HYUNDAI AUTRON

지능형 자동차 개발환경 설정

차량융합신기술센터 운전편의제어팀
이준묵 책임연구원, 양지혁 책임연구원
2015. 07. 23

목차

HYUNDAI AUTRON

목차

HYUNDAI AUTRON

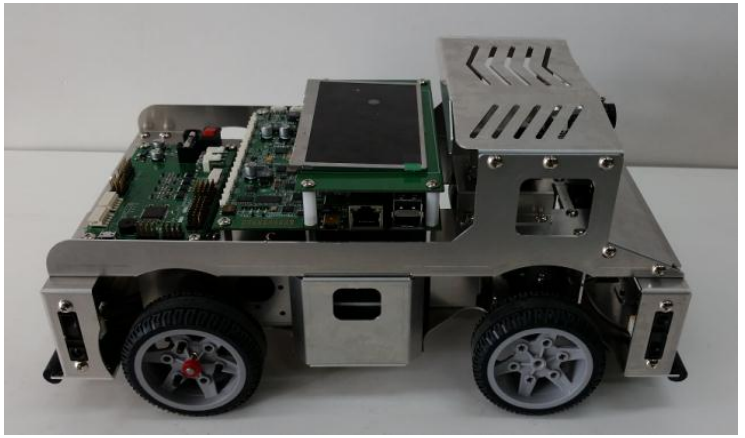
지능형 자동차 개발환경 소개

Host computer

Ethernet

Serial

Target board



개발환경

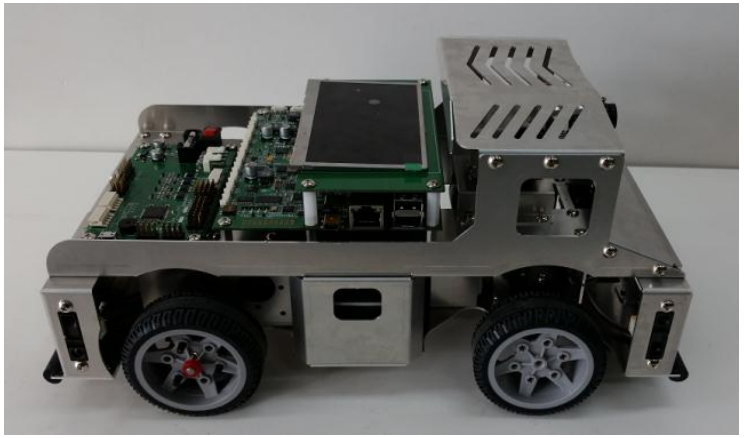
- OS : PC-Linux, target board-embedded linux
- 터미널(USB cable - serial 통신용)
- Network File System(Ethernet cable)

프로세서	NVIDIA Tegra3 (ARM Cortex-A9 Quadcore 900MHz)
메모리	DDR3 2GB
	NOR Flash 64MB
	eMMC 8GB
카메라	640x480
USB	USB2.0 port 2EA
AD	AD port 6EA (PSD sensor용)
LCD	4.3" (해상도 480x272)
디버깅	Ethernet (10/100) 및 mini USB
보드간 통신	UART

HYUNDAI AUTRON

지능형 자동차 개발환경 소개

■ 지능형 자동차 본체



■ 케이블 부품류

보드 전원 케이블

이더넷 케이블

USB 젠더 케이블

USB 케이블

배터리 충전기

리눅스 관련 용어정리 (1/3)

■ 부트로더

- 부트로더란 임베디드 시스템의 전원을 켰을 때 가장 먼저 실행되는 프로그램이다.

이곳에서 RAM을 사용할 수 있도록 초기화하고, 파일 시스템을 준비한다.

또한, 자신을 RAM에 복사한 후 RAM에서 실행하며, 네트 워크를 초기화하고, 커널을 RAM에 올린 후 실행 권한을 커널에게 주는 것이다.

- Hardware 초기화
  - Memory setting
  - CPU Clock setting
  - Serial setting
  - MAC address 획득 및 Ethernet port setting

- Flash Rom에서 RAM으로 메모리 복사
  - Kernel image 복사
  - Ramdisk image 복사
  - Bootloader image(자기 자신) 복사
  - Kernel booting
  - Command mode 제공
  - 포팅의 편의를 위한 디버깅 함수 제공

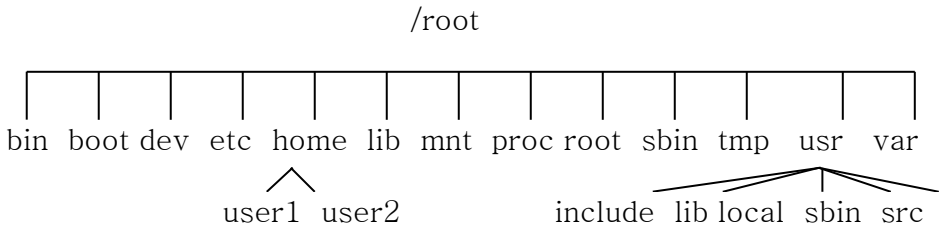
리눅스 관련 용어정리 (2/3)

■ 커널

- 커널이란 운영체제의 핵심을 이루는 요소로서 **컴퓨터내의 자원을 사용자 프로그램이 사용할 수 있도록 관리**하는 프로그램이다. 이런 커널의 역할은 사용자가 작동시키는 응용프로그램과 하드웨어간의 조정자 역할을 맡는다. 동시에 수행되는 여러 응용프로그램들을 위해 메모리 관리를 해주며 컴퓨터 자원을 배분하는 역할을 해준다.

■ 파일시스템

- 파일 시스템이란 운영체제가 파티션이나 디스크에 파일들이 연속 되게 하기 위해 사용하는 방법과 자료구조이다.

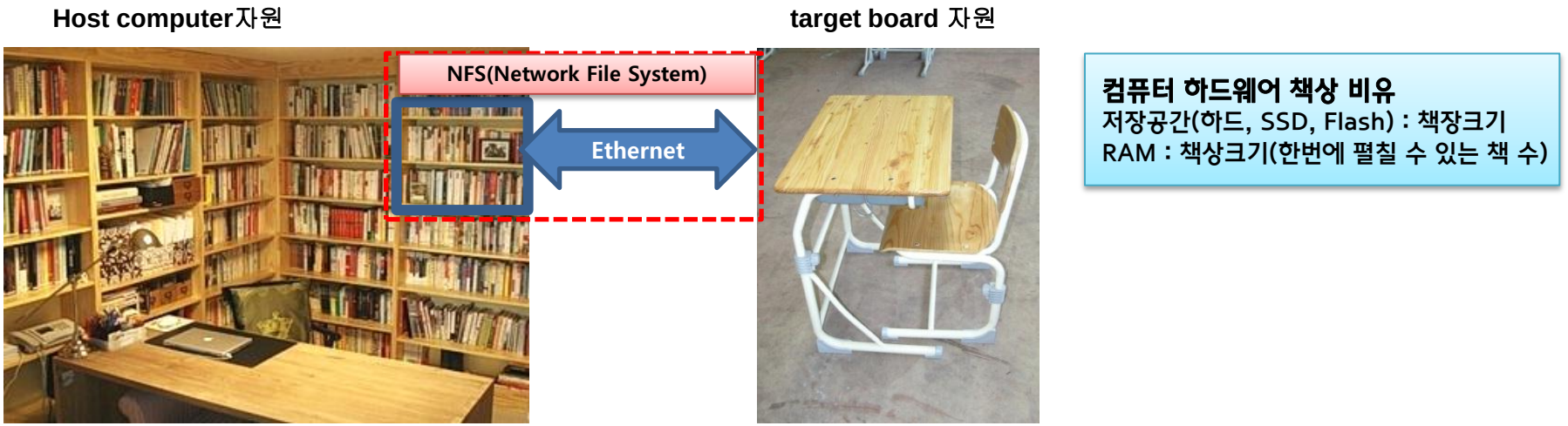


디렉토리	설 명
/	최상위 디렉터리인 루트 디렉터리
/bin	중요하고 꼭 필요한 기본 명령어가 위치한 디렉터리
/boot	커널 시스템 부팅에 관련이 있는 디렉터리
/dev	시스템 디바이스 드라이버 파일을 저장하는 디렉터리
/etc	시스템의 각종 환경 설정 파일을 저장하는 디렉터리
/home	여러 사용자의 홈 디렉터리
/lib	프로그램(C, C++)등에 필요한 각종 라이브러리 저장
/mnt	Floppy, CDROM 등을 마운트하는 디렉터리
/proc	실행중인 프로세스나 현재 시스템의 정보를 파일 형태로 보여주는 가상 디렉터리
/root	root의 홈 디렉터리
/sbin	시스템 관리자용 명령어를 저장하는 디렉터리
/tmp	임시 파일을 저장하는 디렉터리
/usr	사용자 응용프로그램이 설치되는 디렉터리
/usr/include	C프로그램에 필요한 헤더 파일 저장 디렉터리
/usr/lib	사용자 추가 라이브러리
/usr/local	사용자 추가 응용 프로그램 설치 디렉터리
/usr/sbin	/bin에 제외된 사용자 추가 명령어
/usr/src	프로그램 소스 저장 디렉터리
/var	시스템 운용에 필요한 생성 삭제 파일 저장하는 디렉터리

리눅스 관련 용어정리 (3/3)

■ NFS(Network File System)

- Host computer의 저장공간 빌려쓰는 것
- Target 보드의 file system을 host computer에 설치하고 target 보드 부팅시에 이더넷으로 접속하는 방법
- Target 보드가 꺼져있거나 연결되어 있지 않더라도 target 보드의 파일을 편집가능
- Target board 에 NFS 클라이언트 설치, host computer에는 NFS 서버 설치.



컴퓨터 하드웨어 책상 비유
저장공간(하드, SSD, Flash) : 책장크기
RAM : 책상크기(한번에 펼칠 수 있는 책 수)

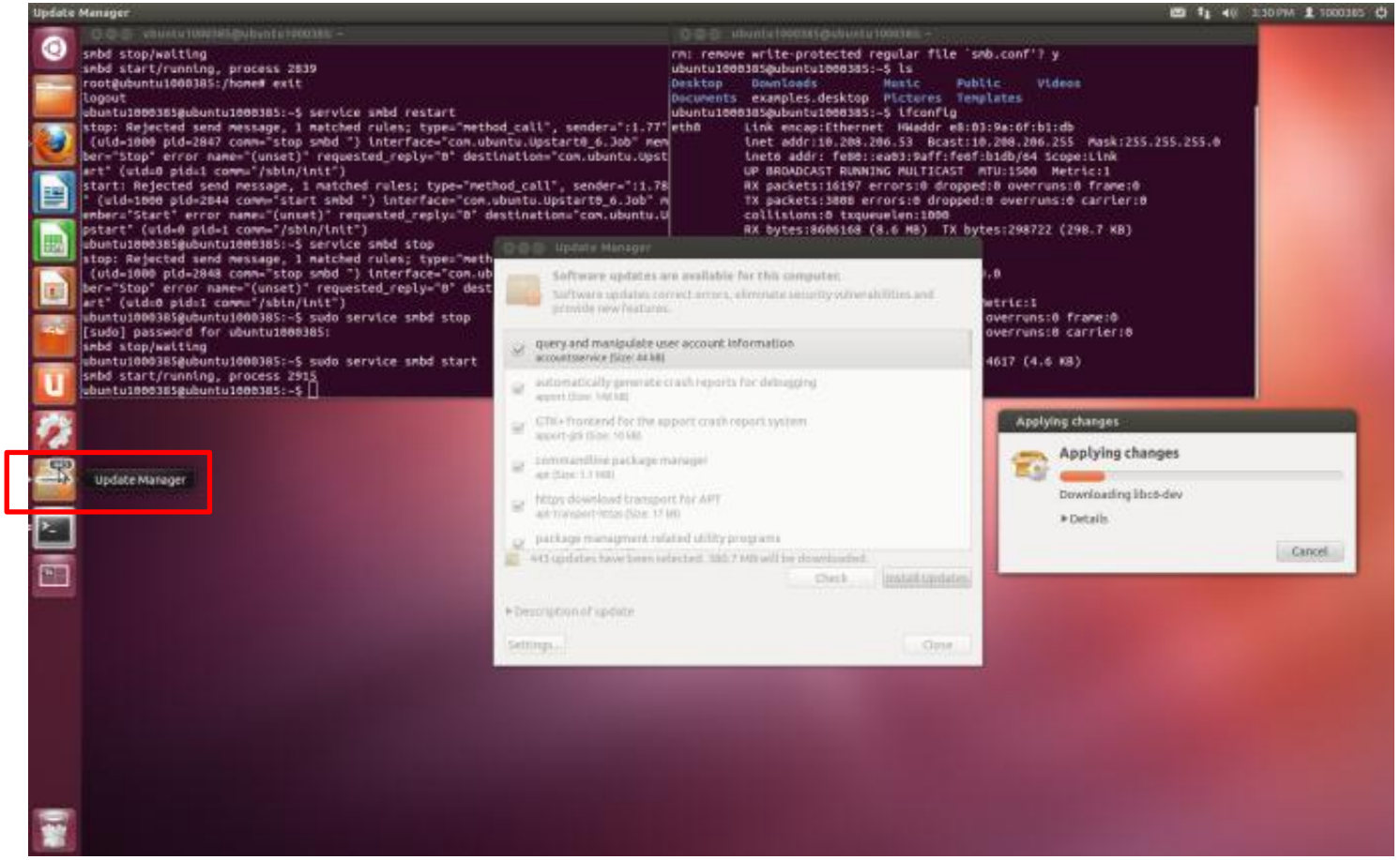
목차

지능형 자동차 개발환경 설정 방법

(1) Install Ubuntu 12.04 (32bit 또는 64bit)

- Web Site : www.ubuntu.com

(2) Update Ubuntu to the latest version using the Update Manager



지능형 자동차 개발환경 설정 방법

(3) apt-get install 이용 개발에 필요한 패키지 Host PC에 설치

■ **패키지 인덱스 정보 업데이트**

```
Host$ sudo apt-get update
```

■ **프로그램 개발에 필요한 헤더파일과 라이브러리 설치**

```
Host$ sudo apt-get install build-essential
```

■ **Minicom 설치**

```
Host$ sudo apt-get install minicom
```

■ **DHCP 서버 설치**

```
Host$ sudo apt-get install isc-dhcp-server
```

■ **NFS 서버 설치**

```
Host$ sudo apt-get install nfs-kernel-server portmap
```

(4) Install “ia32-libs” : 64bit linux에서 32bit 실행파일 돌아가게 하기 위함 (32bit linux 설치 했을시에는 생략)

■ **On a 64-bit host system, the "ia32-libs" Ubuntu package is required for compatibility.**

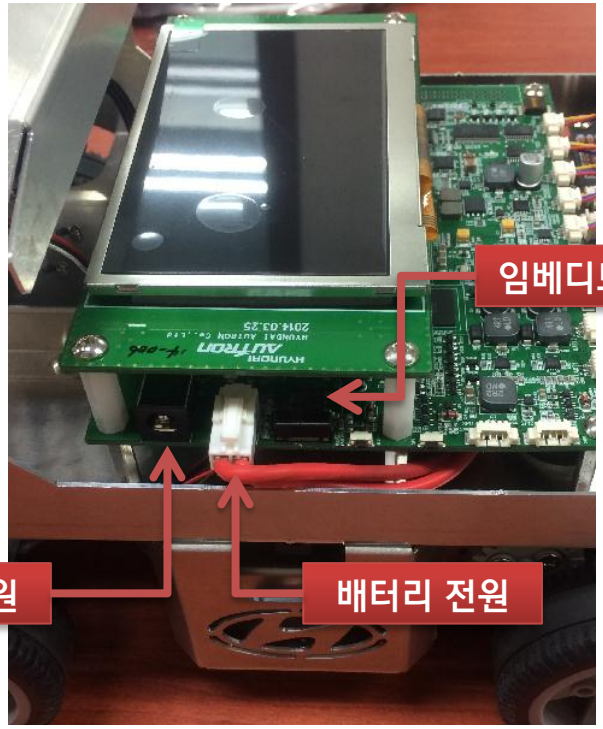
■ **In the host terminal command shell, type and run**

```
Host$ sudo apt-get install ia32-libs
```

지능형 자동차 개발환경 설정 방법

(5) 지능형 자동차 SW보드 전원 및 케이블 연결

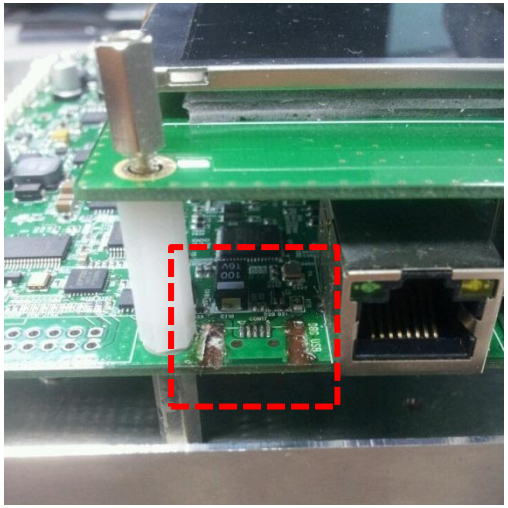
- 호스트 컴퓨터와 SW보드 USB 케이블 연결 (USB 젠더 이용)
- 호스트 컴퓨터와 SW보드 이더넷 케이블 연결
- SW보드에 12V 전원연결(어댑터 전원 또는 배터리 전원 중에 택1)후 임베디드보드 스위치 ON



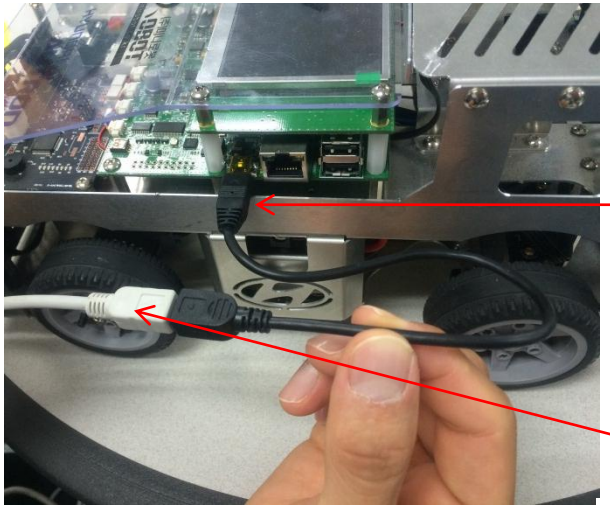
지능형 자동차 개발환경 설정 방법

■ USB 단자의 파손으로 인한 차량 고장 예방을 위한 대책

- 차량의 USB 단자와 USB 젠더를 연결 (이 젠더는 차량으로부터 뽑지 않고 항시 연결시켜서 사용한다.)
- USB 젠더를 통해 USB 케이블과 연결 (USB 케이블은 젠더에 꽂거나 뽑아 사용한다.)



〈 파손 사례 〉



〈 개선안 〉

① USB 젠더 연결
(뽑지 않는다.)

② USB 케이블 연결
(필요시 꽂았다가 빼서 사용한다.)

■ USB 파손 시 수리 기간 약 1~2주일가량 소요됨.

■ 파손시 자체 수리 금지 (납땜 등의 자체 수리 시 페널티 적용)

지능형 자동차 개발환경 설정 방법

(6) Setup serial connection for console (by Minicom)

■ **In the host terminal command shell, type and run**

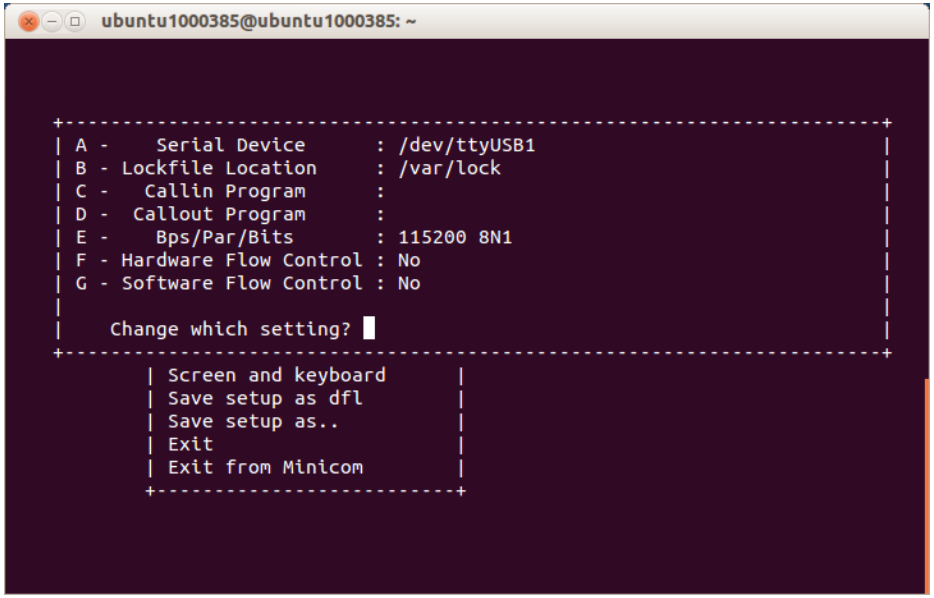
```
Host$ sudo minicom -s
```

- Go to Configuration Serial Port Setup
- Change Serial Device from '/dev/ttyS0' to '/dev/ttyUSB1'
- Check the 'Hardware Flow Control : No'
- Save setup as dfl & Exit

■ **In the host terminal command shell, type and run**

```
Host$ sudo minicom
```

→ 세팅후 바로 실행되면 생략. 중복으로 터미널 접근시 "Device /dev/ttyUSB1 is locked." 메시지 출력됨



지능형 자동차 개발환경 설정 방법

(7) SDK(software development kit) 설치

■ **Target 보드 Linux 설치에 필요한 툴 제공**

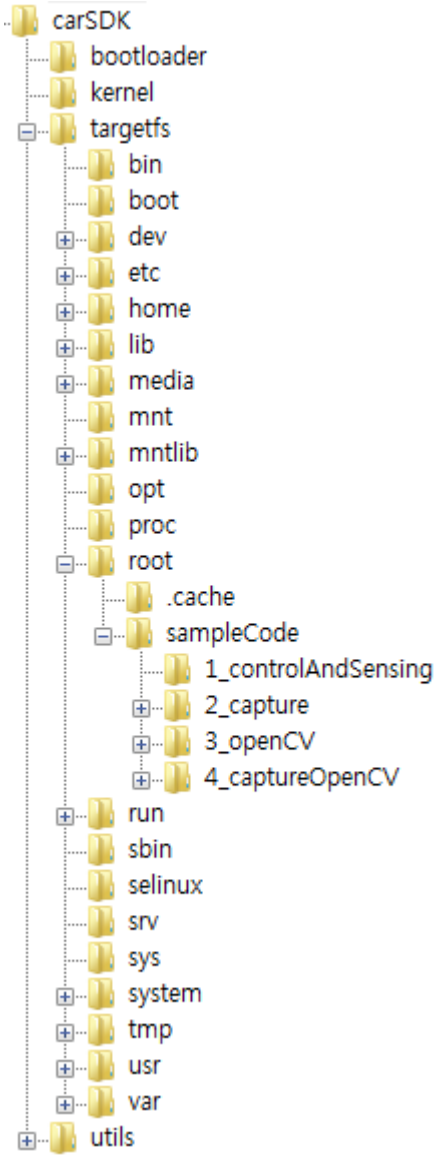
- bootloader : 리눅스 부트로더(타겟에 설치되는 파일)
- kernel : 리눅스 커널(타겟에 설치되는 파일)
- targetfs : 리눅스 파일시스템(타겟에 설치 또는 호스트 컴퓨터로 NFS이용 접근)
- utils : 리눅스 설치관련 파일

■ **SampleCode 제공 : targetfs/root/sampleCode**

- 1_controlAndSensing → 지능형 자동차 기본 제어 및 센서 테스트
- 2_capture → 카메라 입력 및 LCD 출력 테스트
- 3_openCV → openCV 예제 코드
- 4_captureOpenCV → 카메라 입력 openCV 적용 예제

■ **SDK_Autron.tgz 파일 Linux 컴퓨터에서 압축푸는 명령어**

```
Host$ tar xfvz SDK_Autron.tgz
```



(마지막에 tar: Exiting with failure status due to previous errors 라고 나오더라도 무시해도 됨)

지능형 자동차 개발환경 설정 방법

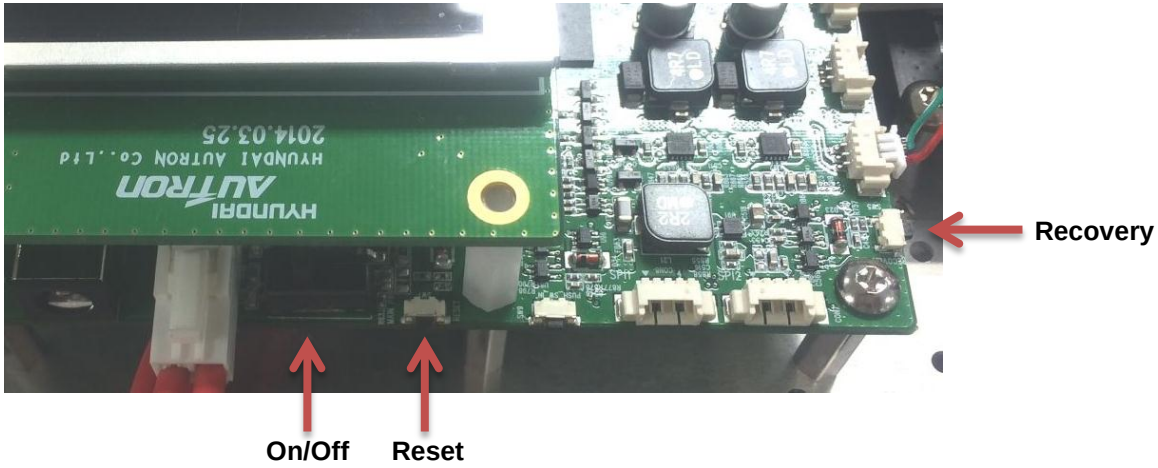
(8) Flashing Bootloader and Kernel ：NVIDIA 보드에 부트로더와 커널 올리는 과정

■ **In the host terminal command shell, type**

```
Host$ cd {설치경로}/carSDK
Host$ bash find_and_replace_in_files.sh
Host$ cd utils/scripts/burnflash
```

■ **다음과 같은 명령어 실행 전 SW 보드의 'Recovery' 버튼을 먼저 누르고 떼지 않은 상태에서 'Reset' 버튼을 눌렀다 뗀 다 그리고 'Recovery' 버튼에서 손 떼기**

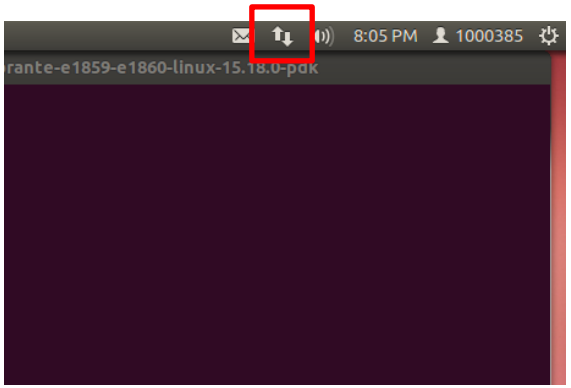
```
Host$ ./burnflash.sh -Z lzf -S 31 -n eth0
```

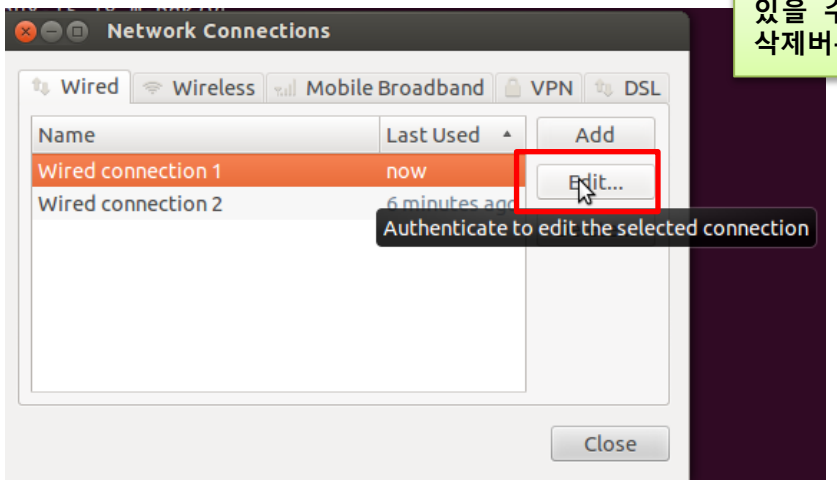


지능형 자동차 개발환경 설정 방법

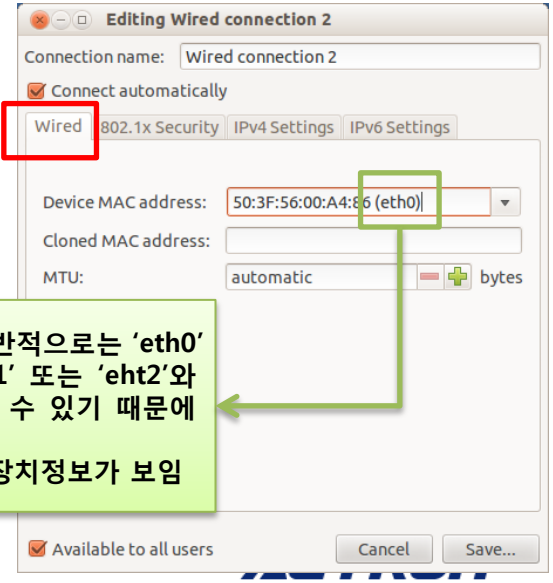
(9) Configure the network interface : NFS로 사용할 이더넷 IP 설정

- Go to System -> Preferences -> Network Connections
- Select Wired connection --> Edit --> IPv4 settings
- Select Method : Manual
- Address : 10.0.0.1
- Netmask : 255.255.255.0
- Gateway : 0.0.0.0





저장 버튼 활성화 안될경우 주소에 빈칸이 있을 수 있으므로, 주소에 빈칸을 클릭후에 삭제버튼 누르고 다시 시도해본다.



연결이 여러 개 보일 경우 타겟보드와 연결된 장치 이름 확인. 일반적으로는 'eth0' 이지만 USB2LAN과 같은 장치로 타겟보드와 연결했다면 'eth1' 또는 'eht2'와 같이 다른 이름으로 보이는 장치가 타겟보드와 연결된 장치 일 수 있기 때문에 확인이 필요하다.
터미널에서 ifconfig 라는 명령어를 치면 PC에서 인식된 이더넷 장치정보가 보임

지능형 자동차 개발환경 설정 방법

(10) Configure the DHCP and NFS server

■ **Edit the '/etc/default/isc-dhcp-server' file**

```
Host$ sudo gedit /etc/default/isc-dhcp-server
```

→ INTERFACES="eth0"

Target 보드와 연결되는 Host PC의 이더넷 장치 이름. 만일 USB2LAN과 같은 장치로 Target 보드를 연결한다면 장치 이름이 "eth0" 가 아닌 "eth1"과 같이 다른 이름이 될 수 있으므로 확인 필요. (9)번 과정에서도 장치가 여러 개 보일 경우에 실제 Target과 연결된 이더넷 장치를 고정IP로 설정한다.
여기서 Host PC의 이더넷 장치 이름이 "eth0"가 아닐 경우라도 (8)번의 `Host$ ./burnflash.sh -Z lzf -S 31 -n eth0` 는 Target 보드의 환경을 설정하는 것이므로 (8)번의 "eth0" 관련이 없다. 따라서 (8)번의 "eth0"는 바꾸지 않는다.

지능형 자동차 개발환경 설정 방법

- 하기 총 4줄을 #으로 커맨트아웃할 것.
option domain-name ~
option domain-name-servers ~

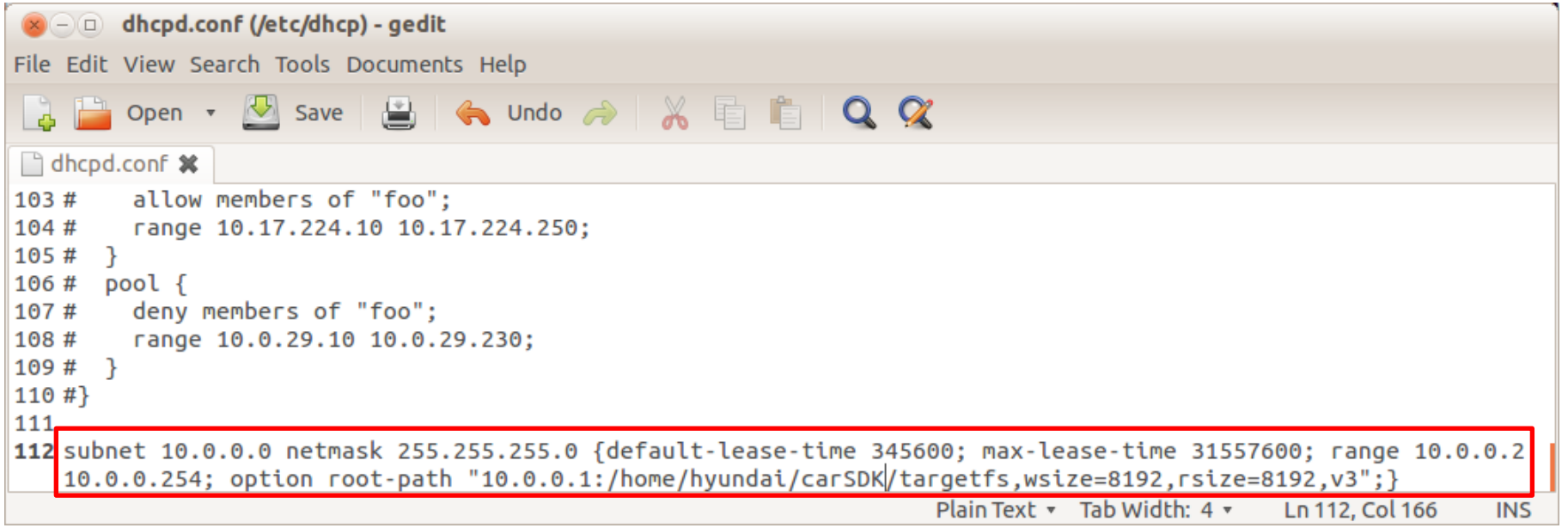
default-lease-time ~
max-lease-time ~

(11) Configure the DHCP and NFS server

■ Edit '/etc/dhcp/dhcpd.conf' file (마지막에 다음의 한줄을 추가하고 나머지는 모두 주석처리(#))

Host$ sudo gedit /etc/dhcp/dhcpd.conf

→ subnet 10.0.0.0 netmask 255.255.255.0 {default-lease-time 345600; max-lease-time 31557600; range 10.0.0.2 10.0.0.254; option root-path "10.0.0.1:{설치경로}/carSDK/targetfs,wsize=8192,rsize=8192,v3";}



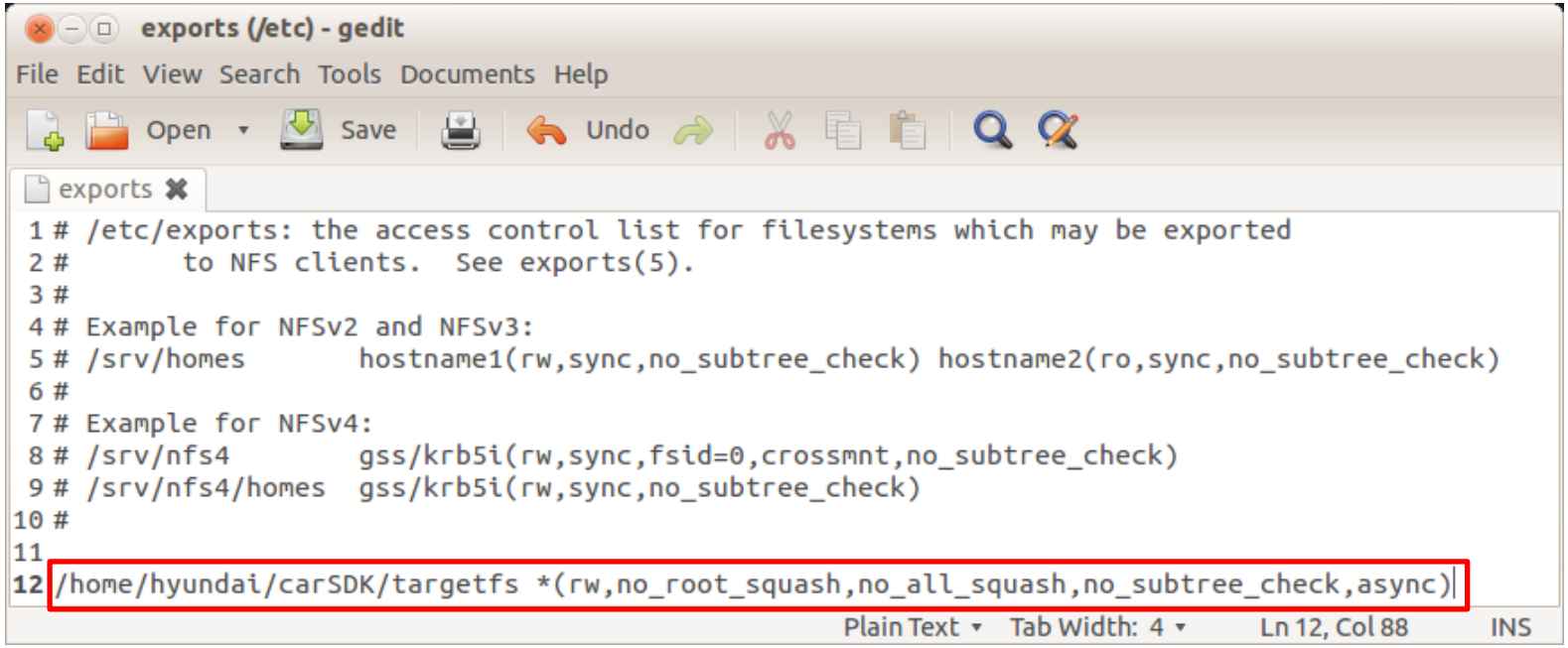
지능형 자동차 개발환경 설정 방법

(12) Configure the DHCP and NFS server

■ Edit '/etc/exports' file (다음과 같이 작성)

Host$ sudo gedit /etc/exports

→{설치경로}/carSDK/targetfs *(rw,no_root_squash,no_all_squash,no_subtree_check,async)



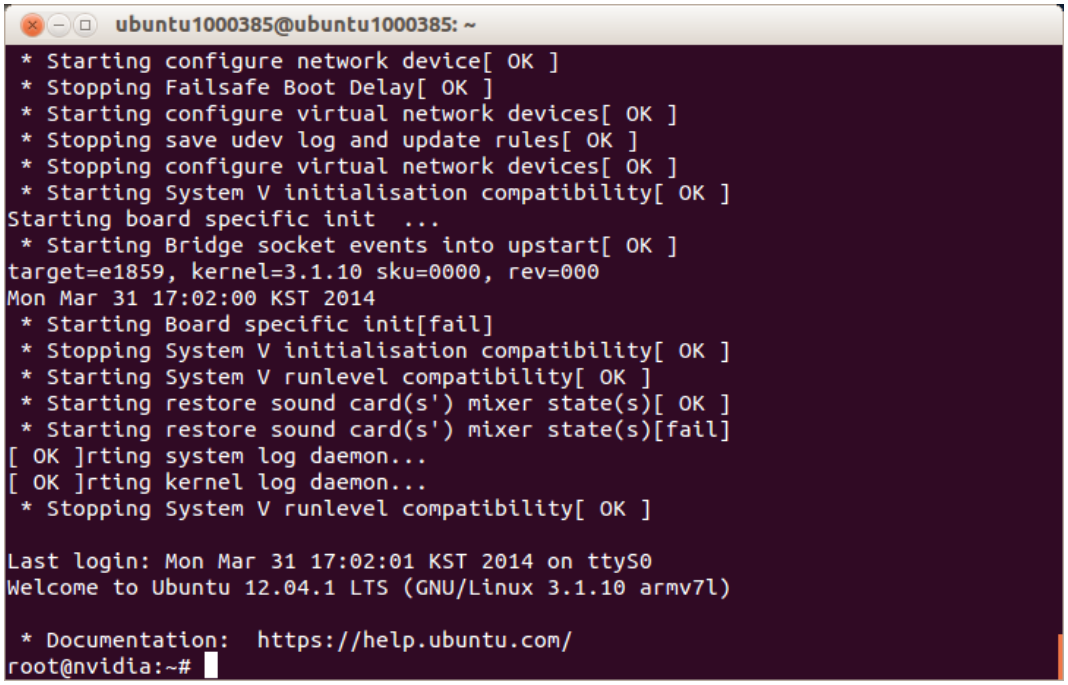
지능형 자동차 개발환경 설정 방법

(13) Run DHCP and NFS server

■ **In the host terminal command shell, type**

```
Host$ sudo /etc/init.d/nfs-kernel-server start
Host$ sudo exportfs -a
Host$ sudo /etc/init.d/isc-dhcp-server start
```

(14) Power On Board(Or Reset) → 첫 부팅시 minicom에 부팅 로그 계속 출력되면서 10분정도 소요



NFS를 이용한 개발 환경 세팅 완료 !!

목차

HYUNDAI AUTRON

NFS 개발환경 세팅 완료후 부팅방법

■ (9)번 과정 확인 : Network Interface 확인 및 편집(이미 설정되어 있다면 생략)

■ NVIDIA board Power ON

■ 터미널 1 → minicom 동작시키기 : target 보드와 통신용

```
HOST$ sudo minicom
```

■ 터미널 2 → Run DHCP and NFS server

```
HOST$ sudo /etc/init.d/nfs-kernel-server start
HOST$ sudo exportfs -a
HOST$ sudo /etc/init.d/isc-dhcp-server start
```

(한번에 잘 안되면 `HOST$ sudo /etc/init.d/isc-dhcp-server restart`)

■ NVIDIA 보드가 실행되는 폴더 : HostPC의 {설치경로}/carSDK/targetfs/root/

HYUNDAI AUTRON

NFS를 사용하지 않고 eMMC에 파일시스템 올리는 방법

■ (8)번 과정에서 burnflash 명령어

Host$ ./burnflash.sh -k quickboot_snor_linux_fs.cfg -S 31

```
ubuntu1000385@ubuntu1000385:~/vACE/vibrante-e1859-e1860-linux/utils/scripts/burn
flash$ ./burnflash.sh -k quickboot_snor_linux_fs.cfg -S 31
burnflash.sh for e1853 board
NOTE: All switches are on E1860/Jetson (base board)
If board is ON, power OFF it by moving switch (S4) to OFF position.

Put the board in force recovery
Press and hold the RECOVER button (S3) while switching POWER (S4) to ON
Press Enter to continue

Bus 002 Device 104: ID 0955:7230 NVidia Corp.
Flashing the bootloader...
Nvflash v1.11.104376 started
Creating Filesystem image for partition P1
```

Burnflash가 정상동작 하는 경우 나타나는 초기 메시지

targetfs 폴더가 P1 파티션 크기의 90%를 넘으면,
(1.07GB * 0.9 = 약 960MB)
하기 에러가 발생하면서 burnflash error 발생하므로 주의 !!

```
Creating Filesystem image for partition P1
Source directory size is more than the 90% of partition size
Error while flashing bootloader, exiting...
```

■ 결과

- Host 컴퓨터의 ‘carSDK/targetfs/’ 파일들이 Target 보드의 eMMC로 올라가며, 재부팅시 NFS를 사용하지 않고 보드의 파일시스템 사용
- 즉, Target 보드가 Host 컴퓨터와 LAN선 연결 없이 stand-alone으로 동작.
  따라서 개발이 완료되면 eMMC에 파일시스템을 올리고 최종대회를 치룬다.

Target 보드 부팅 완료후 sample code 컴파일 및 실행

■ Sample Code

- **자동차 센서 및 제어 테스트 프로그램**

  컴파일 명령어 : root@nvidia:~/sampleCode/1_controlAndSensing# gcc main.c -o main

  실행 명령어 : root@nvidia:~/sampleCode/1_controlAndSensing# ./main

- **Video capture 프로그램**

  컴파일 명령어 : root@nvidia:~/sampleCode/2_capture# make

  실행 명령어 : root@nvidia:~/sampleCode/2_capture# ./nvmedia_capture -vt 10

  10초간 실행하라는 옵션

- **openCV 코드 테스트**

  컴파일 명령어 : root@nvidia:~/sampleCode/3_openCV/1_cannyTest# gcc `pkg-config opencv --cflags` test.c -o test `pkg-config opencv --libs`

  실행 명령어 : root@nvidia:~/sampleCode/3_openCV/1_cannyTest# ./test t.png

- **Video capture에 openCV 적용예제**

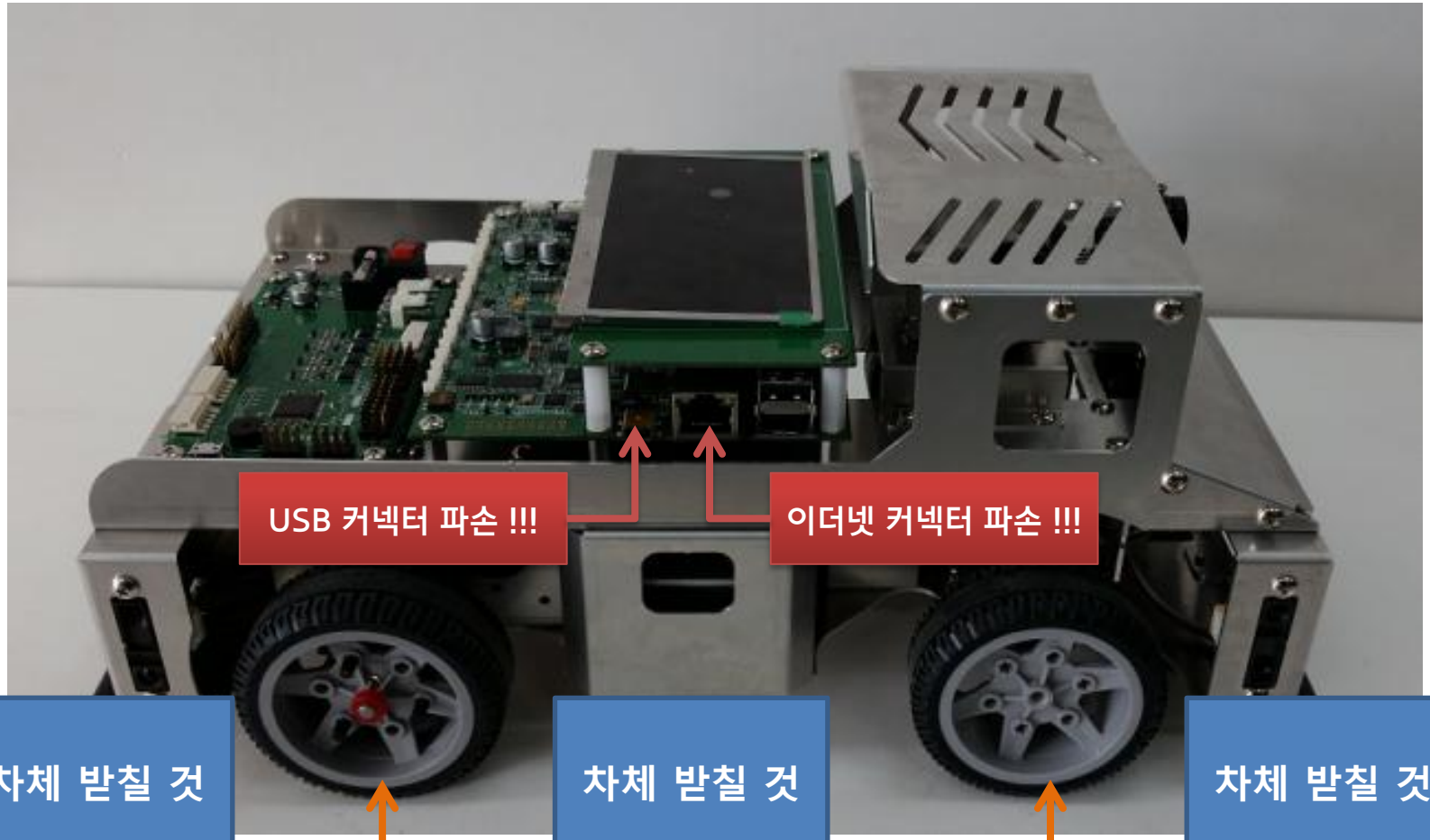
  컴파일 명령어 : root@nvidia:~/sampleCode/4_captureOpenCV# make

  실행 명령어 : root@nvidia:~/sampleCode/4_captureOpenCV# ./captureOpenCV -vt 10

Sample Code 실행시 주의 사항

■ 모터 센서값 미설정시 급발진하여 USB 또는 이더넷 케이블 커넥터가 파손될 수 있음

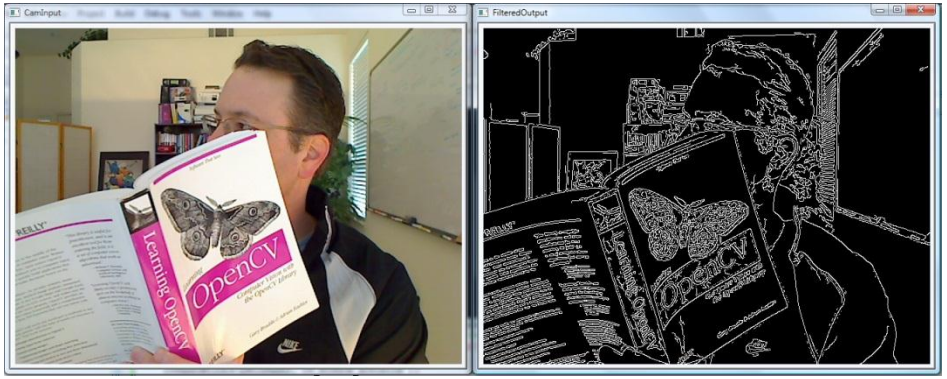
- 테스트 초기에는 차량 바퀴를 허공에서 공회전시킨 상태에서 검증할 것.



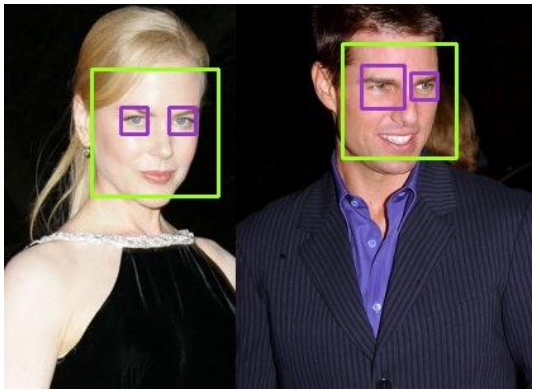
openCV

- 인텔에서 만든 오픈소스(http://opensource.org 참고)로 만들어지고 있는 컴퓨터 비전 라이브러리
- 실시간 연산이 가능하도록 연산의 효율성을 최대한 고려하여 설계
- 최적화된 C언어로 작성, 멀티코어 프로세서의 장점 활용 가능
- 라이브러리는 500개가 넘는 함수들로 구성
- 상업적인 프로그램개발에 사용해도 무방

Edge detection



Face Recognition



HYUNDAI AUTRON

Test Code 1 : 그림파일 확장자 변환

키보드 좌측상단의 "~"와 같은 키에 있는 기호임 !!!

```
// gcc `pkg-config opencv --cflags` image-conversion.c  -o image-conversion `pkg-config
opencv --libs`
// g++ `pkg-config opencv --cflags` image-conversion.cpp  -o image-conversion `pkg-
config opencv --libs`
// http://linuxconfig.org/introduction-to-computer-vision-with-opencv-on-linux
// compile : http://www.ozbotz.org/opencv-installation/

// ./image-conversion t.png image.jpg

#include <highgui.h>

int main( int argc, char** argv )
{
    IplImage* img = cvLoadImage(argv[1], 1);

    cvSaveImage(argv[2] , img, 0);
    cvReleaseImage(&img);
    return 0;
}
```

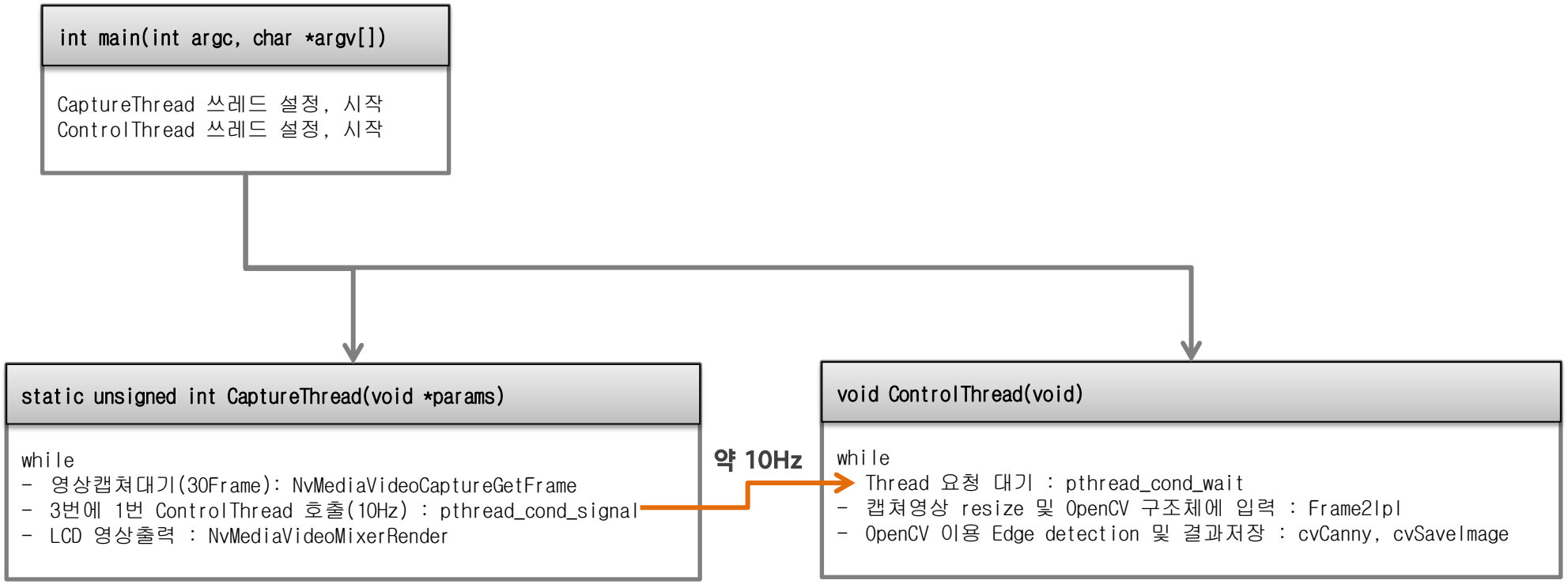
Test Code 2 : Canny edge test

키보드 좌측상단의 "~"와 같은 키에 있는 기호임 !!!

```
// gcc `pkg-config opencv --cflags` test.c  -o test `pkg-config opencv --libs`
// source : http://carstart.tistory.com/188
// compile : http://www.ozbotz.org/opencv-installation/

// ./test t.png

#include <highgui.h>
#include <cv.h>

int main( int argc, char** argv )
{
    IplImage* img = cvLoadImage(argv[1], 1);
    IplImage* cannyImage = cvCreateImage(cvGetSize(img), IPL_DEPTH_8U, 1);

    if(img == NULL)
        return;

    cvCanny(img, cannyImage, 100, 100, 3);
    cvSaveImage("cannyResult.png" , cannyImage, 0);
    cvReleaseImage(&img);

    return 0;
}
```


HYUNDAI AUTRON

Video capture에 openCV 적용예제(4_captureOpenCV) 흐름도



별첨 1 Sample code compile 시 발생하는 사소한 경고해결 방법

■ Sample code compile 시 다음과 같은 경고가 발생할 수 있지만, 컴파일 결과는 정상

```
root@nvidia:~/sampleCode/4_captureOpenCV# make
make: Warning: File `Makefile' has modification time 1.9e+06 s in the future
gcc -Os -O2 -Wall -I. -I./utils `pkg-config opencv --cflags` -I./include  -c -o captureOpenCV.o captureOpenCV.c
gcc -Os -O2 -Wall -I. -I./utils `pkg-config opencv --cflags` -I./include  -c -o nvthread.o nvthread.c
gcc  -o captureOpenCV captureOpenCV.o nvthread.o  -L ./utils -lnvmedia -lnvtestutil_board -lnvtestutil_capture_input -lnvtestutil_i2c -lpthread `pkg-config opencv --libs`
make: warning:  Clock skew detected.  Your build may be incomplete.
root@nvidia:~/sampleCode/4_captureOpenCV#
```

■ 원인 : 타겟보드 시스템의 시간보다 sample code의 시간이 미래로 설정되어 있음

```
root@nvidia:~# date
Wed May  7 14:43:13 KST 2014
root@nvidia:~# ls -l
total 12
drwxr-xr-x 3 root root 4096 May  7 14:31 apt-repos
drwxr-xr-x 7 root root 4096 May 29  2014 sampleCode
drwxr-xr-x 5 root root 4096 May  7 14:31 vibrante-e1859-e1860-linux
root@nvidia:~#
```

■ 해결방법 : sample Code 시간 변경

```
root@nvidia:~# find ./sampleCode -exec touch -t 201401010000 {} ₩;
```

별첨 2 보드 부팅時 원하는 프로그램 자동으로 실행되게 하는 방법

■ '.bashrc' 파일 편집

- NFS를 사용할 경우 Host 입장에서의 '.bashrc' 파일 위치 : {설치경로}/carSDK/targetfs/root
- Target보드 입장에서의 '.bashrc' 파일 위치 : root@nvidia:~#

```
root@nvidia:~# ls .bashrc
.bashrc
root@nvidia:~#
```

- '.bashrc' 파일의 마지막 줄에 원하는 실행 명령어 삽입

```
# some more ls aliases
alias ll='ls -alF'
alias la='ls -A'
alias l='ls -CF'

# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
#if [ -f /etc/bash_completion ] && ! shopt -oq posix; then
#    . /etc/bash_completion
#fi

./sampleCode/4_captureOpenCV/captureOpenCV -vt 20
                                                          100,0-1
```

별첨 3 향후 계획

■ 다음 교육 설명회

: 8월말 경에 예정 (일자는 미정이며, 추후 재공지 예정)

■ 다음 교육 설명회까지의 1차 Mission

: 최종 심사 결과에 Advantage/Disadvantage 포인트 적용

- 전방 카메라를 이용하여 차량의 주행 시작 스위치를 만드세요.

→ 별첨의 무인자동차 명령어 코드표를 참고하여 전/후방 깜박이를 켜고 Beep음을 울린 다음,
전방 카메라 영상의 변화를 인식한 후 차량을 앞으로 전진시키세요.
(예. 카메라로 사람 손가락에 의한 신호를 인식하여 전진 시작)

※ 단, 실행파일을 eMMC에 올리고 부팅후 프로그램이 자동으로 시작되도록 (23 페이지 eMMC 설명 장 참조)
stand-alone으로 차량을 동작시키세요.

※ 앞으로 전진시키는 거리, 속도 등은 임의로 설정하여도 됩니다.

HYUNDAI AUTRON