

TI DSP, MCU 및 Xilinx Zynq FPGA 프로그래밍 전문가 과정

강사 - Innova Lee(이상훈)

gcccompil3r@gmail.com

학생 - GJ (박현우)

uc820@naver.com

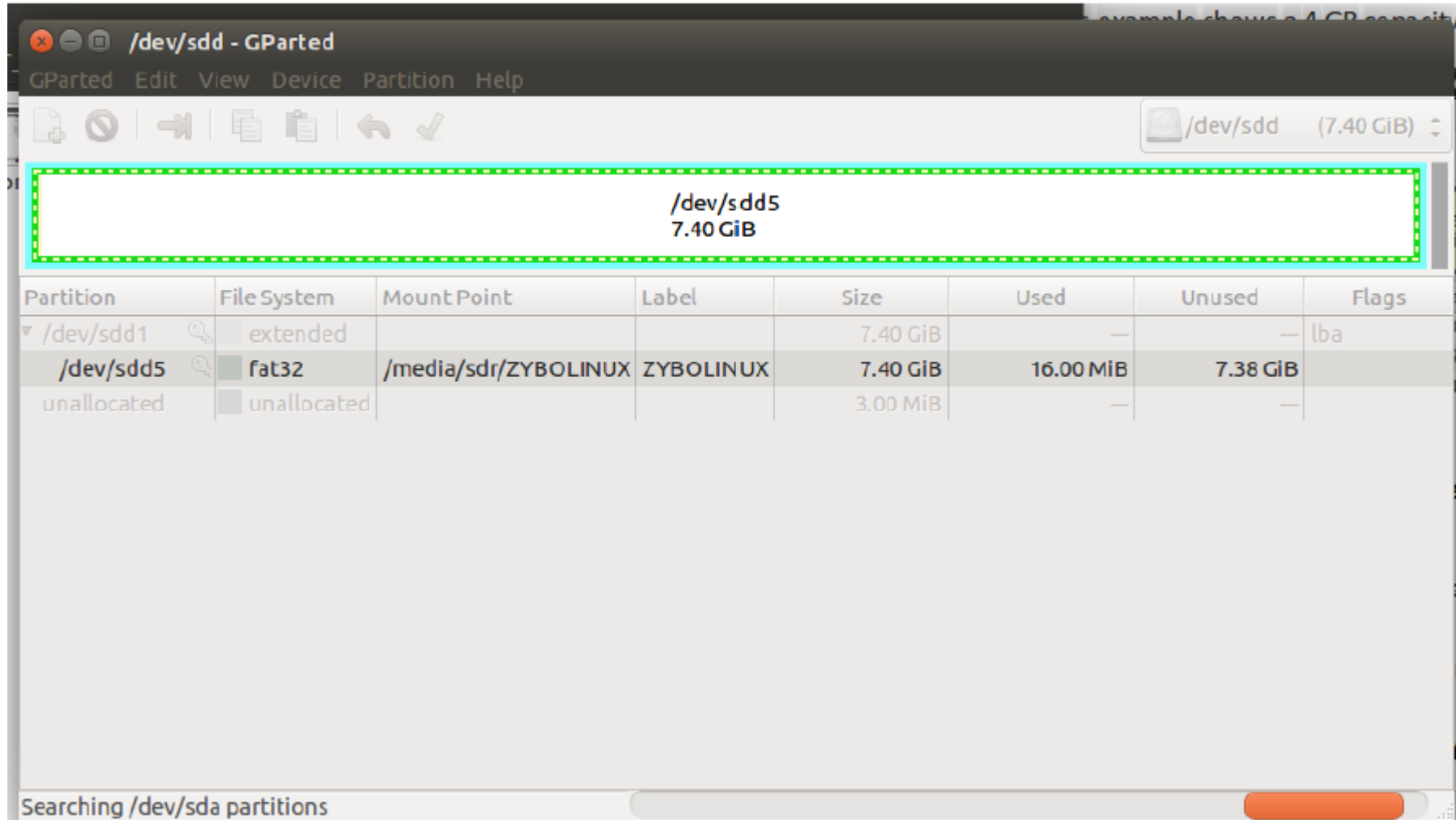
목차

FPGA on Petalinux

- 1) SD카드 설정
- 2) device driver code
- 3) Device Tree 추가

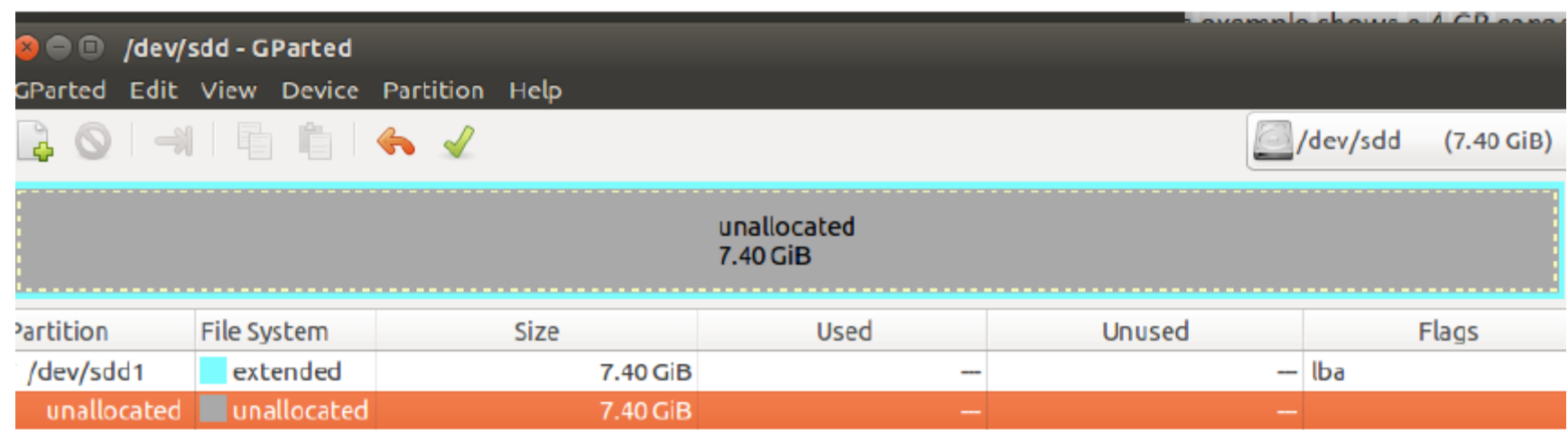
1) SD 카드 설정 1

파일 시스템이 fat32 인 부분을 우클릭하고 umount 한다.

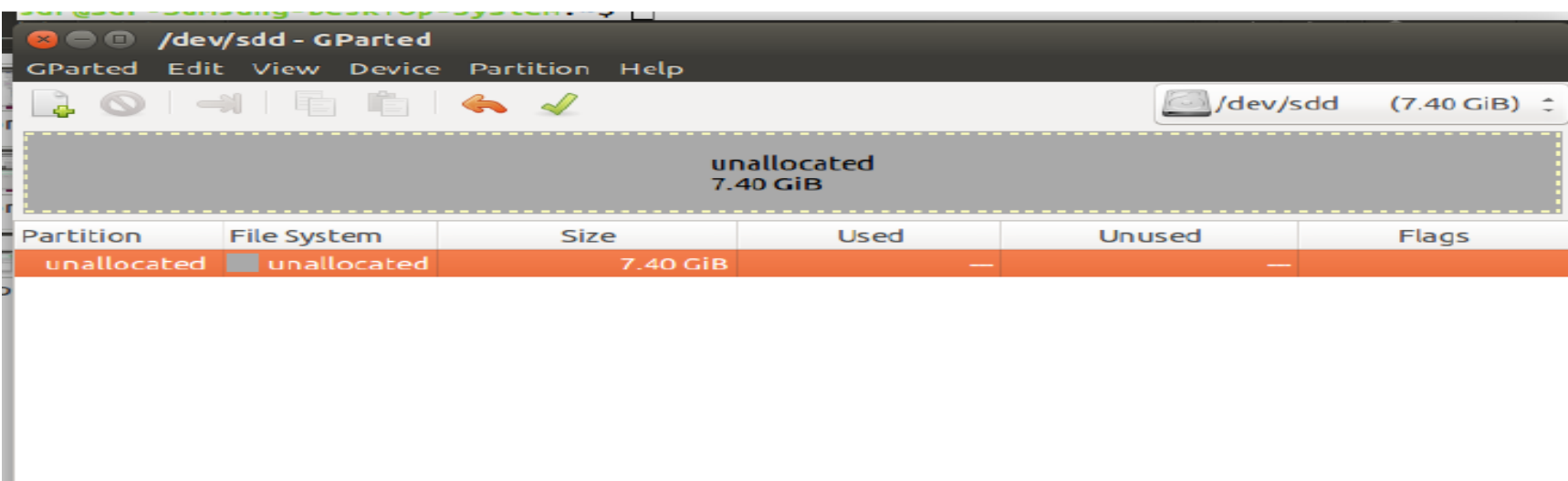


1) SD 카드 설정 2

파일 시스템이 fat32 인 부분을 우클릭하고 delete 한다.

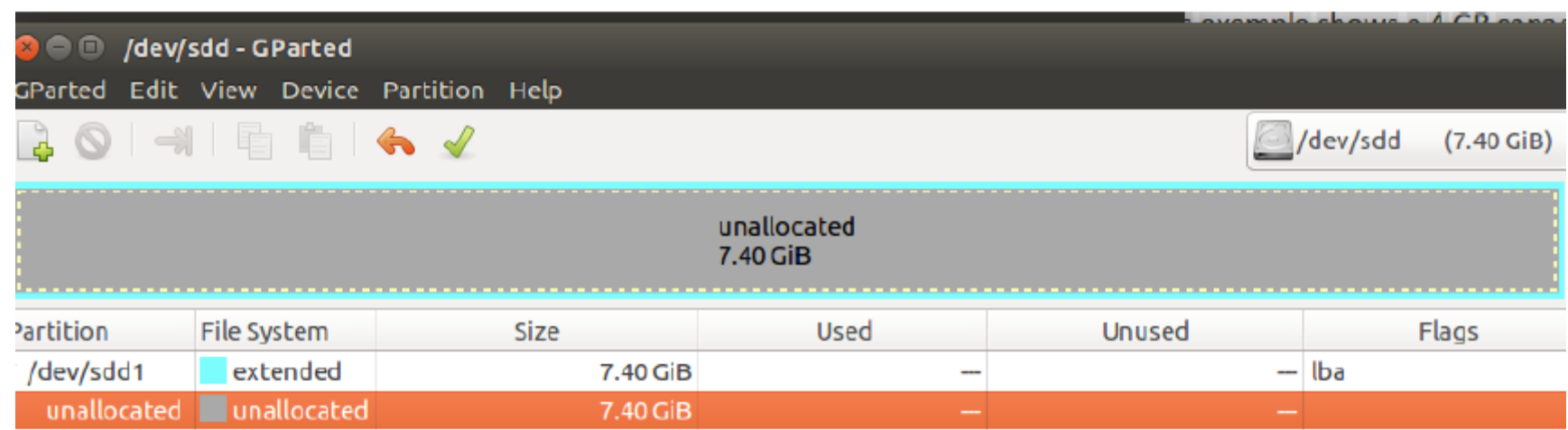


남아 있는 Extended 부분도 마저 delete 한다.

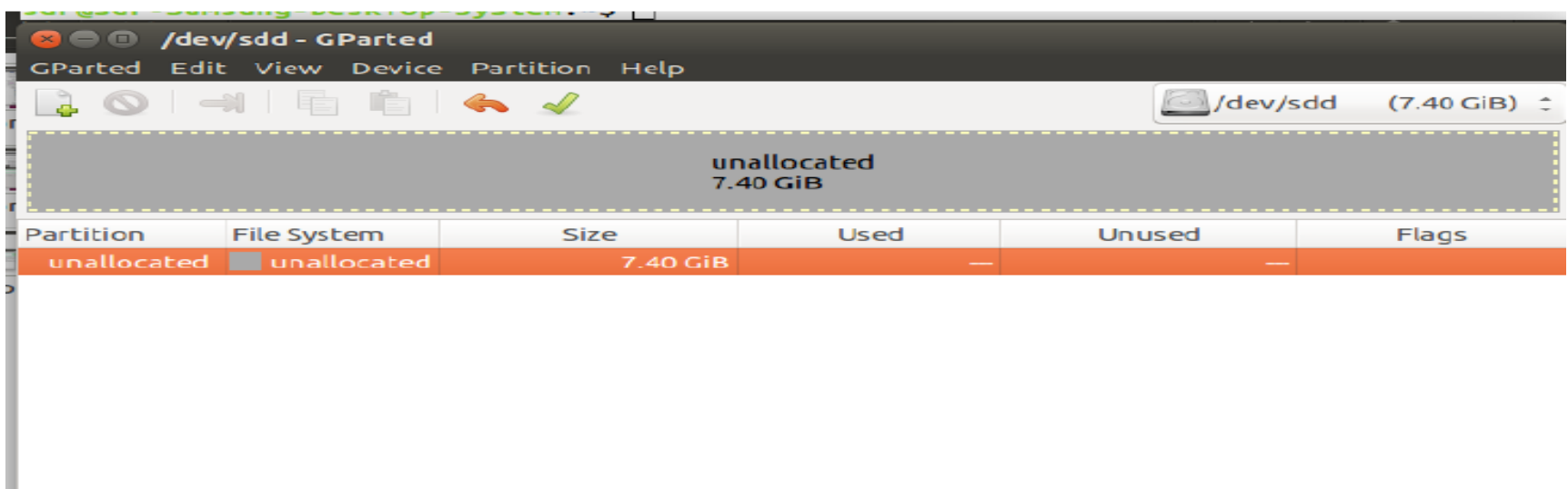


1) SD 카드 설정 3

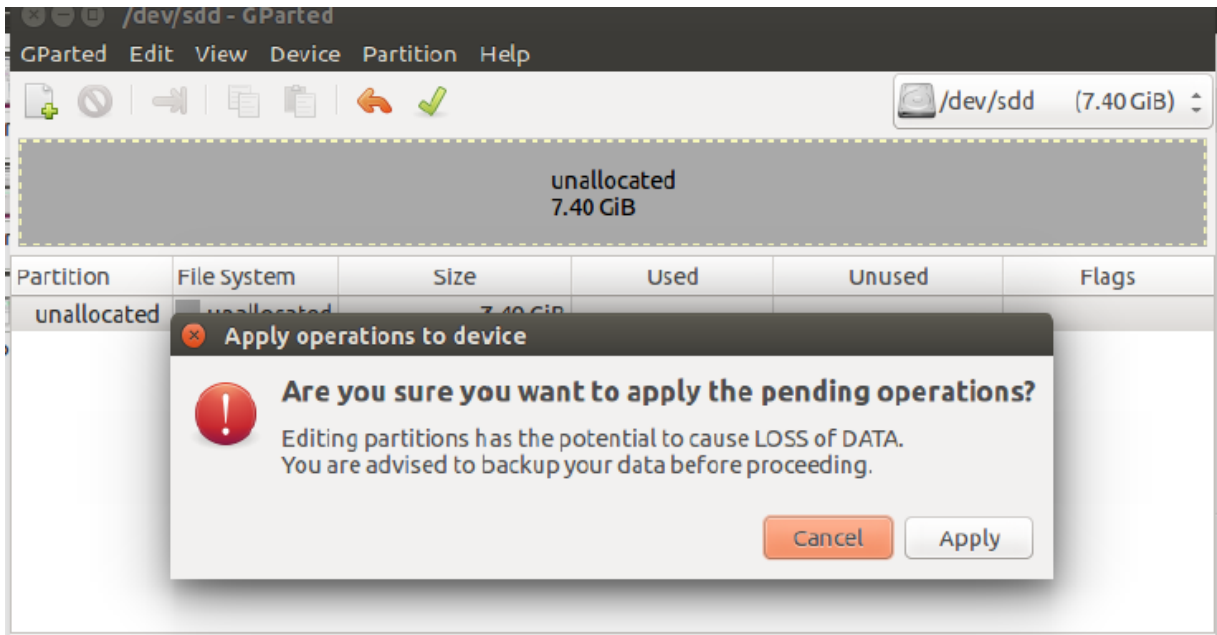
파일 시스템이 fat32 인 부분을 우클릭하고 delete 한다.



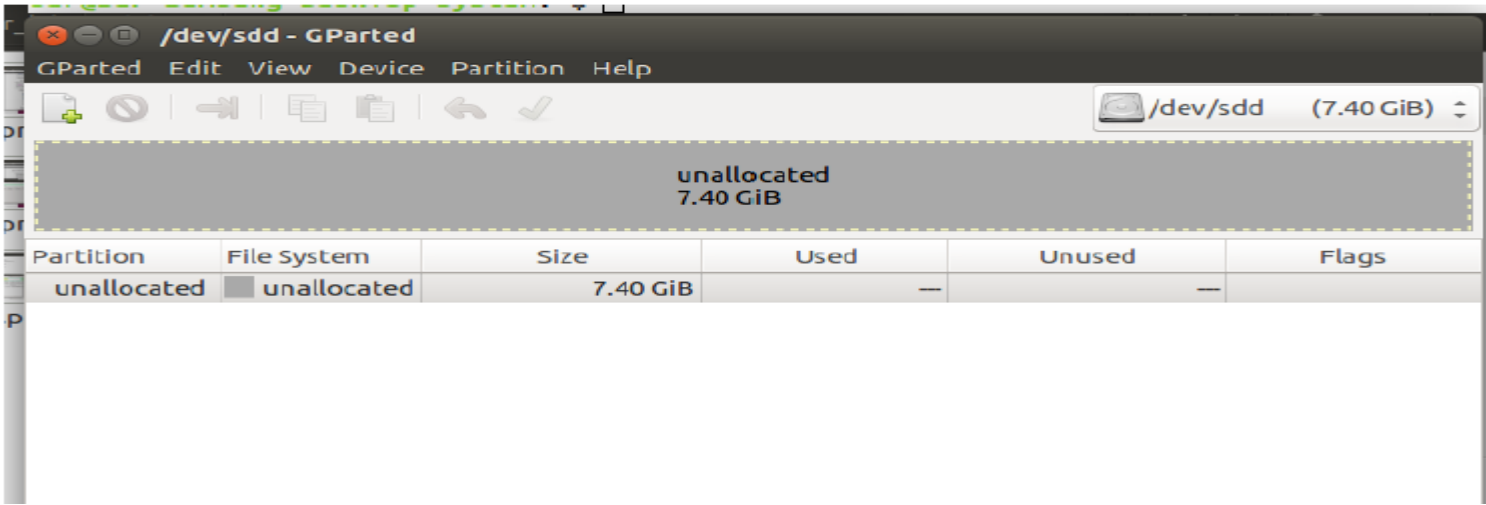
남아 있는 Extended 부분도 마저 delete 한다.



1) SD 카드 설정 4





여기서 이제 우클릭하고 New 를 누른다.



1) SD 카드 설정 5


Create new Partition

Minimum size: 33 MiB Maximum size: 7579 MiB

Free space preceding (MiB):	<input type="text" value="4"/>	Create as:	<input type="text" value="Primary Partition"/>
New size (MiB):	<input type="text" value="52"/>	Partition name:	<input type="text"/>
Free space following (MiB):	<input type="text" value="7524"/>	File system:	<input type="text" value="fat32"/>
Align to:	<input type="text" value="MiB"/>	Label:	<input type="text" value="BOOT"/>

아래쪽 공간이 큰 녀석을 우클릭해서 New 한다.



The screenshot shows the GParted application window. The title bar reads "/dev/sdd - GParted". The menu bar includes "GParted", "Edit", "View", "Device", "Partition", and "Help". The toolbar contains icons for creating, deleting, moving, copying, pasting, and committing changes. The main display shows a disk of 7.40 GiB with a large unallocated area of 7.35 GiB highlighted with a dashed yellow border. Below the visual representation is a table showing the partition details.

Partition	File System	Label	Size	Used	Unused	Flags
unallocated	unallocated		4.00 MiB	—	—	
New Partition #1	fat32	BOOT	52.00 MiB	—	—	
unallocated	unallocated		7.35 GiB	—	—	

Partition	File System	Label	Size	Used	Unused	Flags
unallocated	unallocated		4.00 MiB	—	—	
New Partition #1	Fat32	BOOT	52.00 MiB	—	—	
unallocated	unallocated		7.35 GiB	—	—	

1) SD 카드 설정 6

Create new Partition

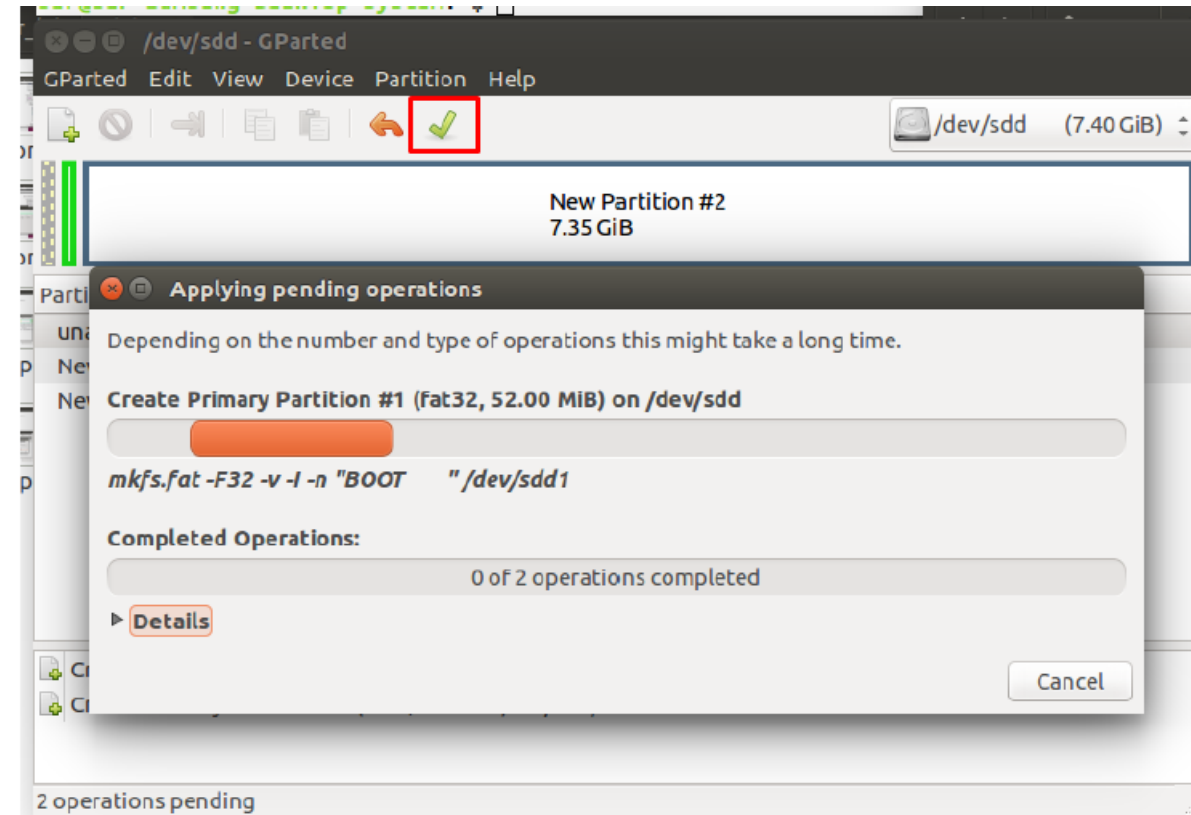
Minimum size: 1 MiB Maximum size: 7524 MiB

Free space preceding (MiB): Create as:

New size (MiB): Partition name:

Free space following (MiB): File system:

Align to: Label:

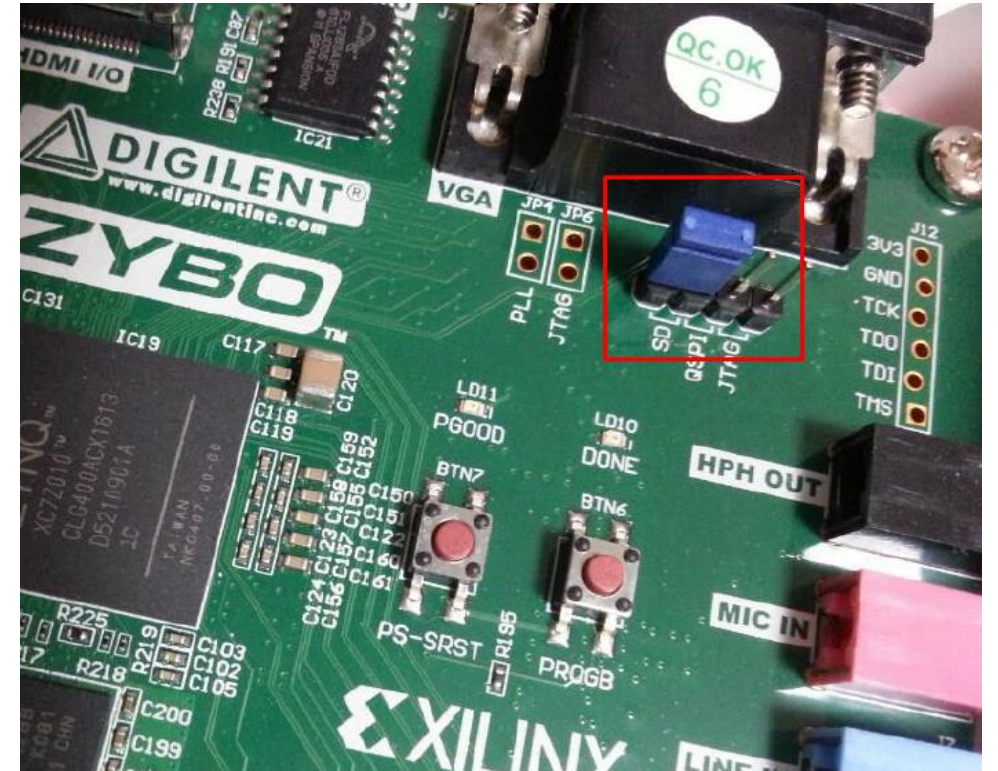


1) SD 카드 설정 7

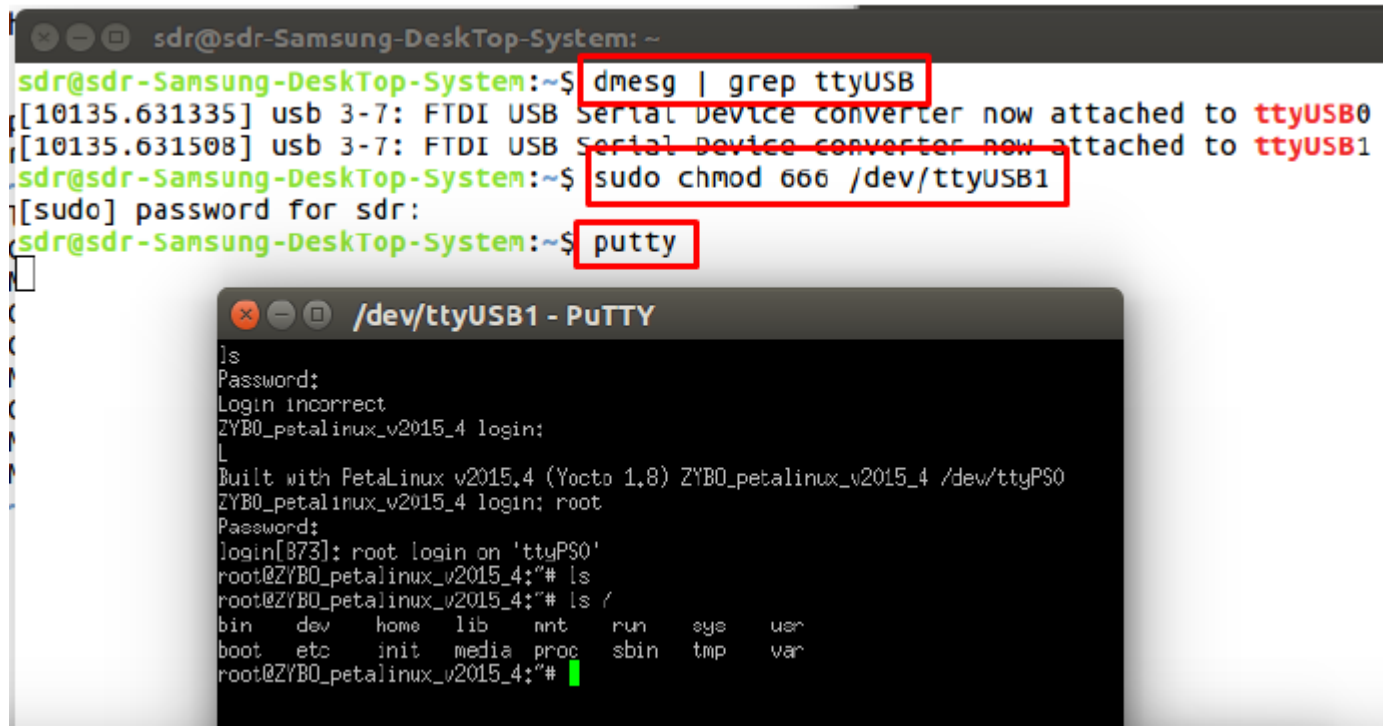


위치 : Software or Kernel/image/linux에 있음

SD 카드의 BOOT 에 위 2 개의 파일을 넣는다.



1) SD 카드 설정 8



The image shows two terminal windows. The top window is a Linux terminal with the prompt 'sdr@sdr-Samsung-DeskTop-System: ~'. It contains the following commands and output:

```
sdr@sdr-Samsung-DeskTop-System:~$ dmesg | grep ttyUSB
[10135.631335] usb 3-7: FTDI USB Serial Device converter now attached to ttyUSB0
[10135.631508] usb 3-7: FTDI USB Serial Device converter now attached to ttyUSB1
sdr@sdr-Samsung-DeskTop-System:~$ sudo chmod 666 /dev/ttyUSB1
[sudo] password for sdr:
sdr@sdr-Samsung-DeskTop-System:~$ putty
```

The bottom window is a PuTTY terminal titled '/dev/ttyUSB1 - PuTTY'. It shows the login process for a device named 'ZYBO_petalinux_v2015_4':

```
ls
Password:
Login incorrect
ZYBO_petalinux_v2015_4 login:
L
Built with PetaLinux v2015.4 (Yocto 1.8) ZYBO_petalinux_v2015_4 /dev/ttyPS0
ZYBO_petalinux_v2015_4 login: root
Password:
login[873]: root login on 'ttyPS0'
root@ZYBO_petalinux_v2015_4:~# ls
root@ZYBO_petalinux_v2015_4:~# ls /
bin  dev  home  lib  mnt  run  sys  usr
boot  etc  init  media  proc  sbin  tmp  var
root@ZYBO_petalinux_v2015_4:~#
```

- 1) dmesg 를 통해 USB Device Driver 가 잘 잡히는지 확인한다.
- 2) sudo apt-get install putty
- 3) putty의 폰트 등 각종 설정을 수행한다.
- 4) sudo chmod 666 /dev/ttyUSB1
- 5) Baud Rate 를 115200 으로 지정하고 연결한다.

2) Device Driver Code 1

(38) 이제 HW 설계와 SW 연동을 수행해보도록 한다.
그 이전에 각종 설계를 수행할 것인데
관리를 위해 디렉토리를 한 단계 더 분할하도록 한다.
`mkdir hw_sw_co_design`

(39) `cd hw_sw_co_design`
(40) FPGA 로 설계한 HW 를 보관할 디렉토리를 만든다.
`mkdir hardware`
(41) Xilinx Vivado 틀을 실행한다.
(42) 수업에서 진행하였듯이 아래와 같은 각종 HW 를 설계한다.
(GPIO, ADC, I2C, SPI, PWM, 기타 전용 HW 등등)
프로젝트 저장장 방금 만든 hardware 디렉토리에 저장하도록 한다.
hardware 의 위치는 아래와 같다.
`fpga_dev_driver/hw_sw_co_design/hardware`

(43) `petalinux-create -t project -n kernel --template zynq`
(44) `cd kernel`
(45) `petalinux-config --get-hw-description=~ /fpga_dev_driver/hw_sw_co_design`
`/hardware/~~~.sdk`
여기서 sdk 는 Vivado 에서 HW 설계한 내용에 해당한다.
(46) `cd components/bootloader/zynq_fsbl`
(47) `ls`
FPGA 베이스의 Cortex-A9 부트 코드를 볼 수 있다
(48) `cd ../../..`
(49) `petalinux-config -c u-boot`
(50) `petalinux-build`
(51) `petalinux-create -t apps -n device_driver --enable`
(52) `cd components/apps/device_driver`
(53) `vi device_driver.c` (다음 페이지 에서 코드 작성 해야 함.)

HW 를 제어하기 위한 SW 코드인 Device Driver 코드를 작성한다.

(54) `cd ~/fpga_dev_driver/hw_sw_co_design/kernel/images/linux`
(55) `ls`

여기에 부트 로더와 리눅스 이미지가 있는 것을 볼 수 있을 것이다.

(56) `petalinux-build`
(57) `petalinux-package --boot --fsbl zynq_fsbl.elf --fpga ./비트스트림 --u-boot -force`

[출처] [64 회차 교육 로그](#) | 작성자 [silenc3502](#)

2) Device Driver Code 2

순서대로 코드 입력

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/mman.h>
#include <fcntl.h>

#define IN 0
#define OUT 1

#define GPIO_MAP_SIZE 0X10000

#define GPOP_DATA_OFFSET 0X00
#define GPIO_TRI_OFFSET 0X04
#define GPIO2_DATA_OFFSET 0X00
#define GPIO2_TRI_OFFSET 0X04

void usage(void){

    printf("*argv[0] -d <UIO_DEV_FILE> -i | -o <VALUE>Wn");
    printf(" -d UIO device file - ex) /dev/uio0");
    printf(" -i Input from GPIOWn");
    printf(" -o <VALUE> Output to GPIOWn");

}
```

```
int main(int argc, char *argv[]){

    int c, fd, value, direction = IN;
    char *uiod;
    void *ptr;

    printf("GPIO UIO TestWn");

    while((c = getopt(argc, argv, "d:io:h")) != -1){

        wsitch(c){

            case 'd' :
                uiod = optarg;
                break;
            case 'i' :
                direction = IN;
                break;
            case 'o' :
                direction = OUT;
                value = atoi(optarg);
                break;
            default :
                printf("Invalid Option: %cWn", (char)c);
                usage();
                return -1;

        }

    }

}
```

```
fd = open(uiod, O_RDWR);

if(fd < 1){

    perror(argv[0]);
    printf("Onvalid UIO Device File: %sWn", uiod);
    usage();
    return -1;
}

ptr = mmap(NULL, GPIO_MAP_SIZE, PROT_READ | P
ROT_WRITE, MAP_SHARED, fd, 0);

if(direction == IN){

    *((unsigned *)(ptr + GPIO_TRI_OFFSET)) == 255;
    value = *((unsigned *)(ptr + GPIO_DATA_OFFSET));
    printf("%s: Input: %08xWn", argv[0], value);

}else{

    *((unsigned *)(ptr + GPIO_TRI_OFFSET)) = 0;
    *((unsigned *)(ptr + GPIO_DATA_OFFSET)) = value;

}

munmap(ptr, GPIO_MAP_SIZE);

return 0;

}
```

3) Device-tree 추가

```
sw -> petalinux-config -c rootfs    ( Kernel 에서 해도 됨.)  
petalinux-config -c kernel  
cd ../hardware/driver_lab.sdk
```

```
petalinux-config --get-hw-description -p ~/~~~[sw location] ( software 위치 )  
Cd ~~[sw location]    ( Kernel에서 밑에 것 수행 )  
petalinux-config  
petalinux-config -c kernel  
petalinux-config -c rootfs
```

```
vi subsystems/linux/configs/device-tree/system-top.dts
```

```
petalinux-build
```

```
cd images/linux
```

```
petalinux-package --boot --fsbl zynq_fsbl.elf --fpga ~/zynq_zybo/~~~/hardware/driver_lab.runs/impl_1/system_wrapper.bit --u-boot --force
```

BOOT에 파일 다시 올리고

putty 들어가고 device_driver -d /dev/uio0 -o -0 , 1

```
/dts-v1/;  
/include/ "system-conf.dtsi"  
/  
{  
};  
  
&clk {  
    ps-clk-frequency = <500000000>;  
};  
  
&flash0{  
    compatible = "s25fl128s1";  
};  
  
&usb0{  
    dr_mode = "otg";  
};  
  
&gem0{  
    phy-handle = <&phy0>;  
    mdio{  
  
#address-cells = <1>;  
#size-cells = <0>;  
    phy0: phy@1{  
        compatible = "realtek,RTL8211E";  
        device_type = "ethernet-phy";  
        reg = <1>;  
    };  
};  
  
&led_pin{  
    compatible = "generic-uio";  
};
```