※ 참고: CD(V1.6.0.1)에서 복사하는 파일들은 버전에 따라 파일명이 다를 수 있으니 ls 명령어를 사용하여 확인하고 복사

또한 CD의 사용 대신에 PC 하드에 CD를 복사하고 폴더 공유 기능을 이용해 작업하면 편리함

0. 개발환경 구축

0.1 VMware Player 설치

0.2 Ubuntu 12.04 64bit 설치 (약 15~20분)

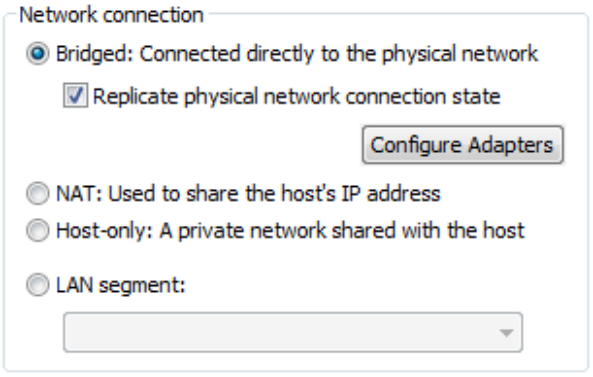
반드시 이 버전을 다운로드해 사용. 중간에 업데이트도 금지

→ 모든 실습환경이 여기에 맞추어 설정되어 있음

안될 경우, BIOS -> CPU -> Virtualization ->

Intel Virtualization Technology [Enable] 설정

Network Setup



Configure Adapters를 선택 → PC LAN 카드만 마크함

(Default로 NAT 인 네트워크를 **Bridged** 모드로 변경)

※ 네트워크 상태에 따라 네트워크가 불안할 경우 NAT로 설정을 하고 진행 (2. 디바이스 드라이버까지는 문제 없음. 이후에도 Bridged 모드가 불안할 때는 SD카드를 PC로 꼽고 SD카드에 내용을 복사해서 타겟에 다시 꽂은 후 실습 가능)

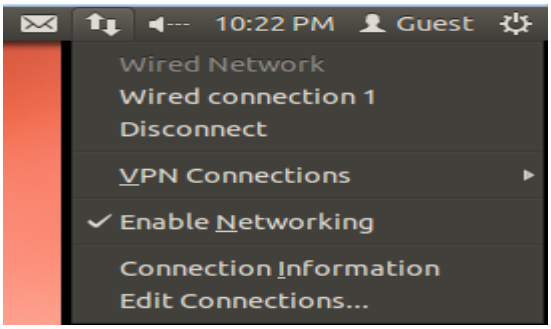
※ 실습에 사용되는 IP 종류

1) PC IP: 자기 PC의 IP → 확인 필요

2) 우분투(리눅스) 호스트 IP: PC IP + 50

3) 타겟 보드 IP: PC IP + 100

Ubuntu IP Address Setup 방법:



[Edit Connections...] 클릭 -> Wired connection 1 선택 후 [Edit] -> IPv4 Settings -> Method [Manual] -> [Add] -> IP 입력, DNS 입력

Address 10.40.1.(본인 PC IP+50)

NetMask 255.255.255.0

Gateway 10.40.1.254

DNS server 210.93.48.196

※ PC와의 공유폴더를 만들고 CD ROM 복사

※ 새로운 터미널 창 열기: Ctrl+Alt+T

0.3 루트 패스워드 설정

sudo passwd root

0.4 루트 계정 로그인 설정

방법-1: 초기부터 root 계정 로그인 하는 방법

sudo vi /etc/lightdm/lightdm.conf

/etc/lightdm/lightdm.conf
.... greeter-show-manual-login=true ....

Login에서 Username (root) –> passwd 입력 로그인

방법-2: 사용자 계정으로 로그인 → root 계정 전환

#su –

※ 모든 작업은 root 계정에서 실시

0.5 32bit 라이브러리 및 vim 설치

apt-get update (패키지 인덱스 정보 업데이트)

(에러 시 /etc/apt/sources.list 수정, DNS변경)

apt-get upgrade (최신 패키지로 업그레이드 → 15분~20분 소요)

apt-get install vim ia32-libs

(32bit lib패키지 미지원 시 gcc-multilib 설치) → 5분 정도 소요. 이 경우 VMtool 업데이트를 안하면 그럴 수도 있음

0.6 minicom 설치

apt-get install minicom

minicom –s

Serial port setup
A – Serial Device : /dev/tty~~~~~
E – Bps/Par/Bits : 115200 8N1
F – Hardware Flow Control : No
G – Software Flow Control : No)

(미니컴 환경설정,

통신 포트 선택 COM1 → /dev/ttyS0

COM2 → /dev/ttyS1

USB-to-Serial → /dev/ttyUSB0

* #dmesg | grep serial 명령어를 사용해서 나오는 장치이름 (ttyS0 또는 ttyS1을 사용하여 설정해 볼 것)

* Save setup as dfl 로 설정값 저장

▶ 보드의 CONSOLE(Ethernet 바로 옆)에 호스트 PC의 시리얼 포트를 연결한다

0.7 TFTP 설치

apt-get install tftp tftpd xinetd

mkdir /tftpboot

vi /etc/xinetd.d/tftp (정확히 타이핑 할 것!!)

/etc/xinetd.d/tftp

```
service tftp
{
    socket_type     = dgram
    protocol        = udp
    wait            = yes
    user            = root
    server          = /usr/sbin/in.tftpd
    server_args     = -s /tftpboot
    disable         = no
    per_source      = 11
    cps             = 100 2
    flags           = IPv4
}
```

service xinetd restart

▶ TFTP 테스트

(대부분의 error가 /etc/xinetd.d/tftp의 오타로 발생. Time out시 재작성 할 것)

cd /tftpboot

cat > /tftpboot/test.txt

Hello World (Ctrl+D를 눌러 입력 내용 저장)

cd /root

tftp localhost

```
tftp> get test.txt
tftp> quit
```

ls /root

[.... test.txt ....]

cat test.txt

[Hello World]

0.8 NFS 설치

apt-get install nfs-kernel-server

vim /etc/exports (서버에서 공유될 디렉토리 및 옵션들 설정)

/etc/exports

```
....
/nfsroot        *(rw,sync,no_root_squash,no_subtree_check)
....
```

mkdir /nfsroot

service nfs-kernel-server restart

▶ NFS 테스트

mkdir /mnt/nfs (클라이언트에서 서버와 공유할 디렉토리 설정)

mount –t nfs localhost:/nfsroot /mnt/nfs (수초 소요)

cd /nfsroot

touch test1 (서버에서 파일을 하나 생성)

ls /mnt/nfs (test 파일이 있는지 확인)

umount /mnt/nfs (연결 해제)

0.9 작업 디렉토리 생성

mkdir -p /work/achro5250

(-p : 상위 디렉토리까지 포함하여 디렉토리 생성)

0.10 툴체인 설치

dpkg-reconfigure –plow dash ('No'를 선택)

▶실습장비와 함께 제공된 CD 삽입 후 탐색기 화면이 뜬 이후 다음 명령을 실행하여 파일 복사

▶CD 내용을 PC 폴더에 공유했으면 향후 CD가 없어도 /mnt/hgfs/~~ 디렉토리 밑에서 복사 가능

cp –a [CD 위치]/toolchain/arm-2010q1-203-arm-none-linux-gnueabi-i686-pc-linux-gnu.tar.bz2 /work/achro5250 *(같은 줄에 타이핑)* → **파일명이 다를 수 있으니 확인 필요(ls 이용)**

mkdir /opt/toolchains

cd /work/achro5250

tar xvfj arm-2010q1-202-arm-none-linux-gnueabi-i686-pc-linux-gnu.tar.bz2 **–C** /opt/toolchains/ (-C: 경로 지정 옵션)

vi /root/.bashrc

~/.bashrc

```
If[-f ~/.bash_aliases]; then
. ~/.bash_aliases
fi

(파일의 제일 아래에 다음 3라인을 추가)
export ARCH=arm
export CROSS_COMPILE=arm-none-linux-gnueabi-
export PATH=/opt/toolchains/arm-2010q1/bin:$PATH
.....
```

source /root/.bashrc

cd /work/achro5250/

vi helloworld.c (새로운 파일을 만들어 개발환경 확인)

hellowrold.c

```
#include <stdio.h>
int main(int argc, char** argv)
{
   printf("Hello world!!₩n");
   retrun 0;
}
```

arm-none-linux-gnueabi-gcc –o helloworld helloworld.c

file ./helloworld (컴파일된 바이너리의 확인)

[ELF 32-bit LSB executable, ARM, version 1 ~~~~~]

arm-none-linux-gnueabi-gcc –-v (컴파일러 버전 확인 시 실행)

0.11 USB 드라이버 설치 및 smdk-usbdl 복사

apt-get install libusb-dev

cd [CD 위치]/utilities/linux_tools/smdk-usbdl

cp smdk-usbdl.tar.gz /work/achro5250

cd /work/achro5250

tar xvfz smdk-usbdl.tar.gz

cd dltool

cp -a smdk-usbdl /usr/bin

1. 시스템 이미지 제작

1.1 부트로더 컴파일

cd /work/achro5250

cp –a [CD 위치]bootloader/u-boot_140806.tar.gz .

(점 주의)

tar xvfz u-boot_140806.tar.gz

cd u-boot/sd_fuse

make

cp mkbl2 /work/achro5250/u-boot

cd /work/achro5250/u-boot

make distclean (이전 부트로더 설정환경 초기화)

make clean (이전 make 과정에서 생긴 파일들 제거 *.bak, *.o, *,a 등)

make achro5250_config (타겟보드용 환경 설정)

make (타겟보드용 부트로더 바이너리 파일 생성)

1.2 부트로더 퓨징

◈ MicroSD카드로 부팅할 경우 모바일보드 뒷면의 OM핀 중 3번만 ON, 나머지는 모두 OFF한 후 진행할 것

▶호스트 PC에서 진행

▶장치에 있는 system SD 메모리 카드를 빼서 USB 젠더에 연결한 후 호스트 PC의 USB 커넥터에 연결한다.

pwd

[/work/achro5250/u-boot]

ls /dev/sd* (메모리 디바이스 확인. sdb가 있어야 함)

[/dev/sda /dev/sda1 ..... /dev/sdb]

dd if=/dev/zero of=/dev/sdb bs=64k (SD카드 초기화)

※ dd 명령의 진행상황을 알고 싶을 때는 또 다른 터미널을 열고 다음 명령을 실행한다. 30초 마다 상태가 보여진다

4GB SD의 경우 약 15분 정도 소요

#watch -n30 'sudo kill -USR1 $(pgrep ^dd)'

▶완료 후 SD 카드 재연결

./mksdcard-boot.sh /dev/sdb (SD카드 RAW 파일영역 설정/복사)

▶SD 카드 파티션 설정

fdisk /dev/sdb

Command (m for help) : **n**
Partition type:
P primary(0 primary, 0 extended, 4 free)
e extended
Select (default p) : **p**
Partition number (1-4, default 1) : **1**
First sector (2048-**(SD 카드마다 다를 수 있음),** default 2048) : **1050624**
Last sector, +sectors or +size{K,M,G} (1050624-**(SD 카드 끝)**, default **(SD 카드 끝)**): **+512M**

Command (m for help) : **p**
.....
Device Boot Start End Block .....
/dev/sdb1 1050624 2099199 .....
.....
Command (m for help) : **n**
Partition type:
P primary(0 primary, 0 extended, 4 free)
e extended
Select (default p) : **p**
Partition number (1-4, default 2) : **2**
First sector (2048-**(SD 카드마다 다를 수 있음)),** default 2048) : **2099200**
(SD 카드마다 다를 수 있음. End+1 입력,)

Last sector, +sectors or +size{K,M,G} (2026048-**(SD 카드 끝)**, default **(SD 카드 끝)**): **Enter**

Command (m for help) : **w**

mkfs.ext3 –L Achro5250_System /dev/sdb1

mkfs.ext3 –L Achro5250_Data /dev/sdb2

▶ SD카드를 제거해서 보드에 넣는다

1.3 커널 컴파일

cd /work/achro5250

cp –a [CD 위치]/kernel/kernel_140806.tar.gz . (점 주의)

tar xvfz kernel_140806.tar.gz

cd /work/achro5250/kernel

make distclean

make achro5250_defconfig

make (십여분 소요)

1.4 커널 퓨징

▶타겟 보드와 호스트 PC를 USB-Serial 케이블로 연결한다.

▶타겟 보드와 호스트 PC를 USB OTG 케이블로도 연결한다.

▶타겟에서 (부트롬 모드에서) 다음 명령을 실행한다:

dnw 0x40000000

▶호스트에서 다음 명령을 실행한다:

smdk-usbdl –f arch/arm/boot/zImage

▶타겟에서 T-flash로 이미지 쓰기:

movi write kernel S5P_MSHC2 0x40000000

reset

※ 커널 부팅 후 리눅스 파일 시스템을 찾게 되므로 파일 시스템이 없으면 커널 패닉이 발생함.

1.5 리눅스 파일 시스템 퓨징

cd /work/achro5250

cp –a [CD 위치]/filesystem/linux_filesystem/linuxfs_20130312.tar.gz . (점 주의)

tar xvfz linuxfs_20130312.tar.gz

▶ACHRO5250의 전원을 끈다.

▶장치에 있는 system SD 메모리 카드를 빼서 USB 젠더에 연결한 후 호스트 시스템의 USB 커넥터에 연결한다.

▶파일시스템은 시간 관계상 기존 이미지를 그대로 fusing

▶아래 내용은 리눅스 호스트 창에서 실행하는 것임.

cd /media/Achro5250_System

rm –rf *

cd /work/achro5250/linuxfs

cp –a * /media/Achro5250_System

sync

umount /media/Achro5250_System

umount /media/Achro5250_Data

▶마운트된 SD카드를 제거하여 보드의 System SD 소켓에 장착하고 전원을 켠다.

▶ NFS 테스트

(타겟 보드 리눅스 파일시스템 퓨징 후에 시행)

Ethernet으로 타겟과 호스트를 연결(PC와 직접적인 선의 연결은 아님)

(다음 내용들은 타겟에서 실행)

타겟에서도 항상 root 계정으로 입력 (root 입력)

vi /etc/network/interfaces

/etc/network/interfaces

```
# Configure Loopback
auto lo
iface lo inet loopback

# User Network setting
auto eth0
iface eth0 inet static
        address 10.40.1.(PC의 IP 번호 + 100)
        netmask 255.255.255.0
        broadcast 10.40.1.255
        gateway 10.40.1.254
```

reboot

mount -t nfs 10.40.1.(리눅스 호스트번호):/nfsroot /mnt/nfs -o tcp,nolock

(※ 리눅스 호스트 번호 = PC IP 번호 + 50)

ls /mnt/nfs

[... test1 ....] (test1 파일이 출력되는지 확인)

2. 디바이스 드라이버

2.1 모듈 프로그래밍

▶ 모듈 프로그래밍 예제 파일 복사

cd [CD 위치]/examples/linux/module

cp –a module.tar.gz /work/achro5250

cd /work/achro5250

tar xvfz module.tar.gz

cd module

vi Makefile (수정)

Makefile

```
…
KDIR := /work/achro5250/kernel
…
```

make

cp hello_module.ko /nfsroot

▶ 타겟 보드에서 NFS 연결

※ NFS 연결 전, ping 을 통하여 네트워크가 연결 됐는지를 먼저 확인할 것

※ 이후부터 네트워크 상태에 따라 NFS가 안될 경우에는 SD카드를 PC로 꼽고 SD카드에 내용을 복사해서 타겟에 다시 꽂은 후 실습 가능하다. 예) 현재 디렉토리의 파일을 타겟의 root 디렉토리로 복사하는 경우

 # cp <복사할 파일> /media/Achro5250_System/root

ping 10.40.1.(리눅스 호스트 번호)

mkdir /mnt/nfs (디렉토리가 없을 경우 생성)

mount -t nfs 10.40.1.(리눅스 호스트번호):/nfsroot /mnt/nfs -o tcp,nolock

cp /mnt/nfs/hello_module.ko /root

cd root

insmod hello_module.ko

[Hello, Welcome to module!!]

#rmmod hello_module

[GoodBye, Exit Module!!]

2.2 디바이스 드라이버 예제 파일 가져오기

▶ 디바이스 드라이버 예제 파일들 복사 (**리눅스 호스트**)

mkdir /work/achro5250/device_driver

cd /work/achro5250/device_driver

cp –a [CD 위치]/examples/linux/fpga_driver/* .

2.3 디바이스 드라이버 예제 소스 컴파일 및 실행

◈ 디바이스 드라이버 실행시 FPGA 보드내의 스위치 SW7의 1번을 ON으로 하고, 2번은 OFF로 하고 전원 연결 후, 보드 전원 스위치를 켤 것

[예제 공통]

▶각 예제의 Makefile 수정

```
.....
KDIR :=/work/achro5250/kernel
....
app:
      arm-none-linux-gnueabi-gcc –static .....................
....
```

※ 위와 같이 각 디바이스 드라이버 디렉토리 내의 Makefile 을 수정하지 않으려면 다음과 같이 kernel과 toolchain 명령어들에 대해 symbolic link 를 걸어주면 됨

cd /work/achro5250

ln –s kernel kernel-20130306

cd /opt/toolchains/arm-2010q1/bin

ftp computer.kpu.ac.kr

Name (computer.kpu.ac.kr: root): **anonymous**

Passwd: **anonymous**

ftp> **cd achro5250**

ftp> **get cross-symlink.sh**

ftp> **bye**

sh cross-symlink.sh

※ sh 명령에서 실행이 안되고 ^M 관련 에러가 발생하면 'dos2unix' 방법으로 해결

① fpga_led

cd /work/achro5250/device_driver

tar xvfz fpga_led.tar.gz

cd fpga_led

▶ Makefile 수정 *(symbolic link 설정 시 안 해도 됨)*

make

make install (/nfsroot로 실행파일들 복사)

(# make && make install 로 동시에 실행이 가능)

▶ 타겟 보드에서 NFS 마운트 후 진행

mount -t nfs 10.40.1.(리눅스 호스트번호):/nfsroot /mnt/nfs -o tcp,nolock

(복사하지 않고 /mnt/nfs 디렉토리로 가서 실행해도 무방)

cp –a /mnt/nfs/fpga_test_led /root

cp –a /mnt/nfs/fpga_led_driver.ko /root

insmod fpga_led_driver.ko

mknod /dev/fpga_led c 260 0

(기존에 존재한다고 나오면 무시하고 진행)

./fpga_test_led 1 (타겟 보드의 LED 확인)

② fpga_FND (Seven Segment)

cd /work/achro5250/device_driver

tar xvfz fpga_fnd.tar.gz

cd fpga_fnd

▶ Makefile 수정 *(symbolic link 설정 시 안 해도 됨)*

make && make install

▶ 타겟 보드에서 NFS 마운트 후 진행

mount -t nfs 10.40.1.(리눅스 호스트번호):/nfsroot /mnt/nfs -o tcp,nolock

(복사하지 않고 /mnt/nfs 디렉토리로 가서 실행해도 무방)

cp –a /mnt/nfs/fpga_test_fnd /root

cp –a /mnt/nfs/fpga_fnd_driver.ko /root

insmod fpga_fnd_driver.ko

mknod /dev/fpga_fnd c 261 0

(기존에 존재한다고 나오면 무시하고 진행)

./fpga_test_fnd 1234 (타겟 보드의 FND 확인)

③ fpga_dot (Dot Matrix)

cd /work/achro5250/device_driver

tar xvfz fpga_dot.tar.gz

cd fpga_dot

▶ Makefile 수정 *(symbolic link 설정 시 안 해도 됨)*

make && make install

▶ 타겟 보드에서 NFS 마운트 후 진행

mount -t nfs 10.40.1.(리눅스 호스트번호):/nfsroot /mnt/nfs -o tcp,nolock

(복사하지 않고 /mnt/nfs 디렉토리로 가서 실행해도 무방)

cp –a /mnt/nfs/fpga_test_dot /root

cp –a /mnt/nfs/fpga_dot_driver.ko /root

insmod fpga_dot_driver.ko
mknod /dev/fpga_dot c 262 0
(기존에 존재한다고 나오면 무시하고 진행)
./fpga_test_dot 1 (타겟 보드의 Dot Matrix 확인)

④ fpga_text_lcd

cd /work/achro5250/device_driver
tar xvfz fpga_text_lcd.tar.gz
cd fpga_text_lcd
▶ Makefile 수정 *(symbolic link 설정 시 안 해도 됨)*
make
make install
(# make && make install)

▶ 타겟 보드에서 NFS 마운트 후 진행
mount -t nfs 10.40.1.(리눅스 호스트번호):/nfsroot /mnt/nfs -o tcp,nolock

(복사하지 않고 /mnt/nfs 디렉토리로 가서 실행해도 무방)
insmod fpga_text_lcd_driver.ko
mknod /dev/fpga_text_lcd c 263 0
(기존에 존재한다고 나오면 무시하고 진행)
./fpga_test_text_lcd hello world
(타겟 보드의 LCD 확인)

⑤ fpga_dip_switch

cd /work/achro5250/device_driver
tar xvfz fpga_dip_switch.tar.gz
cd fpga_dip_switch
▶ Makefile 수정 *(symbolic link 설정 시 안 해도 됨)*
make && make install

▶ 타겟 보드에서 NFS 마운트 후 진행
mount -t nfs 10.40.1.(리눅스 호스트번호):/nfsroot /mnt/nfs -o tcp,nolock

(복사하지 않고 /mnt/nfs 디렉토리로 가서 실행해도 무방)
cp –a /mnt/nfs/fpga_dip_switch /root
cp –a /mnt/nfs/fpga_dip_switch_driver.ko /root

insmod fpga_dip_switch_driver.
mknod /dev/fpga_dip_switch c 266 0
(기존에 존재한다고 나오면 무시하고 진행)
./fpga_test_dip_switch
(타겟 보드의 dip switch(SW1)를 변경 해보면서 확인)
(종료시 Ctrl+C 입력)

⑥ fpga_push_switch

cd /work/achro5250/device_driver
tar xvfz fpga_push_swtich.tar.gz
cd fpga_push_swtich
▶ Makefile 수정 *(symbolic link 설정 시 안 해도 됨)*
make && make install

▶ 타겟 보드에서 NFS 마운트 후 진행
mount -t nfs 10.40.1.(리눅스 호스트번호):/nfsroot /mnt/nfs -o tcp,nolock

(복사하지 않고 /mnt/nfs 디렉토리로 가서 실행해도 무방)
cp –a /mnt/nfs/fpga_test_push_swtich /root
cp –a /mnt/nfs/fpga_push_swtich_driver.ko /root

insmod fpga_push_swtich_driver.ko
mknod /dev/fpga_push_swtich c 265 0
(기존에 존재한다고 나오면 무시하고 진행)
./fpga_test_push_switch
(타겟 보드의 push switch를 눌러보며 확인)

⑦ fpga_buzzer

cd /work/achro5250/device_driver
tar xvfz fpga_buzzer.tar.gz
cd fpga_buzzer
▶ Makefile 수정 *(symbolic link 설정 시 안 해도 됨)*
make && make install

▶ 타겟 보드에서 NFS 마운트 후 진행
mount -t nfs 10.40.1.(리눅스 호스트번호):/nfsroot /mnt/nfs -o tcp,nolock

(복사하지 않고 /mnt/nfs 디렉토리로 가서 실행해도 무방)

cp –a /mnt/nfs/fpga_buzzer /root

cp –a /mnt/nfs/fpga_buzzer_driver.ko /root

insmod fpga_buzzer_driver.

mknod /dev/fpga_buzzer c 264 0

(기존에 존재한다고 나오면 무시하고 진행)

./fpga_test_buzzer (비프음 출력)

⑧ fpga_step_motor

cd /work/achro5250/device_driver

tar xvfz fpga_step_motor.tar.gz

cd fpga_step_motor

▶ Makefile 수정 *(symbolic link 설정 시 안 해도 됨)*

make && make install

▶ 타겟 보드에서 NFS 마운트 후 진행

mount -t nfs 10.40.1.(리눅스 호스트번호):/nfsroot /mnt/nfs -o tcp,nolock

(복사하지 않고 /mnt/nfs 디렉토리로 가서 실행해도 무방)

cp –a /mnt/nfs/fpga_test_step_motor /root

cp –a /mnt/nfs/fpga_step_motor_driver.ko /root

insmod fpga_step_motor_driver.ko

mknod /dev/fpga_step_motor c 267 0

(기존에 존재한다고 나오면 무시하고 진행)

./fpga_test_step_motor 1 1 50

./fpga_test_step_motor 0 0 0

(타겟 보드의 step motor가 움직이는지 확인)

3. mmap()을 이용한 디바이스 응용 어플리케이션

▶ 예제 가져오기

cd /work/achro5250

ftp computer.kpu.ac.kr

Name (computer.kpu.ac.kr: root): **anonymous**

Passwd: **anonymous**

ftp> **cd achro5250**

ftp> **get mmap.tar.gz**

ftp> **bye**

tar xvf mmap.tar.gz

cd mmap

① mmap_led

arm-linux-gcc –o mmap_led mmap_led.c

(arm-linux-gcc가 동작되지 않으면 심볼릭 링크 설정이 안된 것임. arm-none-linux-gnueabi-gcc 명령을 사용할 것)

cp mmap_led /nfsroot

▶ 타겟 보드에서 NFS 마운트 후 진행

mount –t nfs 10.40.1.*51*:/nfsroot /mnt/nfs –o tcp,nolock *(같은 줄에 타이핑)*

cp –a /mnt/nfs/mmap_led /root

cd /root

./mmap_led

② mmap_fnd

arm-linux-gcc –o mmap_fnd mmap_fnd.c

(arm-linux-gcc가 동작되지 않으면 심볼릭 링크 설정이 안된 것임. arm-none-linux-gnueabi-gcc 명령을 사용할 것)

cp mmap_fnd /nfsroot

▶ 타겟 보드에서 NFS 마운트 후 진행

mount –t nfs 10.40.1.*51*:/nfsroot /mnt/nfs –o tcp,nolock *(같은 줄에 타이핑)*

cp –a /mnt/nfs/mmap_fnd /root

cd /root

./mmap_fnd

▶ 다른 문자 디바이스들에 대해서도 mmap_led.c나 mmap_fnd.c를 수정하여 테스트 프로그램을 작성해 볼 것.

4. 디바이스 응용 어플리케이션

3.1 Frame Buffer

▶ 프레임 버퍼 예제 가져오기

cd [CD 위치]/examples/linux/application

cp –a frame_buffer.tar.gz /work/achro5250

cd /work/achro5250

tar xvfz frame_buffer.tar.gz

3.2 프레임 버퍼 예제 소스 컴파일 및 실행

pwd

[/work/achro5250/framebuffer]

① fbinfo

arm-none-linux-gnueabi-gcc –o fbinfo fbinfo.c

cp fbinfo /nfsroot

▶ 타겟 보드에서 NFS 마운트 후 진행

cp –a /mnt/nfs/fbinfo /root

./fbinfo

② fbset

▶ **fbset**을 실행하여 화면의 해상도를 변경한 경우에는 다른 예제 프로그램의 실행에 문제가 있을 수 있으니 해상도를 원래의 값으로 되돌려 주고 다른 예제 프로그램을 실행할 것 (자신 없으면 fbset 은 실행하지 말기 바람!!)

arm-none-linux-gnueabi-gcc –o fbset fbset.c

cp fbset /nfsroot

▶ 타겟 보드에서 NFS 마운트 후 진행

cp –a /mnt/nfs/fbset /root

./fbset

③ fbpixel

arm-none-linux-gnueabi-gcc –o fbpixel fbpixel.c

cp fbpixel /nfsroot

▶ 타겟 보드에서 NFS 마운트 후 진행

cp –a /mnt/nfs/fbpixel /root

./fbpixel

④ fbrect

arm-none-linux-gnueabi-gcc –o fbrect fbrect.c

cp fbrect /nfsroot

▶ 타겟 보드에서 NFS 마운트 후 진행

cp –a /mnt/nfs/fbrect /root

./fbrect

⑤ fbranrect

arm-none-linux-gnueabi-gcc –o fbranrect fbranrect.c

cp fbranrect /nfsroot

▶ 타겟 보드에서 NFS 마운트 후 진행

cp –a /mnt/nfs/fbranrect /root

./fbranrect

⑥ fbmranrect

arm-none-linux-gnueabi-gcc –o fbmranrect fbmranrect.c

cp fbmranrect /nfsroot

▶ 타겟 보드에서 NFS 마운트 후 진행

cp –a /mnt/nfs/fbmranrect /root

./fbmranrect

※ 본 자료는 공기석/전광일교수님께서 작성하신 내용을 새로운 Achro-5250 CD 버전에 맞게 수정한 내용입니다.