for parts used in the evaluation board, send e-mail to linudix@linuxonchip.com.

Appendix 1: Pin Allocation and Description



The pin names used for LOSA are as the above figure. If you see the LOSA module from the top, the location of con1 and con2 of the connector is in order as shown in the figure with the distance of 1600 mil between them.

The connector used is KEL 8901 series. If you want to use a different pin that not applied to the evaluation board, see the spec sheet for 8901 that you can find at www.kel-system.co.jp.

Every pin has the name corresponding to the Strong ARM processor and is matched in 1:1.

* 100 pin connector specifications

- KEL CORPORATION
  - 8900 series
  - (1) 8901-100-177s (Receptacle) which are used in the LOSA EB.
  - (2) 8911-100-178s (Plug) which are used in the LOSA CPU module.
- Distance between PCB con1 and con2: 1600 mil

Appendix 2: Creating RAM Disk

Upon completing development using the NFS mount, you need to create new disk images including application programs and libraries required. The disk image will be compressed in gzip and stored in a flash memory of LOSA through using an equipped tool in LOSA, named flashloader. To create, you can use either a RAM disk or loop device. Here we are looking at the way to use RAM disk to create a disk image. The work must be processed in a development host with Linux system. For you information, this process is assumed to be underway using redhat 5.2. Redhat 5.2 and higher will be appropriate.

1. Preparations

1) A standard disk image file for LOSA (8 Mbyte disk image) with file name, **ramdisk8M.gz**

- The file is included in the CDROM for LOSA development .
- The image in the CD is an old version, so you need to update some commands.

(Commands to be updated are flashloader, netconfig, and envconfig under /usr/bin that must be updated. If requested, we will send you the files by e-mail immediately.)

2) The Linux being used as a development host much support ramdist with maximum ramdisk size as the same with the disk image intended to create or more. (In this example, 8 Mbytes)

- Execute the command, # **cat /proc/devices** and see the followings are presented:

Character devices:
1 mem
2 pty
3 …..
Block devices:
1.ramdisk means that RAM disk is supported.
2.fd
3. …..

Most of distributed versions will support ramdisk device as a standard, but if ' ramdisk' is not registered in you block device item, you need to do kernel re-compiling. In this case, please ask a technical support.

3) To create a disk image, you can use following tools.

dd - disk dump utility

zcat - uncompress and cat

mount gzip

2. RAM disk reconfiguration (size adjustment)

1) Check maximum RAM disk capacity available in your PC.

- By running #dmesg | grep RAM, you can check your maximum available RAM disk capacity .

…Ramdisk driver initialized : 16 ramdisks of xxxxK size

If the figures in xxxx is smaller than 8192, you need to reconfigure the RAM disk.

In this case, we are going to proceed with creating 3 Mbyte RAM disk. If you have smaller RAM disk than 8M as a result of (2) (e.g. 4096 Kbytes), you can follow the below steps to increase the size.

2) PC RAM disk capacity adjustment

Open the /etc/lilo.conf file using the editor and add option line (arrow-marked) as below:

The following lilo.conf is seen generally, but your lilo.conf file can be slightly different with that.

 # vi /etc/lilo.conf

boot=/dev/ had

map=/boot/map

install=/boot/ boot.b

image=/boot/vmlinuz-2.0.36-lrp-flash

label=linux

ramdisk =16384 (RAM disk size assigning option)

root=/dev/hda5

read-only

append="mem=64m ether=0,0,eth1"

#lilo

Added linux

#

Now, reboot your computer and recheck the maximum size of your RAM disk adjusted referring to the " Preparation (2)" .

3. Creating disk image

You can make a new disk image from the scratch, but it is recommended to use an existing disk image and add required executable files and libraries to it.

1) Mounting existing disk image
You can mount current disk on the development host as follows:

 # mkdir / mnt/ramdisk
 # zcat ramdisk8M.gz > /dev/ram
 # mount /dev/ram / mnt/ramdisk
#

Now, the disk image that can be seen currently while LOSA is on booting mode is mounted at /mnt/ramdisk.
If you add an application **hello** and a library **libmylib.so**, for example, the application will be stored in /usr/bin and library is in /lib (both paths are directories seen at the LOSA side.), and it will be /mnt/ramdisk/usr/bin, / mnt/ramdisk/lib each for the development host.

 # cp hello / mnt/ramdisk/usr/bin
 # cp libmylib.so /mnt/ramdisk/lib

Confirm whether they are correctly copied again and go to the next step.

 # ls / mnt/ramdisk/usr/bin/hello
 # ls / mnt/ramdisk/lib/ libmylib.so

In addition to application and library, setup files such as startup script also can be adjusted to fit to each application. For example, if you want to activate a program hello when booting LOSA, add an item /usr/bin/hello to the end of /mnt/ramdisk/etc/rc.d/rc.local, which is a startup script.
If required process is completed, you need to make a compressed image as follows:

 # umount / mnt/ramdisk
 # dd if=/dev/ram of=ramdisk8Mnew bs=1k count=8192.

 # gzip -9 ramdisk8Mnew

 # ls -l

ramdisk8Mnew.gz

#

This RAM disk image created can be stored and used again for the next adjustment.
To mount the already compressed ramdisk8Mnew.gz file to the flash in LOSA, reboot LOSA and log in, and copy the ramdisk8Mnew.gz file to the NFS shared directory in the development host. Then, do NFS mount the disk in the development host on the LOSA. (e.g., mounting at /mnt directory)

 # mount -t nfs 192.168.1.2:/home2 /mnt

If the mounting is completed normally, update the disk image as below:

 # flashloader ramdisk /mnt/ramdisk8Mnew.gz

-..

#

Now, if you reboot LOSA, it will be started with the new disk including hello and libmylib.so.
You can also download the new disk image in monitor mode through x-modem to use. However, the flash should be turned to unprotect to write the ramdisk image on the flash in monitor mode. If a massage, "unprotect sector" is not shown when you are converting LOSA into monitor mode, you have an old version boot loader image, which is needed to update. If you changed it to a new boot loader image, it would do unprotect process automatically whenever monitor mode is be turned on.
If you want to change to new boot load image, request it to us by e-mail.

4. Creating larger disk image than 8M
You may need to recreate disk image in order to increase RAM disk availability to help creating temporary files after booting, or cannot store applications and libraries in the standard 8 Mbytes disk. Assuming this, here is the way to create any size of disk image (e.g. 16 Mbytes). The process is unexpectedly easy to follow with using 2 RAM disks.

1) Undo the standard disk at ramdisk and mount.

 # zcat ramdisk8M.gz > /dev/ram
 # mount /dev/ram / mnt/ramdisk

2) Create an 16 Mbytes empty disk at ramdisk and mount.

 # mkdir / mnt/ramdisknew
 # dd if=/dev/zero of=/dev/ram2 bs=1k count=16384
 # mke2fs -m 0 /dev/ram2 16384
 # mount /dev/ram2 /mnt/ramdisknew

3) Now copy the standard disk as follows. Do not forget –d option in cp. (an option to copy a symbolic link as it is)

 # cp -Rd /mnt/ramdisk/* /mnt/ramdisknew

4) You can copy or adjust any applications required or files desired to add to or at the new disk sized 16 Mbytes (/mnt/ramdisk2)

5) Create a compressed image as follows:

 # umount / mnt/ramdisk
 # umount / mnt/ramdisknew
 # dd if=/dev/ram2 of=ramdisk16M bs=1k count=16384
 # gzip -9 ramdisk16M
#

6) Write down the size of newly created ramdisk16M.gz file.

 # ls -l ramdisk16M.gz

7) Before rebooting, you need to do another process described in the (8) to do write the new 16 Mbyte disk at the flash in LOSA.

 # flashloader ramdisk /mnt/ramdisk16M.gz.

8) You need to assign the size of ramdisk in the environmental parameter using the setenv command in monitor mode on the LOSA.

(1) Enter the LOSA into the monitor mode. (Reset LOSA while pressing the ESC key on minicom.)
(2) Record the size of Initial ramdisk (in K bytes) in the parameter, initrd_size.
mon> setenv initrd_size xxxxxxxx
mon>
(3) In the xxxxxxxx, you need to record a number calculated as follows: divide the size written in (6) by 1024 and raise decimals to the next whole number, and then record in 8 digit number by converting to be a hex value. For example, the size of ramdisk16M.gz is 2235474, first divide it by 1024 and raise the decimals to be 2184 and convert it in hex to get the result - 0x00000888. Therefore it is likely to be done accurately as follows:
(4) **mon> setenv initrd_size 00000888**
You do not need to stick to the rules but at least to enter a higher valued that its actual size for initrd_size in Kbyte. So, you can put **mon> setenv initrd_size 00001000** instead.

9) Next, you also need to assign the ramdisk size in the command line argument, which forwards data to kernel in monitor mode, so that the LOSA kernel will secure corresponding RAM disk size. First write down current command line element.

mon>setenv
……
cmdline mem=32m ramdisk_size=8192
……

You can enter current cmdline parameters as the same except of ramdisk_size. In the above case, mem=32m means that the standard LOSA has 32 Mbyte SDRAM. If there are any other options, let them be as they are. If the image is sized as 16 Mbytes, you can adjust the cmdline using ramdisk_size=16384 rather than ramdisk_size=8192.

mon> setenv cmdline mem=32m ramdisk_size=16384.

mon>

10) Now, restart the LOSA. If every setting is correctly done, the LOSA will have 16 Mbyte disk capacities. You can check it by using df command as follows:

 # df
Filesystem 1024-blocs Used Available Capacity Mounted on
/dev/ram 16384 xxx xxx xxx /

Appendix 3 : Evaluation board circuitry

- For the development board circuitry, please visit our website and download to see it.
- As the circuitry stored in CD cannot be clearly seen, please visit our website and download to see it.