Title:

LB-100AN Development Procedures Manual

Drawing Type:	Manual
Drawing No.:	TA103890XX
Date:	November 02, 2016

Rev.	Description	Date	Prepared	Checked	Approved
XX	First Edition	Nov 02, 16	Y.Ueda	K.Nishihara	E.Hayashi

1. Overview

This document describes how to use software of LB-100AN and the customization procedures.

2. Development Procedures

2.1. Development Environment

LB-100AN is designed for a development on Debian 7.1. The procedures in this document use Oracle VM VirtualBox.

2.1.1. Add Virtual Machine

Follow procedures below to add a virtual machine to VirtualBox.

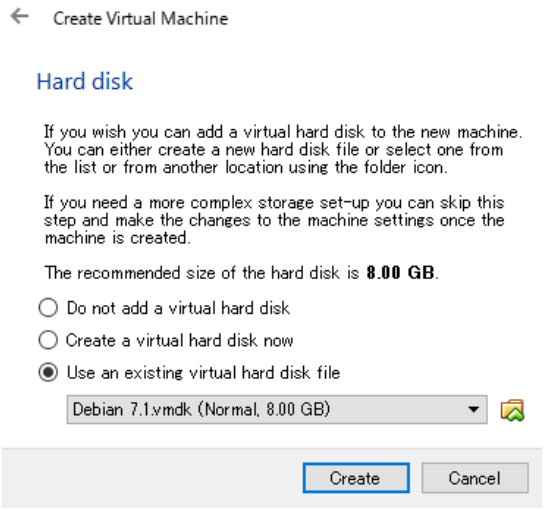
(1) Select "New" to add a virtual machine.

(2) Set the virtual machine name. Make sure that the type and version are configured as follows:

Item	Value
Type	Linux
Version	Debian(32-bit)

(3) Set the memory size.

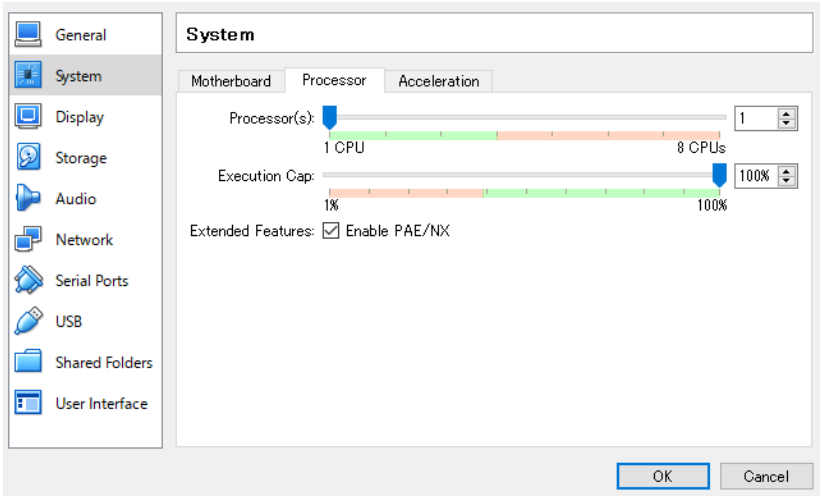
(4) Select "Use an existing virtual hard disk file" for the hardware drive, and choose the extracted environment file provided by Silex.



2.1.2. Configure Virtual Machine

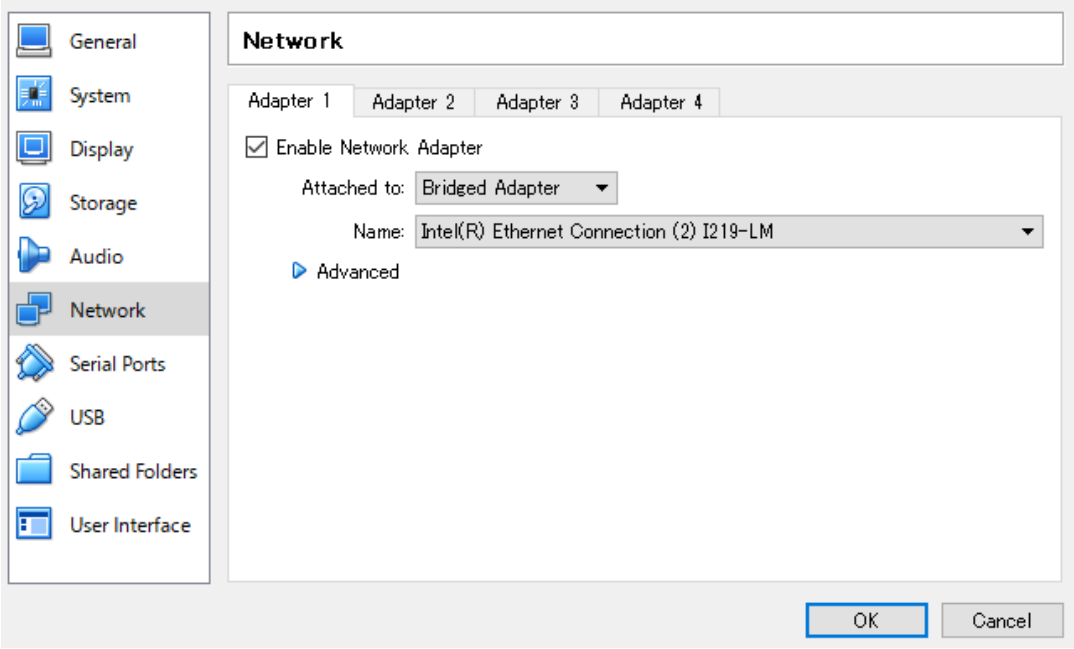
Select the virtual machine you added, and make the following setting.

(1) Click *System* - *Processor* and check the check box at **Enable PAE/NX**.



(2) In *Network*, change the adapter setting.

For LB-100AN development, TFTP server is used in the development environment to implement the network booting. It is recommended to change the network adapter to **Bridge Adapter** on virtual machine setting. Make sure that a wired LAN interface is used as a network interface for the development environment.



2.1.3. Boot Virtual Machine

To start the virtual machine, double click the virtual machine added to VirtualBox or select the virtual machine and click the start button.

2.1.4. User Information

The following shows the user name and password of the virtual environment.

User Name	Password
admin	password
root	password

2.1.5. Network Setting

eth0 is configured as the DHCP client for the network setting of virtual environment. Change the network setting according to your environment since DNS and Proxy are not set. The following file controls the network setting.

```
/etc/network/interfaces
```

To set a static IP address, open the file under the super user authority and change it as follows:

(Original File)

```
auto eth0
iface eth0 inet dhcp
```

(Example of Change; Set a value appropriate for your network environment)

```
auto eth0
iface eth0 inet static
address 192.168.0.1
        netmask 255.255.255.0
        gateway 192.168.0.254
```

2.1.6. Service

The followings are working services:

- SSH Server

  Refer to Section 2.1.4 for user information.

- TFTP Server

  The following shows the root directory. It is a read-only server.

  ```
  /srv/tftp
  ```

- SAMBA (Windows Network Sharing)

  The shared folder is the admin user's home directory and accessible using the admin user authority. The following file controls these setting. To change the setting, edit the file as the super user.

  ```
  /etc/samba/smb.conf
  ```

  If the admin user cannot access the file from Windows, use one of methods below to configure the user information from Windows.

  a. Go to *Add a Windows credential* under *Control Panel – User Account – Credential Manager* and register an admin user in your development environment.
  b. Go to *This PC – Map Network Driver*, tick "Connect using different credentials", and connect to LB-100AN as the admin user.
  c. Use *Run* (windows program), specify the shared folder name after the IP address of the virtual machine to access, and verify the user authentication.

     ```
     ¥¥192.168.0.1¥admin
     ```

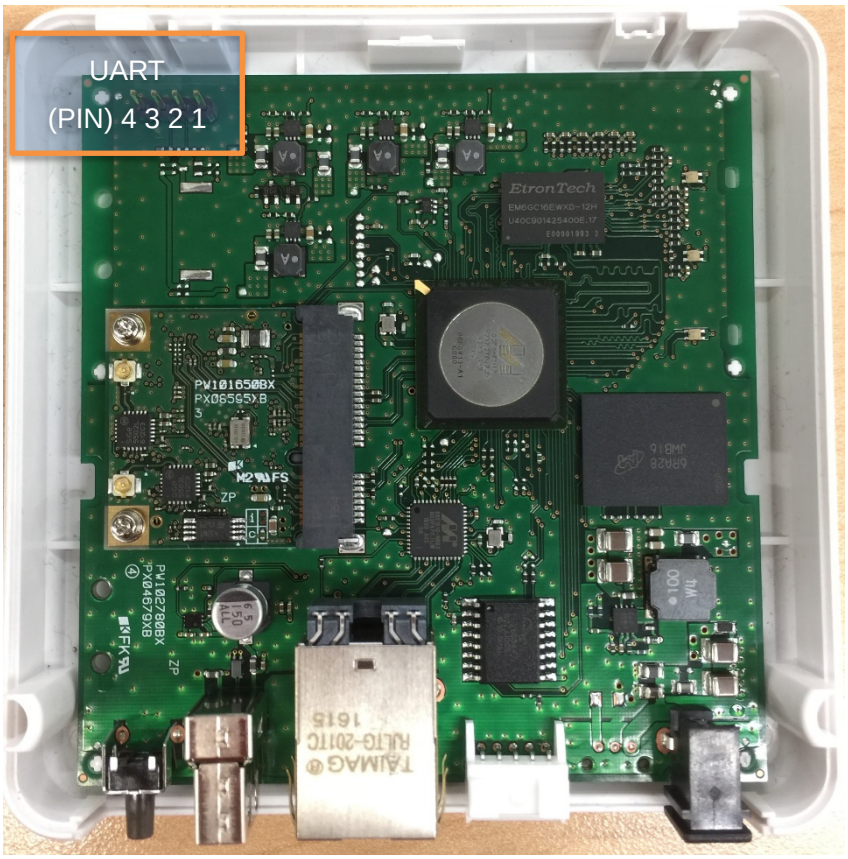
  Note: The above procedures are for Windows7/10. Menus, notations, and operation steps may vary by OS.

  If you edit a file from Windows using the file sharing function, execution permissions for shell scripts may be lost. In such a case, issue the following command to configure the execution permission (Specify a file name for XXX.sh to change the permission).

  ```
  chmod 755 XXX.sh
  ```

2.2. Console

U-Boot and Linux console can be accessed from UART of debug.



The following table shows the PIN configuration. The most right of the above PIN photo shows the PIN 1. Since Vcc carries output power from the PCB, do not apply power externally.

PIN	Function	Direction
1	Vcc	+3.3V Power Output
2	TXD	Output (3.3V TTL)
3	RXD	Input (3.3V TTL)
4	GND	GND

The following shows the console setting.

Item	Value
Baud Rate	115200
Data	8 bits
Parity	None
Stop Bit	1 bit
Flow Control	XON/XOFF

2.3. Source Code Tree

The following table shows source code of LB-100AN.

Directory		Description
apps	/	Directory to locate application source code developed/customized by Silex
	preinstall	Directory to have setting files and startup scripts to be implemented to rootfs
	sample	Sample applications to show how to install applications on LB-100AN
	sxcontext	Directory to locate Silex configuration files
	sxmisc	Directory to have applications unique to Silex devices
	sxutils	Directory to locate commands common to Silex products
	wireless_tools	Application to check wireless LAN setting and status
bootloader	/	Directory to locate source code of the boot loader
	u-boot	Boot loader
build	/	Directory to create the firmware. The complete firmware is created in this directory.
	staging	Directory generated when the firmware is built, and to have the interim firmware.
buildroot	/	Package manager for embedding open source software
kernel	/	Directory to locate Kernel and Kernel modules
	linux-3.2.58	Linux Kernel
	wlan-10.2-00082-4	Wireless LAN driver including configuration tools and supplicants
libs	/	Directory to locate source code of Silex libraries
	libsx	Directory to have Silex standard libraries
	libsxsys	Directory to have Silex platform-dependent libraries
toolchain	/	Toolchain. To build LB-100AN source code, use the toolchain wrapper generated by buildroot.

2.4. How to Build Source Code

Follow the procedure below to create the firmware:

(1) Extract LB-100AN source code.

x.y.z shows the version number of source code.

```
tar zxpf lb-100an-x.y.z.src.tar.gz
```

(2) Move to the build directory to build LB-100AN.

```
cd lb-100an/build
```

(3) Load environmental variables.

```
source setenv.sh
```

(4) Build the firmware.

```
make configure
make
make archive
```

The created firmware is located in build directory.

File Name	Description
initrd.img	initrd for RAM disk. Download the file from the network when LB-100AN is started and use it for the network booting.
rootfs.squashfs	Raw data to be written in the rootfs domain of Flash ROM
rootfs.tar	File of rootfs compressed by tar
uImage	Raw data to be written in the kernel domain of Flash ROM
FIRMWARE.sum	MD5 checksum of the firmware file. LB-100AN gets the firmware file from network and writes it in Flash ROM during the production. It is used to verify integrity of the obtained data.
u-boot.bin	Raw data to be written in the U-Boot domain of Flash ROM
LOADER.sum	MD5 checksum of u-boot.bin

2.4.1. Attention

If buildroot or apps/preinstall is changed, restart from **make** after executing **make clean**.

The following are reasons of the process:

- buildroot copies apps/preinstall in the target root file system only for the first building *(See Note 1)*.
- Another build process for buildroot will not be executed though the buildroot setting is changed when a built object is exist

Note 1: The following files in the apps/preinstall directories are updated when the command of make archive is executed.

Directory	Description
etc/default	Directory to save default value other than LB-100AN's setting that is included in apps/sxcontext
etc/init.d	Directory to locate the startup script
etc/ssl	Directory to save the openssl setting
usr/sbin	Directory to save the shell script

2.5. How to Add Application

This section describes how to add new source code to the firmware. **Before implement the following, finish the build procedure in Section 2.4.**

See also Section 4.1 and 4.3 for OSS and adding commands.

This section shows how to compile using the following as a sample:

```
apps/sample/hello.c
```

2.5.1. Read Environmental Variable

By issuing the following command under the build directory of LB-100AN source code, environmental variables for cross-compiling are read.

```
source setenv.sh
```

This procedure does not need to be executed if it has already been done on a terminal that you logged on to development environment. If the path to the LB-100AN source code is changed, issue the command again. The following table shows some of environmental variables to be added or changed.

Variable	Description
CROSS_COMPILE	Configures prefix of the cross compiler. Use it as follows: $(CROSS_COMPILE)gcc $(CROSS_COMPILE)ar
PATH	Adds the path of cross compiler to the path.
PS1	Adds the LB-100AN name to the shell prompt. It enables the user to see if environmental variables are loaded.

2.5.2. Cross Compile Application

Issue the command as follows to compile the source code directly by using the cross compiler.

(Compile Example)

```
${CROSS_COMPILE}gcc -o hello hello.c
```

The following is a sample to use Makefile.

(Makefile Example)

```
CC      = $(CROSS_COMPILE)gcc
CFLAGS = -Wall -O2 -D_GNU_SOURCE
TARGET = hello

# objects
TARGET_OBJ = hello.o
all: $(TARGET)

hello: $(TARGET_OBJ)
        $(CC) $(CFLAGS) -o $@ $^
clean:
        $(RM) $(TARGET) *.o
distclean: clean

%.o: %.c
        $(CC) $(CFLAGS) -c -o $@ $<
```

The Makefile (above sample) is located in the following path.

```
apps/sample/Makefile
```

2.5.3. Temporarily Embed Application in Firmware

The following procedure creates the firmware which has a cross-compiled application.

(1) Copy the cross-compiled application in the following directory.

```
build/staging/target/usr/bin
```

(2) Move to the build directory of LB-100AN source code and issue the command.

```
make archive
```

2.5.4. Build and Embed Application together with Firmware Build

The following procedure builds an application in the build directory and loads it to the firmware. Please also refer to a comment in the file to edit.

(1) Move to the build directory.

(2) Specify the absolute path to a directory that has Makefile and the application source code to be loaded in system.conf.

```
HELLO_DIR=${APPS_DIR}/sample
```

(3) Specify a rule to Makefile to add the application.

The following is a sample to create rules of **hello** and **hello-clean** and add them to PACKAGE_LIST and CLEAN_LIST. If a process of **configure** or **distclean** is needed before compiling, create and add another rule to CONFIGURE_LIST and DISTCLEAN_LIST.

```
PACKAGE_LIST = ¥
        buildroot ¥
        :(omitted)
        wpa_supplicant ¥
        hello ¥

CLEAN_LIST = ¥
        buildroot-clean ¥
        : (omitted)
        wpa_supplicant-clean ¥
        hello-clean ¥

(…)
hello:
        make -C $(HELLO_DIR)
hello-clean:
        make -C $(HELLO_DIR) clean
```

(4) Add a process to embed the binary to firmware after it is compiled to mkarchive.sh. Then, create a function to copy the file using a name of the rule that has been added to PACKAGE_LIST.

```
hello() {
        install_bin ${HELLO_DIR}/hello ${TARGET_DIR}/usr/bin/hello

        return 0
}
```

By making these changes to execute a process to build a firmware, added application is compiled as well as implemented to the firmware.

2.6. How to Update Firmware and Network Boot

This section describes how to update the firmware built in Section 2.4.

For firmware update and network booting, save the following files in a directory, GX01, in the TFTP server.

Directory	File	Description
GX01	uImage	Kernel program image
	initrd.img	RAM disc
	u-boot.bin	U-Boot image file
	rootfs.squashfs	File of rootfs compressed by **squashfs** to be written in ROM

To skip copying files, create a symbolic link to the build directory as GX01 in the root directory of the TFTP server.

Note: GX01 is version information of Silex loader.

2.6.1. How to Update Firmware (Linux)

Linux in Flash ROM can be edited using TFTP from the U-Boot terminal.

The following shows the update procedure:

(1) Connect LB-100AN and PC with a serial cable.

(2) Enter any key from the terminal after turning on LB-100AN before Linux boots up and enter into the terminal U-Boot.

(3) Set the following environmental variables.

Environmental Variable	Description
netmask	Net mask of LB-100AN
ipaddr	IP address of LB-100AN
serverip	TFTP server address

The following shows the sample setting.

```
setenv netmask 255.255.255.0
setenv ipaddr 192.168.0.20
setenv serverip 192.168.0.1
```

Issue the following command to keep the environmental variables after restart of LB-100AN.

```
saveenv
```

(4) Issue the following commands.

```
sf probe 0:0
tftpboot 0x2000000 GX01/uImage
sf erase 0x180000 0x280000
sf write $fileaddr 0x180000 $filesize

tftpboot 0x2000000 GX01/rootfs.squashfs
sf erase 0x400000 0xC00000
sf write $fileaddr 0x400000 $filesize
```

The following table shows details of above values and variables.

Variable/Value	Description
0x2000000	First address to save a file obtained by TFTP in RAM
0x180000	First address of Flash ROM to write Linux Kernel
0x280000	Data size saved for Linux Kernel in Flash ROM
0x400000	First address of Flash ROM to write the firmware
0xC00000	First address of Flash ROM secured for the firmware
$fileaddr	Variable to show the first address which saves a file obtained by TFTP
$filesize	Variable to show the size of file obtained by TFTP

(5) Restart LB-100AN.

```
reset
```

2.6.2. How to Update Firmware (U-Boot)

U-Boot in Flash ROM can be edited using TFTP from the U-Boot terminal. This section describes the update procedure. Please note that if the update failed, U-Boot will not be able to start up. In such a case, please contact Silex.

(1) Connect LB-100AN and PC with a serial cable.

(2) Enter any key from the terminal to log in the console of U-Boot after turning on LB-100AN and until Linux starts up.

(3) Set the following environmental variables.

Environmental Variable	Description
netmask	Net mask of LB-100AN
ipaddr	IP address of LB-100AN
serverip	TFTP server address

This is a setting example.

```
setenv netmask 255.255.255.0
setenv ipaddr 192.168.0.20
setenv serverip 192.168.0.1
```

Issue the following command to keep the environmental variables after restart of LB-100AN.

```
saveenv
```

(4) Give the following command.

```
bubt GX01/u-boot.bin
```

(5) Restart LB-100AN.

```
reset
```

LB-100AN operates with the updated U-Boot after restart. Make sure U-Boot update is appropriately finished before its restart.

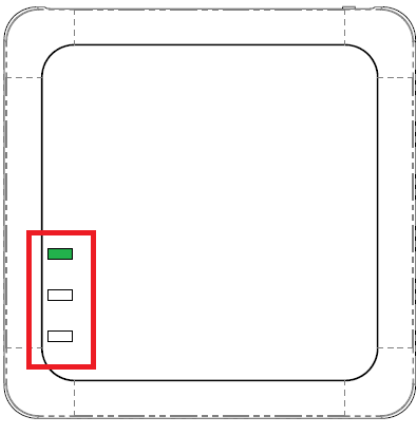
2.6.3. Network Booting Process

If the firmware is not written in Flash ROM, LB-100AN starts the network booting. The following procedures enable the user to run network booting of LB-100AN though the firmware is written. To use the process below, size of initrd.img must be 32 MBytes or smaller.

2.6.3.1. Start Up with Push Switch

(1) Turn on LB-100AN while holding down the push switch.

(2) Confirm LED3 (See the figure below) flashes green, and then release the push switch within 2 seconds.



(3) U-Boot starts downloading a file from TFTP server and Linux boots up in approximately 30 seconds.
The IP address of TFTP server is configured as one of the U-Boot environmental variables. (Default: 192.168.0.1)

2.6.3.2. Network Boot Configuration using Serial Console

(1) Connect LB-100AN and PC using a serial cable.

(2) Enter any key from the terminal to log in the console of U-Boot after turning on LB-100AN before Linux starts up.

(3) Set the following environmental variables.

Environmental Variable	Description
Netmask	Net mask of LB-100AN
Ipaddr	IP address of LB-100AN
serverip	Server address of TFTP

The following is a setting example.

```
setenv netmask 255.255.255.0
setenv ipaddr 192.168.0.20
setenv serverip 192.168.0.1
```

Issue the following command to keep the environmental variables after restart of LB-100AN.

```
Saveenv
```

(4) Issue the following command.

```
run bootcmd_net
```

To change the default startup mode to the network booting, use the following commands to rewrite the environmental variables of U-Boot. These commands saves default environmental variables as different ones.

```
setenv bootcmd_rom $bootcmd
setenv bootcmd 'run bootcmd_net'
saveenv
reset
```

To allow LB-100AN to boot from Flash ROM again, issue the following commands using the created environmental variable using the above sample.

```
setenv bootcmd $bootcmd_rom
saveenv
reset
```

3. How to Use LB-100AN

This chapter describes on initial configuration of LB-100AN. For details of wireless LAN setting, refer to the other document, *Wireless LAN Driver Specifications*. Please note that description of items not related to the configuration files are omitted. For example, the configuration method of operation mode is not mentioned for the IP address setting.

3.1. Configuration File

The startup process of LB-100AN implements the network setting and wireless LAN applications using the following files.

File Name	Description
/etc/sysconfig/system.conf	Configuration file having setting values of the operation mode and the IP address for LB-100AN
/etc/wpa_supplicant.conf	Configuration file of wpa_supplicant (application used in the STA mode). It is a symbolic link and the actual file is /etc/sysconfig/wpa_supplicant.conf.
/etc/hostapd.conf	Configuration file of hostapd (application used in the AP mode). It is a symbolic link and the actual file is /etc/sysconfig/hostapd.conf.

Changing setting values in these files enables the startup script of LB-100AN to change the network setting. Editors such as vi, etc. can be used to edit these files since they are saved as a text format. For details of the file format and setting items, refer to other documents, *Software Functional Specifications* and *Wireless LAN Driver Specification*.

3.2. Configure IP Address

The following shows how to describe the configuration file below to set the IP address.

```
/etc/sysconfig/system.conf
```

- To use DHCP

```
BOOTP_ENABLE=ENABLE
```

- To fix the settings of IP address (192.168.0.10) and default gateway address (192.168.0.1)

```
BOOTP_ENABLE=DISABLE
IPADDR=192.168.0.10
NETMASK=255.255.255.0
GATEWAY=192.168.0.1
```

To enable the settings, issue one of the following commands.

- Restart the startup script

```
/etc/init.d/S40network restart
```

- Restart LB-100AN

```
reboot
```

3.3. Use LB-100AN with Wired LAN

Default values of LB-100AN work for a wired LAN only.

3.4. Use LB-100AN as STA

(1) Set the ship-to country for the wireless LAN.

If the ship-to country is Japan (JP), specify the value as follows. Regarding the supported country codes, refer to the other document, *Software Functional Specifications*. To set the country using U-Boot, refer to Section 4.1.2.

```
fw_setenv country_code JP
```

(2) Set the operation mode to STA in the configuration file, and change wpa_supplicant.conf according to the AP.

i. When LB-100AN operates as STA and connects to AP using OPEN/NONE setting:

Setting	Value
SSID	AP_OPEN
Network Authentication	Open
WEP Encryption	OFF

File Name	Description
/etc/sysconfg/system.conf	NETWORK_MODE=STA

/etc/wpa_supplicant.conf	ctrl_interface=/var/run/wpa_supplicant
	ctrl_interface_group=root
	network={
	scan_ssid=1
	mode=0
	ssid="AP_OPEN"
	auth_alg=OPEN
	key_mgmt=NONE
	pairwise=NONE
	}

/etc/hostapd.conf	(Not in use)

ii. When LB-100AN operates as STA and connects to AP using WPA2-PSK/AES setting:

Setting	Value
SSID	AP_WPA2-PSK-AES
Network Authentication	WPA2-PSK
Encryption Method	AES
Shared Key	12345678

File Name	Description

/etc/sysconfg/system.conf	NETWORK_MODE=STA

/etc/wpa_supplicant.conf	ctrl_interface=/var/run/wpa_supplicant
	ctrl_interface_group=root
	network={
	scan_ssid=1
	mode=0
	ssid="AP_WPA2-PSK-AES"
	proto=RSN
	key_mgmt=WPA-PSK
	pairwise=CCMP
	group=CCMP TKIP
	psk="12345678"
	}

/etc/hostapd.conf	(Not in use)

(3) Enable the settings.

When the configuration file has been edited, enable the settings by using one of the following commands.

To restart only network setting using the startup script

```
/etc/init.d/S40network restart
```

To restart LB-100AN

```
reboot
```

3.5. Use LB-100AN as AP

(1) Set the ship-to country for the wireless LAN.

If the ship-to country is Japan (JP), specify the value as follows. Regarding the supported country codes, refer to the other document, *Software Functional Specifications.* To set the country using U-Boot, refer to Section 4.1.2.

```
fw_setenv country_code JP
```

(2) Set the operation mode to AP in the configuration file, and update hostapd.conf according to the settings you wish to use.

i. When LB-100AN operates as AP using the following OPEN/NONE setting:

Setting	Value
SSID	AP_OPEN
Network Authentication	Open
WEP Encryption	OFF

File Name	Description
/etc/sysconfg/system.conf	NETWORK_MODE=AP
/etc/wpa_supplicant.conf	(Not in use)
/etc/hostapd.conf	interface=ath0 driver=atheros ssid= AP_OPEN max_num_sta=100 auth_algs=1

ii. When LB-100AN operates as AP using the following WPA2-PSK/AES setting:

Setting	Value
SSID	AP_WPA2-PSK-AES
Network Authentication	WPA2-PSK
Encryption Method	AES
Shared Key	12345678

File Name	Description
/etc/sysconfg/system.conf	NETWORK_MODE=AP
/etc/wpa_supplicant.conf	(Not in use)
/etc/hostapd.conf	interface=ath0 driver=atheros ssid=AP_WPA2-PSK-AES

	max_num_sta=100 wpa=2 wpa_key_mgmt=WPA-PSK wpa_pairwise=CCMP wpa_passphrase=12345678

(3) Enable the settings.

Use one of the following commands after editing the configuration files to enable the settings.

To restart only network setting by using the startup script only:

```
/etc/init.d/S40network restart
```

To restart LB-100AN:

```
reboot
```

4. TIPS

4.1. U-Boot Environmental Variables

4.1.1. Default of Startup Command

The following shows the default of U-Boot startup command.

```
setenv bootargs $console $mtdparts $bootargs_root; bootm $bootaddr;
```

Default values allow the following command to start LB-100AN from Flash ROM.

```
setenv bootcmd 'setenv bootargs $console $mtdparts $bootargs_root; bootm $bootaddr;'
saveenv
reset
```

4.1.2. Setting of MAC Address and Ship-to Country for Wireless LAN

If U-Boot environmental variables are initialized by using the following command, **MAC address of LB-100AN and the ship-to country code for wireless LAN will be reset.**

```
resetenv
```

If values are reset, write the setting values from U-Boot as follows.

- When MAC Address is 84:25:3f:00:11:fa:

```
setenv ethaddr  84:25:3f:00:11:fa
saveenv
```

- When the ship-to country code is Japan (JP) for wireless LAN:

```
setenv country_code  JP
saveenv
```

See also Section 4.6.2 for how to refer and configure environmental variables of U-Boot from Linux.

4.1.3. Boot Delay

It takes a longer period of time to detect PHY link and the network booting may fail depending on a network connected to LB-100AN. Adjust Boot Delay using the example below when the network booting fails to get uImage.

- To set Boot Delay to 3 seconds:

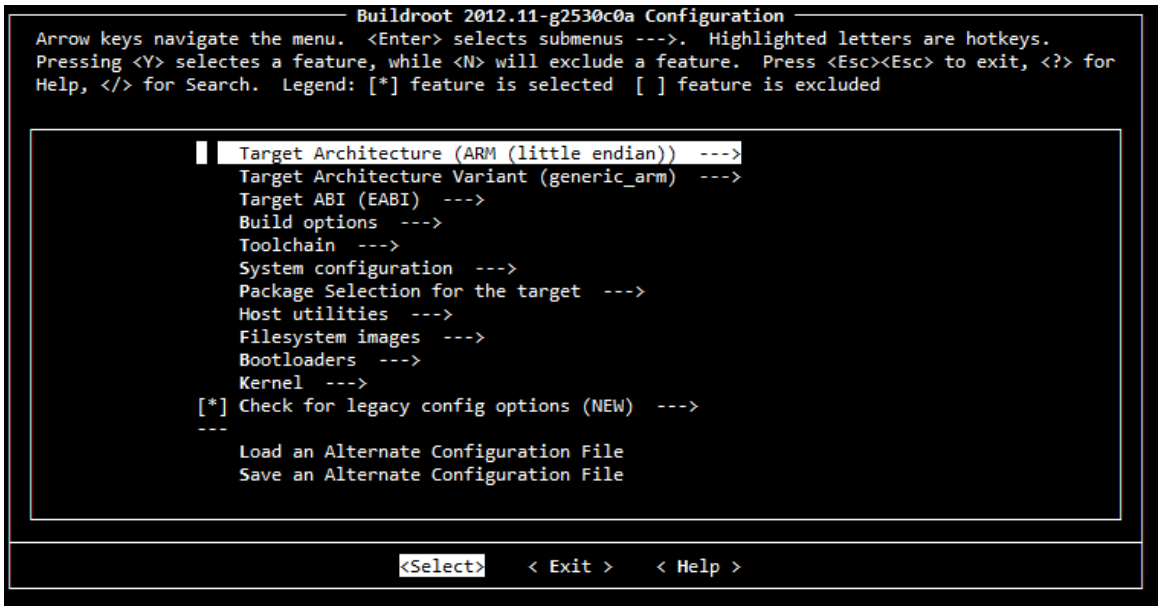
```
setenv bootdelay  3
saveenv
```

4.2. How to Customize buildroot

By executing the following command in the build directory of the source code, the buildroot setting can be changed. **Make sure to load environmental variables (setenv.sh) beforehand**.

```
make buildroot-menuconfig
```

Issue the command and the following menu shows up.



Select and save a package to add. As it temporarily outputs the setting information to build/staging/.config, save the file to the source code using the following commands.

```
cd staging/
cp .config ../../buildroot/configs/LB100AN_defconfig
```

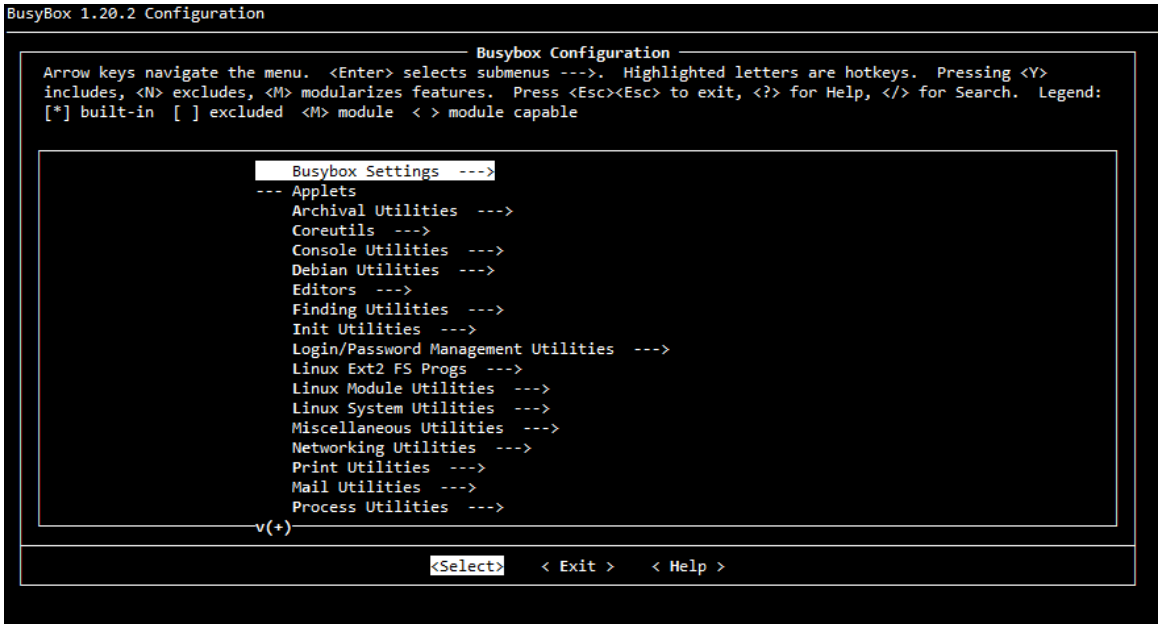
To enable the setting in the firmware, execute rebuild at Section 2.4.1. If a new package is added, **buildroot will download the necessary source code for the package from the Internet.** Make sure your development environment is connected to the Internet.

4.3. How to Customize busybox

By executing the following command in the build directory of the source code, the busybox setting can be changed. The command is for adding and deleting busybox commands. **Make sure to load the environmental variables (setenv.sh) beforehand.**

```
make buildroot-busybox-menuconfig
```

Issue the command and the following menu shows up.



Save the settings. As it temporarily outputs the setting information to staging/build/busybox-1.20.2/.config, save the file to the source code using the following commands.

```
cd staging/build/busybox-1.20.2
cp .config ../../../../buildroot/package/busybox/busybox-1.20.x.config
```

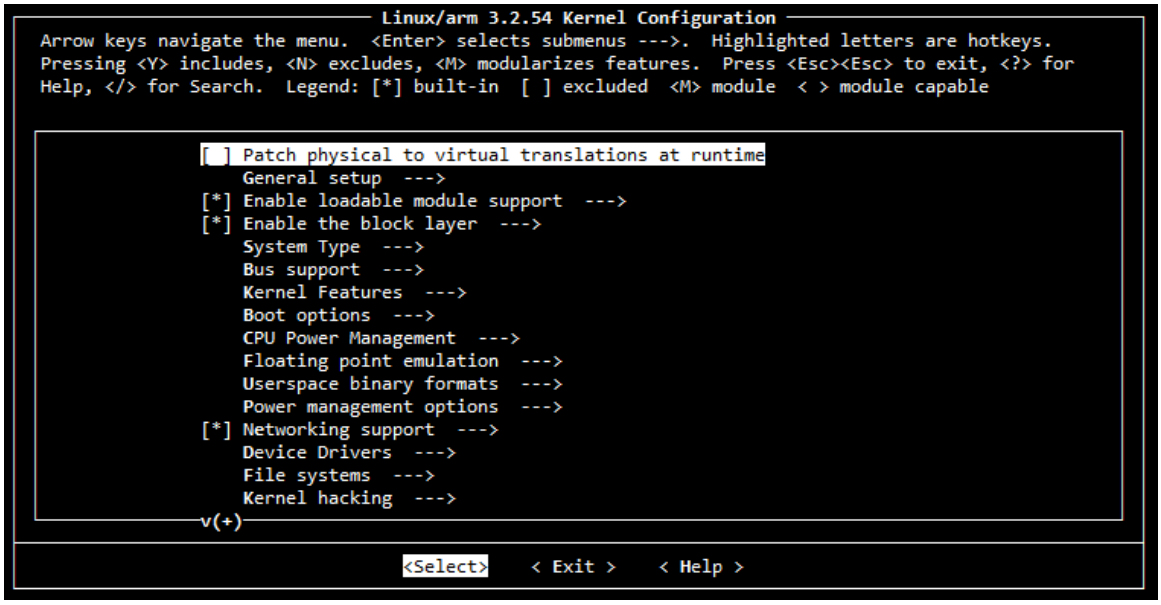
To enable the setting in the firmware, execute rebuild at Section 2.4.1.

4.4. How to Customize Linux

By executing the following command in the build directory of the source code, the Linux setting can be changed. **Make sure to load the environmental variables (setenv.sh) beforehand.**

```
make linux-menuconfig
```

Issue the command and the following menu shows up.



Save the settings. As it temporarily outputs the setting information to kernel/linux-3.2.58/.config, save the file to the source code using the following commands.

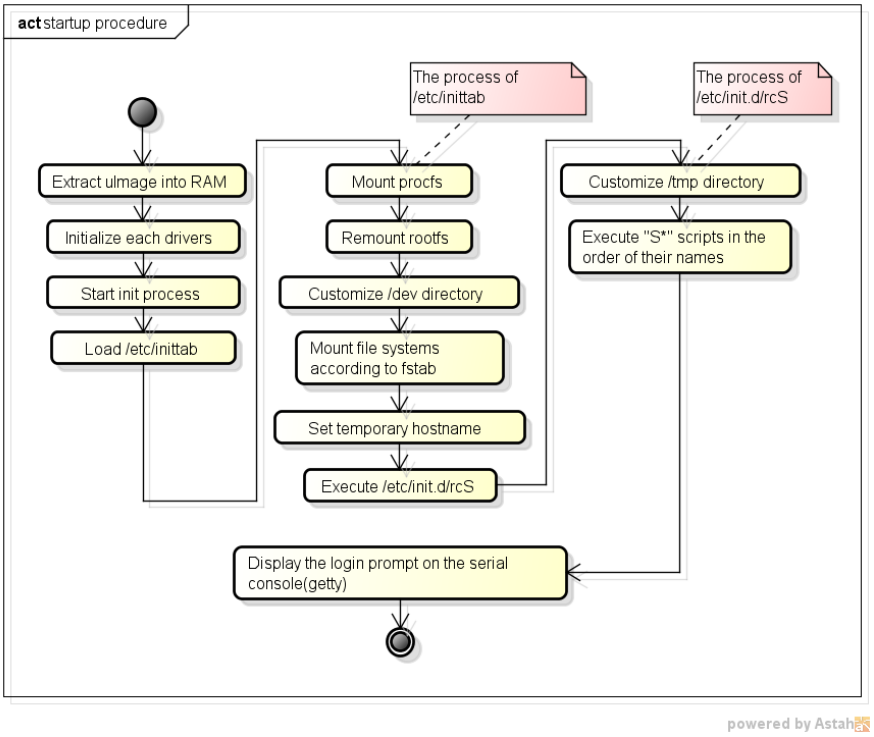
```
cd ../kernel/linux-3.2.58/
cp .config ../../build/def_linux-config
```

Issue the following commands to rebuild linux and rootfs.

```
make linux-clean
make linux
make archive
```

4.5. How to Start Linux

LB-100AN starts Linux by using the following procedure.



LB-100AN executes a script when it is started if shell scrip is allocated on /etc/init.d with a name of SXYfunc(S, 2-digit number (XY), function (func)). By customizing apps/preinstall/etc/init.d or lower level, new applications can be started.

S* scripts run in ascending order, and perform the following.

File Name	Description
S01logging	Process of starting the syslog function
S02init	Initialization process for LB-100AN Preforms the following: ● Mount the setting domain in the following directory: /etc/sysconfig ● Change the root user password
S10udev	Process of starting Kernel event control function
S12usbocd	Service initialization process for USB overcurrent detection
S20urandom	Random number initialization process
S40network	Process of starting and configuring the network interface
S50dropbear	Process of starting SSH server

A script may automaticallybe added to the target rootfs if a service other than the above is added to buildroot.

4.6. How to Use Interface

4.6.1. Network

The table below shows some of supported commands for network configuration. For wireless LAN setting, refer to the other document, *Wireless LAN Driver Specifications*.

Command	Description
ifconfig	Configures the network interface. It is used for IP address and MTU settings.
ethtool	Configures the network device driver and hardware. It is used for the link speed setting.
brctl	Configures and manages the Ethernet bridge for Linux Kernel.

When LB-100AN uses the startup script, the following interfaces are used depending on the network operation mode.

Operation Mode	Interface	Description
WIRED	eth0	Wired LAN interface
STA	ath0	Virtual interface to make wireless LAN communications (IP address is set to the interface)
	wifi0	Physical interface to control the wireless module
AP	eth0	Wired LAN interface
	br0	Bridge interface (IP address is set to the interface)
	ath0	Virtual interface to make wireless LAN communications
	wifi0	Physical interface to control the wireless module
Common	lo	Local loopback

4.6.1.1. Command Example

This section shows how to use commands to configure the network interface.

- To set the IP address of the wired LAN (192.168.0.10)

```
ifconfig eth0 192.168.0.10
```

- To fix the network link speed to 100MBASE-TX Half for wired LAN

```
ethtool -s eth0 speed 100 duplex half autoneg off
```

- To bridge wired LAN and wireless LAN (ath0) by creating a bridge interface (br0)

```
brctl addbr br0
brctl addif br0 eth0
brctl addif br0 ath0
```

4.6.1.2. Source Code Location

LB-100AN uses the wired LAN driver in the following directory.

```
kernel/linux-3.2.58/arch/arm/plat-armada/mv_drivers_lsp/mv_neta/net_dev/
```

The wireless LAN driver is based on Qualcomm Atheros's, and provided as a binary format.

4.6.2. Flash ROM

This table shows some of commands to access Flash ROM data.

Command	Description
fw_printenv fw_setenv	Shows and configures the environmental variables of U-Boot. Since LB-100AN controls the MAC address and ship-to country information for wireless LAN using the environmental variables of U-Boot, the setting can be referred and changed from Linux using this command.
flash_eraseall	Deletes the entire data in Flash ROM.
flashcp	Copies data to Flash ROM.

4.6.2.1. Command Example

This section shows how to use the commands.

- Get the ship-to country code for wireless LAN:

```
fw_printenv  country_code
```

- Change the ship-to country code to US:

```
fw_setenv country_code US
```

- Initialize the setting domain by jffs2:

```
flash_eraseall -j /dev/mtd2
```

- Update Kernel image in Flash ROM:

```
flashcp uImage /dev/mtd3
```

4.6.2.2. Source Code Location

LB-100AN uses the driver in the following directory.

```
kernel/linux-3.2.58/arch/arm/plat-armada/mv_hal/sflash
```

4.6.3. eMMC

Since Linux Kernel recognizes eMMC as a block device, eMMC can format and mount devices just like a general storage device.

4.6.3.1. Command Example

- Format the partition 1 by EXT3:

```
mkfs -t ext3 /dev/mmcblk0p1
```

- Mount the partition 1 in /tmp/mmc:

```
mkdir /tmp/mmc
mount -t ext3 /dev/mmcblk0p1 /tmp/mmc
```

4.6.3.2. eMMC Boot

This section describes setting examples to use LB-100AN using the eMMC booting.

4.6.3.2.1. Load eMMC-saved Firmware on RAM

The following is a setting example to save and start uImage and initrd.img on RAM. To use the procedure, the size of initrd.img should be 32 MBytes or smaller.

(1) Start the firmware.

(2) Format eMMC as EXT2.

```
mkfs -t ext2 /dev/mmcblk0p1
```

(3) Mount eMMC.

```
mkdir /tmp/mmc
mount -t ext2 /dev/mmcblk0p1 /tmp/mmc
```

(4) Save uImage and initrd.img on eMMC.

When TFTP server (Section 2.6) is used (xxx.xxx.xxx.xxx: IP address of TFTP server):

```
cd /tmp/mmc
tftp -g xxx.xxx.xxx.xxx -r GX01/uImage
tftp -g xxx.xxx.xxx.xxx -r GX01/ initrd.img
```

When a FAT-formatted USB memory is used:

```
mkdir /tmp/usb
mount -t vfat /dev/sda1 /tmp/usb
cp /tmp/usb/uImage /tmp/mmc
cp /tmp/usb/initrd.img /tmp/mmc
```

(5) Restart LB-100AN and start U-Boot from the serial console.

(6) Set the environmental variables.

Change and save bootcmd (environmental variable) to always start LB-100AN from eMMC.

For firmware update, save a variable separately to start LB-100AN from Flash ROM.

```
setenv bootcmd_rom $bootcmd
setenv bootcmd_mmc 'mmc rescan; ext2load mmc 0 0x2000000 uImage; ext2load mmc 0 0x2400000 initrd.img; setenv bootargs $console $mtdparts initrd=0x2400000,32M $user_debug; bootm 0x2000000; '
setenv bootcmd 'run bootcmd_mmc'
saveenv
```

Note: Make sure that environmental variables do not have a line feed character, and are

properly input on a single line.

(7) Start the firmware.

```
boot
```

4.6.3.2.1.1. Update Firmware on eMMC

The following procedure updates the firmware.

(1) Start U-Boot from the serial console.

(2) Change environmental variable to allow boot from Flash ROM and then start LB-100AN.
Section 4.1.1 also describes environmental variables to start LB-100AN from Flash ROM.

```
setenv bootcmd 'run bootcmd_rom'
boot
```

(3) Rewrite the firmware with Step (3) and (4) in Section 4.6.3.2.1.

(4) Restart LB-100AN.

```
reboot
```

4.6.3.2.2. Implement rootfs on eMMC

This section shows setting examples to create partitions in eMMC and start LB-100AN as follows. When you use this procedure, it is recommended to implement additional processes to check and rebuild rootfs before mounting rootfs to ensure completeness of roofts.

Partition	Description
mmcblk0p1	Saves uImage. U-Boot reads out uImage from the domain and loads it on RAM.
mmcblk0p2	Loads rootfs. Linux mounts the domain as rootfs.

(1) Start the firmware.

(2) Delete the partition of eMMC, and create two partitions.
Refer to the command help for fdisk operation.

```
fdisk /dev/mmcblk0
Command (m for help): d
Selected partition 1
Command (m for help): n
Select (default p):
Partition number (1-4, default 1):
First sector (2048-7553023, default 2048):
Last sector, +sectors or +size{K,M,G} (2048-7553023, default 7553023): 34815
Command (m for help): n
Select (default p):
Partition number (1-4, default 2):
First sector (34816-7553023, default 34816):
Last sector, +sectors or +size{K,M,G} (34816-7553023, default 7553023):
Command (m for help): w
```

(3) Format the partition 1 as EXT2 and the partition 2 as EXT3.

```
mkfs -t ext2 /dev/mmcblk0p1
mkfs -t ext3 /dev/mmcblk0p2
```

(4) Mount eMMC.

```
mkdir /tmp/mmc1
mkdir /tmp/mmc2
mount -t ext2 /dev/mmcblk0p1 /tmp/mmc1
mount -t ext3 /dev/mmcblk0p2 /tmp/mmc2
```

(5) Save uImage and rootfs.

When TFTP server (Section 2.6) is used (xxx.xxx.xxx.xxx: IP address of TFTP server):

```
cd /tmp/mmc1
tftp -g xxx.xxx.xxx.xxx -r GX01/uImage
cd /tmp/mmc2
tftp -g xxx.xxx.xxx.xxx -r GX01/ rootfs.tar
tar xpf rootfs.tar
```

When a FAT-formatted USB memory is used:

```
mkdir /tmp/usb
mount -t vfat /dev/sda1 /tmp/usb
cp /tmp/usb/uImage /tmp/mmc1
tar xpf /tmp/usb/rootfs.tar -C /tmp/mmc2
```

(6) Restart LB-100AN, and start U-Boot from the serial console.

(7) Set the environmental variables.

Save environmental variables separately to start LB-100AN from Flash ROM for firmware update. Change and save bootcmd (environmental variable) to always start LB-100AN from eMMC.

```
setenv bootcmd_rom $bootcmd
setenv bootcmd_mmc 'mmc rescan; ext2load mmc 0 0x2000000 uImage; setenv bootargs $console $mtdparts root=/dev/mmcblk0p2 rootfstype=ext3 rootwait; bootm 0x2000000; '
setenv bootcmd 'run bootcmd_mmc'
saveenv
```

Note: Make sure that environmental variables do not have a line feed character, and are properly input on a single line.

(8) Start the firmware.

```
boot
```

4.6.3.2.2.1. Update Firmware on eMMC

Implement the following procedure to update the firmware.

(1) Start U-Boot from the serial console.

(2) Change environmental variable to allow boot from Flash ROM and then start LB-100AN. Section 4.1.1 also describes the environmental variables to start LB-100AN from Flash ROM.

```
setenv bootcmd 'run bootcmd_rom'
boot
```

(3) Rewrite the firmware with Step (4) and (5) in Section 4.6.3.2.2.

(4) Restart LB-100AN.

```
reboot
```

4.6.3.3. Source Code Location

LB-100AN uses the driver saved in the following directory.

```
kernel/linux-3.2.58/drivers/mmc
```

The following tables show the related Linux Kernel settings.

- Device Driver

Description	Definition	Value
MMC/SD/SDIO card support	CONFIG_MMC	y
MMC block device driver	CONFIG_MMC_BLOCK	y

- File System

Description	Definition	Value
Second extended fs support	CONFIG_EXT2_FS	y
Ext3 journalling file system support	CONFIG_EXT3_FS	y

4.6.3.4. How to Change eMMC to Pseudo SLC

eMMC of LB-100AN can change to pseudo SLC (pSLC). Issue the following commands from the console of U-Boot, turn off and on LB-100AN; and MLC will change to pSLC. As eMMC requires the hardware reset, wait for a few seconds after removing the power cable from LB-100AN until feeding power again.

```
mmc rescan
mmchwpart
```

The above process does not allow pSLC to change back to MLC.

4.6.4. UART for External Interface

Linux Kernel recognizes UART for an external interface as /dev/ttyS1. The table below shows one of commands used to configure UART.

Command	Description
stty	Changes control settings of an associated device depending on a specified argument. For the supported setting range, refer to the other document, *Software Functional Specifications*.

4.6.4.1. Command Example

- Get a list of /dev/ttyS1 settings.

```
stty -F /dev/ttyS1 –a
```

- Change the /dev/ttyS1 speed to 9600.

```
stty -F /dev/ttyS1 speed 9600
```

4.6.4.2. Source Code

LB-100AN uses the driver in the following directories.

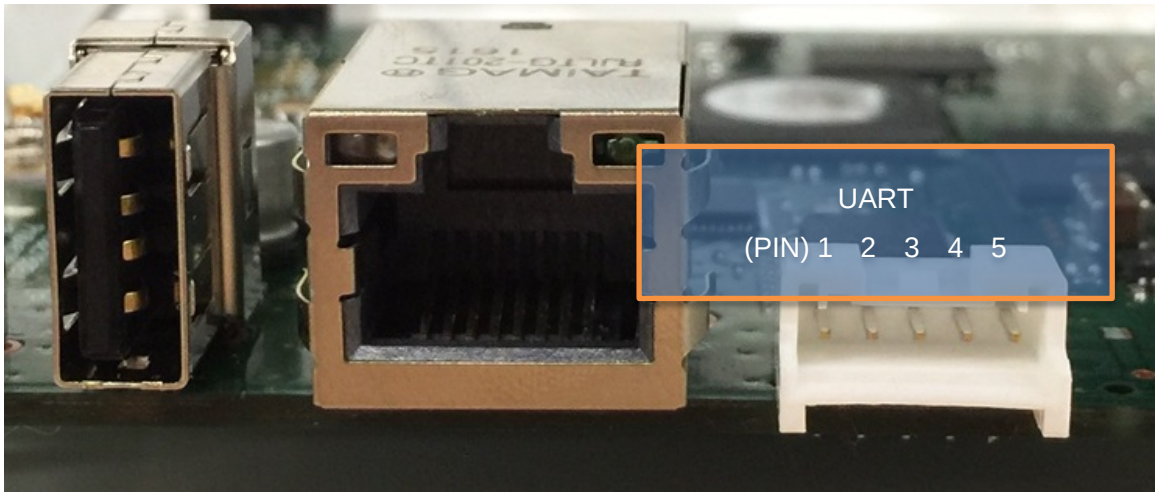
```
kernel/linux-3.2.58/drivers/tty
kernel/linux-3.2.58/drivers/tty/serial
```

The table shows the related Linux Kernel setting.

Description	Definition	Value
8250/16550 and compatible serial support	CONFIG_SERIAL_8250	y

4.6.4.3. PIN Location

The photo below shows the PIN location of UART for an external interface. The leftmost pin is PIN 1.



PIN	Function
1	TXD
2	RTS
3	RXD
4	CTS
5	GND

4.6.4.4. Attention

When RTS/CTS are enabled, the flow control does not work properly with the serial cable between RTS and CTS pins.

4.6.5. USB

Bus power and overcurrent are configured and detected by using GPIO. For GPIO and MPP settings, refer to the other document, *Software Functional Specifications*.

4.6.5.1. Source Code Location

The driver of host controller is saved in the following directory.

kernel/linux-3.2.58/drivers/usb/host/ehci-hcd.c

The following table shows the related Linux Kernel settings. LB-100AN uses Full/Low-speed devices with EHCI host controller driver.

Description	Definition	Value
Support for Host-side USB	CONFIG_USB	Y
EHCI HCD (USB 2.0) support	CONFIG_USB_EHCI_HCD	Y
Root Hub Transaction Translators	CONFIG_USB_EHCI_ROOT_HUB_TT	Y

4.6.6. GPIO

4.6.6.1. Configuration Method Using sysfs

The following functions show how to control an interface connected to GPIO via sysfs. For other functions controlled by GPIO, refer to the other document, *Software Functional Specifications*.

Function	GPIO Number	Direction
Push Switch (1: OFF, 0: ON)	6	IN
LED2 (Red) (1: ON, 0: OFF)	45	OUT

4.6.6.1.1. Push Switch

(1) Export GPIO control of the push switch.

```
echo 6 > /sys/class/gpio/export
```

(2) Get the status of push switch.

If the push switch is ON, the command outputs 0.

```
cat /sys/class/gpio/gpio6/value
```

(3) To finish control via sysfs, use the command below to release the GPIO control.

```
echo 6 > /sys/class/gpio/unexport
```

4.6.6.1.2. LED2 (Red)

(1) Obtain GPIO control of LED2 (Red).

```
echo 45 > /sys/class/gpio/export
```

(2) Change the GPIO direction of LED2 (Red) to OUT.

```
echo out > /sys/class/gpio/gpio45/direction
```

(3) Turn on LED2 (Red).

```
echo 1 > /sys/class/gpio/gpio45/value
```

(4) Turn off LED2 (Red).

```
echo 0 > /sys/class/gpio/gpio45/value
```

(5) To finish controlling via sysfs, use the following command to release the GPIO control.

```
echo 45 > /sys/class/gpio/unexport
```

4.6.6.2. Initialization Process

Default values of the GPIO register are set using the startup process of Linux.

If you wish to change the default, update the following Linux Kernel file.

arch/arm/mach-armada370/armada_370_family/boardEnv/mvBoardEnvSpec.h

The following shows the settings.

```
/* 0-31 */
#define ALMACH_88F6710_GPP_OUT_ENA_LOW      (~(BIT31 | BIT16 | BIT15 | BIT14 | ¥
                                            BIT13 | BIT12 | BIT11 | BIT10 | ¥
                                            BIT9 | BIT8 | BIT7 | BIT5 | ¥
                                            BIT4 | BIT3 | BIT2))
/* 32-63 */
#define ALMACH_88F6710_GPP_OUT_ENA_MID      (~(BIT31 | BIT30 | BIT29 | BIT28 | ¥
                                            BIT27 | BIT26 | BIT25 | BIT24 | ¥
                                            BIT23 | BIT22 |¥
                                            BIT14 | BIT13 | BIT12 | ¥
                                            BIT11 | BIT7 | BIT5 | BIT0))
/* 64-65 */
#define ALMACH_88F6710_GPP_OUT_ENA_HIGH     (~(BIT1 | BIT0))

#define ALMACH_88F6710_GPP_OUT_VAL_LOW      0x0
#define ALMACH_88F6710_GPP_OUT_VAL_MID      BIT12 | BIT28
#define ALMACH_88F6710_GPP_OUT_VAL_HIGH     0x0

#define ALMACH_88F6710_GPP_POL_LOW          0x0
#define ALMACH_88F6710_GPP_POL_MID          0x0
#define ALMACH_88F6710_GPP_POL_HIGH  0x0
```

Definition	Description
ALMACH_88F6710_GPP_OUT_ENA _(LOW/MID/HIGH)	Definition to set the GPIO direction. Sets MPP to OUT by bit.
GPP_OUT_VAL _(LOW/MID/HIGH)	Definition to set the GPIO default. Sets MPP to 1 by bit.
GPP_POL _(LOW/MID/HIGH)	Definition to set the polarity of IN signals for the GPIO. Sets MPP to reverse the value by bit.

LOW/MID/HIGH have the following MPP.

Definition	MPP
LOW	0 to 31
MID	32 to 63
HIGH	64 to 65

4.6.6.2.1. GPIO Default Setting Example

The following functions show examples of how to configure the GPIO default values. For other functions controlled by GPIO, refer to the other document, *Software Functional Specifications*.

Function	GPIO Number	MPP	Direction
Push Switch (1: OFF, 0: ON)	6	MPP[6]	IN
LED2(Red) (1: ON, 0: OFF)	45	MPP[45] 13	OUT

4.6.6.2.2. Push Switch

This section describes how to get the push switch status by reversing it.

(1) Change the definition to set the polarity of the push switch.

Since MPP[6] is categorized as LOW, change the setting as follows.

```
#define ALMACH_88F6710_GPP_POL_LOW BIT6
```

(2) Rebuild Linux, and update the firmware.

After the firmware updates, the register value of push switch is reversed. Values shown in the procedure of Section 4.6.6.1.1 will not be changed because of the GPIO driver specifications.

4.6.6.2.3. LED2 (Red)

This section describes how to light on LED2 (Red) when Linux starts.

(1) Change the definition to set the GPIO value of LED2 (Red).

Since MPP[45] is categorized as MID and MID Bit is 13, change the definition as follows.

```
#define ALMACH_88F6710_GPP_OUT_VAL_MID      BIT12 | BIT28 | BIT13
```

(2) Rebuild Linux, and update the firmware.

When LB-100AN starts up after updating the firmware, LED2 turns red.

4.6.6.3. Source Code Location

The GPIO library driver is saved in the following directory.

kernel/linux-3.2.58/drivers/gpio

The following table shows the related Linux Kernel setting.

Description	Definition	Value
GPIO Support	CONFIG_GPIOLIB	y
/sys/class/gpio/... (sysfs interface)	CONFIG_GPIO_SYSFS	y

4.7. SSH Password

LB-100AN, as written in Section 4.5, changes the root user password using the following file.

apps/preinstall/etc/init.d/S02init

To change SSH password, change the PASSWD variable in the file. The following example changes the password to "password".

```
SetRootPasswd(){
        if [ ! -f /tmp/etc/passwd ]; then
                cp /etc/default/passwd /tmp/etc/passwd
        fi

        if [ ! -f /tmp/etc/shadow ]; then
                cp /etc/default/shadow /tmp/etc/shadow
        fi

        PASSWD=password
        echo "root:${PASSWD}" | chpasswd --md5
}
```

4.8. Production Process

For LB-100AN, the PCB is verified using Silex original application after the network booting. After the verification, your product firmware is written and checked on the PCB. This chapter gives precautions you may need to consider before the mass production of your product using LB-100AN.

4.8.1. Attention to Change U-Boot

If you do not plan to use Silex's U-Boot image in your actual product, you need to provide your customized specifications and U-Boot image file (u-boot.bin) in advance.

The network booting function is used for verifications during the production process in Silex. **If the function does not work, the production may stop. For the production using pSLC instead of eMMC, the change procedure written in *Software Functional Specifications* should be enabled.** If eMMC needs to be changed to pSLC in a way not described in the document, please consult with Silex.

4.8.2. Ship-to Country Setting for Wireless LAN

To sell the created products in US, you need to design them not to allow the users to change the ship-to country setting on their own in order to conform to FCC regulation. **The ship-to country information must be a text string** that is written in the other document, *Software Functional Specifications*. If you wish to write the ship-to country information using a different method, please consult with Silex.

4.8.3. Setting Domain

Please consult with Silex for the mass production procedures if partitions and the format of setting domain need to be changed from those described in the functional specification document. If the following file, written in the functional specification document, does not exist in the setting domain, please ask Silex for support.

/etc/sysconfig/system.conf

4.8.4. eMMC Booting

Silex's mass production test is designed to write the firmware on Flsh ROM. **If you want to write the firmware on eMMC, please consult with Silex.**

4.8.5. eMMC Default Setting

For mass production procedure at Silex, if eMMC partition and format need to be changed from those specified in functional specification document, please consult with Silex.

4.8.6. Firmware Writing

The firmware is written using the following procedure on Flash ROM during the production process at Silex.

(1) Use the test tool and start the firmware update checking.
(2) LB-100AN downloads shell scripts from TFTP server of the test PC by TFTP, and execute them on the RAM disk.
(3) The shell scripts download and write the firmware in Flash ROM.
(4) If the shell scripts is successfully executed, it returns "Success" to the test tool.
(5) Restart LB-100AN.

If you wish to change the partitions or mapping in Flash ROM, review the shell scripts used for Silex production tool. Refer to the following file in the source code for a shell script sample implemented by the procedure herein.

build/Script

4.8.7. Operation after Firmware Writing

Silex production conducts the following processes after writing your product firmware.

- Re-testing of PCB by sampling
- Confirming of your firmware if it is properly written

The processes above require the following procedures.

- To start network booting of LB-100AN when the firmware has already been written
- To acquire information that can give the firmware version or verify if the appropriate firmware is written