

마이크로프로세서 응용 설계
(Microprocessor Application Design)
Spring 2019

실습 1
모듈 및 간단한 디바이스 드라이버 작성

전북대학교 컴퓨터공학부

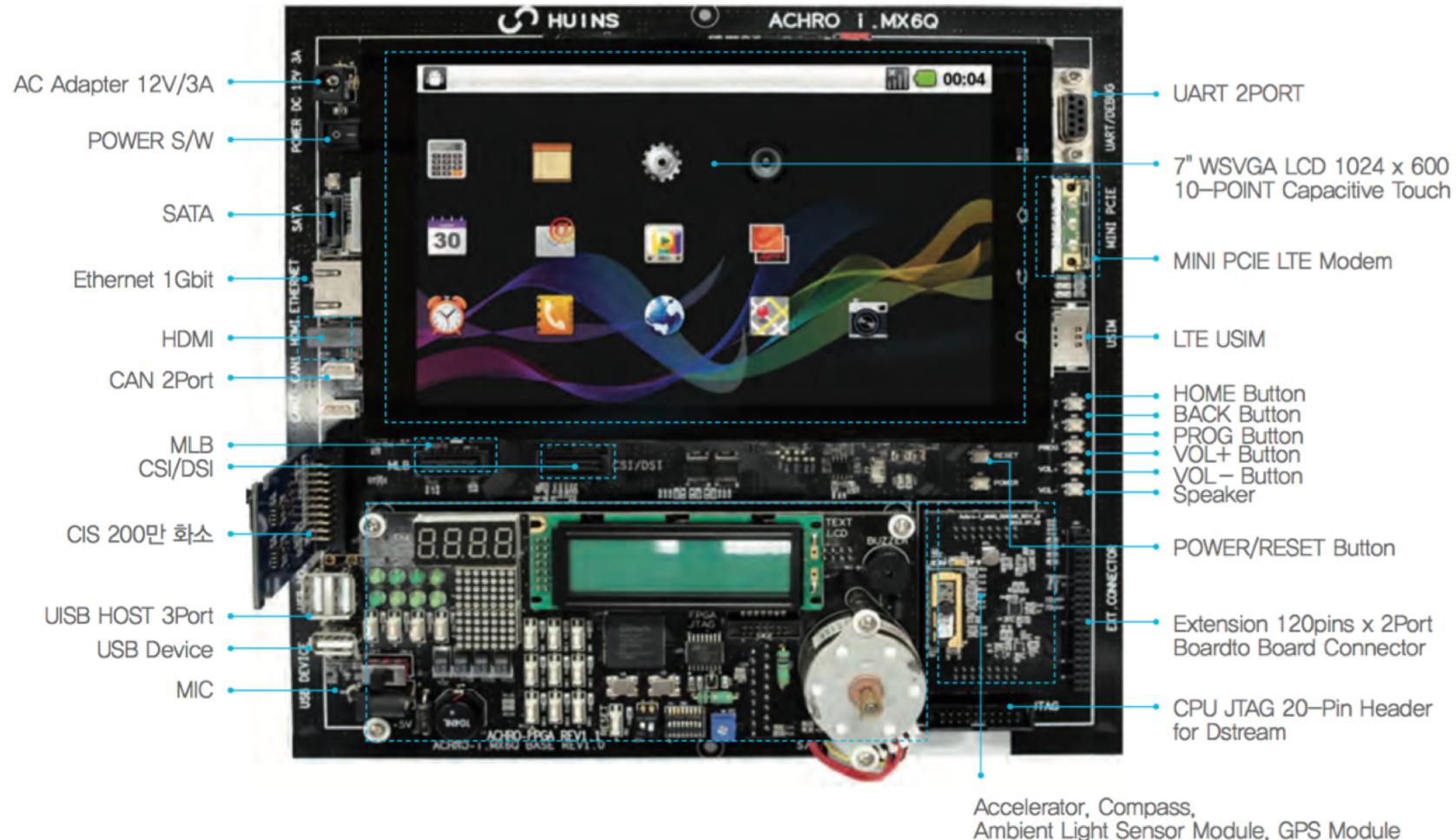
Achro I.MX6Q Hardware Overview

- Freescale I.MX6Q
- Cortex-A9 Quad core
- USB 2.0, SATA
- 1.0/2.0/3.0 Interface
- eMMC 4.4 / T-FLASH
- Wifi / Bluetooth / GPS
- Ethernet 10/100M bps
- HDMI mirror screen
- 10 points touchscreen
- Flexcan

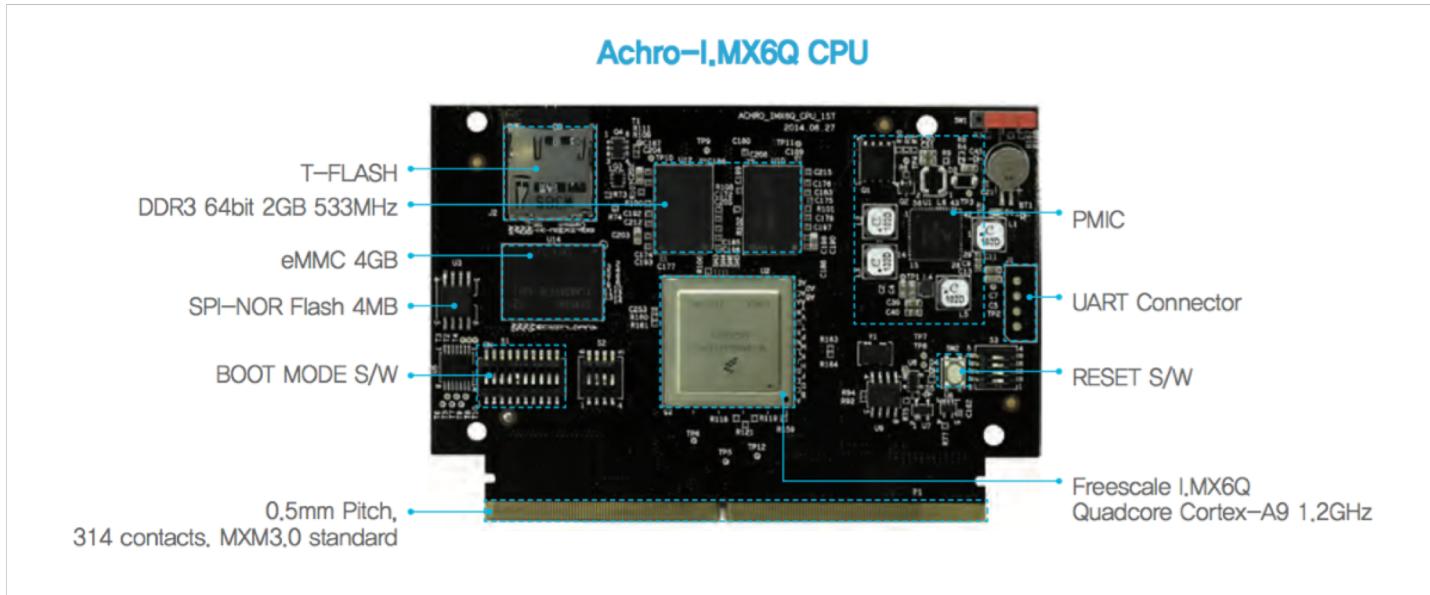


Achro I.MX6Q 베이스 보드

Achro-I.MX6Q



Achro I.MX6Q CPU, FPGA 보드



Achro I.MX6Q 베이스 보드 사양

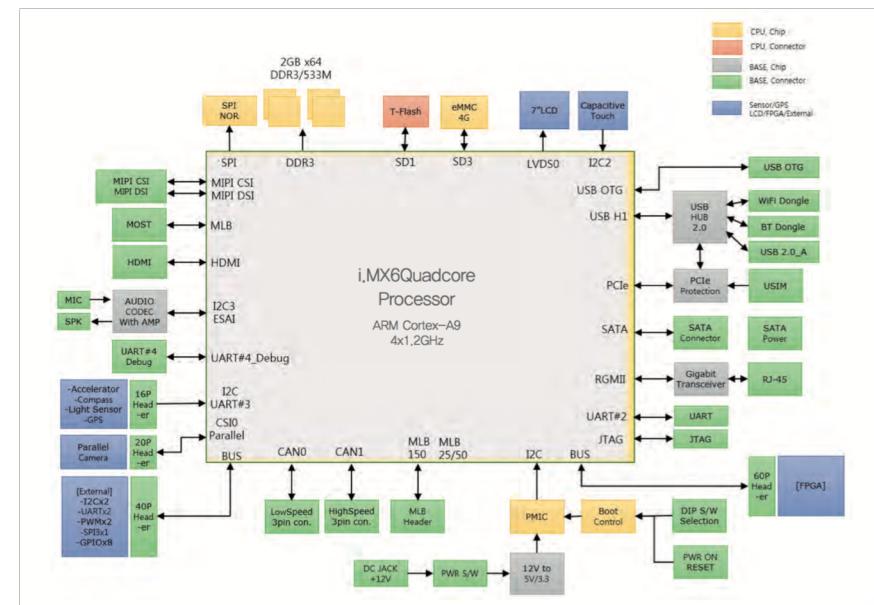
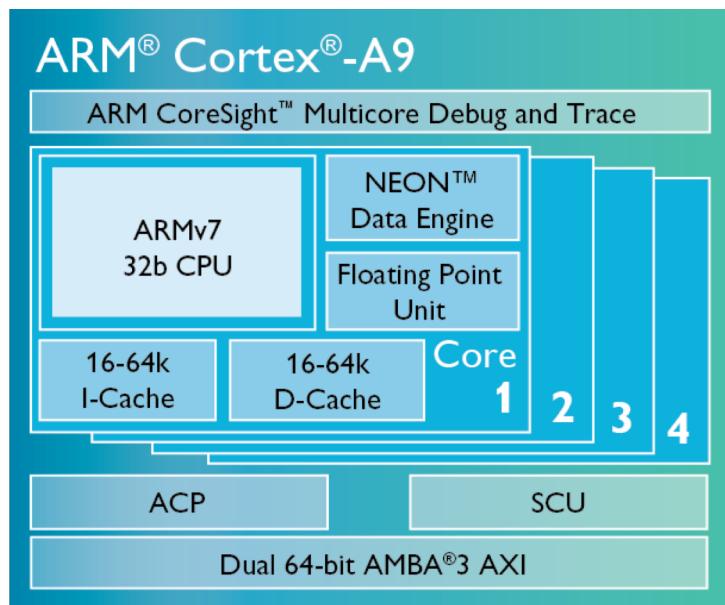
Achro-I.MX6Q Base Board

구 분	규 格
Board to Board Connector	0.5mm Pitch, 314 Contacts, MXM3.0 Standard
LCD	7" WSVGA LCD 1024 x 600 (TFT LCD)
Touch Screen	10–Point Capacitive Multi–Touch Screen Panel
USB	Microchip USB 2514, hi-Speed USB Hub Controller, 3 x USB HOST Connector, USB ITG Connector
Speaker / MIC	1W, Speaker/Microphone With Amplifier
HDMI	1080p, HDMI v1.4, CEC Interface
SATA	SATA2, SATA 1.5 Gb/s and SATA 3.0 Gb/s Speed With Power Port
Ethernet	Gigabit Ethernet (10/100/1000Mbps Support)
Mini PCI Express	PCI Express 2.0 Support with USB2.0 Host, USIM Connector
Sensor/GPS Module	Accelerator, Compass, Ambient Light Sensor, GPS Module
Parallel Camera	5MPixel Parallel Camera
MLB Port	MLB (Media Lab Bus) Support The Standardized on-PCB, Inter-Chip Communication Bus for MOST® Based Devices
CAN	2 x FlexCAN v2.0B
2 UART	9pin d-SUB(Debug), EXT UART
Keys	Power/Reset, Android Key (Home, Back, PROG, Volume)
Power	AC Adapter 12V/3A

Achro I.MX6Q CPU 보드 사양

Achro-I.MX6Q CPU

구 분	규 격
CPU	Freescale I.MX6Q Quad core Cortex-A9 1.0/1.2GHz
Memory	DDR3 64bit 2GB 533MHz
PMIC	Freescale MMPF0100(6 Buck, 6 LDO)
Storage	4GB eMMC, 4MB SPI-NOR Flash, T-FLASH
ETC	3.3V Power IN, UART Connector, RESET S/W 등



Achro I.MX6Q FPGA 보드 사양

규격

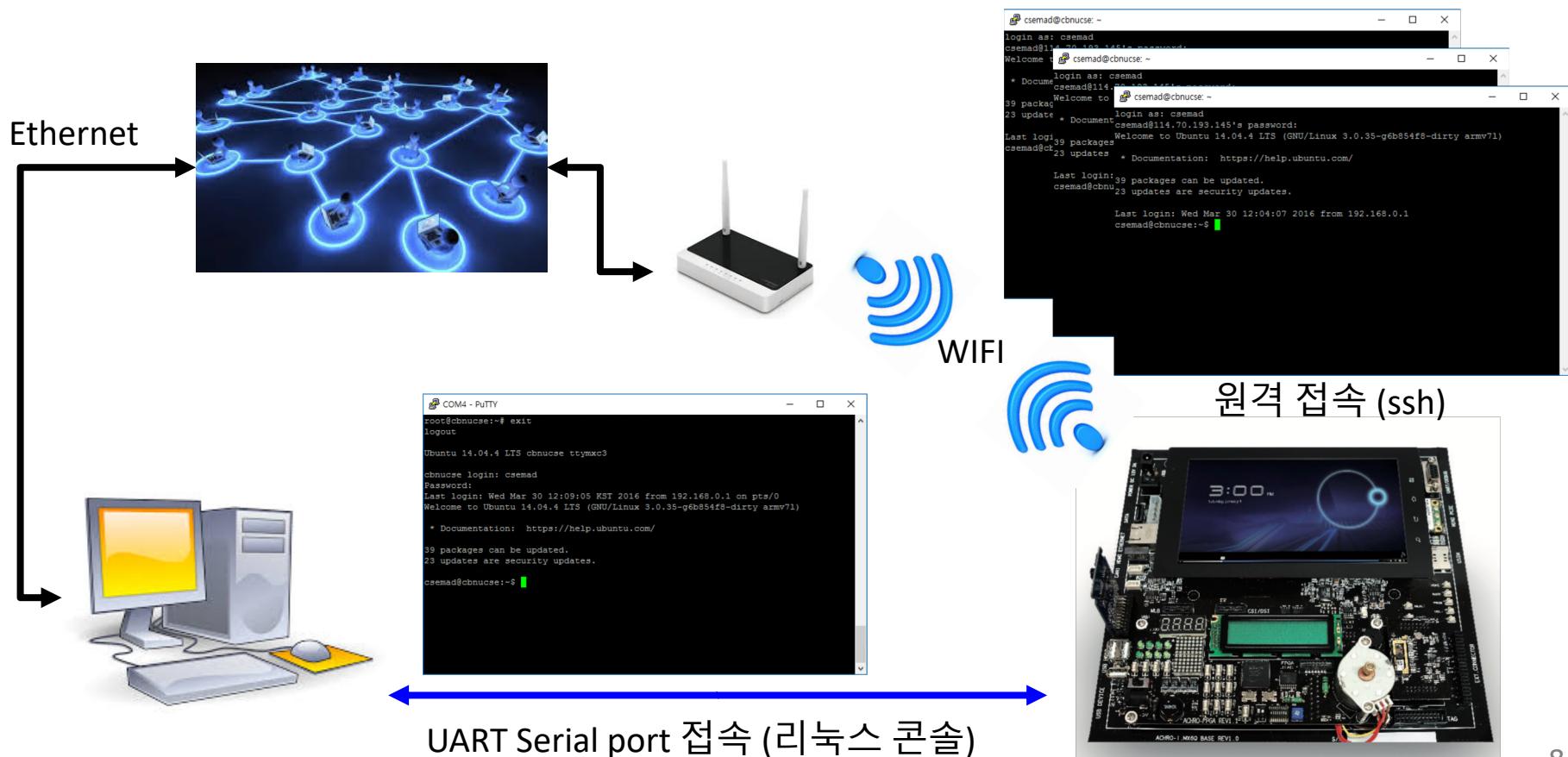
Achro 본체 확장 60핀에 연결하여 사용 I/O 확장하여 컨트롤
전원내장으로 FPGA 단독으로 사용 가능

FPGA : Xilinx Spartan6 FPGA XC6SLX16-2ftg256c –기본 160만 gate

TEXT-LCD (16X2 Character)	LED 8EA
DOT-MATRIX 7 x 5 : 2EA	FND 4 DIGIT
PIEZ0 BUZZER	DIP 8EA
PUSH Button S/W 4EA	STEPPING MOTOR
PUSH Button S/W 9EA	Oscillator 2EA (33MHz, 12MHz)
Reset Push Button	FPGA JTAG (Xilinx platform cable 별도)

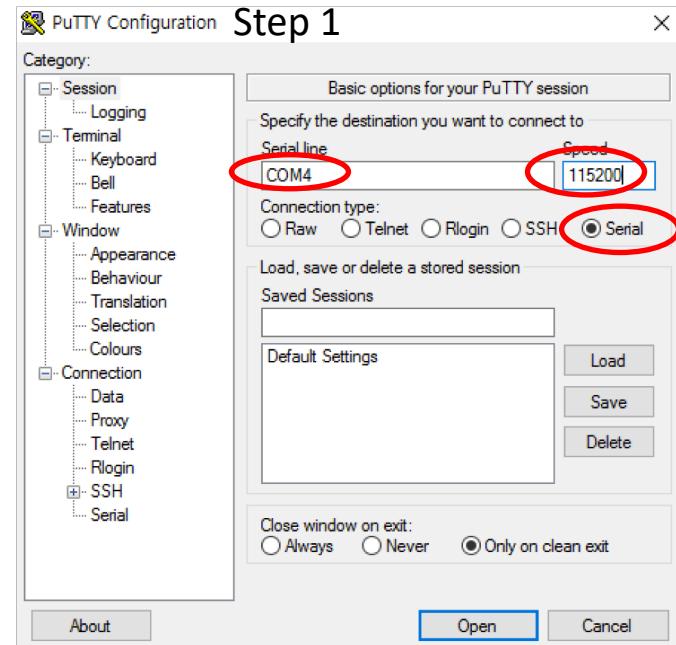
실습 환경

- 일반적으로 리눅스 머신에 원격 접속해서 프로그램을 개발하는 절차와 동일



호스트와 보드 연결

- Putty 사용
- 1. 보드 연결
 - 보드 전원을 끈 상태에서 UART 케이블로 PC와 보드 연결
- 2. 시리얼 포트를 통한 콘솔 접속
 - Putty 설정
 - Serial line: PC마다 사용하는 포트가 다를 수 있음
 - 내 PC → 장치관리자 → 포트 → USB Serial Port 번호 확인
 - 보드 전원 인가
 - 부팅되면서 콘솔에서 부팅 메세지 출력
 - 로그인 계정/암호: **csemad / 0000**
 - 콘솔에서 발생하는 timeout 메세지는 무시해도 됨



The screenshot shows the 'COM4 - PUTTY' terminal window. It displays the system boot logs from 'Ubuntu 14.04.4 LTS'. The logs show various services starting up, such as the NM status monitor, System V initialisation compatibility, and the OpenSSH server. The terminal prompt 'cnuce login:' is visible at the bottom, with a red circle highlighting it.

```
* Starting NM status monitor [ OK ]
* Starting System V initialisation compatibility [ OK ]
* Stopping Block the mounting event for NFS filesystems until statd is run [ OK ]
* Stopping Mount network filesystems [ OK ]
* Setting up X socket directories... [ OK ]
* Stopping System V initialisation compatibility [ OK ]
* Starting System V runlevel compatibility [ OK ]
* Starting save kernel messages [ OK ]
* Starting configure network device security [ OK ]
* Starting OpenSSH server [ OK ]
* Starting regular background program processing daemon [ OK ]
* Starting deferred execution scheduler [ OK ]
* Starting ACPI daemon [ OK ]
* Starting CPU interrupts balancing daemon [ OK ]
* Stopping save kernel messages [ OK ]
* Starting automatic crash report generation [ OK ]
* Starting configure virtual network devices [ OK ]
* Stopping Mount filesystems on boot [ OK ]
* Restoring resolver state... [ OK ]
* Stopping System V runlevel compatibility [ OK ]

Ubuntu 14.04.4 LTS cnuce ttymxcs3
cnuce login:
```

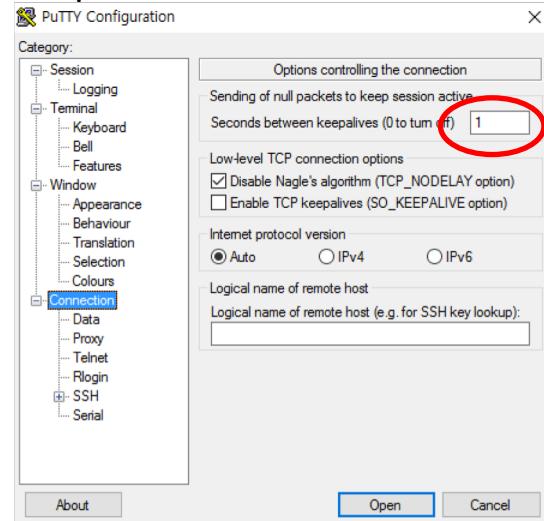
The screenshot shows the 'COM4 - PUTTY' terminal window again. This time, the screen is mostly blank except for the bottom few lines which show a series of 'timeout' messages from the 'accelerometer' service on 'devices/virtual/input*' devices. A red circle highlights the first message.

```
[ 42.735167] systemd-udevd[2079]: timeout 'accelerometer /devices/virtual/input/input3'
[ 42.743163] systemd-udevd[2116]: timeout 'accelerometer /devices/virtual/input/input4'
[ 42.751152] systemd-udevd[2149]: timeout 'accelerometer /devices/virtual/input/input5'
```

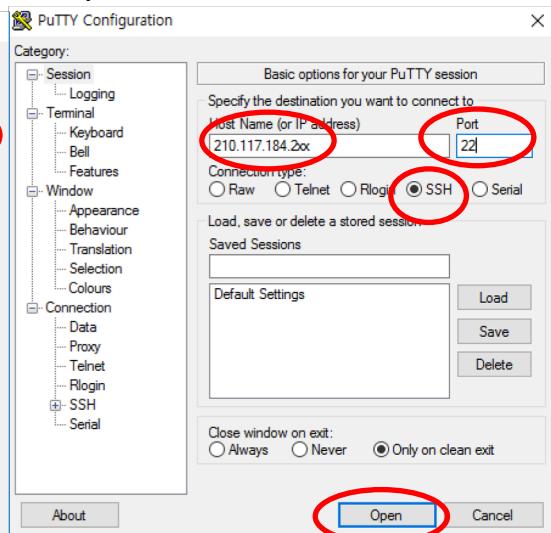
호스트에서 보드 원격 접속

- Putty에서 ssh로 접속
 - 접속이 지속되게 하기 위해 keepalives 주기를 1로 설정
 - 접속 방법 (ssh)
 - 보드 접속 IP, 포트
 - 실습시간에 별도로 안내하는 IP에 반드시 자리마다 지정된 포트 번호로 접속할 것
 - 수업 홈페이지 참고
- 필요에 따라 여러 개의 창을 띠워서 작업할 수 있음

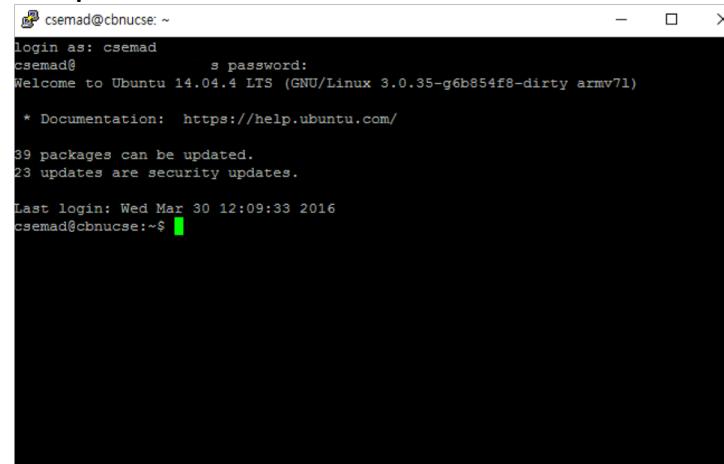
Step 1



Step 2



Step 3



실습 1.1 - 모듈 프로그램

- 간단한 모듈 프로그램 작성 및 빌드
- 홈디렉토리(/home/csemad) 밑에 lab1_module 디렉토리를 만들고 디렉토리 안으로 이동
- hello_module.c와 Makefile 파일 작성

```
1 #include <linux/kernel.h>
2 #include <linux/module.h>
3 #include <linux/init.h>
4
5 static int module_begin()
6 {
7     printk(KERN_ALERT "hello, wellcome to module\n");
8     return 0;
9 }
10
11 static void module_end()
12 {
13     printk(KERN_ALERT "Good bye, exit module\n");
14 }
15
16 module_init(module_begin);
17 module_exit(module_end);
~
```

```
obj-m += hello_module.o

all: driver

driver:
    make -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules

clean:
    make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean
```

모듈 프로그램 빌드

- 다음과 같이 make 수행

```
csemad@cbnucse:~/lab1_module$ ls
Makefile  hello_module.c
csemad@cbnucse:~/lab1_module$ make
make -C /lib/modules/3.0.35-g6b854f8-dirty/build M=/home/csemad/lab1_module modules
make[1]: Entering directory `/mnt/nfs/kernel'
  CC [M]  /home/csemad/lab1_module/hello_module.o
/home/csemad/lab1_module/hello_module.c:5:12: warning: function declaration isn't a prototype [-Wstrict-prototypes]
 static int module_begin()
           ^
/home/csemad/lab1_module/hello_module.c:11:13: warning: function declaration isn't a prototype [-Wstrict-prototypes]
 static void module_end()
           ^
Building modules, stage 2.
MODPOST 1 modules
CC      /home/csemad/lab1_module/hello_module.mod.o
 LD [M]  /home/csemad/lab1_module/hello_module.ko
make[1]: Leaving directory `/mnt/nfs/kernel'
csemad@cbnucse:~/lab1_module$ ls
Makefile      hello_module.c  hello_module.mod.c  hello_module.o
Module.symvers  hello_module.ko  hello_module.mod.o  modules.order
csemad@cbnucse:~/lab1_module$
```

모듈 프로그램 실행

- insmod, lsmod, rmmod 명령어로 모듈 적재, 확인, 제거

```
csemad@cbnucse:~/lab1_module$ sudo insmod hello_module.ko
csemad@cbnucse:~/lab1_module$ sudo lsmod
Module           Size  Used by
hello_module      802   0
csemad@cbnucse:~/lab1_module$ sudo rmmod hello_module
csemad@cbnucse:~/lab1_module$
```

- 출력 메세지 확인

- 부팅 콘솔(serial port에 연결된 putty 화면)에는 직접 출력됨



COM4 - PuTTY

```
csemad@cbnucse:~$ [ 2772.084452] hello, wellcome to module
[ 2776.128207] Good bye, exit module
```

- ssh 접속 창에서는 dmesg 명령어로 출력 메세지 확인

```
csemad@cbnucse:~/lab1_module$ dmesg
[    0.000000] Initializing cgroup subsys cpuset
[    0.000000] Initializing cgroup subsys cpu
[    0.000000] Linux version 3.0.35-g6b854f8-dirty (suhak@suhak-B85M-D3H) (gcc version 4.8.3 20140320 (prerelease) (So
urcery CodeBench Lite 2014.05-29) ) #1 SMP PREEMPT Mon Aug 31 11:44:05 KST 2015
[    0.000000] CPU: ARMv7 Processor [412fc09a] revision 10 (ARMv7), cr=10c53c7d
...
...
```

```
[ 2772.084452] hello, wellcome to module
[ 2776.128207] Good bye, exit module
csemad@cbnucse:~/lab1_module$
```

실습 1.2 – 간단한 디바이스 드라이버 및 테스트 프로그램 작성

■ 디바이스 드라이버 - skeleton.c

```
1 #include <linux/module.h>
2 #include <linux/fs.h>
3 #include <linux/kernel.h>
4 #include <linux/init.h>
5 #include <linux/major.h>
6
7 MODULE_LICENSE("GPL");
8
9 int result;
10
11 int skeleton_open(struct inode *inode, struct file *filp) {
12     printk("Device Open!!\n");
13     return 0;
14 }
15
16 int skeleton_release(struct inode *inode, struct file *filp) {
17     printk("Device Release!!\n");
18     return 0;
19 }
20
21 int skeleton_read(struct file *filp, const char *buf, size_t count, loff_t *f_pos){
22     printk("Device Read!!\n");
23     return 0;
24 }
25
26 int skeleton_ioctl(struct file *filp, unsigned int cmd, unsigned long arg)
27 {
28     printk("Device Ioctl!!\n");
29     return 0;
30 }
31
32 int skeleton_write(struct file *filp, unsigned int *buf, size_t count, loff_t *f_pos)
33 {
34     printk("Device Write!!\n");
35     return 0;
36 }
37
```

```
38 struct file_operations skeleton_fops = {
39     .open = skeleton_open,
40     .release= skeleton_release,
41     .read = skeleton_read,
42     .unlocked_ioctl = skeleton_ioctl,
43     .write = skeleton_write,
44 };
45
46 static int skeleton_init(void)
47 {
48     printk("skeleton module init!!\n");
49     result = register_chrdev(248, "skeleton", &skeleton_fops);
50     printk("return value=%d\n", result);
51     return 0;
52 }
53
54 static void skeleton_exit(void)
55 {
56     printk("skeleton module exit!!\n");
57 }
58
59 module_init(skeleton_init);
60 module_exit(skeleton_exit);
```

실습 1.2

■ 디바이스 드라이버 빌드용 Makefile

```
1 obj-m += skeleton.o
2
3 all: driver
4
5 driver:
6   make -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules
7
8 clean:
9   make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean
10
```

실습 1.2

■ 테스트용 응용 프로그램 - userapp.c

```
1 #include <stdio.h>
2 #include <fcntl.h>
3 #include <unistd.h>
4 #include <stdlib.h>
5 #include <string.h>
6 #include <sys/ioctl.h>
7
8 int main()
9 {
10    int fd;
11
12    if (0 > (fd = open("/dev/csemad", 0_RDWR))) {
13        printf("failed to open\n");
14        return -1;
15    }
16
17    printf("Device Driver Test Application\n");
18
19    read(fd, 0, 0);
20    ioctl(fd, 0, 0);
21    close(fd);
22
23    return 0;
24 }
```

디바이스 드라이버 및 응용프로그램 빌드

■ 디바이스 드라이버

```
csemad@cbnucse:~/lab1_simple_device_driver$ make
make -C /lib/modules/3.0.35-g6b854f8-dirty/build M=/home/csemad/lab1_simple_device_driver modules
make[1]: Entering directory `/mnt/nfs/kernel'
CC [M]  /home/csemad/lab1_simple_device_driver/skeleton.o
/home/csemad/lab1_simple_device_driver/skeleton.c:41:3: warning: initialization from incompatible pointer type [enabled by default]
    .read = skeleton_read,
    ^
/home/csemad/lab1_simple_device_driver/skeleton.c:41:3: warning: (near initialization for 'skeleton_fops.read') [enabled by default]
/home/csemad/lab1_simple_device_driver/skeleton.c:42:3: warning: initialization from incompatible pointer type [enabled by default]
    .unlocked_ioctl = skeleton_ioctl,
    ^
/home/csemad/lab1_simple_device_driver/skeleton.c:42:3: warning: (near initialization for 'skeleton_fops.unlocked_ioctl') [enabled by default]
/home/csemad/lab1_simple_device_driver/skeleton.c:43:3: warning: initialization from incompatible pointer type [enabled by default]
    .write = skeleton_write,
    ^
/home/csemad/lab1_simple_device_driver/skeleton.c:43:3: warning: (near initialization for 'skeleton_fops.write') [enabled by default]
Building modules, stage 2.
MODPOST 1 modules
CC      /home/csemad/lab1_simple_device_driver/skeleton.mod.o
LD [M]  /home/csemad/lab1_simple_device_driver/skeleton.ko
make[1]: Leaving directory `/mnt/nfs/kernel'
csemad@cbnucse:~/lab1_simple_device_driver$
```

■ 응용 프로그램

```
csemad@cbnucse:~/lab1_simple_device_driver$ gcc -Wall userapp.c -o userapp
csemad@cbnucse:~/lab1_simple_device_driver$
```

■ 빌드 결과

```
csemad@cbnucse:~/lab1_simple_device_driver$ ls
Makefile Module.symvers modules.order skeleton.c skeleton.ko skeleton.mod.c skeleton.mod.o skeleton.o userapp userapp.c
csemad@cbnucse:~/lab1_simple_device_driver$
```

응용 프로그램 실행

- mknod 명령어로 디바이스 생성
- insmod 명령어로 디바이스 드라이버 적재
- userapp 실행
- lsmod 명령어로 디바이스 드라이버 확인
- rmmod 명령어로 디바이스 드라이버 제거

```
csemad@cbnucse:~/lab1_simple_device_driver$ sudo mknod /dev/csemad c 248 0
csemad@cbnucse:~/lab1_simple_device_driver$ sudo insmod skeleton.ko
csemad@cbnucse:~/lab1_simple_device_driver$ sudo ./userapp
Device Driver Test Application
csemad@cbnucse:~/lab1_simple_device_driver$ lsmod
Module           Size  Used by
skeleton        1741  0
csemad@cbnucse:~/lab1_simple_device_driver$ sudo rmmod skeleton
csemad@cbnucse:~/lab1_simple_device_driver$
```

- dmesg 명령어로 디바이스 드라이버단에서의 출력 메세지 확인

```
csemad@cbnucse:~/lab1_simple_device_driver$ dmesg | tail -f
[ 2733.134611] hello, wellcome to module
[ 2743.103118] Good bye, exit module
[ 2772.084452] hello, wellcome to module
[ 2776.128207] Good bye, exit module
[ 4866.848215] skeleton module init!!
[ 4866.848231] return value=-16
[ 5664.806145] skeleton module exit!!
[ 6336.884238] skeleton module init!!
[ 6336.884253] return value=-16
[ 6366.668900] skeleton module exit!!
csemad@cbnucse:~/lab1_simple_device_driver$
```

■ 실습 과제

- 실습 1.1
 - 모듈 컴파일시 출력되는 경고 메세지를 없애도록 코드 수정
- 실습 1.2
 - 디바이스 드라이버 빌드시 출력되는 경고 메세지를 없애도록 코드 수정
 - 디바이스 드라이버 적재, 제거시 `printf()` 함수의 출력 메세지가 부팅 콘솔에서도 출력되도록 코드 수정