# KUROBOX/PRO Product Specifications

Product name: KUROBOX/PRO

# 1. Contents

## 2. Product Overview

The Kurobox pro is a Linux-Box self-work kit that enables access from multiple PCs connected to a LAN, and is equipped with 1 10/100/1000Mbps Ethernet port and 2 USB2.0 A connectors.

## 3. Special Features

- GigaBit Ethernet is supported.
- Auto MDIX is supported.
- Equipped with 2 USB 2.0 ports
- System temperature and fan revolutions can be obtained via the microcomputer. Also, fan revolutions can be controlled as desired.
  (However, this requires creation of the application by the user.)
- The following can be performed easily by removing the front panel.
  (a.) UART can be accessed.
  (b.) PCI-Express x1 can be used.
  (c.) SATA ports for extension can be accessed.
  (d.) I2C can be accessed.
  (e.) Microcomputer port can be accessed.
  (f.) GPIO can be accessed.
- you can create a spindle-less NAS by connecting a USB silicone disc and 2.5 inch SATA Flash and configuring shared settings.

***If you want to enjoy hacking this product, please do not read this product specification document.***

## 4. Individual Specifications

| Item | Details |
|---|---|
| **<Control Section>** | |
| CPU | Marvell 88F5182 (400MHz) |
| Memory | 128MB (32Mbits x 16bit x 2) |
| Data Storage | NOR Flash 256KiB – Buswidth 8bit(for storing u-boot)<br>NAND Flash 256MiB – Buswidth 8bit(for rootfs) |
| Interface | |
| **<Network Interface Section>** | |
| Interface | (10BASE-T)IEEE802.3<br>(100BASE-TX)IEEE802.3u<br>Conforms to (1000BASE-T)IEEE802.3ab |
| Type of Connector | RJ-45 Type 8 pole connector (NIC Mode) x1 |
| Schematic Diagram | Fig. 5.4.1 |
| Transmission Channel Coding Method | (10BASE-T) Manchester Coding<br>(100BASE-TX) 4B5B + MLT-3<br>(1000BASE-T ) 8B1Q4 + PAM5 |
| Access Method | CSMA/CD |
| Transmission Rate | (Auto sense, auto MDIX)<br>10Mbps HDX/FDX<br>100Mbps HDX/FDX<br>1000Mbps FDX |
| Type of Cable | (10BASE-T) category 3,4,5 2-pair or 4-pair UTP cable<br>(100BASE-TX) category 5 2-pair or 4-pair UTP cable<br>(1000BASE-T) Enhanced category 5 4-pair UTP cable |
| **<SATA Interface section>** | |
| Interface | Complies with SATA1.0 |
| Data Transmission Speed | Maximum transmission speed 3 Gbps |
| **<HDD>** | |
| Unit | None (3.5inch SATA HDD can be installed) |
| **<USB Interface section>** | |
| Interface | USB2.0/1.1 |
| Data Transmission Speed | 480Mbps |
| Type of Connector | Series A Rear ×2 |
| **<LED Section>** | |
| PowerLED | Green |

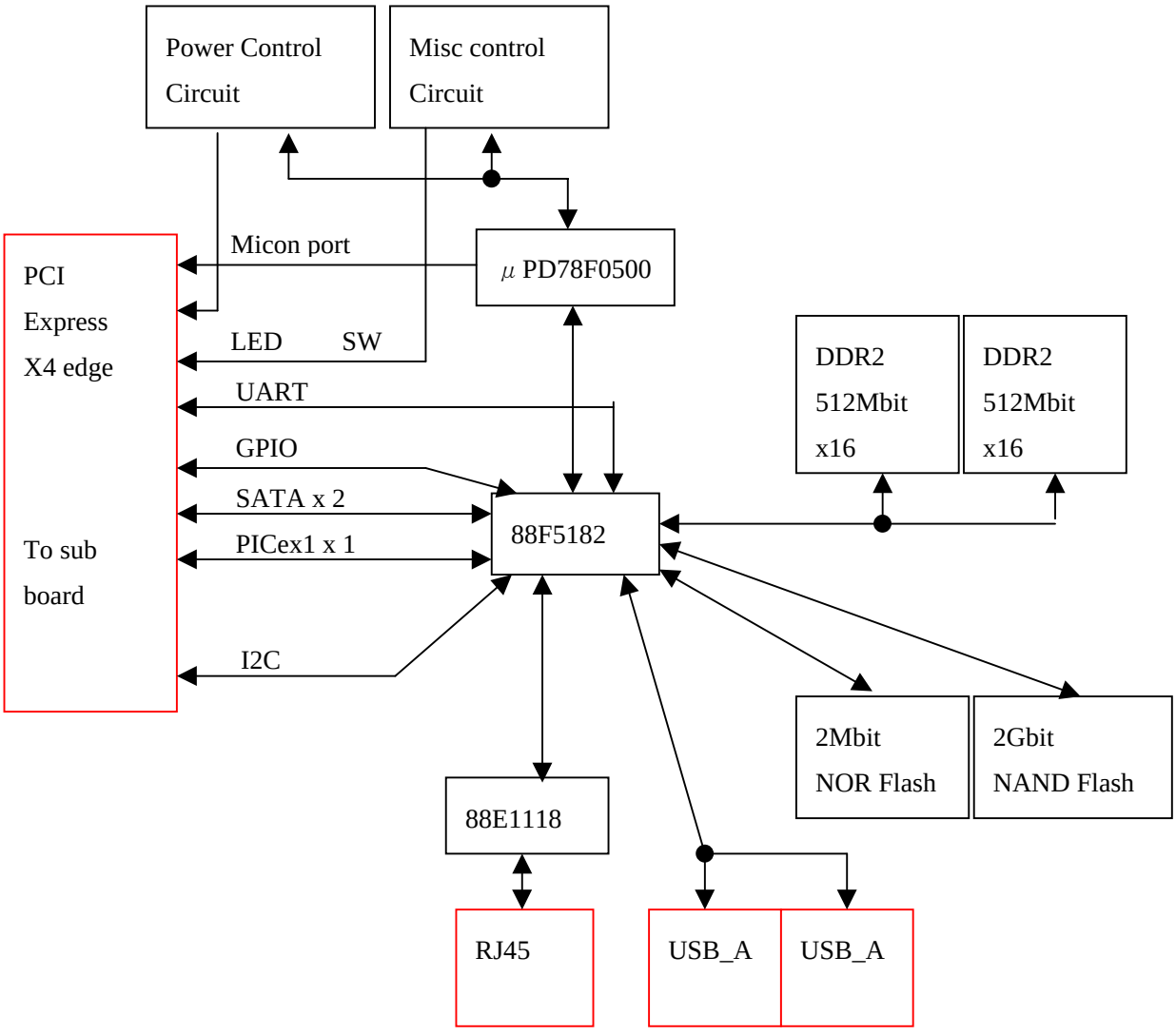
| | |
|---|---|
| LINK/ACT LED | Green |
| INFORMATION | Yellow |
| ERROR | Red |
| **<Other I/F>** | |
| SATA (Front) | x1 (CN1) |
| I2C | x1 (CN2) |
| PCI Express x1 | x1 (CN5) |
| UART | x1 (CN6) |
| GPIO(09, 10) | x1 (CN7) |
| Microcomputer port | x1 (SW3) |

# 5. HW Specifications

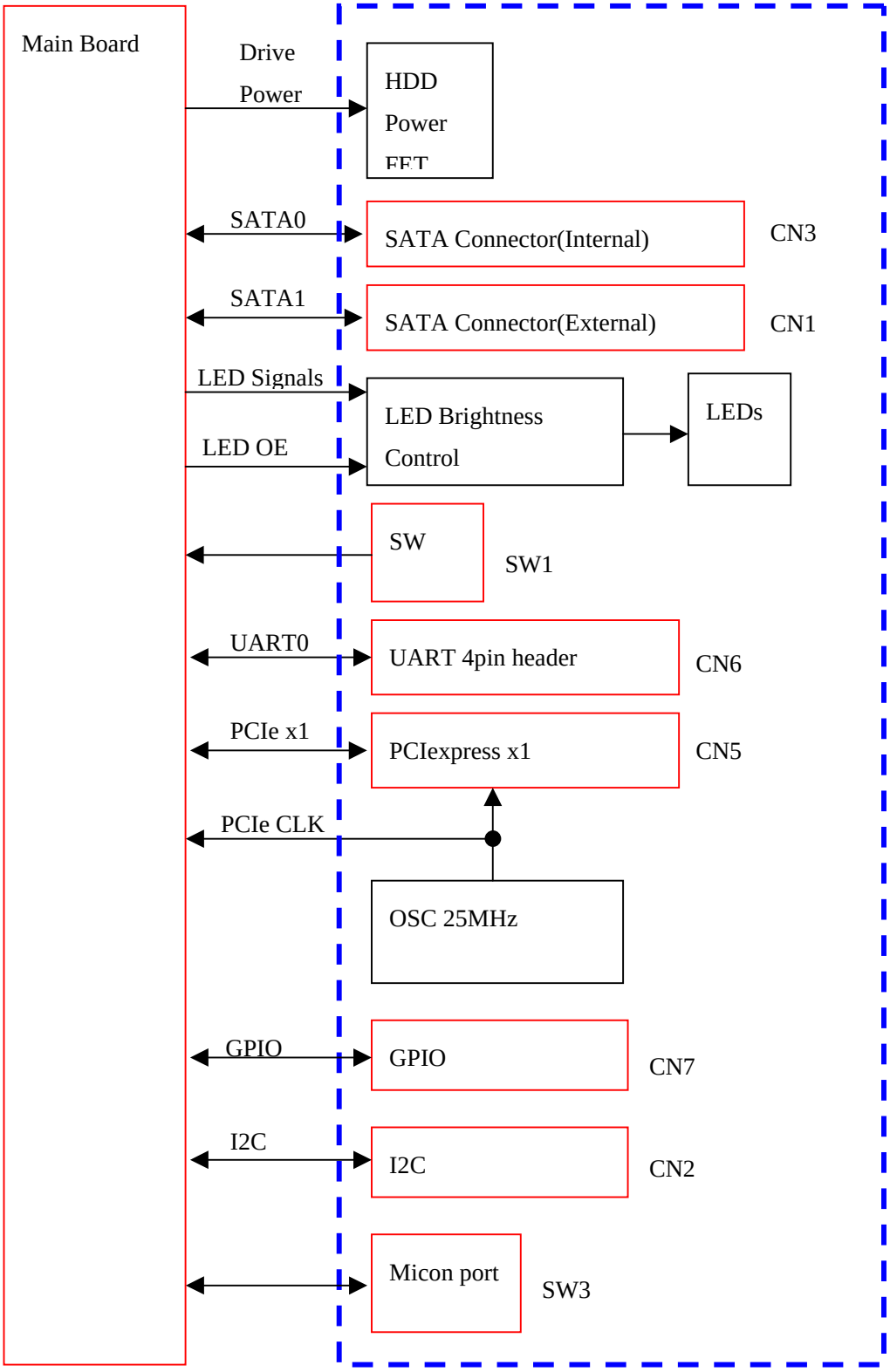
## 5.1. HW Block Diagram

### 5.1.1. Function Block Diagram

**Main Board**

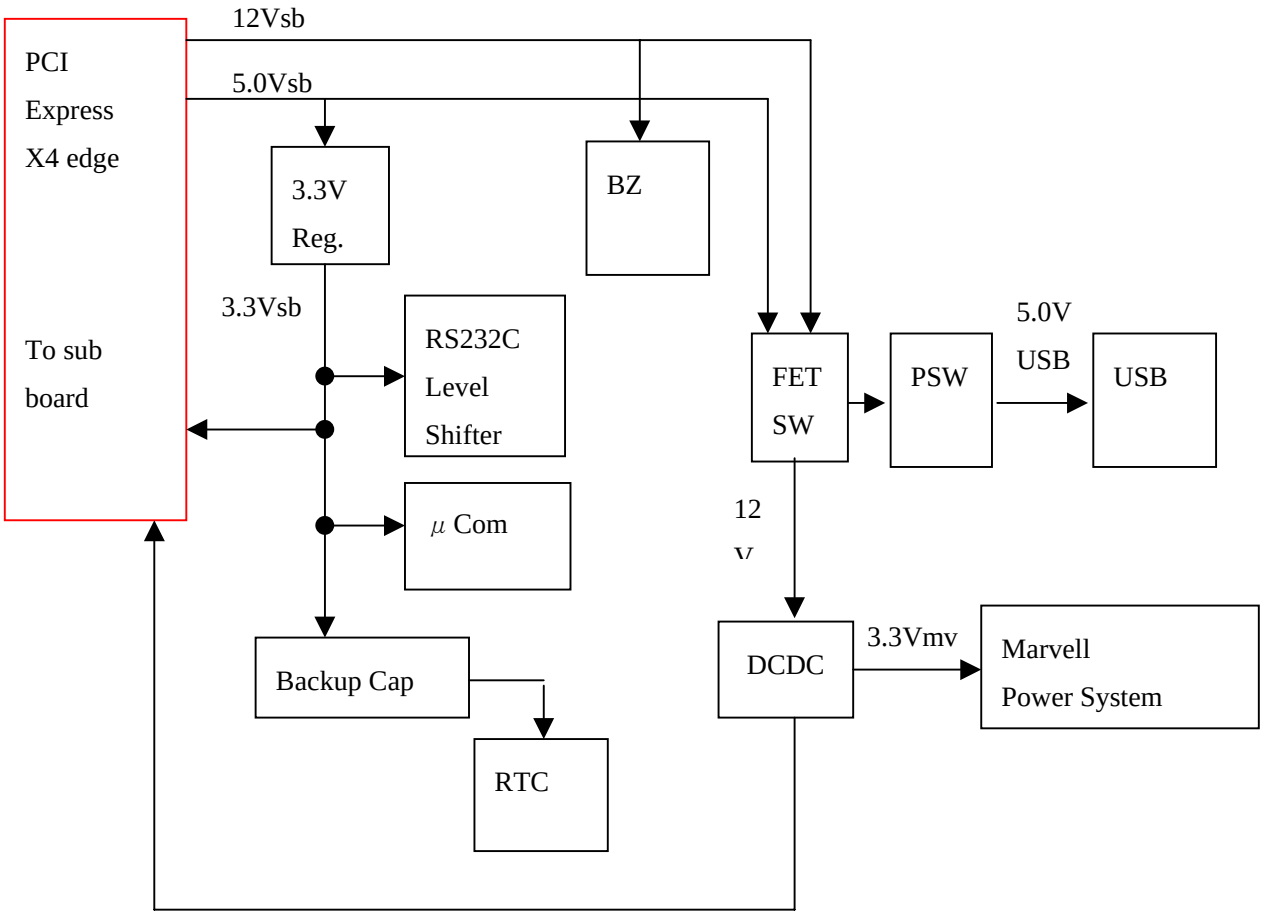


**Sub Board**

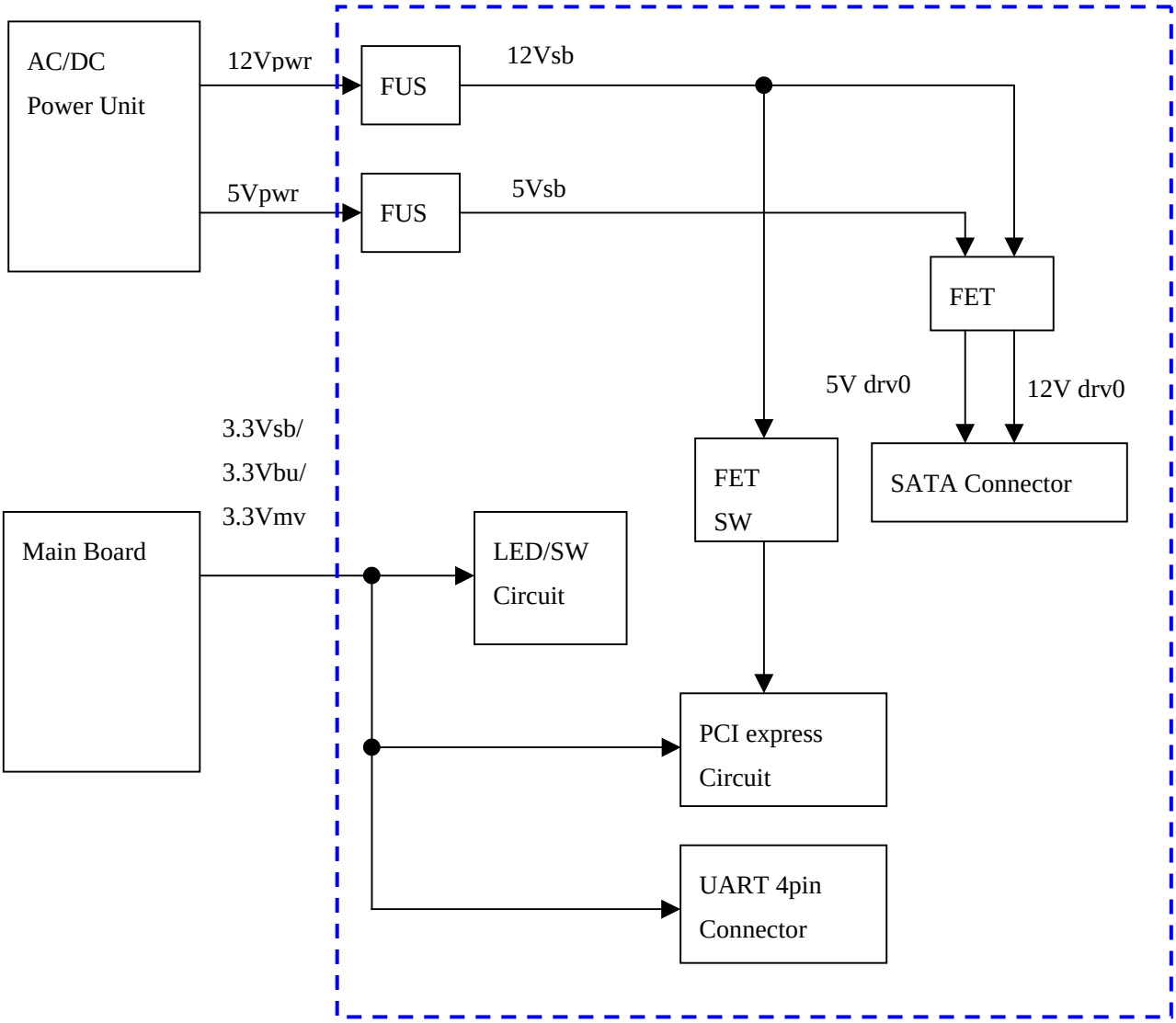

Main Board
Drive
Power
HDD
Power
SATA0
SATA Connector(Internal)
CN3
SATA1
SATA Connector(External)
CN1
LED Signals
LED OE
LED Brightness
Control
LEDs
SW
SW1
UART0
UART 4pin header
CN6
PCIe x1
PCIexpress x1
CN5
PCIe CLK
OSC 25MHz
GPIO
GPIO
CN7
I2C
I2C
CN2
Micon port
SW3
CN4

5.1. 2 Power Block Diagram

**Main Board**

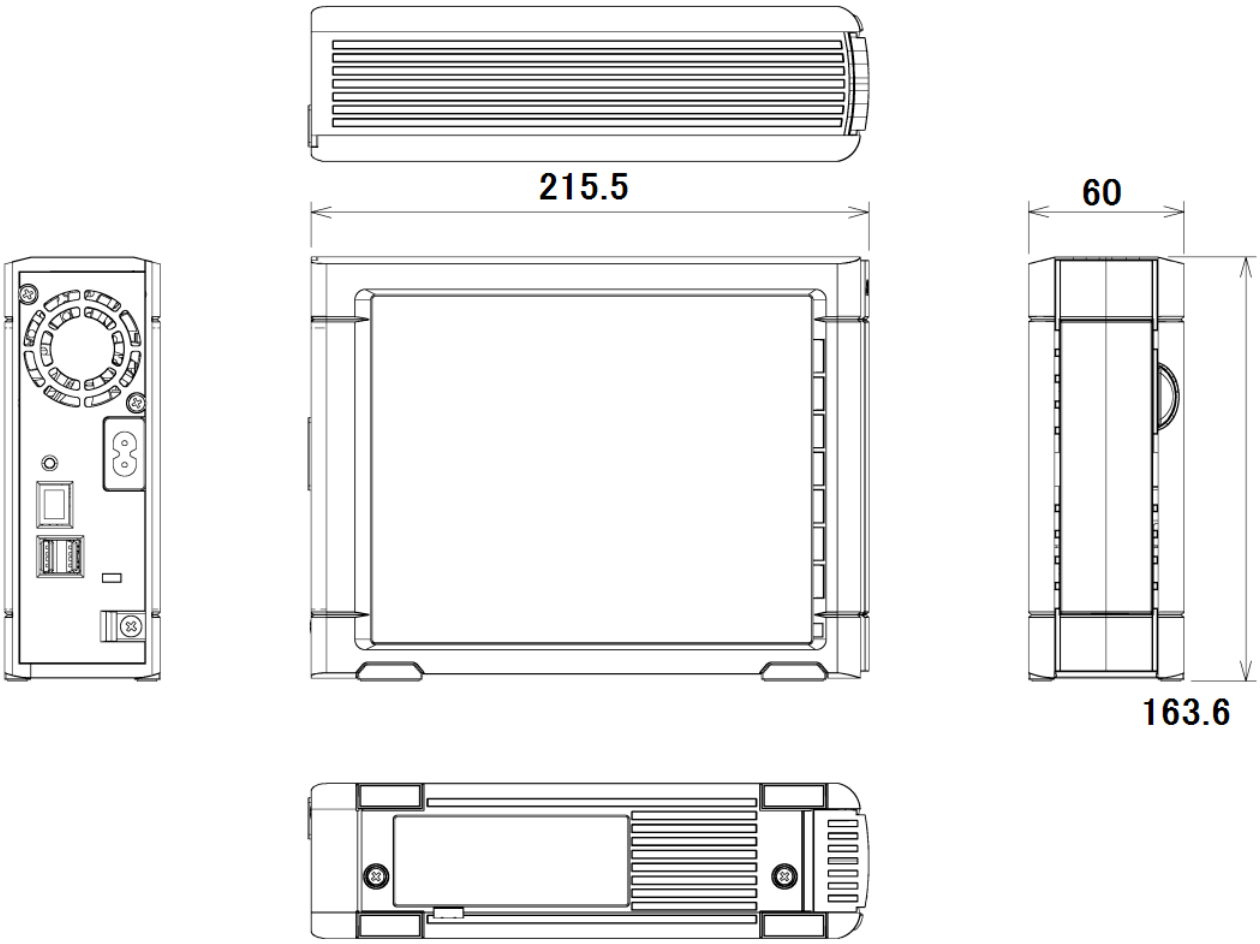


**Sub Board**


AC/DC
Power Unit
12Vpwr
FUS
12Vsb
5Vpwr
FUS
5Vsb
FET
5V drv0
12V drv0
3.3Vsb/
3.3Vbu/
3.3Vmv
FET
SW
SATA Connector
Main Board
LED/SW
Circuit
PCI express
Circuit
UART 4pin
Connector

## 5.2. External Appearance

3-Side Expanded Diagram



## 5.3. Switches

### 5.3.1. Power Switch

| | |
|---|---|
| Outline | Software switch for switching the power on/off |
| Method for switching power ON | • Connect the power cable correctly, and hold down the power switch for 0.3 seconds when the power is OFF to start-up the system |
| Method for forcing power OFF | • When the power is on (possible even if the system is not started), hold down the switch for 9 seconds to force the power to switch OFF |

### 5.3.2. Initialize Switch

| | |
|---|---|
| Outline | • Switch for initializing the HDD |
| Method for initializing the HDD | • After the system has started normally, hold down the initialize switch for 5 seconds to initialize and mount the internal HDD. |
| Notes | • If the internal HDD (/dev/sda) is already formatted in xfs format, formatting will not be performed. However, if you set force_format=yes in /etc/melco/info, you can format irrespective of whether the HDD is currently formatted in xfs format or not.<br>• If the internal HDD (/dev/sda) already has partitions, those partitions will be deleted.<br>• You can easily NAS-ize a usb-disk by connecting a usb-disk when the internal HDD is not connected, and performing the operation described above. (However, caution is required as the usb-disk will be formatted in xfs format, irrespective of the usb-disk format.) ) |

## 5.4. Ports

### 5.4.1. Ethernet Ports

| | |
|---|---|
| No. of Ports | 1 port on back of product |
| Type of Connector | Fig. 5.4.1 |

- Connector connection 1000BASE-T

| Pin Number | Signal | Signal Function |
|---|---|---|
| 1 | DA+ | Data A(+) |
| 2 | DA- | Data A(-) |
| 3 | DB+ | Data B(+) |
| 4 | DC+ | Data C(+) |
| 5 | DC- | Data C(-) |
| 6 | DB- | Data B(-) |
| 7 | DD+ | Data D(+) |
| 8 | DD- | Data D(-) |

- Connector connection 10BASE-T/100BASE-TX

| Pin Number | Signal | Signal Function |
|---|---|---|
| 1 | TD+ | Transmit data (+) |
| 2 | TD- | Transmit data (-) |
| 3 | RD+ | Receive data (+) |
| 4 | (Not Use) | Not use |
| 5 | (Not Use) | Not use |
| 6 | RD- | Receive data (-) |
| 7 | (Not Use) | Not use |
| 8 | (Not Use) | Not use |

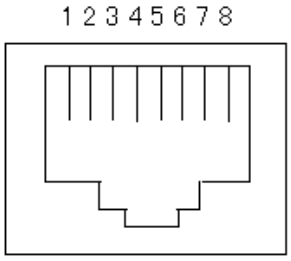



Fig. 5.4.1

Ethernet Port Pin Arrangement (from cross sectional view)

**Network Specifications**

| | |
|---|---|
| Physical Layer | |
| Supported Link Speeds | 10Mbps(HALF/FULL) (10BASE-T IEEE802.3)<br>100Mbps(HALF/FULL) (100BASE-TX IEEE802.3u)<br>1000Mbps (FULL)(1000BASE-Tconforms to IEEE802.3ab) |
| AutoNegotiaion | Supported (Link speed decided automatically) |
| Auto MDIX | Supported (cross/straight detected automatically) |

5.4.2. USB Ports

| | |
|---|---|
| Number of Ports | 2 port on back of product |
| Connectable Devices | USB 2.0 compatible devices |
| Type of Connector | Series A (Refer to figure 9-2 for pin arrangement) |

- Connector Connection

| Pin Number | Signal | Signal Function |
|---|---|---|
| 1 | Vbus | 5V Power supply |
| 2 | D- | Data (-) |
| 3 | D+ | Data (+) |
| 4 | GND | Earth |

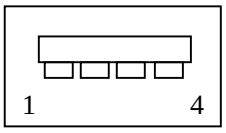



Fig. 5.4.2
USB connector - Pin Arrangement
(from cross sectional view)

5.4.3. I2C Port (CN2)

| | |
|---|---|
| Number of Ports | 1 port on sub board |
| Type of Connector | Serial pin header (4 pin) |

- Connector Connection

| Pin Number | Signal | Signal Function |
|---|---|---|
| 1 | SCK | Clock |
| 2 | SDA | Data |
| 3 | VCC | 3.3V Power supply |
| 4 | GND | Earth |

### 5.4.4. GPIO Port (CN7)

| | |
|---|---|
| Number of Ports | 1 port on sub board |
| Type of Connector | Serial pin header (4 pin) |

- Connector Connection

| Pin Number | Signal | Signal Function |
|---|---|---|
| 1 | IO(09) | GPIO9 |
| 2 | IO(10) | GPIO10 |
| 3 | VCC | 3.3V Power supply |
| 4 | GND | Earth |

### 5.4.5. UART Port (CN6)

| | |
|---|---|
| Number of Ports | 1 port on sub board |
| Type of Connector | Serial pin header (4 pin) |

- Connector Connection

| Pin Number | Signal | Signal Function |
|---|---|---|
| 1 | TXD | Transmit |
| 2 | RXD | Receive |
| 3 | VCC | 3.3V Power supply |
| 4 | GND | Earth |

## 5.5. Power Supply Specifications

| | |
|---|---|
| Input Rating | 100-220[V] (50Hz/60Hz) |
| Rated Output | 12 [V]<br>5 [V] |
| UPS Compatibility | Compatible with the rectangular wave type output that is output when a commercial power type UPS etc is on battery backup (15 minutes) |
| UL Information | |

### 5.5.1. Power Mode

| | |
|---|---|
| Outline | This product has 2 conditions, power ON and power OFF |
| Power OFF | When the power supply is off to devices other than the microcomputer and RTC |
| Power ON | When all of the power is ON |

# 6. Software Specifications

## 6.1. Main Software Versions

| | |
|---|---|
| Kernel Version | 2.6.12.6 |
| gcc version | 3.4.4 (CodeSourcery ARM 2005q3-2) |
| glibc | 2.3.6<br>However, only the following libraries are installed.<br>ld-2.3.6.so<br>libBrokenLocale-2.3.6.so<br>libSegFault.so<br>libanl-2.3.6.so<br>libc-2.3.6.so<br>libcidn-2.3.6.so<br>libcrypt-2.3.6.so<br>libdl-2.3.6.so<br>libm-2.3.6.so<br>libnsl-2.3.6.so<br>libnss_compat-2.3.6.so<br>libnss_dns-2.3.6.so<br>libnss_files-2.3.6.so<br>libnss_hesiod-2.3.6.so<br>libnss_nis-2.3.6.so<br>libnss_nisplus-2.3.6.so<br>libpthread-2.3.6.so<br>libresolv-2.3.6.so<br>librt-2.3.6.so<br>libutil-2.3.6.so |
| busybox | 1.4.1<br>The following applets are enabled.<br>• archival utilities<br>ar, bunzip2, cpio, gunzip, gzip, rpm2cpio, rpm, tar, uncompress, unlzma, unzip<br>• coreutils<br>basename, cal, cat, catv, chgrp, chmod, chown, chroot, cksum, cmp, comm, cp, cut, date, dd, df, diff, dirname, dos2unix, du, echo, env, expr, false, fold, head, hostid, id, install, length, ln, logname, ls, md5sum, mkdir, mkfifo, mknod, mv, nice, nohup od, printenv, printf, pwd, realpath, rm, rmdir, seq, sha1sum sleep, sort, stat, stty, sum, sync, tail, tee, test, touch, tr, |

| | |
|---|---|
| | true, tty, uname, uniq, usleep, uudecode, uuencode, watch, wc, who, whoami, yes<br>• console utilities<br>chvt, clear, deallocvt, dumpkmap, loadfont, loadkmap, openvt reset, resize, setconsole, setkeycodes, setlogcons<br>• debian utilities<br>mktemp, pipe_progress, readlink, run_prrts, start_stop_daemon, which<br>• editors<br>awk, ed, patch, sed, vi<br>• finding utilities<br>find, grep, xargs<br>• init utilities<br>init, reboot, halt<br>• login/password management utilities<br>addgroup, delgroup, adduser, deluser, getty, login, passwd, su sulogin, vlock<br>• ext2 fs progs<br>chattr, fsck, lsattr<br>• linux module utilities<br>insmod, rmmod, modprobe<br>• linux system utilities<br>dmesg, fbset, fdflush, fdformat, fdisk, freeramdisk, fsck.minix mkfs.minix<br>• minix filesystem support<br>getopt, exdump, hwclock, ipcrm, ipcs, losetup, mdev, mkswap more, mount, pivot_root, rdate, readprofile, setarch, swaponoff switch_root, umount<br>• miscellaneous utilities<br>adjtimex, bbconfig, crond, crontab, dc, eject, last, less hdparm, makedevs, mountpoint, mt, nmeter, raidautorun readahead, runlevel, rx, strings, setsid, taskset, time, watchdog<br>• networking utilities<br>arp, arping, dnsd, ether-wake, fakeidented, ftpget, ftpput hostname, httpd, ifconfig, ifupdown, inetd, ipaddr, iplink iproute, iptunnel, iprule, ipcalc, nameif, nc, netstat, nslookup ping, route, telnet, telnetd, tftp, traceroute, vconfig, wget zcip<br>• process utilities |

| | |
|---|---|
| | free, fuser, kill, killall, killall5, pidof, ps, renice,<br>bb_sysctl, top, uptime<br>• shells<br>ash<br>• runit utilities<br>chpst, setuidgid, envuidgid, envdir<br>softlimit |
| mtd-utils | 1.00 |
| bash | 2.05b |
| xfs-progs | 2.5.6 |
| e2fsprogs | 1.27 |
| samba | 3.0.21c |
| libtermcap | 2.0.8 |
| libncurses | 5.2 |
| sl | -- |

# 7. Setup Procedure

## 7.1. Flow of Operation for First Startup

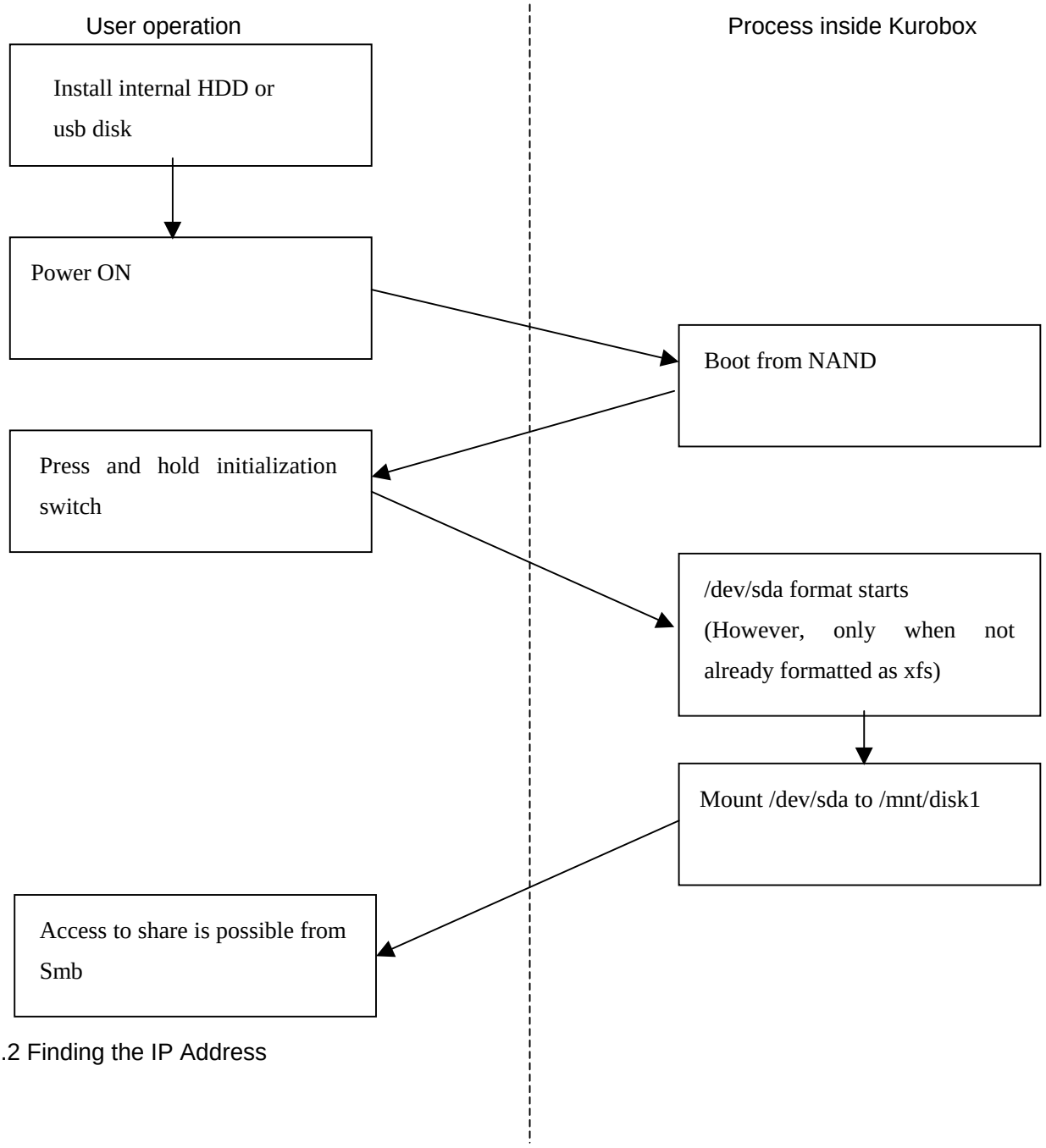
After purchasing your Kurobox and switching the power on, it will start up in the following condition.

| | |
|---|---|
| uImage | /dev/mtdblock1(NAND device) |
| rootfs | /dev/mtdblock2 (NAND device) |
| smb shared directory | mtd_device (/dev/mtdblock3 mountpoint: /mnt/mtd)<br>share(/dev/sda mountpoint /mnt/disk1) |
| telnet | Connection possible |

However, at this point only mtd_device is actually accessible.

The following operation is required to make the share directory useable.

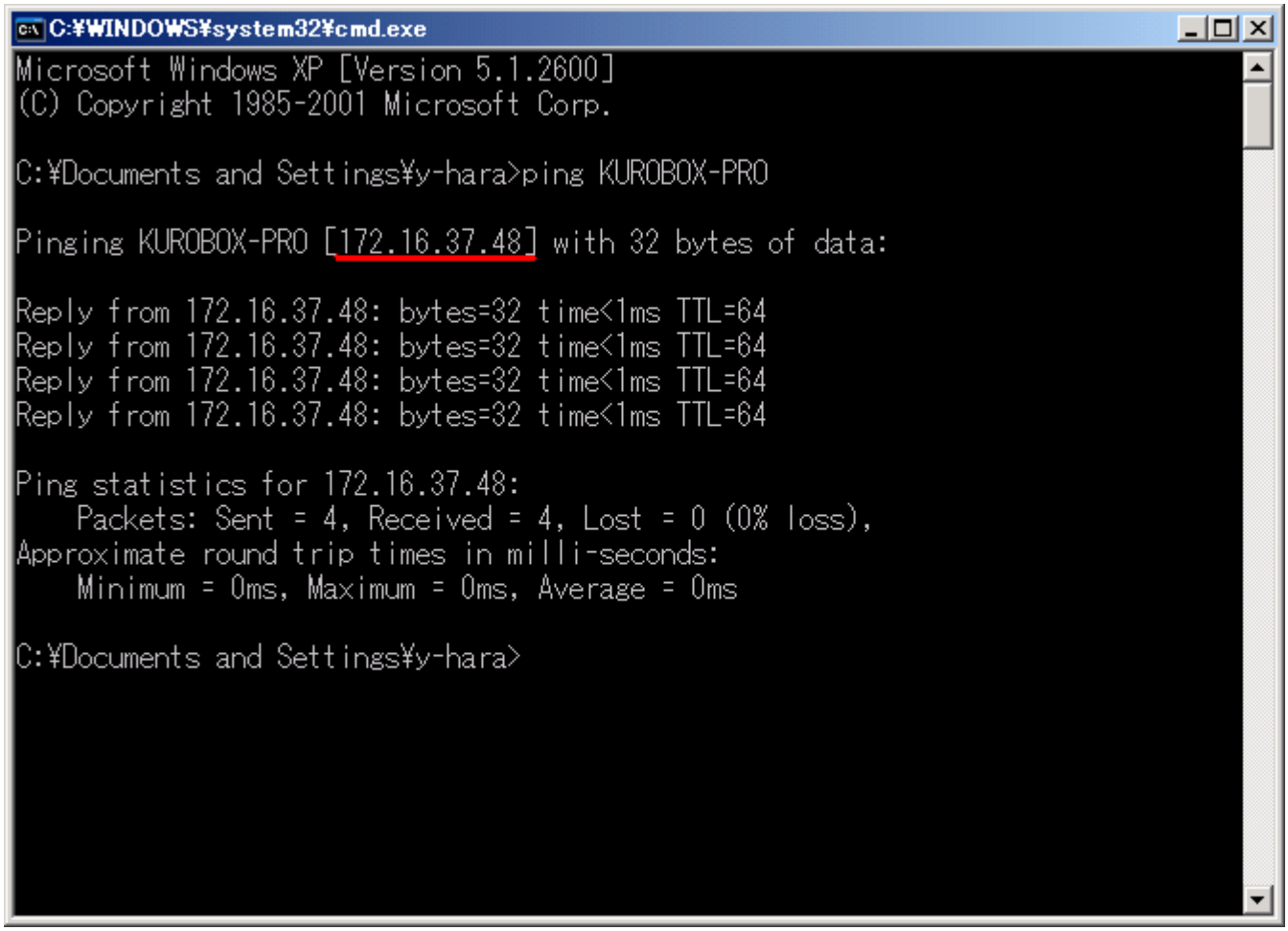
1. Connect either the internal disk or usb disk, and boot up.
2. After boot-up is completed, press and hold the initialization switch on the back of the Kurobox to format the internal disk or usb disk, and mount in xfs.
3. After performing the operation in 2, it is possible to access share.




## 7.2 Finding the IP Address

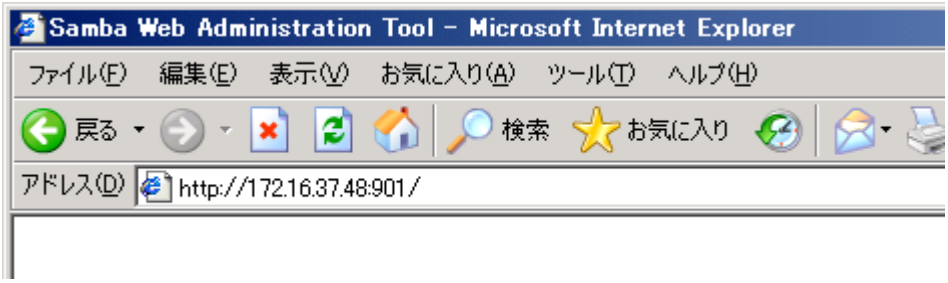

1. Type [\\KUROBOX-PRO] into the address bar of explorer, and check that you can access it.

2. After confirming that connections are possible, enter [Start] –> [Run] –> [cmd], and after the command prompt has started, check the IP address using the [ping KUROBOX-PRO] command. The underlined section is the KUROBOX-PRO IP address.



7.3. Setting the Shared Folder

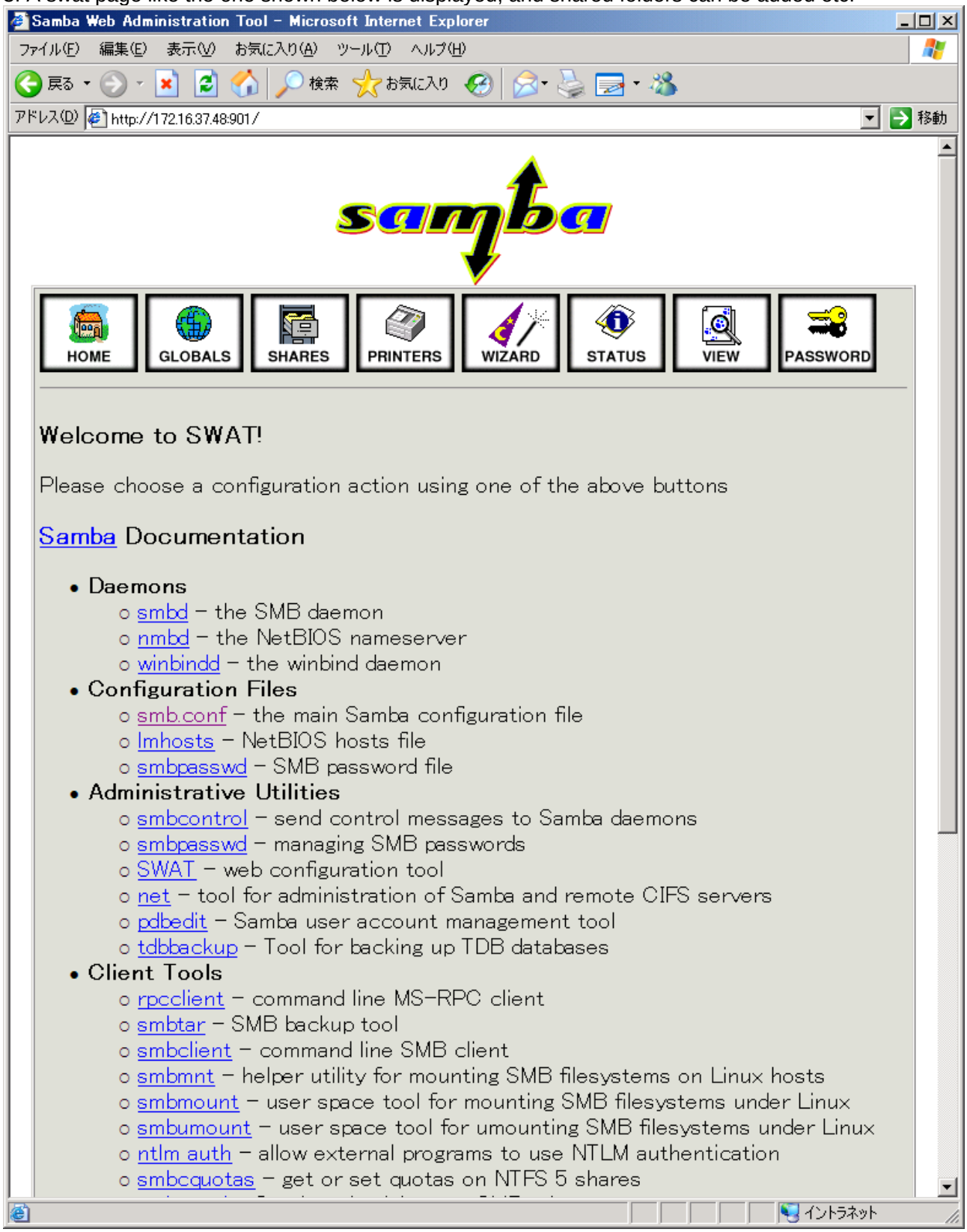
1. Enter a URL as shown below to open the KUROBOX-PRO web setting page. (In the example, the KUROBOX-PRO IP address is 192.168.11.150)




2. A basic authorization dialog box like the one below is displayed. Enter your username and password.

| User name | Default password |
|---|---|
| root | kuroadmin |

3. A swat page like the one shown below is displayed, and shared folders can be added etc.



# 8. Others

## 8.1. Factory Settings (Not Setup Condition)

| Item | Value |
|---|---|
| Firmware mode | Normal startup |
| root password | kuroadmin |
| KURO-BOX name | KUROBOX-PRO |
| Time | Not set |
| IP address/subnet mask | Obtained from the DHCP server<br>(If cannot be obtained, 192.168.11.150) |
| Ethernet frame size (MTU size) | 1518 bytes (1500 bytes) |
| Smb shared functions | Use |
| TELNET server functions | Use |

## 8.2. Error Code List

| Error code | Details of error |
|---|---|
| E00 | MPU malfunction Main MPU not responding |
| E01 | DRAM DATA LINE error |
| E02 | DRAM ADDRESS LINE error |
| E04 | Kernel expansion failed |
| E05 | Watchdog Timer |
| E06 | Kernel image read failed (initrd does not exist on the HDD) |
| E07 | Could not confirm connection of HDD |

* E00 is expressed by all the lights switching off. LEDs are off, and only an alarm sounds

# Appendix

## Appendix A: Example of u-boot environment variable settings

### A.1. Boot settings when uImage(nand /dev/mtdblock1), rootfs(nand /dev/mtdblock2)

| | |
|---|---|
| uImage | /dev/mtdblock1 (0x00020000~0x00400000)<br>4MB arrangement excluding head 128KB |
| initrd | -----(Not use) |
| rootfs | /dev/mtdblock2 |

u-boot environment variable settings

```
setenv default_kernel_addr 0x00100000
setenv bootargs_base console=ttyS0,115200
setenv bootargs_root root=/dev/mtdblock2 rw
setenv bootargs $(bootargs_base) $(bootargs_root) $(buffalo_ver)
setenv uImage_block 0
setenv uImage_offset 0x00020000
setenv bootcmd ‘nboot $(default_kernel_addr) $(uImage_block) $(uImage_offset); bootm $(default_kernel_addr)’
```

### A.2. Boot settings when uImage(nand /dev/mtdblock1), rootfs(/dev/sda2)

| | |
|---|---|
| uImage | /dev/mtdblock1 (0x00020000~0x00400000)<br>4MB arrangement excluding head 128KB |
| initrd | -----(Not use) |
| rootfs | /dev/sda2 |

u-boot environment settings

```
setenv default_kernel_addr 0x00100000
setenv bootargs_base console=ttyS0,115200
setenv bootargs_root root=/dev/sda2 rw
setenv bootargs $(bootargs_base) $(bootargs_root) $(buffalo_ver)
setenv uImage block 0
setenv uImage_offset 0x00020000
setenv bootcmd ‘nboot $(default_kernel_addr) $(uImage_block) $(uImage_offset); bootm $(default_kernel_addr)’
```

A.3. Boot settings when uImage(/dev/sda1), rootfs(/dev/sda2)

| | |
|---|---|
| uImage | /dev/sda1 |
| initrd | -----(Not use) |
| rootfs | /dev/sda2 |
| Notes | /dev/sdal is formatted in ext2, and uImage.buffalo is stored. |

u-boot environment settings

setenv default_kernel_addr 0x00100000

setenv bootargs_base console=ttyS0,115200

setenv bootargs_root root=/dev/sda2 rw

setenv bootargs $(bootargs_base) $(bootargs_root) $(buffalo_ver)

setenv bootcmd ‘ext2load ide 0:1 $(default_kernel_addr)/$(kernel); bootm $(default_kernel_addr)’

A.4. Boot settings when uImage(tftp load), rootfs(nand /dev/mtdblock2)

| | |
|---|---|
| uImage | Load from tftp |
| initrd | -----(Not use) |
| rootfs | /dev/mtdblock2 |
| tftpserver | (e.g.) 192.168.11.100 |
| KUROBOX-PRO | (e.g.) 192.168.11.150 |
| Notes | uImage.buffalo exists in the tftp root directory of tftpserver. |

u-boot environment variable settings

setenv default_kernel_addr 0x00100000

setenv bootargs_base console=ttyS0,115200

setenv bootargs_root root=/dev/mtdblock2 rw

setenv bootargs $(bootargs_base) $(bootargs_root) $(buffalo_ver)

setenv serverip 192.168.11.100

setenv ipaddr 192.168.11.150

setenv bootcmd ‘tftp $(default_kernel_addr) $(kernel); bootm $(default_kernel_addr)

A.5. Boot settings when uImage(tftp load), initrd(tftp load)

| | |
|---|---|
| uImage | Load from tftp |
| initrd | Load from tftp |
| rootfs | Exists in initrd |
| tftpserver | (e.g.) 192.168.11.100 |
| KUROBOX-PRO | (e.g.) 192.168.11.150 |
| Notes | • uImage.buffalo and initrd.buffalo exist in the tftp root directory of tftpserver.<br>• rootfs exists in initrd.<br>• After expanding rootfs in initrd, it is within 32MB. |

u-boot environment variable settings

setenv default_kernel_addr 0x00100000

setenv default_initrd_addr 0x02000000

setenv bootargs_base console=ttyS0,115200

setenv bootargs_root root=/dev/mtdblock2 rw

setenv bootargs $(bootargs_base) $(bootargs_root) $(buffalo_ver)

setenv serverip 192.168.11.100

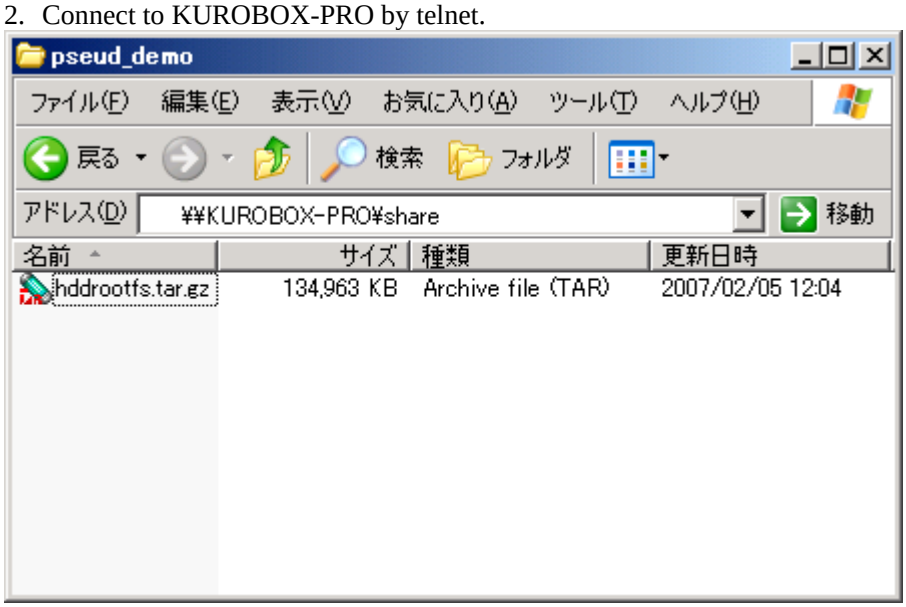
setenv ipaddr 192.168.11.150

setenv bootcmd ‘tftp $(default_kernel_addr) $(kernel); tftp $(default_initrd_addr) $(initrd); bootm $(default_kernel_addr) $(default_initrd_addr)’

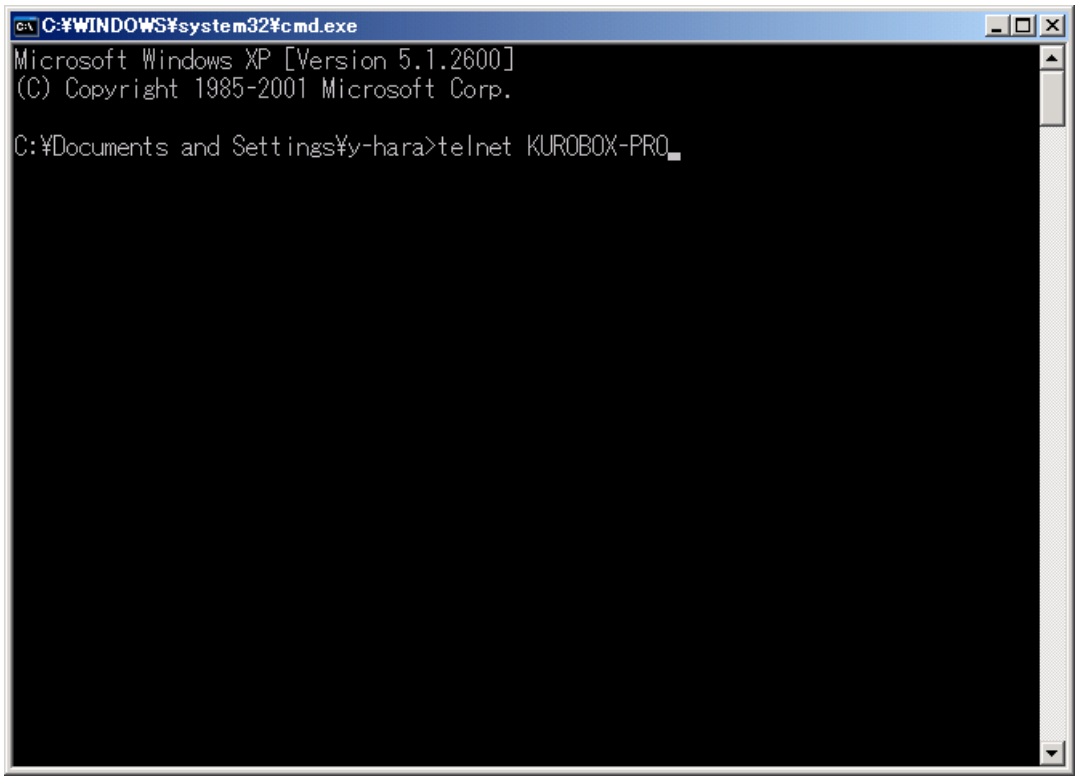
## Appendix B: Constructing the Development Environment

You can create the gcc environment by expanding the [development_kit\hddrootfs.tar.gz] file stored on the supplied CD-ROM. However, in order to avoid too many accesses to the NAND flash, it is recommended that you prepare the development environment on the HDD. You can prepare the development environment on the HDD by using the following procedure.

1. Copy hddrootfx.tar.gz to KUROBOX-PRO via the smb.
2. Connect to KUROBOX-PRO by telnet.



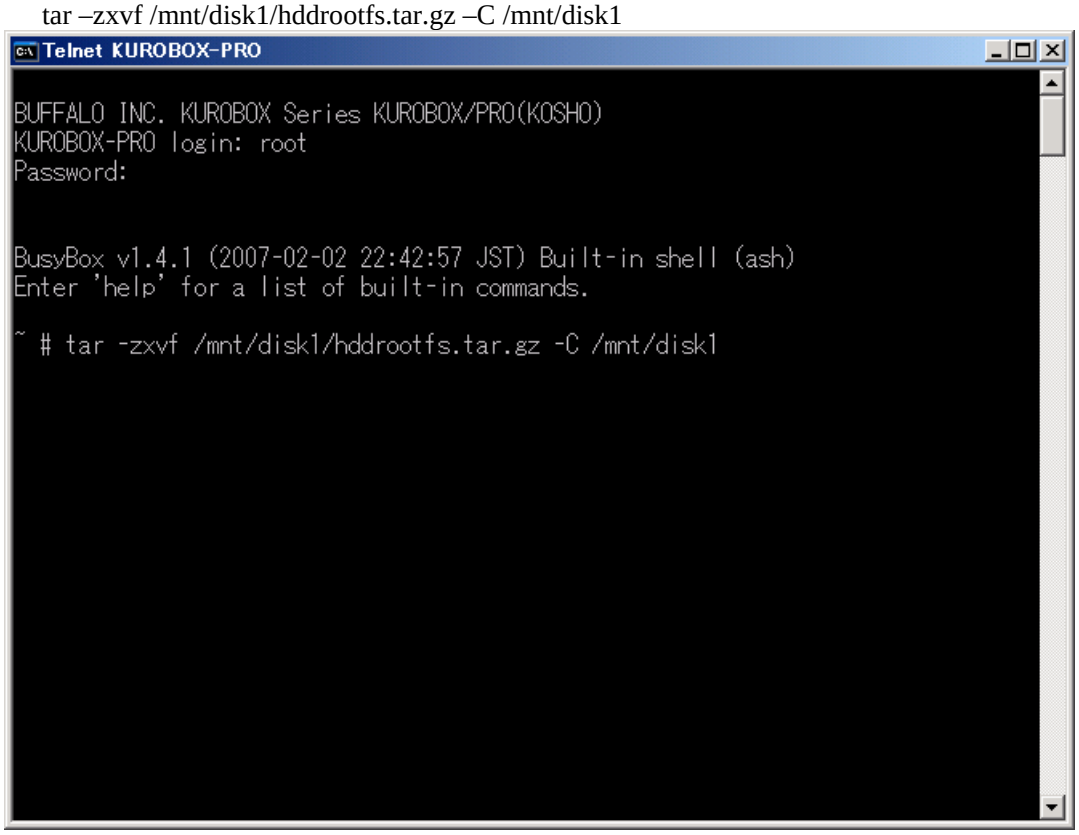

2. Connect to KUROBOX-PRO by telnet, and login as root.



3. Expand in the HDD.

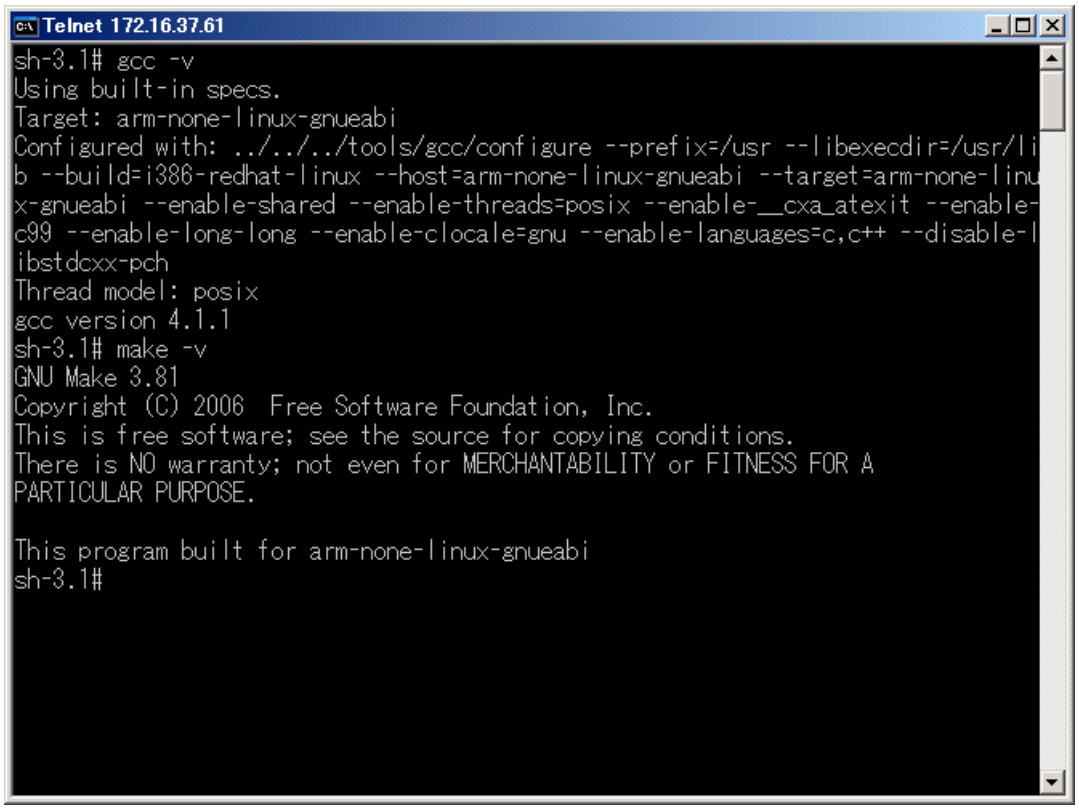
   The commands you will use are shown below.

   tar –zxvf /mnt/disk1/hddrootfs.tar.gz –C /mnt/disk1




4. Execute the following commands, and construction of the development environment is completed if gcc and make operate.

chroot /mnt/disk1



Also, although operation is not guaranteed, the HDD boot environment can be created using the following procedure.

1. Connect an unformatted HDD, and boot up normally.
2. Copy the following files in development_kit of the CD-ROM to [mtd device] in the shared directory.

| | |
|---|---|
| hddrootfs.tar.gz | hddrootfs for development environment |
| ChangeMeDevHDD | Purpose-displayed file for setting the HDD partition for development |
| ChangeMyUbootEnv | Purpose-displayed file for rewriting the u-boot environment variable<br>* If the u-boot environment variable is set to an invalid value, it may not be able to start up again.<br>If there are no problems in the chroot environment, do not copy this file. |
| uImage.buffalo | Kernel |

3. In the above condition, press the initsw on the back of the product.
4. Infoled flashes, and formatting, expansion of the development environment, and rewriting of the u-boot environment variable are performed.

# Appendix C: Kernel Compile Procedure

At present the kernel cannot be compiled on the target. Therefore, you need to install Linux onto an I386 system PC and construct a cross compile environment before building the kernel. That procedure is shown below.

## C.1. Constructing a Cross Compile Environment

1. Download the cross compiler from the sites shown below, and install it on your PC-Linux.

http://www.codesourcery.com/

http://www.codesourcery.com/gnu_toolchains/arm/releases.html

| | |
|---|---|
| Release Version | 2005q3-2 |
| Target Platform | ARM GNU/Linux |
| Host Platform | IA32 GNU/Linux |

2. By expanding the downloaded binary, the environment for generating binaries for arm is prepared.

## C.2. Compiling the Kernel

1. Set the path to the cross compiler.

   e.g.: mkdir /opt/arm-2005q3-2

   tar –jxvf arm-2005q3-2-arm-none-linux-gnueabi-i686-pc-linux-gnu.tar.bz2 \

   –C /opt/arm-2005q3-2

   If the above example is extracted, execute the following command.

   PATH=${PATH}:/opt/arm-2005q3-2/arn-none-linux-gnueabi/bin

3. You can compile the kernel by extracting the kernel source included in the CD-ROM of this product and executing make in the top directory.

4. Execute the make uImage command to create a u-boot image. At that time, the mkimage package is required.

Appendix D: Operation check completed PCIe card

| Type | Vendor | Part number |
|---|---|---|
| Network Card | Kuroutoshikou | GBE-PCIe |