

# Embedded System

## - Ch.7 FPGA Device Driver -

Su-il Choi

Department of Electronics Engineering  
Chonnam National University

### Contents

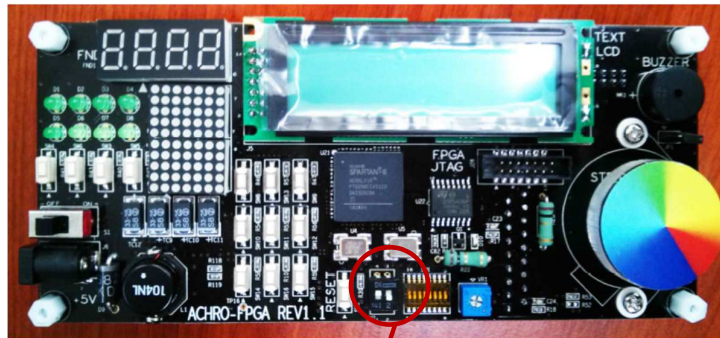
---

- FPGA Device Driver
  - External LED
  - External FND
  - Dot Matrix
  - Text LCD
  - Buzzer
  - DIP Switch
  - PUSH Switch
  - STEP Motor

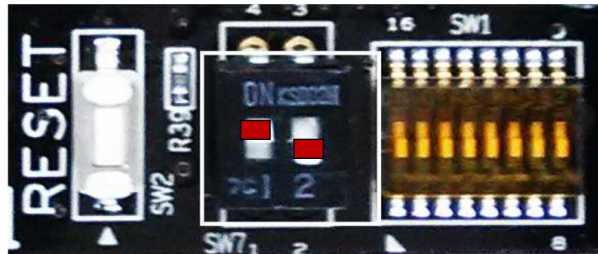
# 1. External FPGA Device Driver



- Acro-4210의 FPGA 모듈을 제어하는 Device Driver



- Achro-4210의 FPGA 확장 모듈을 사용자 모드로 설정
  - SW7의 Dip switch 1번(ON) 2번(OFF)



## External FPGA Module



- Address MAP

번호	장치	어드레스	Node	Major
1	LED	0x0400_0016	/dev/fpga_led	260
2	Seven Segment (FND)	0x0400_0004	/dev/fpga_fnd	261
3	Dot Matrix	0x0400_0210	/dev/fpga_dot	262
4	Text LCD	0x0400_0100	/dev/fpga_text_lcd	263
5	Buzzer	0x0400_0020	/dev/fpga_buzzer	264
6	Push Switch	0x4000_0017	/dev/fpga_push_switch	265
7	3Dip Switch	0x0400_0000	/dev/fpga_dip_switch	266
8	Step Motor	0x0400_000C	/dev/fpga_step_motor	267
EN	Demo Register	0x4000_0300	N/A	N/A

- Demo Register

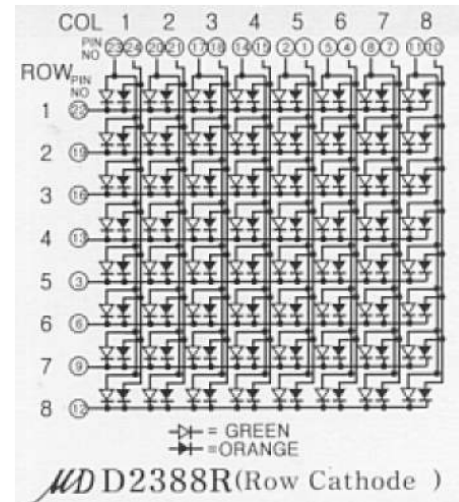
- 사용자 모드를 SW로 ON/OFF 제어하기 위함 (사용자 모드: ON)

### 3) Dot Matrix



- Dot Matrix
  - LED Array 구조로 되어 있으며, 여기서 쓰이는 UD-D2388R은 Row Cathode 방식
  - Row에 Low가 입력 했을 경우 COL의 입력상태에 따라 출력을 나타냄
  - Achro-4210에서 보내는 데이터는 High를 보내야 한다.
  - 그러면 FPGA에서는 해당 값을 Low로 변경하여 LED에 불이 켜짐.

- IOM\_FPGA\_DOT\_ADDRESS: 0x04000210
- IOM\_FPGA\_DOT\_NAME: fpga\_dot
- IOM\_FPGA\_DOT\_MAJOR: 262



### Dot Matrix



- Dot Matrix Source 준비

```
# cp -a /mnt/hgfs/SharedFolder/examples/linux/fpga_driver/fpga_buzzer.tar.gz ~/temp
# tar zxvf fpga_buzzer.tar.gz
# cd fpga_buzzer
```

- Compile (Host)

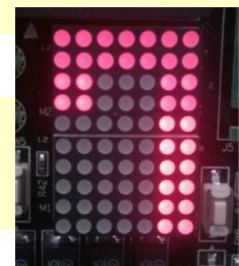
```
# make
# make install (/nfsroot에 driver와 test program을 복사)
```

- Network 설정 (Target)

```
# ifconfig eth0 192.168.11.xx up
# ping 192.168.11.yy
# mount -t nfs 192.168.11.yy:/nfsroot /mnt/nfs -o rw,rsz=1024,nolock
# cd /mnt/nfs
```

- Device Driver 실행 (Target)

```
# mknod /dev/fpga_dot c 262 0
# insmod fpga_dot_driver.ko
# ./fpga_test_dot 7
```





- Dot Matrix Makefile

```
#Makefile for a basic kernel module
obj-m := fpga_dot_driver.o
CC :=arm-none-linux-gnueabi-gcc           // 추가
KDIR :=/root/temp/kernel                 // 수정
PWD :=$(shell pwd)
APP := fpga_test_dot

all: driver app
#all: driver

driver:
    $(MAKE) -C $(KDIR) SUBDIRS=$(PWD) modules

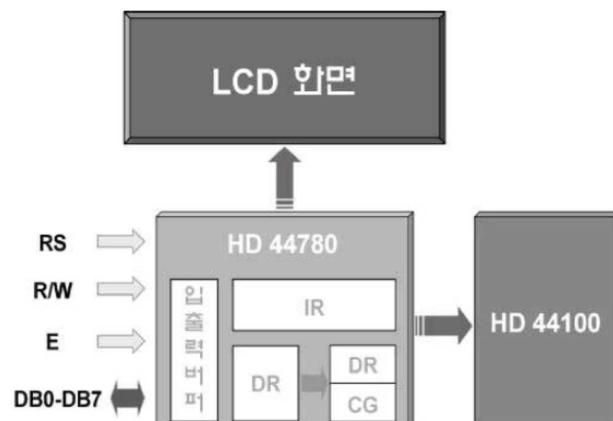
app:
    $(CC) -o $(APP) $(APP).c             // 수정

install:
    cp -a fpga_dot_driver.ko /nfsroot
    cp -a $(APP) /nfsroot
clean:
    rm -rf *.ko
    rm -rf *.mod.*
...
```

## 4) Text LCD



- Text-LCD Module
  - 8bit의 Micro-process, 2개의 Register (DR, IR)
- IR (instruction register)
  - LCD 화면 clear, 커서 시프트, 글자 on/off 등 LCD 제어와 관련된 명령 정보 저장
- DR (data register)
  - D.D. RAM 또는 C.G.RAM에 써 넣을 데이터를 저장하거나, D.D.RAM 또는 C.G.RAM으로 읽어진 데이터가 일시적으로 저장되는 Register



# Text LCD



- Text LCD Source 준비

```
# cp -a /mnt/hgfs/SharedFolder/examples/linux/fpga_driver/fpga_text_lcd.tar.gz ~/temp
# tar zxvf fpga_text_lcd.tar.gz
# cd fpga_text_lcd
```

- Compile (Host)

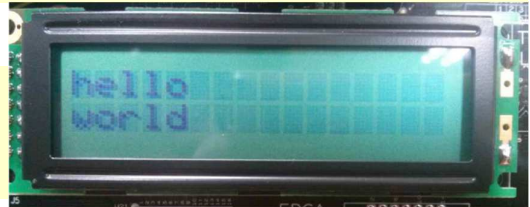
```
# make
# make install (/nfsroot에 driver와 test program을 복사)
```

- Network 설정 (Target)

```
# ifconfig eth0 192.168.11.xx up
# ping 192.168.11.yy
# mount -t nfs 192.168.11.yy:/nfsroot /mnt/nfs -o rw,rsz=1024,nolock
# cd /mnt/nfs
```

- Device Driver 실행 (Target)

```
# mknod /dev/fpga_text_lcd c 263 0
# insmod fpga_text_lcd_driver.ko
# ./fpga_test_text_lcd hello world
```



# Text LCD: Makefile



- Text LCD Device Driver Makefile

```
#Makefile for a basic kernel module
obj-m := fpga_text_lcd_driver.o
CC :=arm-none-linux-gnueabi-gcc // 추가
KDIR :=/root/temp/kernel // 수정

PWD :=$(shell pwd)
APP := fpga_test_text_lcd

all: driver app
#all: driver

driver:
$(MAKE) -C $(KDIR) SUBDIRS=$(PWD) modules

app:
$(CC) -o $(APP) $(APP).c // 수정

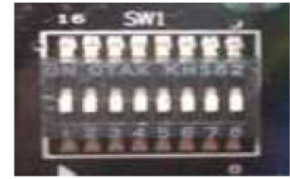
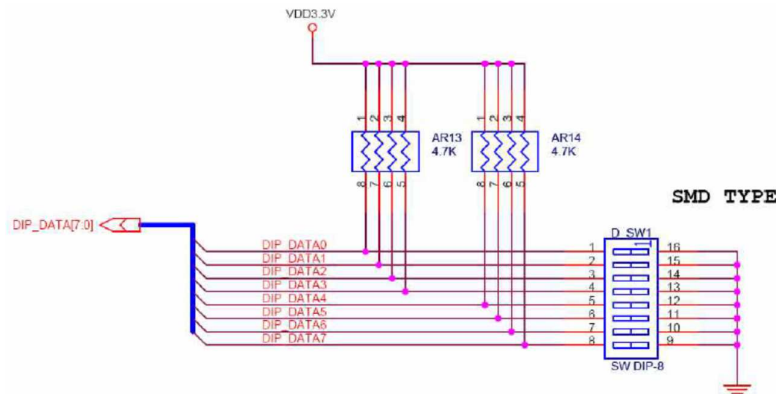
install:
cp -a fpga_text_lcd_driver.ko /nfsroot
cp -a $(APP) /nfsroot

clean:
rm -rf *.ko
rm -rf *.mod.*
...
```

## 5) DIP Switch



- 회로도



- IOM\_FND\_ADDRESS: 0x04000000
- IOM\_FND\_NAME: fpga\_dip\_switch
- IOM\_FND\_MAJOR: 266

## DIP Switch



- DIP Switch Source 준비

```
# cp -a /mnt/hgfs/SharedFolder/examples/linux/fpga_driver/fpga_dip_switch.tar.gz ~/temp
# tar zxvf fpga_dip_switch.tar.gz
# cd fpga_dip_switch
```

- Compile (Host)

```
# make
# make install
```

(/nfsroot에 driver와 test program을 복사)

- Network 설정 (Target)

```
# ifconfig eth0 192.168.11.xx up
# ping 192.168.11.yy
# mount -t nfs 192.168.11.yy:/nfsroot /mnt/nfs -o rw,rsz=1024,nolock
# cd /mnt/nfs
```



- Device Driver 실행 (Target)

```
# mknod /dev/fpga_dip_switch c 266 0
# insmod fpga_dip_switch_driver.ko
# ./fpga_test_dip_switch
```

```
root@ubuntu: /
File Edit View Terminal Help
[root@Achro4210 nfs]# insmod fpga_dip_switch_driver.ko
[ 3399.628790] init module: fpga_dip_switch major number : 266
[root@Achro4210 nfs]# mknod /dev/fpga_dip_switch c 266 0
[root@Achro4210 nfs]# ls
[root@Achro4210 nfs]# ./fpga_test_dip_switch
Press <ctrl+c> to quit.
Read dip switch: 0xFF
Read dip switch: 0xFF
Read dip switch: 0xFF
Read dip switch: 0xFF
Read dip switch: 0xFF
CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.4 | VT102 | Offline
```

# DIP Switch: Makefile



- DIP Switch Driver Makefile

```
#Makefile for a basic kernel module
obj-m := fpga_dip_switch_driver.o
CC :=arm-none-linux-gnueabi-gcc // 추가
KDIR :=/root/temp/kernel // 수정
PWD :=$(shell pwd)

all: driver app
#all: driver

driver:
    $(MAKE) -C $(KDIR) SUBDIRS=$(PWD) modules

app:
    $(CC) -o fpga_test_dip_switch fpga_test_dip_switch.c // 수정

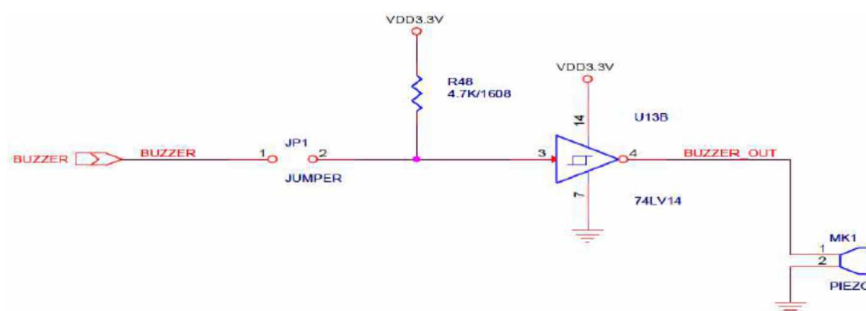
install:
    cp -a fpga_dip_switch_driver.ko /nfsroot
    cp -a fpga_test_dip_switch /nfsroot

clean:
    rm -rf *.ko
    rm -rf *.mod.*
    rm -rf *.o
    ...
```

## 7) Buzzer



- 회로도



- IOM\_FND\_ADDRESS: 0x04000020
- IOM\_FND\_NAME: fpga\_buzzer
- IOM\_FND\_MAJOR: 264



- Buzzer Source 준비

```
# cp -a /mnt/hgfs/SharedFolder/examples/linux/fpga_driver/fpga_buzzer.tar.gz ~/temp
# tar zxvf fpga_buzzer.tar.gz
# cd fpga_buzzer
```

- Compile (Host)

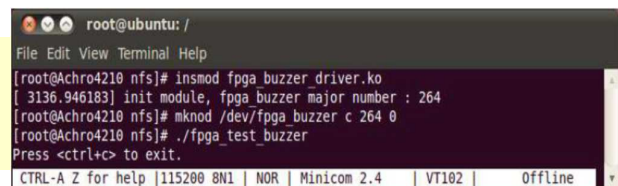
```
# make
# make install (/nfsroot에 driver와 test program을 복사)
```

- Network 설정 (Target)

```
# ifconfig eth0 192.168.11.xx up
# ping 192.168.11.yy
# mount -t nfs 192.168.11.yy:/nfsroot /mnt/nfs -o rw,rsiz=1024,nolock
# cd /mnt/nfs
```

- Device Driver 실행 (Target)

```
# mknod /dev/fpga_buzzer c 264 0
# insmod fpga_buzzer_driver.ko
# ./fpga_test_buzzer
```



```
root@ubuntu: /
File Edit View Terminal Help
[root@Achro4210 nfs]# insmod fpga_buzzer_driver.ko
[ 3136.946183] init module, fpga_buzzer major number : 264
[root@Achro4210 nfs]# mknod /dev/fpga_buzzer c 264 0
[root@Achro4210 nfs]# ./fpga_test_buzzer
Press <ctrl+c> to exit.
CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.4 | VT102 | Offline
```

## Buzzer: Makefile



- Buzzer Driver Makefile

```
#Makefile for a basic kernel module
obj-m := fpga_buzzer_driver.o
CC :=arm-none-linux-gnueabi-gcc // 추가
KDIR :=/root/temp/kernel // 수정
PWD :=$(shell pwd)

all: driver app
#all: driver

driver:
$(MAKE) -C $(KDIR) SUBDIRS=$(PWD) modules

app:
$(CC) -o fpga_test_buzzer fpga_test_buzzer.c

install:
cp -a fpga_buzzer_driver.ko /nfsroot
cp -a fpga_test_buzzer /nfsroot

clean:
rm -rf *.ko
rm -rf *.mod.*
rm -rf *.o
...
```



## 8) Step motor

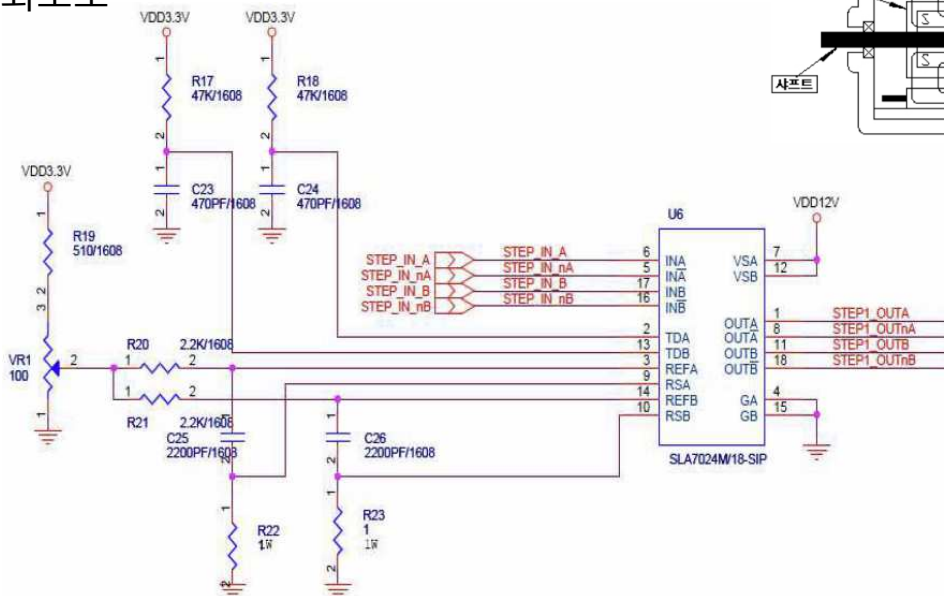


- 스텝 모터(steping motor, pulse motor, stepper motor)
  - AC servo, DC servo motor에 비해 정확한 각도 제어에 유리
  - 디지털 펄스를 기계적인 축 운동으로 변화시키는 변환기
- 장점
  - 디지털 신호를 직접 open loop 제어를 할 수 있고, 전반적으로 간단하게 구동
  - 회전 속도: 펄스 신호의 주파수에 비례
  - 모터 회전각: 입력 pulse 수에 비례
  - 기동, 정지, 정-역 회전, 변속이 용이하며, 응답 특성도 양호
  - 1 스텝 당 가도 오차가 +5% 이내, 회전각의 오차는 step 마다 누적된다
  - 초 저속으로 높은 토크(torque) 운전이 가능
- 단점
  - 고속 운전 시 탈조가 쉽다
  - 특정 주파수에서는 진공, 공진 현상이 발생하기 쉽고, 관성이 있는 부하에 약하다
  - Pulse 비가 상승함에 따라 Torque가 저하하며, DC motor에 비해 효율이 떨어진다.

## Step Motor



- IOM\_FPGA\_STEP\_MOTOR\_ADDRESS: 0x0400000C
- IOM\_FPGA\_STEP\_MOTOR\_NAME: fpga\_step\_motor
- IOM\_FPGA\_STEP\_MOTOR\_MAJOR: 267
- 회로도



# Step Motor



- Buzzer Source 준비

```
# cp -a /mnt/hgfs/SharedFolder/examples/linux/fpga_driver/fpga_step_motor.tar.gz ~/temp
# tar zxvf fpga_step_motor.tar.gz
# cd fpga_step_motor
```

- Compile (Host)

```
# make
# make install (/nfsroot에 driver와 test program을 복사)
```

- Network 설정 (Target)

```
# ifconfig eth0 192.168.11.xx up
# ping 192.168.11.yy
# mount -t nfs 192.168.11.yy:/nfsroot /mnt/nfs -o rw,rsz=1024,nolock
# cd /mnt/nfs
```

- Device Driver 실행 (Target)

```
# mknod /dev/fpga_step_motor c 267 0
# insmod fpga_step_motor_driver.ko
# ./fpga_test_step_motor 1 1 10
```

# Step Motor



- Step Motor 테스트 Program

- 실행시 세 개의 argument 입력
- 작동에 관한 argument, 방향에 관한 argument, 속도에 관한 argument
- 1) motor ON/OFF

	Argument	Action
Motor Action	0	정지
	1	구동

- 2) Motor의 방향

	Argument	Action
Motor Direction	0	시계방향
	1	반시계방향

- 3) Motor 속도

	Argument	Action
Motor Speed	5	255