How to get USB devices working under Linux
리눅스에서 USB 장치 돌리기

Brad Hards , brad.hards@dao.defence.gov.au Revision: 0.7

김기현, arbeit95@chollian.net Revision: 0.7 1999년 10월 9일

이 문서는 USB 장치들을 리눅스 시스템에서 사용하기 위한 단계적 안내서의 초안이다.

목 차

제 1 절 기본적인 USB 설정 1

제 2 절 마우스 설정 3

제 3 절 키보드 설정 4

제 4 절 허브 지원 4

제 5 절 프린터 지원 4

제 6 절 Abstract Control Model 지원 5

제 7 절 CPiA imager 지원 5

제 8 절 대용량 저장장치 5

제 9 절 EZUSB 드라이버 5

제 10 절 USS720 드라이버 6

제 11 절 USB /proc 드라이버 6

제 12 절 자주 질문 받는 것들(FAQs) 8

제 13 절 정정 9

제 1 절 기본적인 USB 설정

최신 버전의 커널이 있어야 한다. 버전 2.2.7 이나 이 이후버전의 커널은 USB 코드를 포함하고 있다. 이상적으로는 다음버전의 모든 pre-patches를 포함한 최신 2.3.x 커널을 사용하고 있으면 좋다. 2.2.x 커널에서 2.3.x USB 코드를 사용하는 것이 가능하다 - 자세한 것은 이 장의 뒷부분을 살펴 보라. USB 코드는 사실 초기 개발단계에 있기 때문에 각 버전마다의 변경(및 버그들)은 매우 빨리 변하는 경향이 있다. 최신 버전에 관한 것은 드물지만 메일링리스트는 거의 대부분에 관한 내용을 다루고 있다.

메일링리스트는 linux-usb@suse.com이다. 가입하려면 majordomo@suse.com에게 **subscribe linux-usb** 라고 쓴 메일을 보내라. 메일링리스트의 글들을 그만 보고 싶으면 majordomo@suse.com에게 **unsubscribe linux-usb** 라고 써서 보내라.

커널에 USB 설정을 해야 한다. **make menuconfig**가 좋을 것이다. **USB drivers - not for the faint of heart** 항목에서 **Support for USB(EXPERIMENTAL!)**를 선택해야 한다. 또한 **UHCI (intel PIIX4 and others) support, OHCI (compaq and some others) support,**또는 **OHCI-HCD (other OHCI opt. Virt. Root Hub) support** 가운데 하나를 선택한다. UHCI, OHCI 및 OHCI-HCD를 동시에 같이 사용하면 제대로 작동하지 않을 수 있다(UHCI는 작동할 것이다). 어느 것을 고를 것이냐는 당신이 어떤 마더보드나 어댑터를 갖고 있느냐에 달려 있다. Intel과 Via 마더보드는 UHCI이다. 컴팩과 NEC 마더보드, iMac과 Opti 칩(모든 칩 종류를 포함)을 쓰는 모든 어댑터는 OHCI이며 OHCI나 OHCI-HCD 가운데 하나를 선택할 수 있다. 어떤 컨트롤러를 선택해야할지 모르겠다면 마더보드 문서를 보라. 또한 /proc/pci에서 힌트를 얻을 수 있다. USB 항목이 0xHHHH형식으로 되어 있을 때는(HHHH는 16진수이다 - 예를 들어 **I/O at 0xe400**) UHCI이다. **32 bit memory at 0xHH000000** 일 때는(HH는 16진수이다 - 예를 들어 **32 bit memory at 0xee000000**) OHCI이다. 이것도 불가능하면 아무거나 하나 시도해보라.

다음으로 어떤 장치를 사용할 것인지를 선택해야 한다. 예를 들어 **xUSB mouse support**는 USB 마우스, **USB keyboard support**는 USB 키보드, **USB hub support**는 허브, **USB communications Devices Class (ACM) support**는 POTS 또는 ISDN모뎀, **USB Printer support**는 USB 프린터, **USB SCSI Support**는 대용량 저장장치용, **USB CPiA Camera support**는 Vision CPiA 칩셋을 사용한 카메라, **EZUSB Firmware downloader**는 Anchor ChipsUSB 마이크로콘트롤러킷에 대한 다운로드, **USS7270 parport driver**는 병렬 포트 어댑터용 등이다. 모듈과 커널 직접 지원(kernel only), 별도의(split) 모듈, 커널코드 등을 사용할 수 있어야 한다. 장치들을 루트 허브에 꽂아 사용해서 허브가 꼭 필요하지 않다 하더라도 허브 지원을 선택하는 것이 좋다.

USB audio parsing support는 현재까지는 지원되지 않는다. 이 문서에 열거되지 않은 장치는 이 글을 쓰는 시점에서는 지원되지 않는 것들이다. 그러나 개발자들은 현재 코드에 도움을 주는 것(contribute)은 언제든지 환영한다.

안정 커널(**2.2.x**)을 쓰고자 한다면 개발커널(**2.3.x**)의 코드를 drivers/usb 디렉토리에 복사해 넣으면 된다. arch/i386/config.in에서 아래와 같은 행의 주석을 없애야 한다.

```
# source drivers/usb/Config.in
```

다른 아키텍처도 비슷할 것이다. drivers/usb/usb.h의 첫부분에 아래와 같은 몇 줄의 정의(definition)를 추가해야 한다.

```
#if LINUX_VERSION_CODE < 0x020300
#define DECLARE_WAIT_QUEUE_HEAD(w)      struct wait_queue *w = NULL
#define DECLARE_WAITQUEUE(w,c)          struct wait_queue w = {(c), NULL}
#define wait_queue_head_t               struct wait_queue *
#define init_waitqueue_head(w)          *(w) = 0
#define DECLARE_MUTEX_LOCKED(sem)       struct semaphore sem = MUTEX_LOCKED
#endif
```

커널을 재컴파일하고 (모듈로 설정했다면) 모듈도 컴파일 한 다음 새로운 커널과 모듈을 설치하라. 시스템을 재부팅한다.

모듈을 사용하고 있다면 모듈을 정해진 순서에 따라 적재(load) 해야 한다. usbcore.o를 먼저 적재하고 mouse.o, keyboard.o, hub.o 등의 드라이버들을 적재한 다음 마지막으로 호스트 컨트롤러(예를 들어 usb-uhci.o 또는 usb-ohci.o)를 적재한다.

커널 로그를 살펴보라. 아래와 같은 내용들을 볼 수 있을 것이다(UHCI와 외부 허브를 사용하는 경우이다).

```
.......
```

```
Jul 19 20:46:02 rachel kernel: USB HID boot protocol mouse registered.
.......
Jul 19 20:46:02 rachel kernel: uhci_control_thread at c01b8c5c
Jul 19 20:46:02 rachel kernel: New bus registered
Jul 19 20:46:02 rachel kernel: USB hub driver registered
Jul 19 20:46:02 rachel kernel: uhci_connect_change: called for 0
.......
Jul 19 20:46:02 rachel kernel: USB hub found
Jul 19 20:46:02 rachel kernel: hub: 4-ports detected
Jul 19 20:46:02 rachel kernel: hub: individual port power switching
Jul 19 20:46:02 rachel kernel: hub: standalone hub
Jul 19 20:46:02 rachel kernel: hub: individual port over current protection
Jul 19 20:46:02 rachel kernel: hub: power on to power good time: 100ms
Jul 19 20:46:02 rachel kernel: hub: hub controller current requirement: 100mA
Jul 19 20:46:02 rachel kernel: hub:  port 1 is removable
Jul 19 20:46:02 rachel kernel: hub:  port 2 is removable
Jul 19 20:46:02 rachel kernel: hub:  port 3 is removable
Jul 19 20:46:02 rachel kernel: hub:  port 4 is removable
Jul 19 20:46:02 rachel kernel: hub: local power source is good
Jul 19 20:46:02 rachel kernel: hub: no over current condition exists
Jul 19 20:46:02 rachel kernel: enabling power on all ports
Jul 19 20:46:02 rachel kernel: uhci_connect_change: called for 1
.......
Jul 19 20:46:02 rachel kernel: hub: port 3 connection change
.......
```

OHCI와 OHCI-HCD는 유사한 기능을 수행할 것이다. 그러나 두 개를 바꾸는 것에 실패한 것에 대해 그리 크게 걱정할 필요는 없다. 별다른 영향을 미치는 않는 사소한 버그이다. USB에 관련된 내용(hubs, usb, ohci, uhci를 표시하는 줄)이 전혀 없다면 아마도 잘못된 드라이버를 사용하고 있거나(OHCI를 써야 하는데 UHCI를 쓰는 경우나 그 반대의 경우), 하드웨어가 물리적으로 설치되지 않은 경우, BIOS에서 USB 기능을 꺼둔 경우, 커널의 잘못된 설정이나 설치 때문일 것이다.

제 2 절 마우스 설정

우선 마우스가 커널에 의해 제대로 인식되는지 확인하라. **more /proc/interrupts**를 실행해 보면 USB에 관한 내용(보통 **ohci-usb** 나 **usb**)을 볼 수 있을 것이다. 마우스를 몇 번 클릭한 다음 **/proc/interrupts**를 살펴보면 해당 카운트가 증가된 것을 볼 수 있을 것이다(클릭 한 번에 2씩 증가하는데 down,up 각 1씩 증가한다). UHCI에서는 카운트가 좀 어려운데 **/proc/interrupts** 를 살펴보는 동안 지연된 시간(초)을 빼야 한다. USB 키보드를 사용하고 있다면 키눌림 또한 감안해야 한다.

/proc/misc를 살펴보면 아래와 비슷한 내용을 볼 수 있을 것이다.

```
[bradh@rachel bradh]$ more /proc/misc
 32 USB Mouse
  1 psaux
```

마우스가 사용할 /dev 항목을 설정해야 한다. 아래의 명령을 사용하라.

```
mknod /dev/usbmouse c 10 32
```

X 윈도우에서 마우스를 사용하려면 아래와 같이 설정해야 한다.

- XF86Config 파일(보통 /usr/X11R6/lib/X11/XF86Config)을 수정한다. 아래의 내용을 추가하라(인식 가능한 곳 어디든 상관없지만 입력장치 항목이 바람직하다).

```
Section "Xinput"
   SubSection "Mouse"
        DeviceName   "USB Mouse"
        Protocol     "IMPS/2"
        Port         "/dev/usbmouse"
        AlwaysCore
   EndSubSection
EndSection
```

- X 서버를 재시작한다. 마우스가 작동되지 않으면 Ctrl-Alt-F1을 눌러 가상 터미널을 띄운 다음 xserver를 kill 명령으로 종료시키고 에러 메세지를 가지고 디버깅하라.

gpm에서 마우스를 사용하려면 **gpm -m /dev/usbmouse -t ps2**와 같이 실행하라(슈퍼 유저로 실행해야 한다). 부팅시 실행되는 파일을 수정하면 이것을 기본값으로 설정할 수 있다. Redhat 배포판에는 기본적으로 /etc/rc.d/ 디렉토리에 이러한 파일들이 있으며 rc.d와 비슷한 이름을 갖고 있다.

제 3 절 키보드 설정

PC를 사용하고 있다면 USB 키보드를 사용하기 위해 운영체제 지원이 전혀 필요 없을 수도 있다. 마더보드의 루트 허브에 연결된 USB 키보드를 지원하는 BIOS들이 있다. 이것은 다른 허브에서 작동할 수도 있고 안할 수도 있으며 내장보드(add-in board)에서는 보통 작동되지 않는다. 따라서 어떻든 간에 지원이 필요할 것이다.

키보드가 커널에서 정확히 인식되는지 확인하라. **more /proc/interrupts**를 실행해 보면 USB에 관한 내용(보통 **ohci-usb** 또는 **usb**)을 볼 수 있을 것이다. 키보드를 누른 다음 **/proc/interrupts**를 살펴보면 카운트가 증가된 것을 볼 수 있다. 만약 UHCI를 사용하고 있다면 지연된 시간(초)을 빼야 한다.

이제 보통의 키보드처럼 USB 키보드를 사용할 수 있을 것이다. LILO가 USB를 인식하지 못한다는 것을 기억하라. BIOS가 USB 키보드를 지원하지 않는다면 USB 키보드를 사용하는 기본 이미지 외의 다른 부트 이미지를 선택할 수 없을 수도 있다. 나는 개인적으로 USB 키보드(그리고 USB 마우스)만을 사용하고 있으며 아무 문제가 없다.

제 4 절 허브 지원

허브는 다른 설정 없이 작동될 것이다. 루트 허브에 연결된 경우에만 작동되고 내장(add-on) 허브에 연결 될 때는 작동되지 않는다면 허브 모듈을 설치했는지 확인하라(모듈로 설정한 경우이다). 또한 버스를 통해 전원이 공급되는 허브에 전력을 많이 사용하는 장치를 연결하지 않도록 한다.

제 5 절 프린터 지원

프린터용 /dev 항목을 설정해야 한다. 아래 명령을 실행하라.

```
mknod /dev/usblp c 63 0
```

이제 이 장치를 /etc/printcap 항목에서 사용할 수 있을 것이다. 약간 느리지만 나로서는 이 드라이버가 아주 훌륭하다고 생각한다. Redhat 제어판의 프린터툴과 같은 자동 설정 툴을 사용해서 장치 파일을 만들 것을 권장한다.

프린터가 작동되지 않으면 프린터의 메이저 넘버(major device number)가 바뀌었는지를 점검하라(63은 실험적 범위의 값이다). 소스코드나 /proc/devices를 살펴보면 된다. 실제로 모듈이 적재되었는지 살펴보고 /etc/printcap 항목을 다시 체크하라. 특히 장치 파일이 당신이 만든 것과 동일한지 확인하라.

제 6 절 Abstract Control Model 지원

다양한 ACM 장치를 사용하기 위해 /dev 항목을 설정해야 한다. 아래의 명령을 사용하라.

```
mknod /dev/ttyACM0 c 166 0
mknod /dev/ttyACM1 c 166 1
mknod /dev/ttyACM2 c 166 2
mknod /dev/ttyACM3 c 166 3
```

이제 이 장치를 사용하기 위한 터미널 애뮬레이터 프로그램을 사용할 수 있을 것이고 모뎀이나 다른 터미널 장치를 사용할 수 있을 것이다. 분명히 이 드라이버는 대부분 작동되지만 아직도 몇 가지 문제가 있다. 내 개인적으로 테스트 해보지는 않았다.

제 7 절 CPiA imager 지원

이 드라이버는 Vision사에서 만든 특정한 칩셋을 지원하며 USB 카메라(대표적으로 Creative WebCamll)에서 사용된다. 이 드라이버를 사용하기 위해서는 **Video For Linux** 항목을 선택해야 한다(**menuconfig**를 사용하고 있다면 **Character Devices** 항목 아래에 있다).

CPiA 카메라를 위해 /dev 항목을 설정해야 한다. 아래 명령을 실행하라.

```
mknod /dev/video0 c 81 0
ln -s /dev/video0 /dev/video
```

장치를 사용하기 위해서는 몇 가지 비디오 툴이 필요하다. 수많은 종류의 툴들이 있다. *http://millenium.diads.com/bdirks* <http://millenium.diads.com/bdirks>은 보통 배포 날짜에 따라 apps19990527.tgz와 같은 이름을 가진 패키지를 제공한다. X 윈도우 및 텍스트 모드 툴을 포함하고 있다. 텍스트 모드 툴을 아래와 같이 사용할 수 있다.

```
./vctrl 320x240x24
./vcat | rawtoppm -bgr 320 240 | xv -
```

이 툴들은 별다른 설정이 없으면 디폴트로 /dev/video를 사용하기 때문에 심볼릭 링크가 그전에 설정되어 있어야 한다.

현재로서는 CPiA 카메라는 흑백이미지만 가능하다. 나는 개인적으로 테스트 한 바 없다. CPiA 카메라를 사용하고자 한다면 Vision사의 CPiA 웹사이트 *http://home.eunet.no/~jtotland/vision* <http://home.eunet.no/~jtotland/vision>을 참고하라.

제 8 절 대용량 저장장치

추가 예정

제 9 절 EZUSB 드라이버

추가 예정

제 10 절 USS720 드라이버

USS720은 병렬 포트용 USB칩으로 Lucent사에서 만들었다. USS720칩은 USB Printer Class 장치처럼 작동한다. 사실 USS720으로 만든 브릿지를 사용하면 병렬 포트 프린터를 앞에서 본 USB 프린터 드라이버로 사용할 수 있다. 또한 USS720을 보통의 병렬포트 하드웨어로 인식하는 모드가 있다(레지스터 모드로 알려져 있다). 이 드라이버는 레지스터 모드를 사용한다.

모듈을 적재하거나 USS720 지원 커널로 재부팅 하기 전에 **/proc/sys/dev/parport**를 살펴보라. 당신의 시스템이 가진 병렬포트 수를 주목하라(보통 **/proc/sys/dev/parport/parport0** 하나이다). USS720 지원항목을 커널에 포함시킨 후에는 또하나의 포트(보통 **/proc/sys/dev/parport1**)가 있을 것이다. 해당 **하드웨어** 항목을 보면 아래와 같은 내용을 볼 수 있을 것이다.

```
[bradh@rachel bradh]$ more /proc/sys/dev/parport/parport1/hardware
base:   0x0
irq:    none
dma:    none
modes:  PCSPP,TRISTATE,COMPAT,EPP,ECP
```

모든 배포판은 병렬 포트용 **/dev** 항목을 포함하고 있다(보통 **/dev/lpx** 또는 **/dev/parx** 이며 x는 임의의 수자이다). 확신이 서지 않는다면 병렬포트는 캐릭터 메이저 넘버(character major number) 6을 사용한다는 것을 기억하라.

USS720을 병렬포트용으로 잘 쓸 수 있을 것이다. 진짜 USB 프린터처럼 프린터 연결에 사용할 **/etc/printcap** 파일을 자동설정파일로 작성하기를 권한다.

설정항목에서 USS720 지원을 선택할 수 없다면 **Parallel port support** 항목을 선택해야 한다(만약 **menuconfig**를 사용한다면 **General setup** 항목 아래에 있다).

제 11 절 USB /proc 드라이버

/proc의 USB 지원을 위해서는 **Preliminary /proc/bus/usb support**를 선택해야 한다. 그리고 위에서 설명한대로 재컴파일하고 재설치하라.

USB의 /proc 지원 항목은 현재 2가지의 인터페이스를 갖고 있는데, `/proc/bus/usb/drivers`와 `/proc/bus/usb/devices`이다.

`/proc/bus/usb/drivers`는 현재 등록된 드라이버 리스트를 보여준다(드라이버가 장치에서 사용되지 않고 있더라도). 모듈 설치 테스트나 특정(unknown) 커널에서 USB 지원 여부를 살펴볼 때 유용하다. 아래의 예를 보라.

```
[bradh@rachel bradh]$ more /proc/bus/usb/drivers
 hub
 printer
 keyboard
 mouse
```

/proc/bus/usb/devices는 현재 USB 버스에 연결된 장치에 대한 정보를 표시한다. 각 장치가 제대로 설정되었는지를 파악하는데 유용하다. 아래의 예는 루트허브, 허브, 키보드, 마우스, 프린터 어댑터 사용하는 경우이다.

```
T:  Lev=00 Prnt=00 Port=00 Cnt=00 Dev#= -1 Spd=12  If#=  0 MxCh= 2 Driver=(root hub)
T:  Lev=01 Prnt=00 Port=00 Cnt=01 Dev#=  1 Spd=12  If#=  0 MxCh= 4 Driver=hub
D:  Ver= 1.00 Cls=09(hub  ) Sub=00 Prot=00 MxPS= 8 #Cfgs=  1
P:  Vendor=0451 ProdID=1446 Rev= 1.00
C:* #Ifs= 1 Cfg#= 1 Atr=e0 MxPwr=100mA
I:  If#= 0 Alt= 0 #EPs= 1 Cls=09(hub  ) Sub=00 Prot=00
E:  Ad=81(I) Atr=03(Int.) MxPS=   1 Ivl=255ms
T:  Lev=02 Prnt=01 Port=00 Cnt=01 Dev#=  2 Spd=12  If#=  0 MxCh= 0 Driver=printer
D:  Ver= 1.00 Cls=00(>ifc ) Sub=00 Prot=00 MxPS= 8 #Cfgs=  1
P:  Vendor=0729 ProdID=1284 Rev= 1.04
C:* #Ifs= 1 Cfg#= 1 Atr=80 MxPwr= 98mA
I:  If#= 0 Alt= 0 #EPs= 1 Cls=07(print) Sub=01 Prot=01
E:  Ad=01(O) Atr=02(Bulk) MxPS=  64 Ivl=  0ms
I:  If#= 0 Alt= 1 #EPs= 2 Cls=07(print) Sub=01 Prot=02
E:  Ad=01(O) Atr=02(Bulk) MxPS=  64 Ivl=  0ms
E:  Ad=82(I) Atr=02(Bulk) MxPS=  64 Ivl=  0ms
I:  If#= 0 Alt= 2 #EPs= 3 Cls=ff(vend.) Sub=00 Prot=ff
E:  Ad=01(O) Atr=02(Bulk) MxPS=  64 Ivl=  0ms
E:  Ad=82(I) Atr=02(Bulk) MxPS=  64 Ivl=  0ms
E:  Ad=83(I) Atr=03(Int.) MxPS=   4 Ivl=  1ms
T:  Lev=02 Prnt=01 Port=01 Cnt=02 Dev#=  3 Spd=1.5 If#=  0 MxCh= 0 Driver=keyboard
D:  Ver= 1.00 Cls=00(>ifc ) Sub=00 Prot=00 MxPS= 8 #Cfgs=  1
P:  Vendor=045e ProdID=000b Rev= 0.82
C:* #Ifs= 1 Cfg#= 1 Atr=a0 MxPwr=100mA
I:  If#= 0 Alt= 0 #EPs= 1 Cls=03(HID  ) Sub=01 Prot=01
E:  Ad=81(I) Atr=03(Int.) MxPS=   8 Ivl= 10ms
T:  Lev=02 Prnt=01 Port=02 Cnt=03 Dev#=  4 Spd=1.5 If#=  0 MxCh= 0 Driver=mouse
D:  Ver= 1.00 Cls=00(>ifc ) Sub=00 Prot=00 MxPS= 8 #Cfgs=  1
P:  Vendor=046d ProdID=c001 Rev= 1.10
C:* #Ifs= 1 Cfg#= 1 Atr=a0 MxPwr= 50mA
I:  If#= 0 Alt= 0 #EPs= 1 Cls=03(HID  ) Sub=01 Prot=02
E:  Ad=81(I) Atr=03(Int.) MxPS=   8 Ivl= 10ms
```

/proc/bus/usb/devices 결과는 몇 가지 그룹으로 나누어진다.

- **T:**로 시작하는 행은 토폴로지(topology)이다. **Lev**는 장치의 레벨을 표시한다. 레벨 00은 루트허브이고 레벨 01은 루트허브에 연결된 장치, 레벨 02는 레벨01인 허브에 연결된 장치 등이다. **Prnt**는 이 장치의 상위 장치이다(00은 루트허브와 루트허브에 연결된 장치를 의미한다). **Port**는 상위 장치의 포트이다. 00은 각 장치의 첫째 포트이다. **Prnt/Port**는 각 버스에서 유일한 값을 갖는다. 토폴로지 레벨 안에서 배열 순서에 따라 **Cnt**는 해당 레벨에서 이 장치의 번호를 표시한다. 첫 번째 장치인 01부터 시작한다. **Dev#**는 버스 배열 순서에 따라 레벨에 관계없는 장치 번호이다. **Spd**는 장치가 작동되는 속도를 Mbps로 표시한다(현재 버전에서는 1.5나 1.2이다). **If#**는 현재 사용되는 인터페이스 번호이다. **MxCh**는 이 장치에 연결될 수 있는 최대 장치 수이다. 허브 외에는 모두 00이다. **Driver**는 이 장치에 사용되고 있는 장치 드라이버를 표시한다('none' 항목은 아무런 드라이버도 사용되지 않음을 의미한다).
- **D:**로 시작되는 행은 장치 디스크립터(descriptor) 정보이다. **Ver**는 장치가 요구하는 USB 사양 버전을 표시한다. **Cls**는 장치가 요청하는 장치 클래스를 표시하며 16진수와 문자열로 표시한다. **Cls** 항목 중

00(>ifc)은 장치 클래스 사양 요구가 인터페이스와 독립적임을 의미하며 인터페이스 디스크립터는 장치 클래스 정보를 포함하고 있어야 한다. **Sub**는 장치가 수용하는 서브 클래스(**Cls** 항목 안에서)를 표시한다. **MxPS**는 엔드포인트(Endpoint) 0으로부터의 패킷의 최대 크기를 표시한다. **Cfgs**는 장치의 설정 개수를 표시한다.

- **D:**와 유사하게 **P:**로 시작하는 행은 장치 디스크립터의 정보를 표시하지만 한 줄에 표시할 수 없기 때문에 분리되어 있다. **Vendor**는 장치의 판매자 고유 코드를 표시하며 **ProdID**는 제품 고유 코드를 표시한다. **Rev**는 제품 개량 번호이다.
- USB 사양서 9.7.1절을 참고하면 장치 디스크립터에 대한 더 많은 정보를 얻을 수 있다.
- **C:**로 시작하는 행은 설정 디스크립터의 정보이다. 각 장치의 **C:**행의 번호는 **#Cfgs**에 의해 할당된다. 아스테리크 문자가 붙은 항목은 현재 설정을 의미한다. **#If**는 장치 인터페이스 수를 표시한다. **Cfg#**는 어떤 설정이 현재 사용되는지를 보여준다. **Atr**은 장치 속성을 16진수로 표시한 것이다(0x80은 전력지원 버스, 0x40은 자체 전원, 0x20은 원격 시동 가능). **MPwr**은 현재 장치 설정에서의 최대 전력 사용 값이며 밀리암페어로 표시한다. USB 사양서 9.7.2절을 참고하면 설정 디스크립터의 추가 정보를 얻을 수 있다.
- **I:**로 시작하는 행은 인터페이스 디스크립터의 정보이다. 각 **C:**행의 **I:**행 번호는 **#Ifs** 항목에 의해 할당된다. **If#**는 해당 장치 설정에서 어느 인터페이스가 사용되는지를 표시한다. **Alt**는 해당 인터페이스에 어떤 대체 설정이 설정되어 있는지를 보여준다. **#EPs**는 해당 엔드포인트의 대체 설정 가운데 얼마나 많은 엔드포인트가 있는지를 보여준다. **Cls**는 대체 인터페이스 설정이 어느 클래스에 해당되는지를 16진수와 문자열로 나타낸다. **Sub**는 인터페이스의 대체 설정이 어느 하위 클래스에 해당되는지를 나타낸다. **Prot**는 대체 인터페이스가 어떤 인터페이스 프로토콜(클래스와 하위클래스 집합 가운데에서)에 적합한지를 보여준다. USB 사양서 9.7.3절을 참고하라.
- **F:**로 시작하는 행은 인터페이스 디스크립터의 정보이다. 각각의 **I:** 행에서의 **E:**행 번호는 **#EPs** 항목에서 주어진다. 엔드포인트 0은 표시되지 않는다. **Ad**는 엔드포인트 주소를 나타내며 엔드포인트가 입력/출력 가운데 어느 것인지를 나타내는 문자를 포함하고 있다. **Atr**은 엔드포인트에 사용되는 속성(전송타입)을 나타내며 전송타입을 설명하는 문자열이 뒤에 나온다. **MxPS**는 해당 엔드포인트가 적절히 송수신할 수 있는 최대 패킷의 크기를 나타낸다. 등시 전송(isochronous transfer)의 경우 **MxPS**는 요구되는 대역폭의 크기를 나타낸다. **Ivl**은 엔드포인트 인터럽트 호출의 간격을 밀리초 단위로 나타낸다. **Ivl**은 벌크와 제어 전송(bulk and control transfers)에서는 무시되며 등시전송에서는 1로 설정된다. 엔드포인트 디스크립터에 관한 자세한 것은 USB 사양서 9.7.4절을 보라.

`/proc/bus/usb`에 관한 더 자세한 내용은 `linux/Documentation/proc_usb_format.txt`를 참고하라.

제 12 절 자주 질문 받는 것들(FAQs)

이장은 반복되는 질문들에 대한 답변이다.

드라이버를 만들려면 어떻게 해야하나요

커널소스를 공부하라. 당신이 원하는 것과 유사한 드라이버가 있으면 작동될 때까지 개조하라. 여기서 유사하다는 것은 당신이 전송하고자 하는 것에 관한 것이지 장치가 어떻게 생겼는지를 말하는 것이 아니다.

USB에 관한 좋은 책이 있다면

나는 단지 한 권을 읽었는데 매우 부실하다고 생각했다. *http://www.usb.org* `<http://www.usb.org>`에서 사양서를 구해보기를 권한다. 그것은 무료이며 읽기 쉽고 최신 내용들이다.

특정 업체가 독자적으로 만든(vendor specific) 장치를 갖고 있어요. 디스크립터는 이렇게 되어 있구요

디스크립터는 드라이버를 만드는 데에 필수적이나 그것만으로 충분하지는 않다. 업체로부터 저수준 설계 세부 내용을 알아야 한다. 게다가 디스크립터가 사양을 준수하지 않는다 하여 놀라지 마라 - 내가 해 줄 수 있는 말은 그 업체의 장치를 사용하지 말라는게 전부다. 디스크립터는 단순한 장치용 데이터 바이트이며, 장치를 재프로그래밍하면 장치의 용도에 관계없이 어떤 문자를 표시되게 할 수 있다.

드라이버에 어떤 메이저 넘버와 마이너 넘버를 사용해야 하나요

실험적 또는 시험용 범위의 번호를 사용하라. 현재로서는 USB에 할당된 공식 메이저 넘버는 없다. 마이너는 아무거나 사용하라.

Sony Vaio laptop에서 USB를 사용하고 싶은데요

BIOS에서 Plug-n-Pray 운영체제 지원 항목을 선택하지 않으면 잘 작동할 것이다.

USB-HOWTO 문서의 최신판은 어디서 구하나요

여깁니다 *http://www.dynamine.net/linux-usb/HOWTO* `<http://www.dynamine.net/linux-usb/HOWTO>`.

제 13 절 정정

이 문서에 관해서 얘기할 것이 있으면 되도록이면 `brad.hards@dao.defence.gov.au`로 메일을 보내는 것이 좋다. 보낼 때는 버전 번호 ($Revision: 0.7 $)를 명시하도록 한다.