Part 2

Embedded Linux

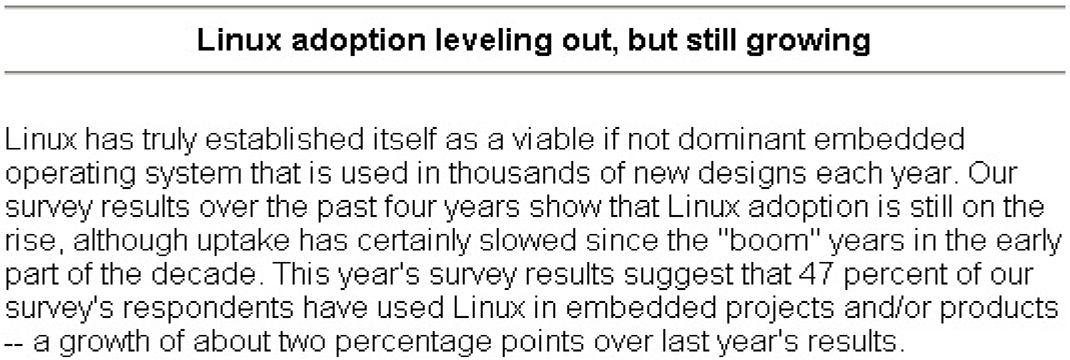
By : H K Sim
(Contact Email : simhkeng@yahoo.com)
March 2007

7.0 Embedded Linux Platform On RISC Processor

Why use embedded Linux

- Embedded application is getting more complex and thus need more powerful operating system
- Free license O.S. is needed to reduce product cost. Network equipments such as Switches, routers , wireless access points, cellular phones, automation server, … uses embedded Linux. Across all markets and technologies
- Ethernet network port (IP enable device) required (instead just Serial port)
- More external data storage required
- mature , high performance, large applications, networking protocol available
- large number of active developers sharing their experiences
- Good for research activities. Able to add/modify Operating System at kernel and driver level to achieve performance requirement (OS source code is available)
- royalty free

Embedded OS & Market Share

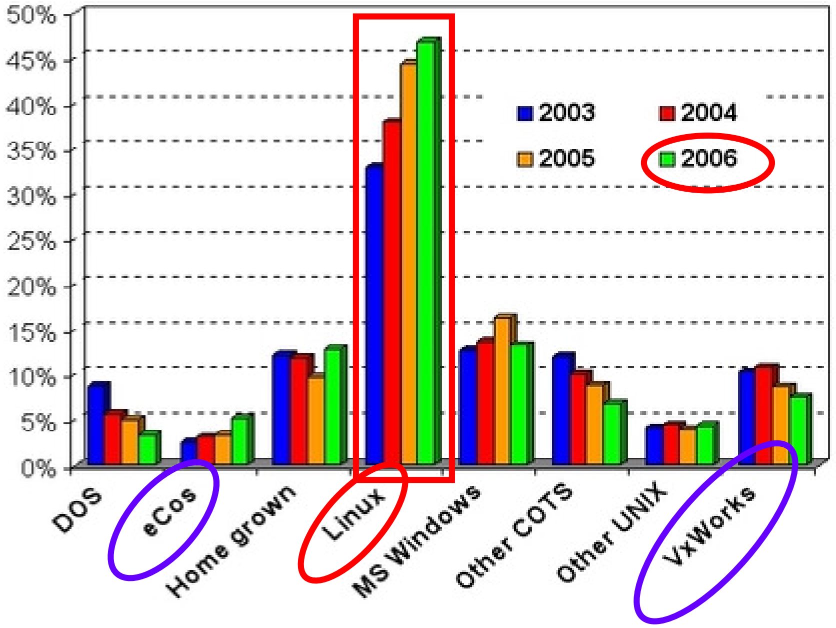


Linux adoption leveling out, but still growing

Linux has truly established itself as a viable if not dominant embedded operating system that is used in thousands of new designs each year. Our survey results over the past four years show that Linux adoption is still on the rise, although uptake has certainly slowed since the "boom" years in the early part of the decade. This year's survey results suggest that 47 percent of our survey's respondents have used Linux in embedded projects and/or products -- a growth of about two percentage points over last year's results.

http://www.linuxdevices.com/articles/AT7070519787.html

Embedded OS sourcing trends



Linux support what processor chip?

Processors that Linux supported

❖Intel Pentium

❖Freescale PowerPC (4th generation)

❖MIPS architecture (BCM 1250 Network processor)

❖ARM architecture

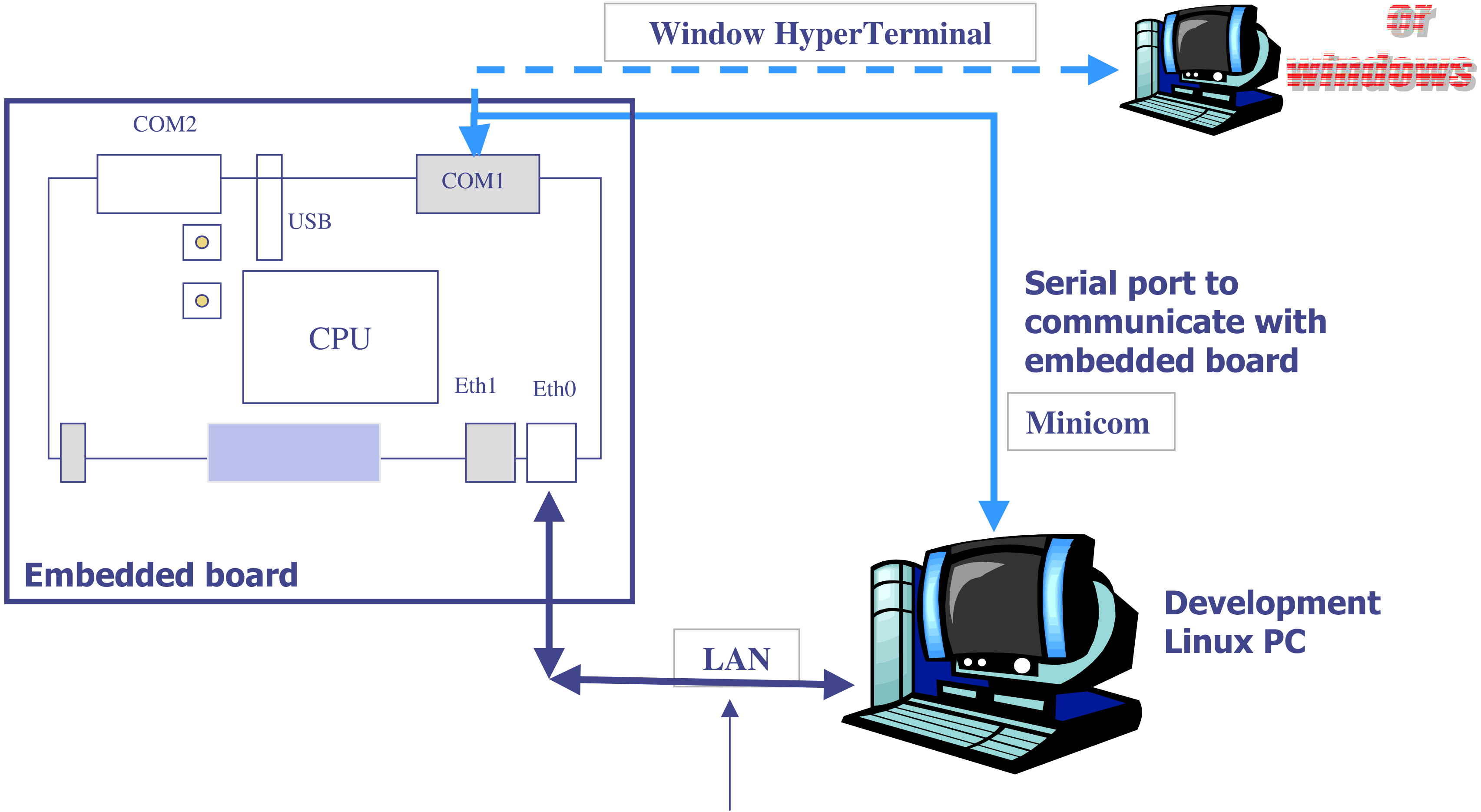
❖Intel ARM XScale (network processor IXP425.. Etc)

… more coming

Embedded system expectation

- Contain a micro- processor, flash memory, DRAM … others
- Design to a specific application (not general purpose computing)
- Simple user interface (Not GUI) via serial port – normally
- Limited resource (small program & data memory, no monitor, no hard disk)
- Communicate with outside world via serial interface (RS232/485),LAN / Wireless, USB, LED, general purpose I/O
- Reset button

Typical Development setup



Embedded board

Serial port to communicate with embedded board

Development Linux PC

LAN to upload Linux OS image or application program

Choose a off-the-shelf embedded Linux Board How-to

❖ Hardware consideration

- CPU speed for a typical application
- external interface : serial , Ethernet port, USB, wireless, memory

❖ Software tools consideration

- working cross- compiler, linker and make file with your hardware
- lot of software examples for application
- able to download program to board via FTP

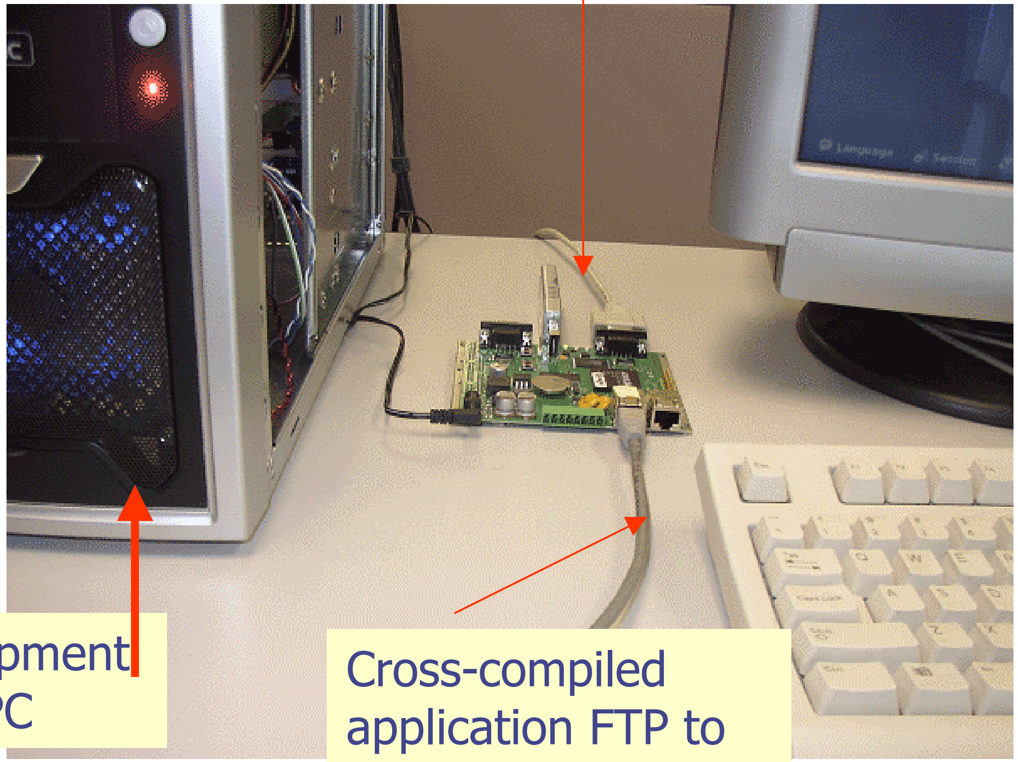
❖ System feature consideration

- default factory setting (useful when enter into endless loop)
- system stability
- monitor program (usually using serial port) available

Why choose A-LinEmb Development Board

- Careful select Hardware system that can demonstrate the complete concept of cross-platform Linux Embedded system
- Hardware consideration
  - 100MIPS , 32 bits RISC Processor
  - more than two serial ports - one use for system Monitoring
  - two LAN ports - able to download program via FTP
  - 1 USB – able to extend system memory using USB memory stick
- Software consideration
  - working cross- compiler, linker and make file to build your kernel and image file
  - able to upload image file, fimage, and application program to board via FTP
  - provide web server
- System feature consideration
  - default factory setting (useful when enter into endless loop)
  - stable system
  - monitor program (usually using serial port) available

A-LinEmb Development Setup



Serial port to communicate to the embedded board

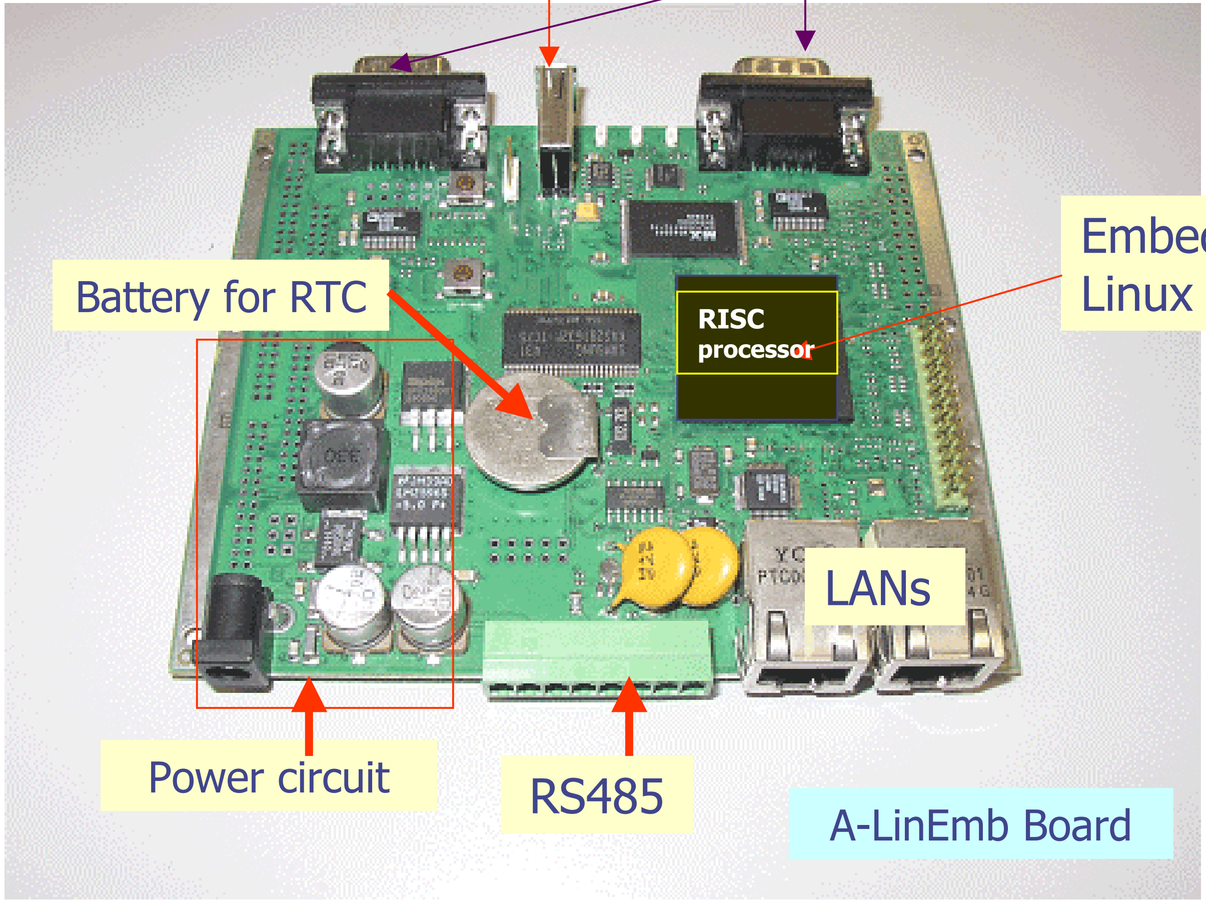
Development Linux PC

Cross-compiled application FTP to embedded board

Close look at A-LinEmb board

USB

Serial ports



Embedded Linux CPU

RISC processor

Battery for RTC

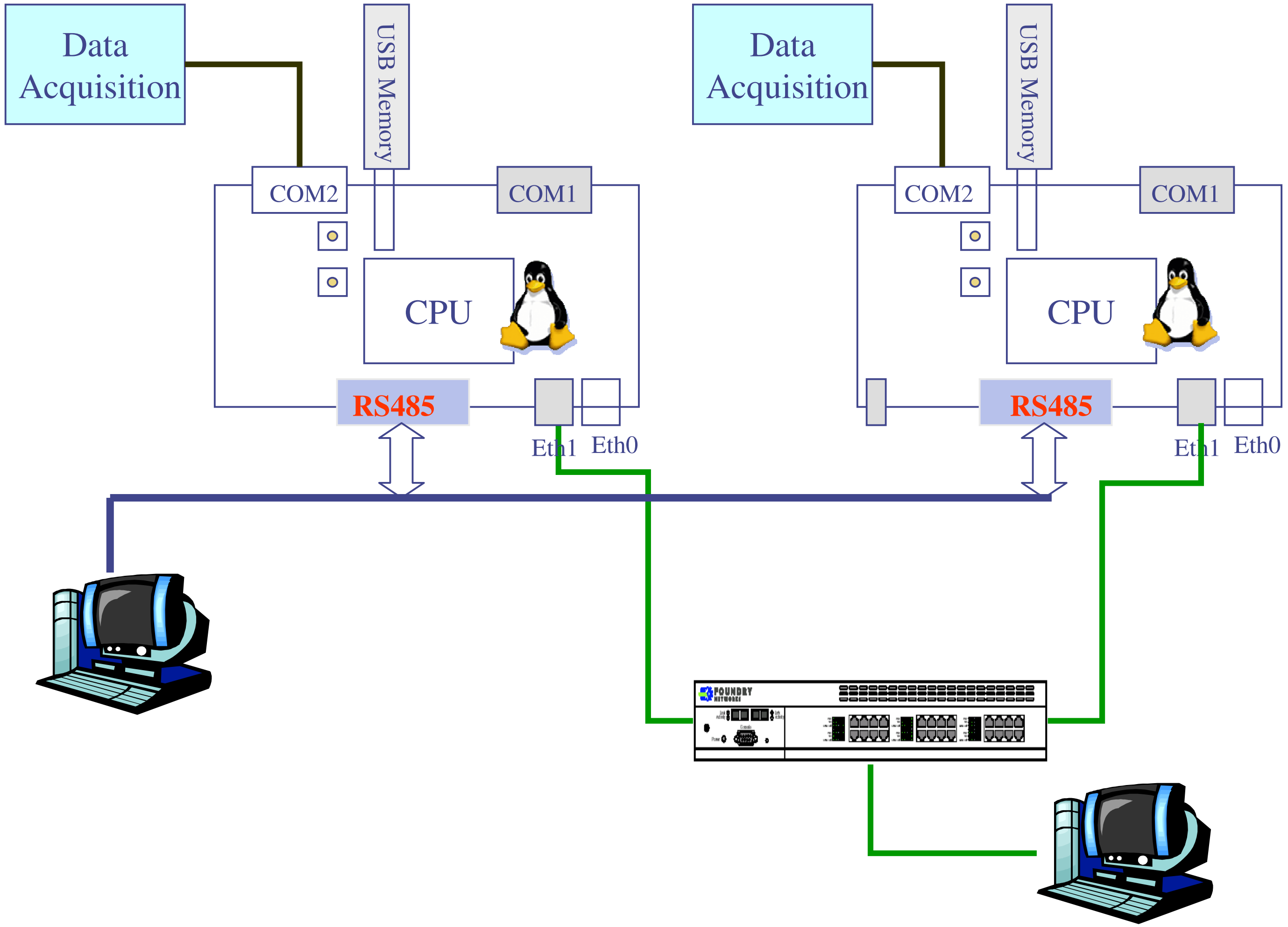
LANs

Power circuit

RS485

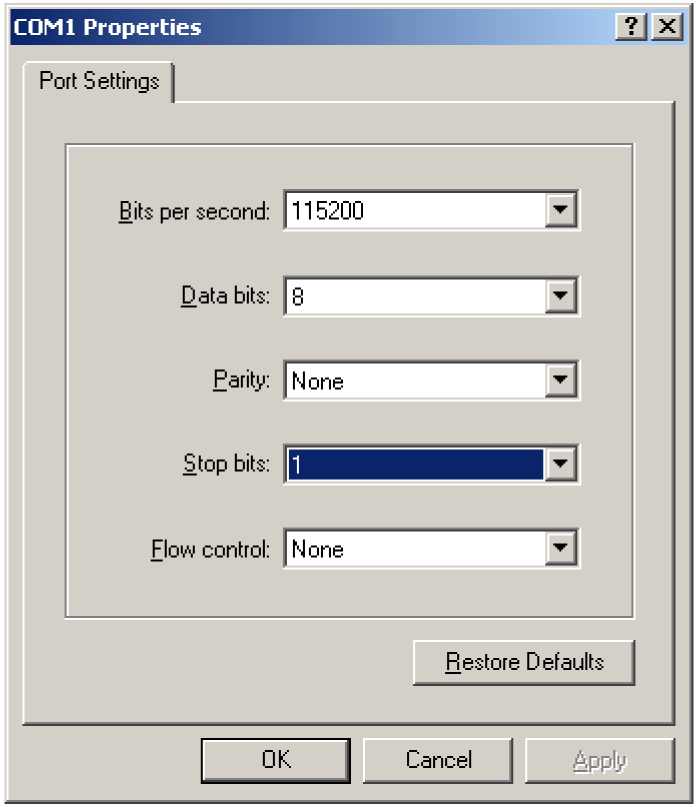
A-LinEmb Board

Application Example of A-LinEmb board



A-LinEmb Monitor Program (Window)

- Before you setup Embedded Linux system, you can use Serial port , on Window PC (Hyper Terminal) or Linux PC (minicom) to view the structure of the A-LinEmb system
- Go to Window Hyper Terminal, set the following :



COM1 Properties

Port Settings

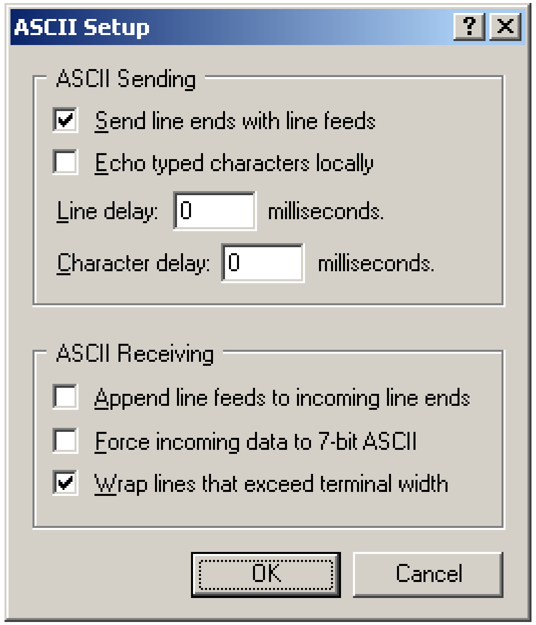
Bits per second: 115200

Data bits: 8

Parity: None

Stop bits: 1

Flow control: None



ASCII Setup

ASCII Sending

☑ Send line ends with line feeds

□ Echo typed characters locally

Line delay: 0 milliseconds.

Character delay: 0 milliseconds.

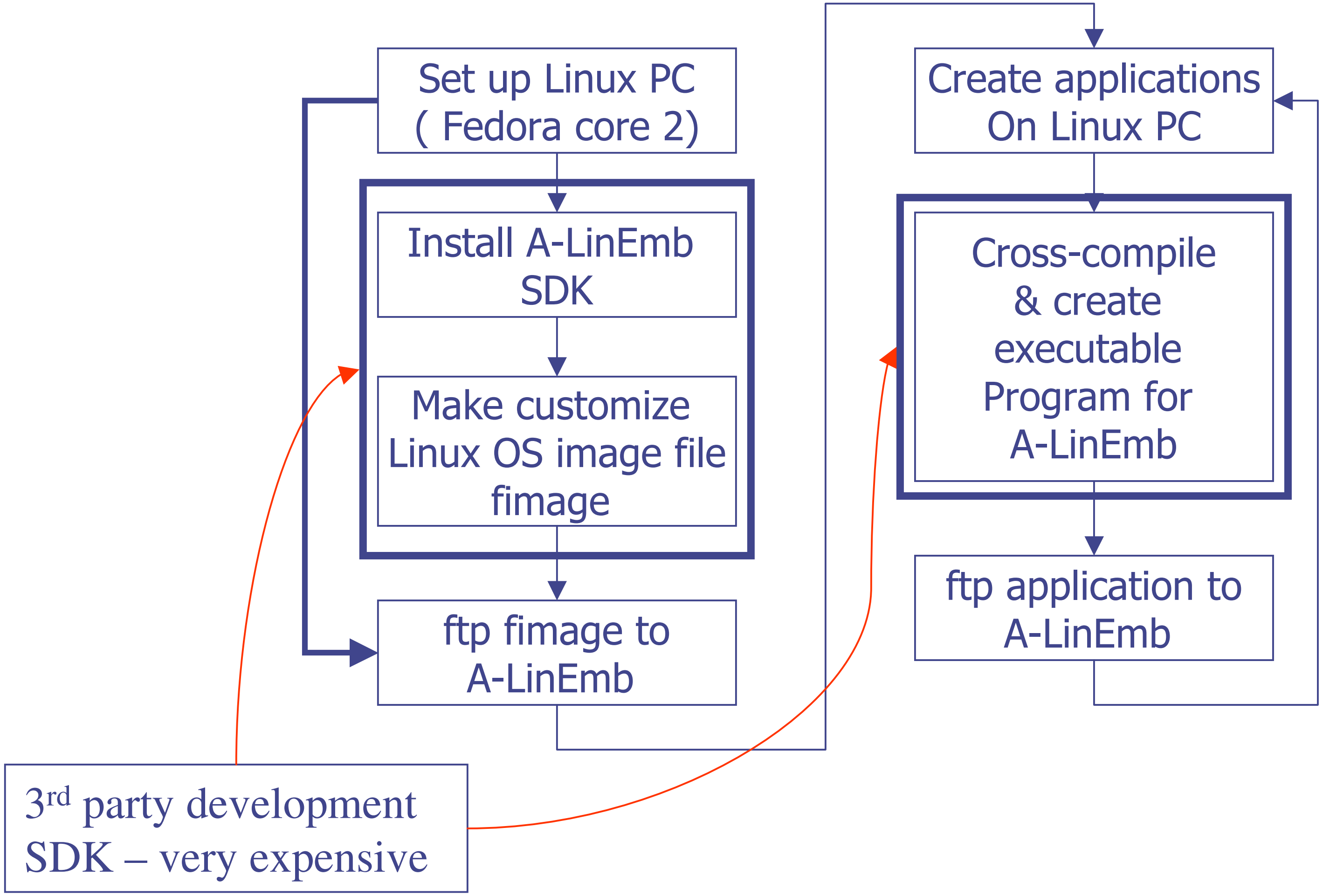
ASCII Receiving

□ Append line feeds to incoming line ends

□ Force incoming data to 7-bit ASCII

☑ Wrap lines that exceed terminal width

A-LinEmb Development Overview


Set up Linux PC
(Fedora core 2)
Install A-LinEmb SDK
Make customize Linux OS image file fimage
ftp fimage to A-LinEmb
Create applications On Linux PC
Cross-compile & create executable Program for A-LinEmb
ftp application to A-LinEmb
3rd party development SDK – very expensive

A-LinEmb Development Overview

- Install Linux OS on the development PC - Fedora 2 & above
- Install board vendor’s development system on the same PC
- Configure Linux OS function and build your own Linux OS image file
- FTP Linux OS file to A-LinEmb
- Develop application program on Linux development PC
- Debug and test the application program at PC level
- When ready, cross-compile/link and create an executable file in A-Emb’s architecture
- FTP the application program to the board’s user memory
- Run the program in the A-LinEmb’s environment and verify it works

What is Cross Compiler ?

- A compiler that run on one computer architecture(i.e i386) but generate executable code that can run on computers with different architecture (Arm, MIPS …etc)
- Cross compiler is hardware architecture dependent
- CRIS ,acronym 'Code Reduced Instruction Set‘, cross compiler is used in A-LinEmb

A-LinEmb Board SDK Setup

A-LinEmb development tool chain installation requirement

- ❖ Install new pmake version 1.45-16

  Pmake-1.45-16.i386.rpm

  A-Emb board need pmake 1.45-16 to work with the cross-compiler
  (Fedora core2's pmake is verion 1.45-14)

- ❖ Install cross-compiler

  Cris-dist-1.63-1.i386.rpm

- ❖ Install SDK release 2.01

  Devboard-R2_01.tar.gz
  Devboard-R2_01-distfiles.tar.gz

- ❖ Create a folder in /root/axis and contain the following files

  Cris-dist-1.63-1.i386.rpm
  Pmake-1.45-16.i386.rpm
  Devboard-R2_01.tar.gz
  Devboard-R2_01-distfiles.tar.gz

A-LinEmb Board SDK Setup

Development Environment Setup in /root/axis

- **Step 1 install pmake**

  # rpm –Uvh pmake-1.45-16.i386.rpm

- **Step 2 Install Cross Compiler**

  # rpm –Uvh cris-dist-1.63-1.i386.rpm

- **Step 3 – install Software Development System (SDK) Release 2.01**

  tar xvfz devboard-R2_01.tar.gz to create devboard-R2_01 folder
  tar xvfz devboard-R2_01-distfiles.tar.gz to create distfiles is then created.

- **Step 4 – go to the just created folder devboard-R2_01**

  - symbolic link inside the devboard-R2_01 folder pointing to this distribution folder using : ***ln –s ../distfiles***

A-LinEmb Board SDK Setup

Create A-LinEmb custom Linux OS

- Go to folder /root/axis/devboard-R2_01
- to define what kind of board you are using

  . /install

- Configuration

  make xconfig

  select Linux 2.6x and others option and save configuration

- Install all packages needed for the configuration

  ./configure

- Build Linux OS Image

  make

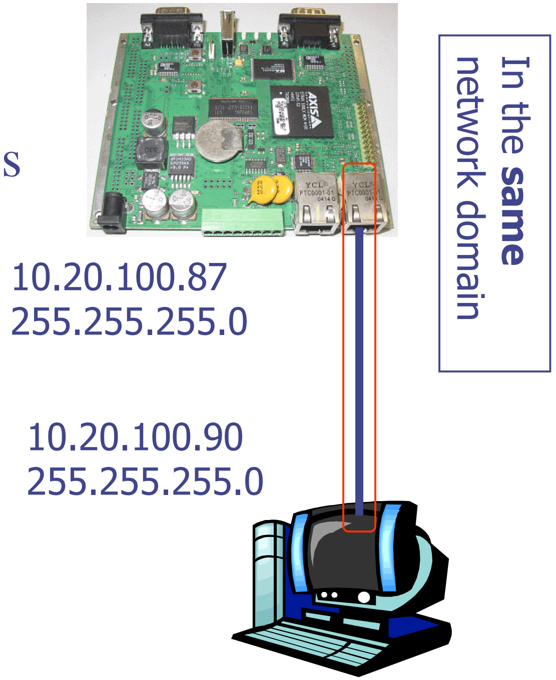
  a file called fimage should be created if no error occur

A-LinEmb Board SDK Setup

Create A-Emb custom Linux OS

- ❖ flash the created fimage file to the development board using ftp
- ❖ While in folder /root/axis/devboard-R2_01 on the development PC

```
]# ftp 10.20.100.87
        > binary
        > hash     - display flashing progress status
        > put fimage flash
            …….. flashing in progress …….
        > quit
]#
```



10.20.100.87
255.255.255.0

10.20.100.90
255.255.255.0

In the **same** network domain

Cross-compile A-LinEmb Board Application Software

- All A-LinEmb application folders will put in the path /root/axis/devboard-R2_01/apps/<myprogram1>
- Make A-LinEmb application program

Convert a xxx.c file from fedora2 platform to custom board platform

Go to folder devboard-R2_01

. init_env

Go to /root/axis/devboard_R2-01/apps/**<myprogram1>**

first create a make file for program myprogram.c on A-LinEmb architecture (see next page for detail)

make cris-axis-linux-gnu

make

Repeated process for new application

- FTP application to embedded board

A-LinEmb make file for CRIS Compiler

```
AXIS_USABLE_LIBS = GLIBC UCLIBC
 include $(AXIS_TOP_DIR)/tools/build/Rules.axis

PROGS = myprogram
INSTDIR = $(prefix)/bin/
INSTMODE = 0755
INSTOWNER = root
INSTGROUP = root

OBJS = myprogram.o

 all: $(PROGS)
$(PROGS): $(OBJS)
     $(CC)$(LDFLAGS) $^ $(LDLIBS) -o $@
 install: $(PROGS)
     $(INSTALL) -d $(INSTDIR)
     $(INSTALL) -m $(INSTMODE) -o $(INSTOWNER) -g $(INSTGROUP) $(PROGS)
     $(INSTDIR)

 clean: rm -f $(PROGS) *.o core
```

Two area need to change

Software Knowledge for embedded Linux development

- ❖ Prefer to use most basic C language (to be transportable to all platform)
- ❖ Need to familiar with Linux system & commands
- ❖ Need to know Linux programming using C language
- ❖ For hardware related programming :
  - Serial port
  - Socket programming – for network programming
  - USB flash memory – for removable data storage

Some Practical Issues

- Hardware (Chip/Board level) vendor do not have control on software due to outsourcing to 3rd party
- Need to pay 3rd party for software development tools , possible license per product shipped
- Vendor's Linux development platform is different from your Linux development platform

  vendor uses redhat9 but you use Fedora core 2
- Even with same platform, the kernel version number must also be the same to minimize error
- Vendor application(specific purpose) example is not available , need lot of resources to research how to do it

… other ?

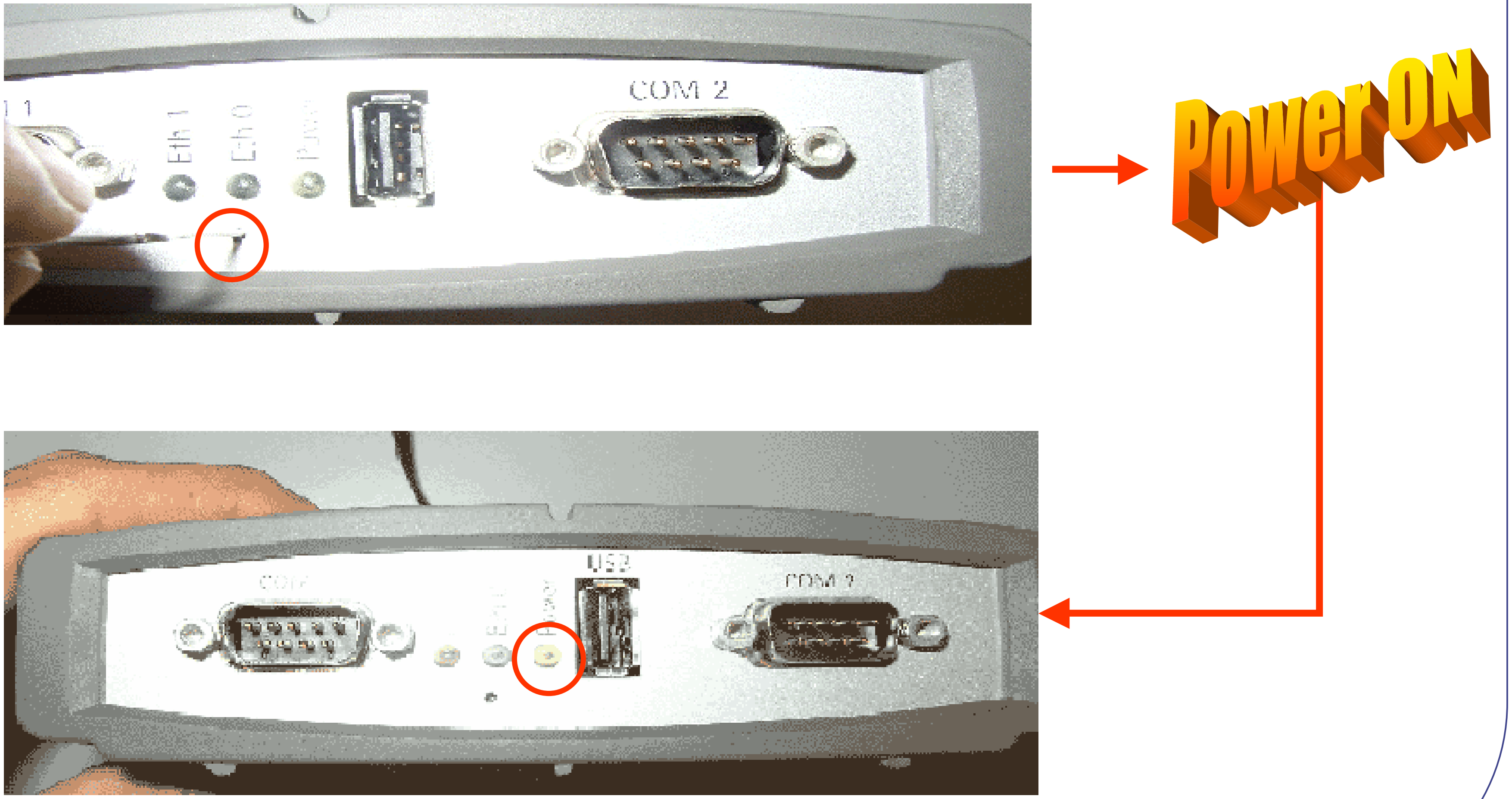
8.0 A-LinEmb Board - How to

Reset to Factory Default

- Reset to factory default setting will be carried out when the A-LinEmb go into a stage that it “hang up” and re-power-up the product has no effect.
- Insert a small pin to the hole at front panel. Power up the A-LinEmb while the button inside the hole still un-release. The front orange color will both on and then one-on and one-off indicate that it is loading the default Linux OS. You can then release the button and wait for the process to complete.

  this process can be monitored as long as your serial port is connected to to COM1
- After the loading of default factory setting, you have to FTP your own fimage file to get back your original setting.

Reset to Factory Default



Power ON

Reset to Factory Default

- After the default setting is loaded, the network ports has the following setting:

  Eth0 : IP= 192.168.0.90 mask = 255.255.255.0 Bcast = 192.168.0.255

  Eth1 : IP= 10.0.0.90 mask = 255.0.0.0 Bcast = 10.255.255.255
- Assuming the Linux development PC is connected to eth0, now you have to temporary change your PC's IP address to be in he same network domain

  e.g. IP address: 192.168.0.1 mask: 255.255.255.0 Bcast: 192.168.0.255
- Do a ping from Linux development PC to A-LinEmb to ensure is linked
- Use flash fimage procedure to update the A-LinEmb OS
- Now you can use the browser on your development PC to edit the /etc/conf.d/net.eth0 and /etc/conf.d/net.eth0 files to set your new IPs
- Do not forget to change back your development PC's IP so that they are in the same network domain

A-LinEmb Editors

- Web based Editor

  This is the best editor to use to change the content of an existing file . e.g /etc/conf.d/ xxx configuration file. It can not be use to create new files. The editor name is called **editcgi** and is reside in the web server. From your PC's web browser use http://<A-LinEmb's IP address> will start the web based editor. You can edit, save, browse the files on the A-LinEmb device

- Serial port , COM1 (not COM2) access editor

  This is the simplified version of the standard Linux editor such as :

  vi editor – default with the Linux OS

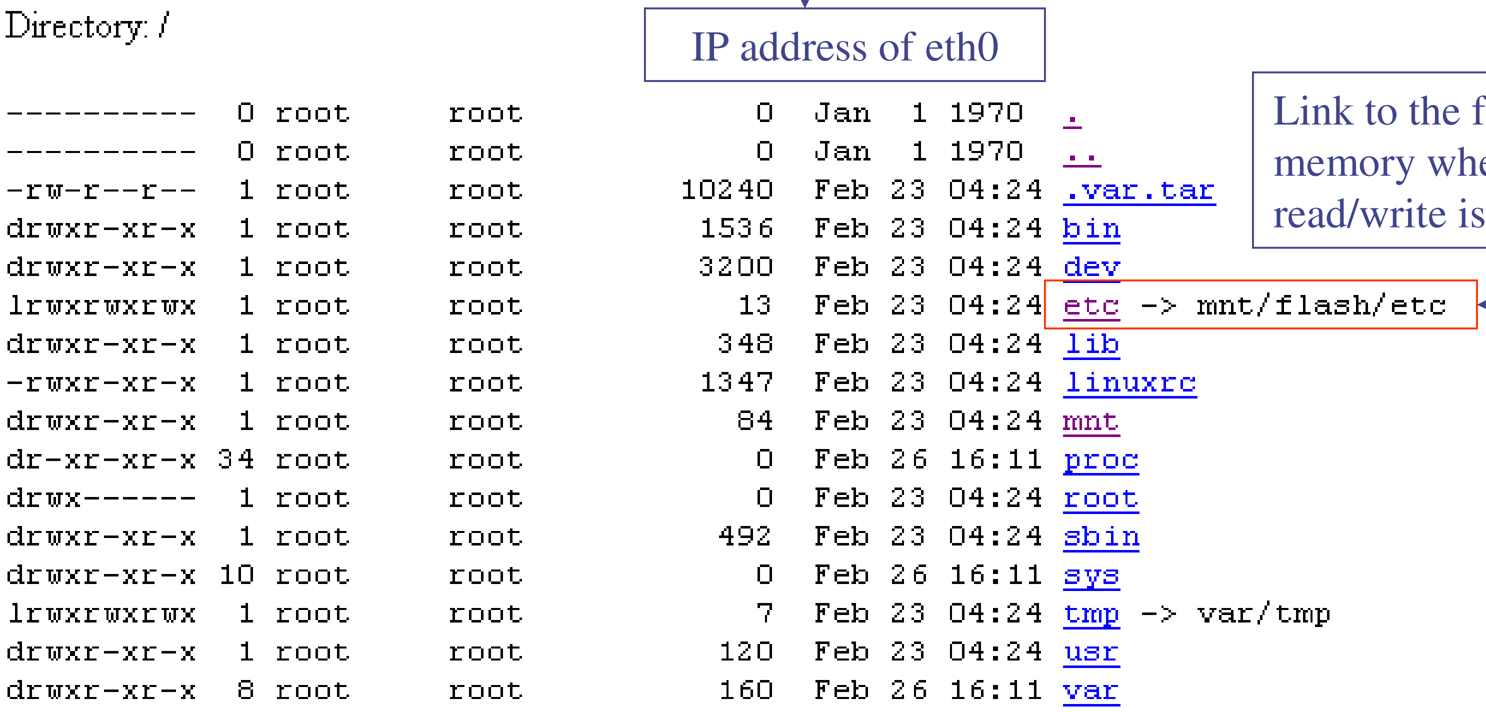
  easyedit (preferred) – need to configure to include

  (Note : on Linux development PC , **gedit** which is very user friendly)

A-LinEmb File Structure – / root folder

❖ At startup, use LAN eth0 , http://10.20.100.87 and go into the root directory

IP address of eth0



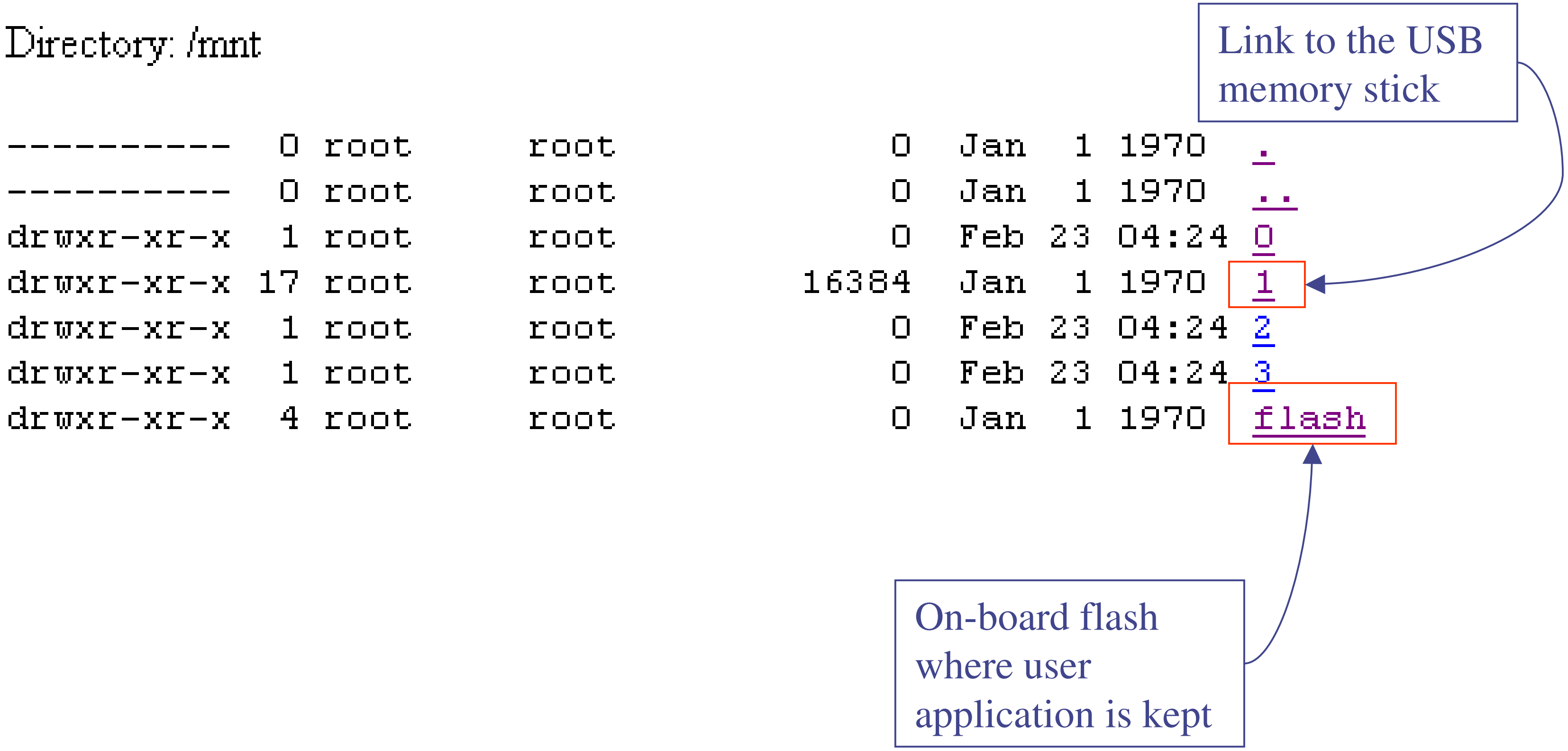
Directory: /

```
----------   0 root     root          0  Jan  1 1970  .
----------   0 root     root          0  Jan  1 1970  ..
-rw-r--r--   1 root     root      10240  Feb 23 04:24 .var.tar
drwxr-xr-x   1 root     root       1536  Feb 23 04:24 bin
drwxr-xr-x   1 root     root       3200  Feb 23 04:24 dev
lrwxrwxrwx   1 root     root         13  Feb 23 04:24 etc -> mnt/flash/etc
drwxr-xr-x   1 root     root        348  Feb 23 04:24 lib
-rwxr-xr-x   1 root     root       1347  Feb 23 04:24 linuxrc
drwxr-xr-x   1 root     root         84  Feb 23 04:24 mnt
dr-xr-xr-x  34 root     root          0  Feb 26 16:11 proc
drwx------   1 root     root          0  Feb 23 04:24 root
drwxr-xr-x   1 root     root        492  Feb 23 04:24 sbin
drwxr-xr-x  10 root     root          0  Feb 26 16:11 sys
lrwxrwxrwx   1 root     root          7  Feb 23 04:24 tmp -> var/tmp
drwxr-xr-x   1 root     root        120  Feb 23 04:24 usr
drwxr-xr-x   8 root     root        160  Feb 26 16:11 var
```

Link to the flash memory where read/write is available

A-LinEmb File Structure - /mnt folder

- ❖ In /mnt folder



Directory: /mnt

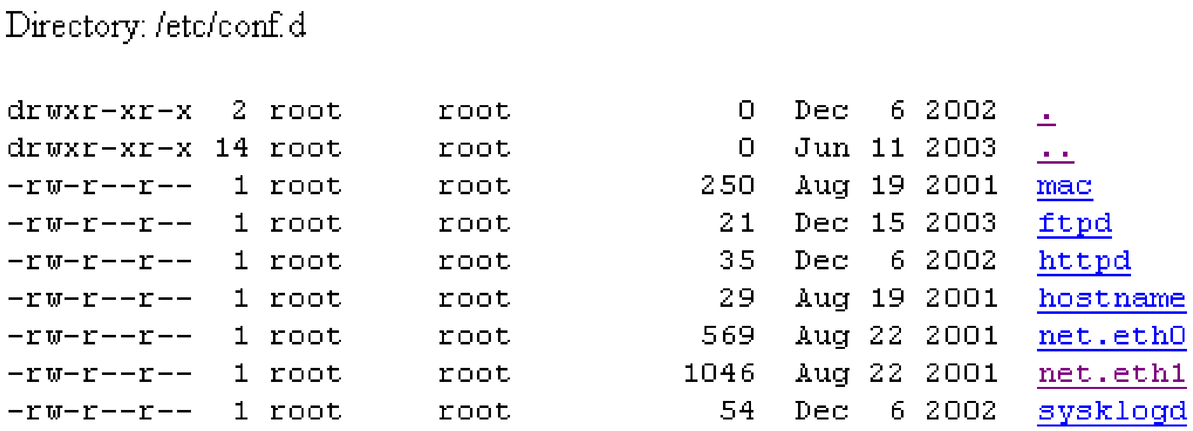
```
----------   0 root      root           0  Jan  1 1970  .
----------   0 root      root           0  Jan  1 1970  ..
drwxr-xr-x   1 root      root           0  Feb 23 04:24 0
drwxr-xr-x  17 root      root       16384  Jan  1 1970  1
drwxr-xr-x   1 root      root           0  Feb 23 04:24 2
drwxr-xr-x   1 root      root           0  Feb 23 04:24 3
drwxr-xr-x   4 root      root           0  Jan  1 1970  flash
```

Link to the USB memory stick

On-board flash where user application is kept

A-LinEmb Network Configuration – /etc/conf.d

- At startup, use LAN eth0.
- /etc/conf.d - all the network configuration files

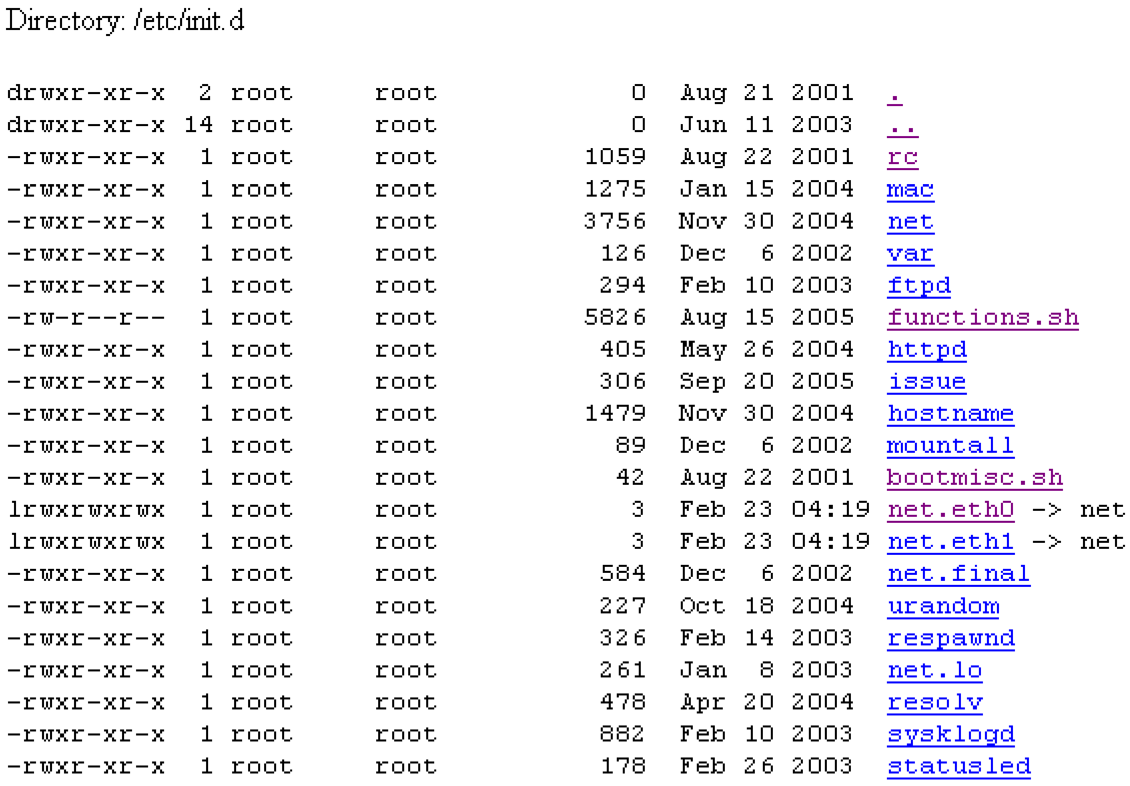


Directory: /etc/conf.d

```
drwxr-xr-x  2 root      root              0  Dec   6 2002  .
drwxr-xr-x 14 root      root              0  Jun  11 2003  ..
-rw-r--r--  1 root      root            250  Aug  19 2001  mac
-rw-r--r--  1 root      root             21  Dec  15 2003  ftpd
-rw-r--r--  1 root      root             35  Dec   6 2002  httpd
-rw-r--r--  1 root      root             29  Aug  19 2001  hostname
-rw-r--r--  1 root      root            569  Aug  22 2001  net.eth0
-rw-r--r--  1 root      root           1046  Aug  22 2001  net.eth1
-rw-r--r--  1 root      root             54  Dec   6 2002  sysklogd
```

A-LinEmb File System Initialization– /etc/init.d

- /etc/init.d - initialization configuration folders

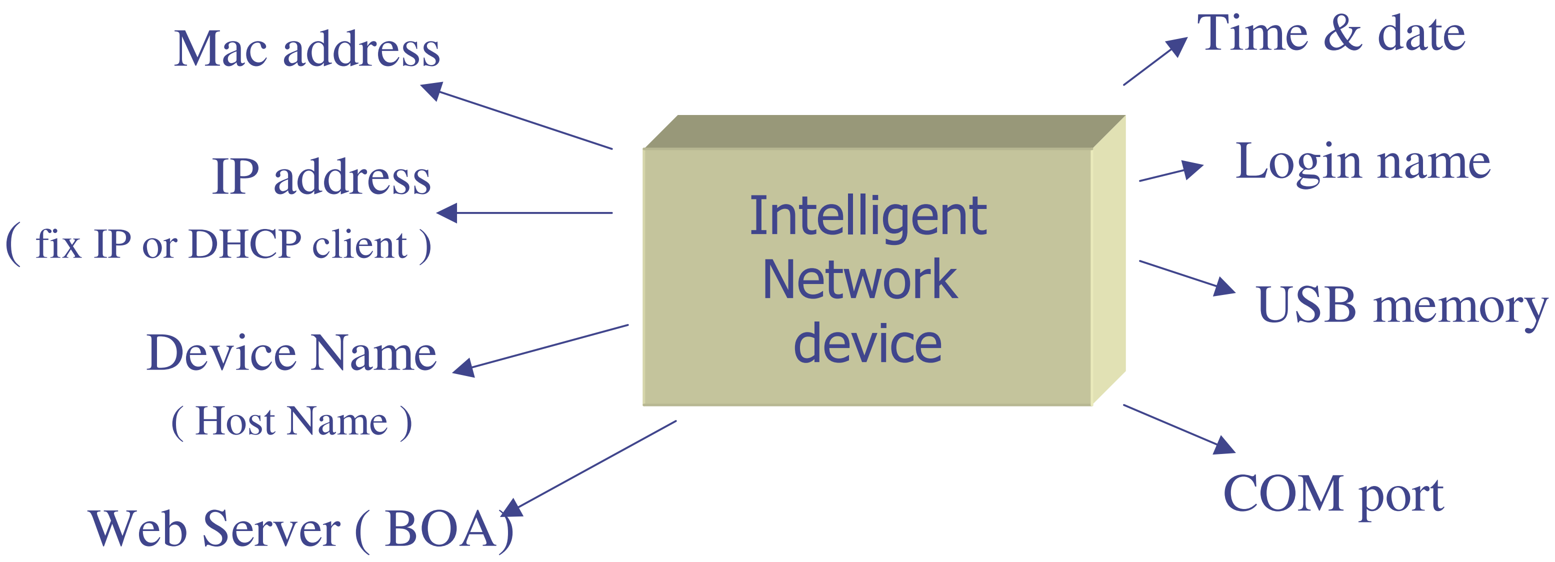


Directory: /etc/init.d

```
drwxr-xr-x  2 root     root           0  Aug 21 2001  .
drwxr-xr-x 14 root     root           0  Jun 11 2003  ..
-rwxr-xr-x  1 root     root        1059  Aug 22 2001  rc
-rwxr-xr-x  1 root     root        1275  Jan 15 2004  mac
-rwxr-xr-x  1 root     root        3756  Nov 30 2004  net
-rwxr-xr-x  1 root     root         126  Dec  6 2002  var
-rwxr-xr-x  1 root     root         294  Feb 10 2003  ftpd
-rw-r--r--  1 root     root        5826  Aug 15 2005  functions.sh
-rwxr-xr-x  1 root     root         405  May 26 2004  httpd
-rwxr-xr-x  1 root     root         306  Sep 20 2005  issue
-rwxr-xr-x  1 root     root        1479  Nov 30 2004  hostname
-rwxr-xr-x  1 root     root          89  Dec  6 2002  mountall
-rwxr-xr-x  1 root     root          42  Aug 22 2001  bootmisc.sh
lrwxrwxrwx  1 root     root           3  Feb 23 04:19 net.eth0 -> net
lrwxrwxrwx  1 root     root           3  Feb 23 04:19 net.eth1 -> net
-rwxr-xr-x  1 root     root         584  Dec  6 2002  net.final
-rwxr-xr-x  1 root     root         227  Oct 18 2004  urandom
-rwxr-xr-x  1 root     root         326  Feb 14 2003  respawnd
-rwxr-xr-x  1 root     root         261  Jan  8 2003  net.lo
-rwxr-xr-x  1 root     root         478  Apr 20 2004  resolv
-rwxr-xr-x  1 root     root         882  Feb 10 2003  sysklogd
-rwxr-xr-x  1 root     root         178  Feb 26 2003  statusled
```

A-LinEmb Device Parameters

- ❖ Parameters that can be altered



USB Memory Stick (use Kernel 2.6)

❖ How to use USB memory stick ?

- in folder /etc/init.d, create a file called bootmisc.sh
- change the file permission to 755 using :

```
chmod 755 bootmisc.sh
```

- inside file bootmisc.sh , insert the following lines :

```
#!/bin/sh
mount -t vfat /dev/sda1 /mnt/1
```

- edit the file /etc/init.d/rc , add the line to the end of file

```
./etc/init.d/bootmisc.sh
```

just before “ exit 0 “

now when you write data to folder /mnt/1 will be automatically write to the usb memory stick

Change network related parameters

❖ Change network related items, mac address, IP address

❖ In folder /etc/conf.d , edit the following files

- Hostname
- mac
- net.eth0
- net.eth1

to change the network parameters

❖ host name :

HOSTNAME="axis-00408c6e14ae"

❖ Mac address :

MAC="00:40:8C:6E:14:AE“

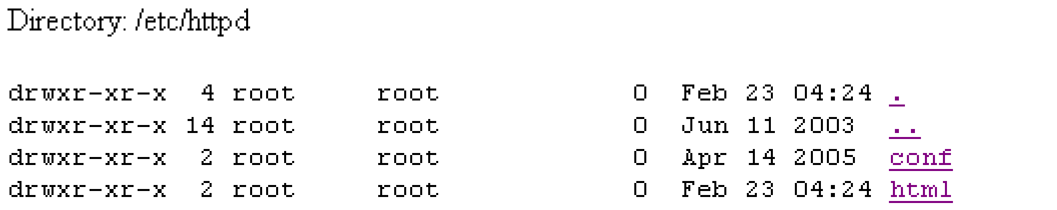
Change network related parameters

- net.eth0 / net.eth1

```
    # MAC="00:40:8C:CD:00:00"
    …
    #BOOTPROTO=‘dhcp”
    BOOTPROTO="none“
    DHCP_CLIENT="/sbin/udhcpc -i eth0"
    …
    MEDIA="auto"
    …
# If you are using DHCP the following variables will not be used.
    IP="10.20.100.87"
    NETMASK="255.255.255.0"
    BROADCAST="10.20.100.255"
    GATEWAY="10.20.100.10"
```

WebServer (http server)

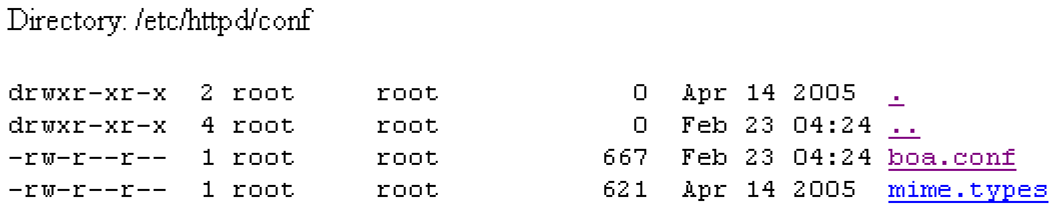
- ❖ A-LinEmb board uses Boa linux web server, is a smaller version of web server compare with Apache Web server.
- ❖ The factory default configuration has only html folder and do not have the CGI (Common Gateway Interface) folder



```
Directory: /etc/httpd

drwxr-xr-x  4 root     root             0  Feb 23 04:24 .
drwxr-xr-x 14 root     root             0  Jun 11 2003  ..
drwxr-xr-x  2 root     root             0  Apr 14 2005  conf
drwxr-xr-x  2 root     root             0  Feb 23 04:24 html
```

- ❖ To configure the Boa server, go to file   /etc/httpd/conf/boa.conf using web browser or monitor program



```
Directory: /etc/httpd/conf

drwxr-xr-x  2 root     root             0  Apr 14 2005  .
drwxr-xr-x  4 root     root             0  Feb 23 04:24 ..
-rw-r--r--  1 root     root           667  Feb 23 04:24 boa.conf
-rw-r--r--  1 root     root           621  Apr 14 2005  mime.types
```

Customize your own web server

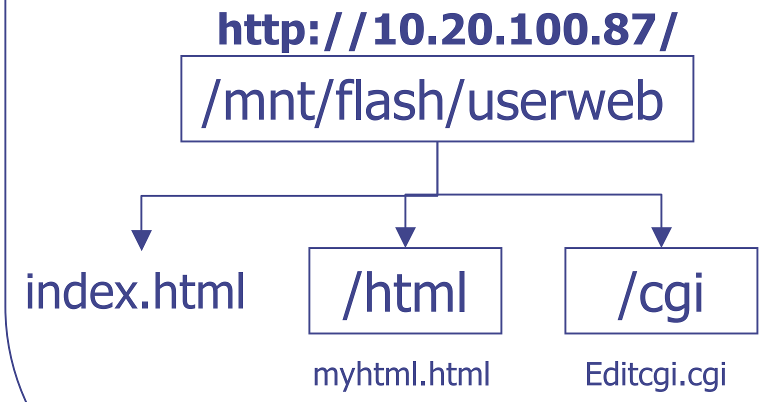
Design condition :

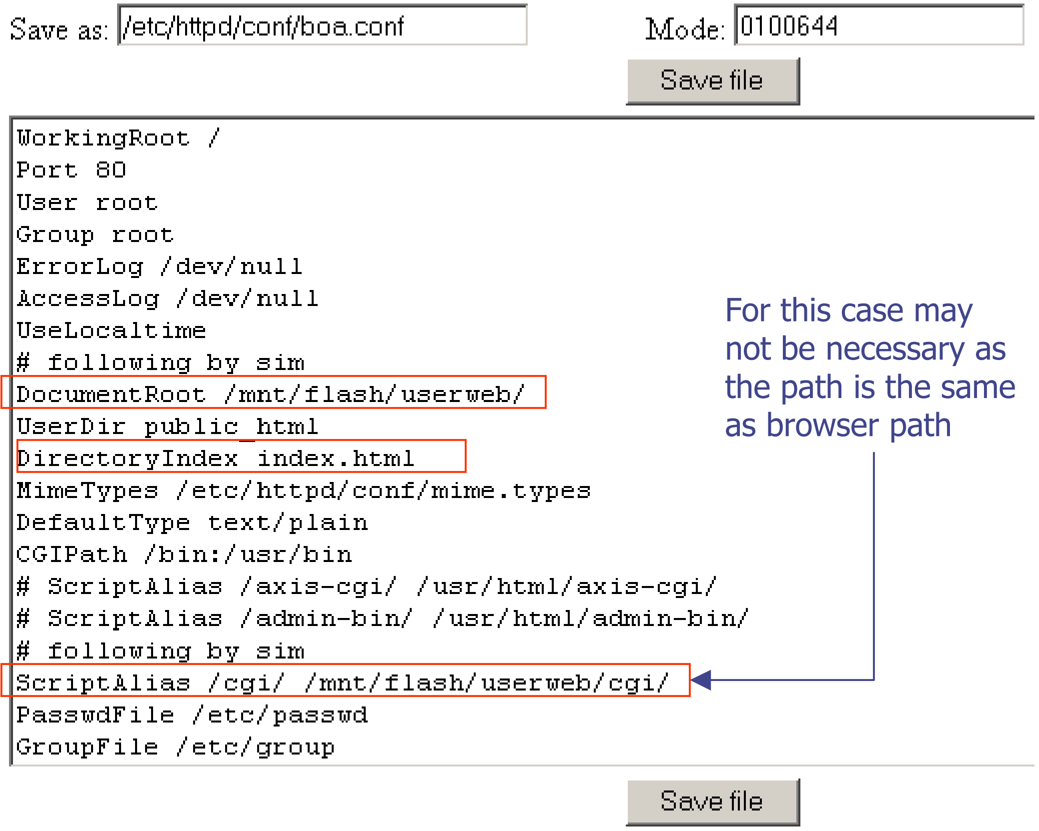
- Location of user web folder is in /mnt/flash/userweb
- Inside the userweb folder, there are 2 folders : html and cgi
- /mnt/flash/userweb/html folder keep all html files include start-up webpage . i.e http://<A-LinEmb IP address> will activate this page

  Start-up page name is : index.html
- /mnt/flash/userweb/html folder keep all the non-html program such as CGI and others
- When http://<A-LinEmb's IP address> will activate the index.html
- When http://<A-LinEmb's IP address>/cgi/mycgi.cgi will run the cgi program called mycgi.cgi

Customize your own web server

- ❖ Using monitor program via serial port, go to /mnt/flash, use mkdir to create a folder called userweb
- ❖ Go to /mnt/flash/userweb folder, create two folders named : cgi & html
- ❖ Go to /etc/httpd/conf folder , edit the boa.conf





Save as: /etc/httpd/conf/boa.conf   Mode: 0100644

```
WorkingRoot /
Port 80
User root
Group root
ErrorLog /dev/null
AccessLog /dev/null
UseLocaltime
# following by sim
DocumentRoot /mnt/flash/userweb/
UserDir public_html
DirectoryIndex index.html
MimeTypes /etc/httpd/conf/mime.types
DefaultType text/plain
CGIPath /bin:/usr/bin
# ScriptAlias /axis-cgi/ /usr/html/axis-cgi/
# ScriptAlias /admin-bin/ /usr/html/admin-bin/
# following by sim
ScriptAlias /cgi/ /mnt/flash/userweb/cgi/
PasswdFile /etc/passwd
GroupFile /etc/group
```

For this case may not be necessary as the path is the same as browser path

Customize your own web server

```
<html>
  <head>
    <title>Linux Experience Sharing</title>
  </head>
  <body bgcolor="#FFFFFF">
  <h1><b>Sharing of Linux development Experience</h1></b><br>
   <a href="/cgi/editcgi.cgi?file=/">
      To Bwowse/Edit/Save of files ( or http://IPaddress/cgi/editcgi.cgi)
   </a>
   <br><br>
   <a href="/html/myhtml.html"> View myhtml.html </a>
   <br>
  </body>
</html>
```

Change System Date & Time

❖ Change the date and time on A-LinEmb board

#] date MMDDHHMMYYYY

e.g. date 030110452006

Your will then be replied with a understandable date/time format

To save it permanently, use :

#] hwclock –w

FAQ

- Does the A-LinEmb Network port's LED works ?

  the A-LinEmb's network port LEDs will not lighted up when the LAN cable is inserted and linked to the network Switch. You should see from the network Switch's LED to know the link is established or not.

  only Power-on LED (Green) will turn on

- How to re-initialize the system at command prompt.

  ]# init 0 or]#init 6 for A-LinEmb board only

  note : when mode change, system will reboot

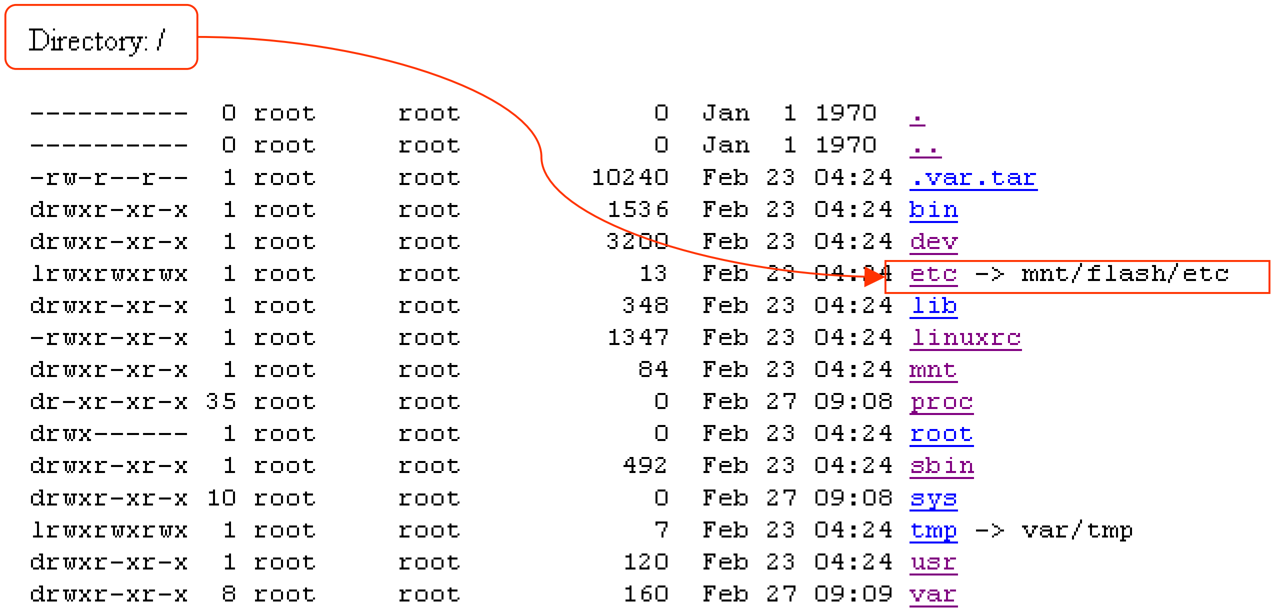
- Can I create new folder using web editor (editcgi) ?

  No. folder can only be created when you are in monitor mode using command]# mkdir <folder name>

FAQ

❖ Which area I can save my programs ?

the area that user can save their program or configuration files is in /mnt/flash. The default folder in this directory is /mnt/flash/etc which store all the configuration and is link to /etc as shown below



Directory: /

```
----------  0 root     root          0  Jan  1 1970  .
----------  0 root     root          0  Jan  1 1970  ..
-rw-r--r--  1 root     root      10240  Feb 23 04:24 .var.tar
drwxr-xr-x  1 root     root       1536  Feb 23 04:24 bin
drwxr-xr-x  1 root     root       3200  Feb 23 04:24 dev
lrwxrwxrwx  1 root     root         13  Feb 23 04:24 etc -> mnt/flash/etc
drwxr-xr-x  1 root     root        348  Feb 23 04:24 lib
-rwxr-xr-x  1 root     root       1347  Feb 23 04:24 linuxrc
drwxr-xr-x  1 root     root         84  Feb 23 04:24 mnt
dr-xr-xr-x 35 root     root          0  Feb 27 09:08 proc
drwx------  1 root     root          0  Feb 23 04:24 root
drwxr-xr-x  1 root     root        492  Feb 23 04:24 sbin
drwxr-xr-x 10 root     root          0  Feb 27 09:08 sys
lrwxrwxrwx  1 root     root          7  Feb 23 04:24 tmp -> var/tmp
drwxr-xr-x  1 root     root        120  Feb 23 04:24 usr
drwxr-xr-x  8 root     root        160  Feb 27 09:09 var
```