PCM5820 & LRP HOWTO

Heimo Schön <*heimo.schoen@gmx.at*> Version 0.1 , 1. November 1999

Setup a *5820/L* with *LRP* the Linux-router-project

1 Overview

1.1 About this HOWTO

This document describes how to setup your *5820/L* with *LRP* the Linux-router-project

If you have any suggestions to this document, than feel free to contact me:

- Heimo Schön <*heimo.schoen@gmx.at*>
- *http://fly.to/heimo.schoen*

Have a lot of fun with your PCM5820/L.

1.2 Version History

- 0.1 01 Nov 1999 first trial

2 Introduction

In this chapter I want to discribe what you can do with LRP on a PCM5820. I try to give you a overview from my test equipment.

2.1 Target

I started to look for small computer solutions when I had to calculate a tender for a passanger-information-system project in 1999. There was the need for translating different protocols between different proprietary protocols.

In the past our company did this job by developing microcontroller boards and developing the applications on this boards for different projects. There was a lot of work to manage the hardware- and software-releases.

the idea was to use small and cheap standard hardware and a releable cheap operating system. After consulting the internet searchmachines I decided that the best choice would be a PC/104 compatible Intel-processor board and a small linux-kernel. Therefore I use the PCM5820/L board and the fantastic small Linux Kernel from the Linux-Router-Project.

2.2 HW-Configuration PCM5820/L

For the PCM5920/L onboard features refer to : *5820/L*

In the PCM5820/L I installed 16 MB Memory, a 20 MB Flashdisk (Sandisk), a 10MBit HUB for a connection to my Linux-Server at home, and a 3 1/2 Floppy Drive for booting the lrp system.

2.3 SW-Configuration LRP

You should refer to *LRP* for more details about how LRP works.

3 Installing LRP on PCM5820/L

3.1 Partitions

In this step we create partitions on the flashdisk installed in the PCM5820/L.

The partitions are named /dev/hdc, /dev/hdc1, ... on yor PCM-5820L. You should create the following partitions on your flashdisk:

```
root@ternitz60:/croot/etc > fdisk /dev/hdc

Command (m for help): p

Disk /dev/hdc: 2 heads, 32 sectors, 612 cylinders
Units = cylinders of 64 * 512 bytes

Device Boot Start End Blocks Id System
/dev/hdc1 * 1 150 4784 83 Linux native
/dev/hdc2 151 612 14784 5 Extended
/dev/hdc5 151 612 14768 83 Linux native

Command (m for help):
```

3.2 PCM-5820 Bootfloppy

Create a PCM5820 bootfloppy and the PCM-5820 LRP-Floppy (you need two floppies)

```
cp pcm5820_dos_floppy.image /dev/fd0
cp pcm5820_lrp_floppy.image /dev/fd0
```

You can download this floppy-images here:

pcm5820_lrp_floppy.image

pcm5820_dos_floppy.image

3.3 Boot from my PCM-5820 Bootfloppy

Now you made your Floppies and can boot now with your dos-floppy

3.4 format flashdisk

Now you can format the boot partition of the flashdisk

```
format c:
```

3.5 bootsector on flashdisk

Now create a bootsektor on the flashdisk. Don´t foreget to unlock the flashdisk before writing the bootsektor.

```
lock c:
          (note: answer the question with y)

syslinux c:
```

3.6 Files to flashdisk

Now insert the lrp-files floppy and copy all lrp files to the flashdisk:

```
copy a:*.* c:
```

3.7 LRP installed

Remove floppy from floppydrive and reboot the 5820. Now 5820 is booting from your new flashdisk!!!

You should see lrp booting.

4 notes

4.1 Watchdog

Die Beschreibung des Watchdog in der 5820 Motherboardbeschreibung ist schlicht falsch. Hier ein User-Space-Programm das ich zum Laufen gebracht habe und das genau das macht was man sich davon erwartet:

```
#include <unistd.h>
#include <stdio.h>
#include <sys/io.h>

#define WD_VERSION "V0.1"

int main(int argc, char ** argv)
{
  int count=0;
  int retval=0;

  // Allow the Process to access the watchdog-port
  retval = iopl(3);
  if (retval == 0) {
    printf("watchdog " WD_VERSION " started!\n");
    // TODO: print to Logfiles !!!

    for (;;) {

      outb_p(0xff ,0x443);
/*        retval = inb(0x443); */
```

```
/*        if ( (  retval != 0) || */
/*         ( (argv[0][0] == '-') && */
/*           (argv[0][1] == 'd') ) ) */
/*      printf("%d. inb(0x443) says %d\n", count, retval); */

      usleep(100000);

      // be a little bit verbose all 10 minutes
      count ++;
      if (!(count % 6000)) {
        // TODO: print to Logfiles !!!
        printf("watchdog " WD_VERSION " running since 10 Minutes\n");
        count = 0;
      }

    }

    // Switch of Watchdog
    outb(1, 0x043);
    retval = inb(0x043);
    if ( (  retval != 0) ||
         ( (argv[0][0] == '-') &&
           (argv[0][1] == 'd') ) )
      printf("inb(0x043) says %d\n", retval);

  } else {
    // TODO: print to Logfiles !!!
    printf("watchdog " WD_VERSION " had problems with iopl(3)\n");
    printf("iopl(3) returned %d\n", retval);
    printf("perhaps watchdog didn't run with root privileges\n");
    printf("Try to set sticky-bit on the watchdog executable\n");
    exit(1);
  }
}
```

Nun will man ja nicht jede Sekunde aus dem Userspace auf das Register 0x443 schreiben und dem Anwenderprozeß root-Rechte verpassen. Daher habe ich nach einer Kerneldriver-Lösung gesucht. Der advantechwdt.c der bei jeder gängigen Linux-Distribution enthalten ist, deutete stark auf einen 5820 kompatiblen Watchdog-Treiber hin. Dieser funktionierte auch nicht auf Anhieb. Um mit z.B.

```
echo 1 > /dev/watchdog
```

und einem sleep 1 in einer Endlosschleife den Watchdog am Leben zu erhalten muß man zuerst in der Datei /etc/modconf folgende Zeile

```
alias char-major-10-130 softdog
```

auskommentieren und gegen den Start des advantechwdt ersetzen. Statt der Zeile sollte das ganze dann so aussehen:

```
alias char-major-10-130 advantechwdt
```

Nun sind wir so weit, daß wir nur durch ein

```
echo 1 > /dev/watchdog
```

schon unseren advantechwdt geladen bekommen. Leider werden Sie vermutlich auch feststellen, daß zwar der Modul geladen ist, in /var/log/messages meldet wer brav seine Version, nur rebootet die PCM5820 Kiste leider nicht. Nun auch das läßt sich beheben. ;-)

Wechseln Sie in das Directory /usr/src/linux (oder wo immer Sie ihre Kernel-Sourcen haben) und editieren Sie dort die Datei drivers/char/advantechwdt.c

Suchen Sie nach einer Zeile die lautet:

```
#define WD_TIMO 60 /* 1 Minute */
```

und ändern Sie diese Zeile in :

```
#define WD_TIMO 1 /* 1 second */
```

Danach bauen Sie Ihre module neu und installieren diese, indem Sie die Befehle:

```
make modules
make modules_install
```

in Ihrem Kernel-Source Diretory eingeben. Danach entfernen Sie den gerade laufenden advantechwdt Modul und laden ihn neu mit:

```
rmmod advantechwdt
modprobe advantechwdt
```

Nun sollten Sie in der Lage sein durch ein echo 1 > /dev/watchdog und danach 1 Sekunde warten, den Rechner zum Rebooten zu bringen. Nun braucht man nur noch in seiner Anwenderapplikation einen Subprozess starten, der jede Sekunde ein Byte auf /dev/watchdog schreibt. Wenn dieser Prozess für mehr als 1,6 Sekunden pausiert, wird der PCM5820 gebootet.

Have a lot of fun ...

4.2 rdev

```
rdev /croot/vmlinuz # Anschauen welche disk als / gemountet wird
rdev /croot/vmlinuz /dev/hdc2 # anderes Device als / dem Kernel beibringen
```

4.3 ramdisk

```
dd bs=1M count=40 if=/dev/zero of=/cuser/dev/ramdisk
mke2fs /cuser/dev/ramdisk
mount /cuser/dev/ramdisk
```

4.4 Pakages

```
/var/lib/lrpkg ......... here you find the pakages on the flashdisk
```

5 Allgemeines

5.1 Copyright