Advanced Card Systems Ltd.
Card & Reader Technologies

eH880 and ACR880

SDE Setup Manual

Subject to change without prior notice

info@acs.com.hk
www.acs.com.hk

Revision History

Version	Date	Prepared By	Description
1.4	06 October 2009	Richelle Arjona	• Changed device photo in front page

Table of Contents

Figures

Tables

1.0. Introduction

1.1. Purpose

The purpose of this documentation is to explain how to setup a software development environment to develop application programs for the eH880 / ACR880 system.

The eH880 / ACR880 device is equipped with a 32-bit CPU running the embedded Linux 2.6.12 OS. For detailed hardware / software capability of the eH880 / ACR880, refer to their respective manuals.

The software development environment consists of a suite of software tools that are typically run on a PC Linux system. Application software can be developed on the PC Linux and then downloaded to the eH880 / ACR880 for testing and running.

This document describes the installation of the software tools on the PC Linux system and also the networking configuration between the PC Linux and the eH880 / ACR880 device.

1.2. Scope and Limitations

This document covers the procedures that interact with the device using the TCP/IP protocol. While there are virtually thousands of Linux kernels available on open source programming, the EH880/ACR880 package includes compressed files that are to be integrated to the device.

Hardware restrictions and Linux OS versions are also detailed and while the device firmware itself is upgradeable, there are limitations on where files should be copied. Folder restrictions as well as some Linux commands are listed on the succeeding chapters. For this setup the Linux OS that the demo will be using is Debian with a Gnome desktop, version 2.14.3.

2.0. Overview

- It is assumed that the user has basic knowledge in using computers and operating Linux OS.
- Section 3.0 discusses the Linux OS limitations and requirements as well as the compiler instructions upon installation.
- Section 4.0 lists the device's connectivity in both the hardware and software aspects.
- Section 5.0 details the folder restrictions when customizing the files inside the device.
- Appendix A shows the processes in connecting the eH880 to the Ethernet network.

3.0. Compiler and Library Installation on PC Linux

3.1. Requirements

3.1.1. Operating System

A Linux based system is required for developing software for eH880. Since the development of the software consists of cross-compilation technique, the developers are expected to have sufficient knowledge to operate Linux system and compile programs using gcc (GNU compiler collection) at least in i386(PC) systems. Developers may use one of the following Linux distributions to develop eH880 software.

Debian-based Linux operating systems emphasize on free software for many hardware platforms. This system uses the .deb package format and dpkg package manager.

http://www.debianhelp.co.uk/debian.htm

Ubuntu-based Linux operating systems, derived from Debian.

http://www.kubuntu.org/download.php

Fedora-based and Red-Hat-Enterprise-based operating system uses Red Hat Package manager.

http://www.fedoralegacy.org/download/

3.1.2. Hardware Requirements

- A modern PC that fulfills the hardware requirements of the above Linux distribution.

3.2. Compiler Installation(Take Fedora as example)

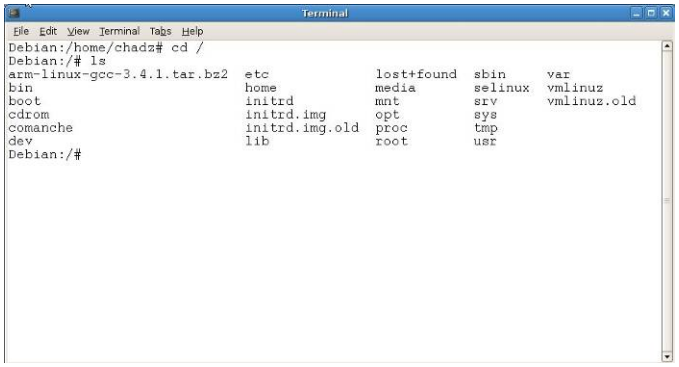
3.2.1. Login your Linux PC as “root”.

Location

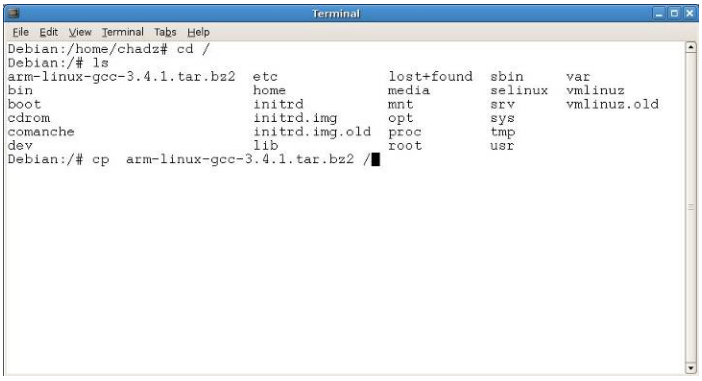
Input the root password in the password box.

3.2.2. Incorporate the necessary files

Go to the root directory “/”, by typing “cd /” on the prompt.



Once inside the root directory, copy the file "arm-linux-gcc-3.4.1.tar.bz2" to "/" directory of your PC Linux

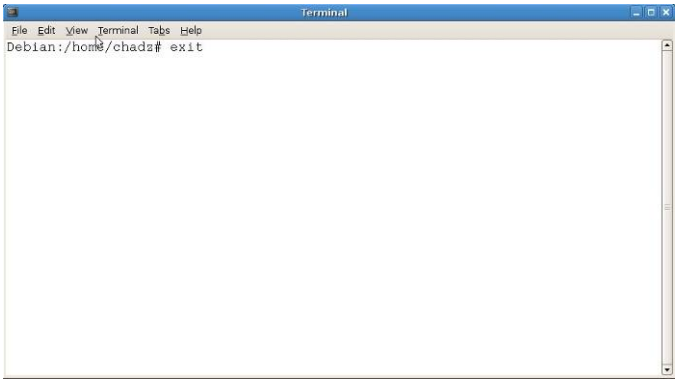


3.2.3. Extract complier files

Extract the complier files by the command "*tar -jxvf arm-linux-gcc-3.4.1.tar.bz2*"

3.2.4. Logout

Input *exit* on the command prompt to logout.



3.3. Setup your environment

3.3.1. Login your PC Linux with your username.

3.3.2. Modify Scripts

Modify your login shell script to include "*/usr/local/arm/3.4.1/bin*" in the search path.

For bash shell, Add the line "export PATH=/usr/local/arm/3.4.1/bin:$PATH" at the end of <home directory>/.bashrc

3.3.3. Logout

3.4. Extract Library

3.4.1. Login your PC Linux with your own username.

3.4.2. Copy Files

Copy the file "*eH880-sdk-yyyymmdd.tgz*" (where yyyymmdd is a date code) to the installation directory

3.4.3. Extract Files

Get into the installation directory and extract the library files by the command "*tar -zxvf A880-sdk-yyyymmdd.tgz*"

Files to verify:

<install_dir>/sdk/.version	: Version log
<install_dir>/sdk/demo	: Some demo program
<install_dir>/sdk/sdk-lib	: Library and Header file for the API

3.4.4. Testing the Extracted Files

To test the installation, go to the directory <install_dir>/sdk/demo/demo1 and type "make".

The "Makefile" will be read and "arm-linux-gcc" will be invoked then the files "demo1" will be generated at <install_dir>/sdk/demo/demo1/bin

4.0. Device Connection to Host Computer

4.1. Ethernet

4.1.1. Cable Restrictions

The eH880/ACR880 uses an RJ45 straight cable to connect to a router/ hub. However, a crossover RJ45 cable is used when the device is directly connected to the host pc. See Appendix A for cable types.

To connect the PC Linux to the device using the RJ45 cable, insert both ends of the cable jack to the designated ports found on each party. From the command prompt or terminal prompt, key-in "*telnet <device IP>*" for Windows/Linux, or "*SSH <IP>*" for Linux.

4.1.2. RJ45 Cable Types

	Cross over cable		Straight cable	
Pin ID	Side A	Side B	Side A	Side B
1	Orange-white	Green-white	Orange-white	Orange-white
2	Orange	Green	Orange	Orange
3	Green-white	Orange-white	Green-white	Green-white
4	Blue	Brown-white	Blue	Blue
5	Blue-white	Brown	Blue-white	Blue-white
6	Green	Orange	Green	Green
7	Brown-white	Blue	Brown-white	Brown-white
8	Brown	Blue-white	Brown	Brown

Table 1: RJ45 Cable Types

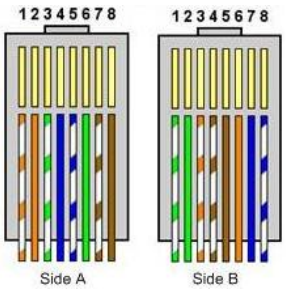


Figure 1: Cross-Over Cable

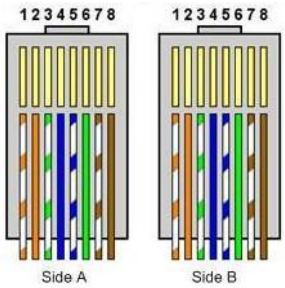


Figure 2: Straight Cable

Figure 3: RJ45 Cables with Jacks Connected

4.1.3. Login Information

The IP can be obtained by pressing <F4> key when the demo program is running

To login and access the eh880, use console (RS232, refer to 4.2) or telnet/ssh (refer to 4.1) to login the device. (username: root, password: cdy123)

4.2. RS232

4.2.1. Serial Port Configuration

Signal Level	RS232 (9 pin)
Baud Rate	115200 bps
Data Bit	8
Parity	None
Stop Bit	1
Flow Control	None

Connect to the device using the RS232 9 pin serial port to the female port located at the back of the device. The male port counterpart is located at the pc. The login, if required, is till *root* and the password is *cdy123*. Certain tools can be used to connect to the device using a serial cable; Hyperterminal is used for Windows and Minicom for Linux.

If you connect the RS232 cable to the PC and eh880 after the eh880 powered up, press “enter” in the terminal window to get the login prompt.

5.0. Program/Files Uploading

5.1. eH880 / ACR880 File Structure

The eH880/ACR880 follows the common Linux file directory structure. The root file system is read-only.

The eH880/ACR880 files are stored in the following directories:

/usr/local/RO/	For read-only files, such as the unit name and serial number, and other system files.
/usr/local/RW	For read-write files. Developers can store their software application and data files in this directory. Multiple subdirectories can also be created through this path.
/tmp	Directory to store temporary files. Data in this directory will be erased after reboot/power-off.

5.2. Ethernet Connection

5.2.1. Device to PC Host

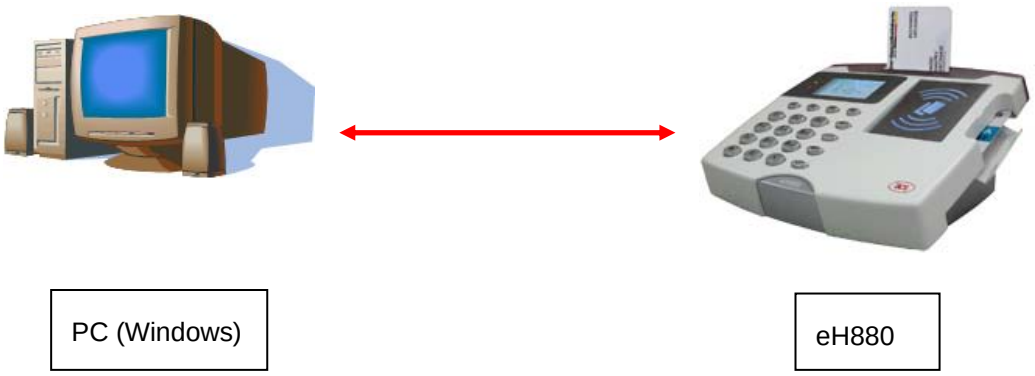


Figure 4: Use the Cross-over Cable for PC Host to Device Setup.

5.2.2. Device to Hub/ Switch to PC Host

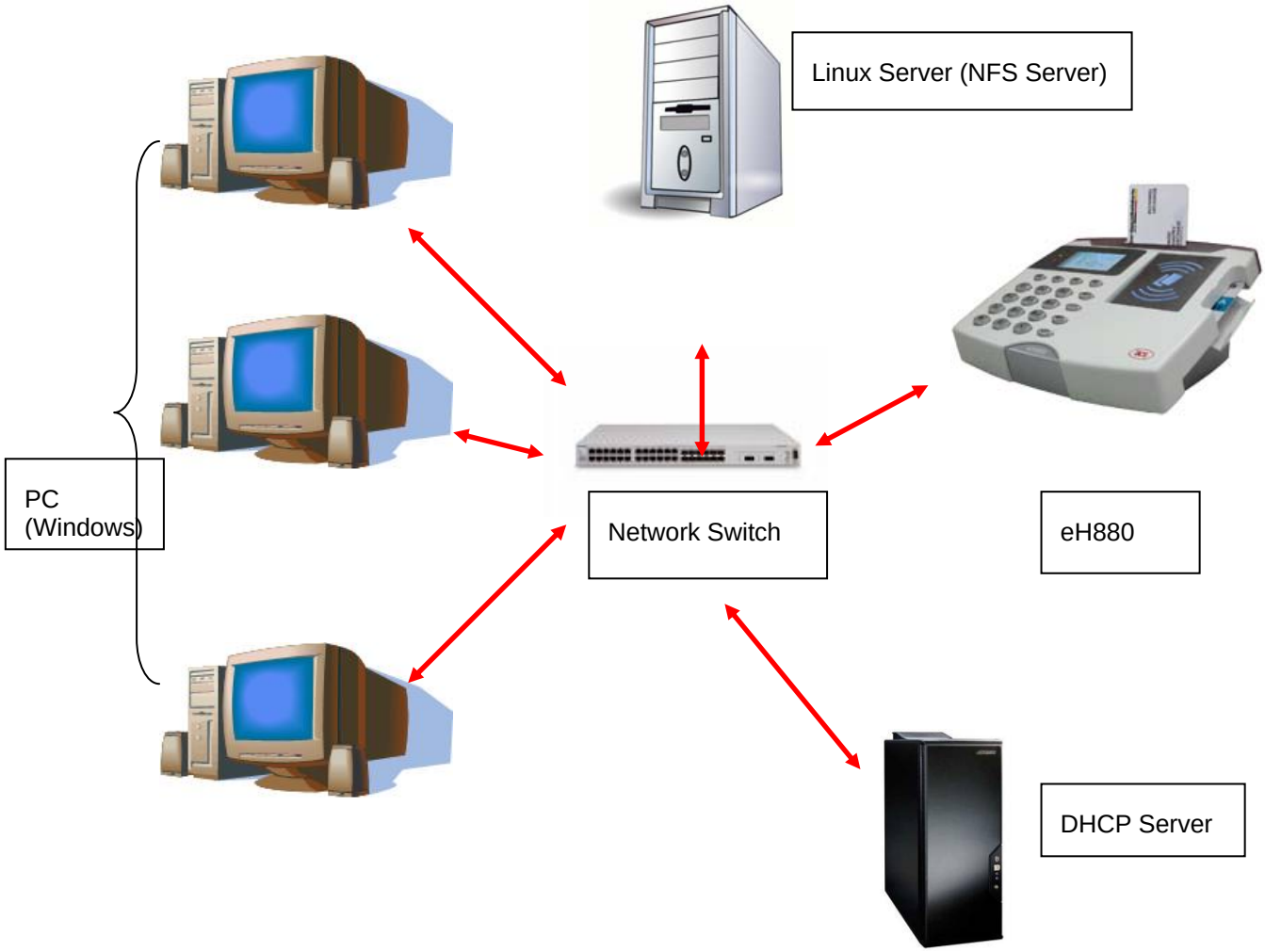


Figure 5: Use the Straight Cable for Device to Hub/Switch Setup

5.3. Upload Types

5.3.1. FTP (File Transfer Protocol)

- "ftp" and "ncftp" are installed in the device
- User can login into the device and initiate FTP connection to a host computer; and user can use these two commands to transfer files.

5.3.2. SFTP (Secure FTP)

- An SFTP daemon is installed in the device.
- User can use an SFTP client to login (with above login info) the device and transfer files.

5.3.3. NFS (Network File Server)

The eH880 supports mounting NFS filesystem and so it is able to mount folders from the remote NFS Severs for more convenient file transfer. NFS file server and Ethernet connection is required. NFS file server can be setup along with the development Linux PC

Procedure:

Login to the eh880 through telnet/sshIssue a command: `mount -o nolock,rsize=1024,wsize=<nfs server ip>:<share directory> /mnt/ext3`

The files in the share directory will be appeared in /mnt/ext3 of the eh880

For details please refer to Annex A2

Appendix A. Connecting eH880 to the Ethernet Network

Appendix A.1. Configuring Ethernet Port

Each eH880 Demo Unit has a built-in Ethernet port. The eH880 unit is pre-configured to get an IP from DHCP servers. After the demo program showed on the screen. Press F4 to view the IP address of the eH880.

It is possible to change the IP manually

After the **telnet** session has been established, an '**ifconfig'** command to view the IP address assignment can be issued. The following is the example output when **ifconfig** is issued:

```
~ # ifconfig
eth0      Link encap:Ethernet  HWaddr 00:02:31:05:0A:01
          inet addr:192.168.61.123  Bcast:192.168.61.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:110 errors:0 dropped:0 overruns:0 frame:0
          TX packets:7 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:37 Base address:0x300

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

On the other hand, users can also configure network parameters. The following sections discuss the commands and procedures to do so. User can specifically choose either static IP or dynamic IP address configuration for the eH880 unit. *However, note that such changes will not be permanent – after the device is re-booted again, its IP address will resume the saved setting in the eH880. To permanently change the IP address, please refer to eH880 tool manual.*

Appendix A.1.1. Setting up Loop Back Device

Loop back device is used to refer to local host of the eH880 unit and it has a default IP address 127.0.0.1 with netmask 255.0.0.0.

1. Input the following command to bring up the loop back device interface:

```
# ifconfig lo 127.0.0.1
```

2. Input **"ifconfig"** command to show the configuration:

```
# ifconfig
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

3. Input the following command to add an entry to routing table.

route add –net 127.0.0.0 netmask 255.0.0.0 lo

4. Input the **"route"** command to show the routing table.

```
~ # route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
127.0.0.0       *               255.0.0.0       U     0      0        0 lo
```

Appendix A.1.2. Setting Up Static IP Address

Assume you want to setup the following IP address for eH880 unit:

IP Address	:	192.168.1.100
Subnet Mask	:	255.255.255.0
Gateway	:	192.168.1.1

1. Input the following command to setup the IP address:

ifconfig eth0 192.168.1.100 netmask 255.255.255.0

2. Input the following command to setup the gateway:

route add default gw 192.168.1.1 eth0

3. Input **"ifconfig"** command to show the configuration:

```
# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0B:51:00:0F:FE
          inet addr:192.168.1.100  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:10155 errors:0 dropped:0 overruns:0 frame:0
          TX packets:22 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:37 Base address:0x300

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

4. Input **"route"** command to show the routing table:

```
# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.1.0     *               255.255.255.0   U     0      0        0 eth0
127.0.0.0       *               255.0.0.0       U     0      0        0 lo
default         192.168.1.1     0.0.0.0         UG    0      0        0 eth0
```

Appendix A.1.3. Setting up Dynamic IP Address

If it is necessary to setup the dynamic IP address for eH880 unit, input the following command to get the IP address from DHCP server:

udhcpc –b –i eth0

Appendix A.2. Setting up Network File System (NFS) Server

Appendix A.2.1 PC Side

The exported directories of NFS server are controlled by **"/etc/exports"** configuration file. To share the directory of Linux Server with eH880 unit, you need to edit the file.

Sample contents of **"/etc/exports"** configuration file:
/share1 192.168.1.0/24(ro)
/share2 192.168.1.100(rw)

For example, directory **"/share1"** can be shared by all devices with IP address starting with **"192.168.1.xxx"**. Note that **"192.168.1.0"** is network address and **"24"** is the short form of **"255.255.255.0"** subnet mask. **"ro"** is used to share directory for read-only while **"rw"** is used to share directory for read/write. Directory **"/share2"** is only shared to the device that has an IP address equal to **"192.168.1.100"** for read/write.

After editing the configuration file, restart NFS server for the changes to take effect. Input the following command to restart NFS server:

/etc/rc.d/init.d/nfs restart

Appendix A.2.2 eH880 side

In the eH880 unit, you can input the following command to use remote directory that is shared by NFS server. Assume that the IP address of Linux server is **"192.168.1.10"**.

mount –o nolock,rsize=1024,wsize=1024 192.168.1.10:/share1 /mnt/ext1

You can change to directory **"/mnt/ext1"** by inputing **"cd /mnt/ext1"** command.

Appendix A.2.3 NFS (Network File System)

After logging in the device (telnet), user can mount a directory structure in his PC Linux to a directory in the device and then use "cp" command to copy files to another directory in the device. To mount a directory on the command prompt:

"*mount -o nolock,rsize=1024,wsize=1024 xxx.xxx.xxx.xxx:/home/user /mnt/ext1*"

"*cp /mnt/ext1 /usr/local/rw*"

User can also execute a program inside the mounted directory directly without implicit transfer.

The mount command will "mirror" the directory /home/user in your PC Linux (with IP = xxx.xxx.xxx.xxx) to /mnt/ext1 in the eH880

Note that "1024" in the above example is necessary to ensure reliability when transferring big files.

Note also that the NFS service in your PC Linux have to be configured correctly to open permission of the directory that is going to be mounted. See "man nfs" and "man exports" in PC Linux for detail.

Appendix A.3. Showing the Status of Processes

To show the status of processes, input **"ps"** (process status) command. The information is arranged by Process ID (PID), User ID (Uid), Virtual Memory Size in bytes (VmSize), Process State (Stat) and command line (Command).

According to the man page of "ps" command, the process state has the following meanings:

Process State	Description
D	uninterruptible sleep (usually IO)
R	runnable (on run queue)
S	Sleeping
T	traced or stopped
Z	a defunct ("zombie") process
W	Has no resident pages
<	High-priority process
N	Low-priority task
L	Has pages locked into memory (for real-time and custom IO)

```
~ $ ps
  PID  Uid     VmSize Stat Command
    1 root       576 S    init
    2 root           SWN [ksoftirqd/0]
    3 root           SW< [events/0]
    4 root           SW< [khelper]
    5 root           SW< [kthread]
   16 root           SW< [kblockd/0]
   19 root           SW  [khubd]
   42 root           SW  [pdflush]
   43 root           SW  [pdflush]
   45 root           SW< [aio/0]
   44 root           SW  [kswapd0]
  125 root           SW  [mtdblockd]
  150 root           SWN [jffs2_gcd_mtd4]
  192 root       612 S    syslogd
  194 root       612 S    klogd
  214 root       636 S    inetd
  217 root      1428 S    webs
  221 root       396 S    /usr/local/eH880/RW/bin/eH880_disc
  222 root       832 S    eH880_Demo4
  224 root       752 S    -sh
  225 root      1360 S    sshd -f /usr/local/eH880/RW/etc/ssh/sshd_config
  228 root       832 S    eH880_Demo4
  229 root       832 S    eH880_Demo4
  230 root       832 S    eH880_Demo4
```

```
  233 root      832 S   eH880_Demo4
  234 root      832 S   eH880_Demo4
  235 root      832 S   eH880_Demo4
  236 root      832 S   eH880_Demo4
  237 root      832 S   eH880_Demo4
  238 root      832 S   eH880_Demo4
  248 root      580 S   udhcpc -i eth0 -b
  252 root      512 S   more
  253 root      480 S   /usr/sbin/telnetd
  254 root      752 S   -sh
  257 root      664 R   ps
~ $
```

Appendix A.4. Miscellaneous Information

Appendix A.4.1 User ID and Password

The unit is setup for the following user:

User ID : root

Password : cdy123

Hence, the user can login as root user to communicate with the eH880 unit.

Warning:

Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.