Xilinx Zynq FPGA, TI DSP, MCU 기반의 프로그래밍 및 회로 설계 전문가 과정

강사 – Innova Lee(이상훈)

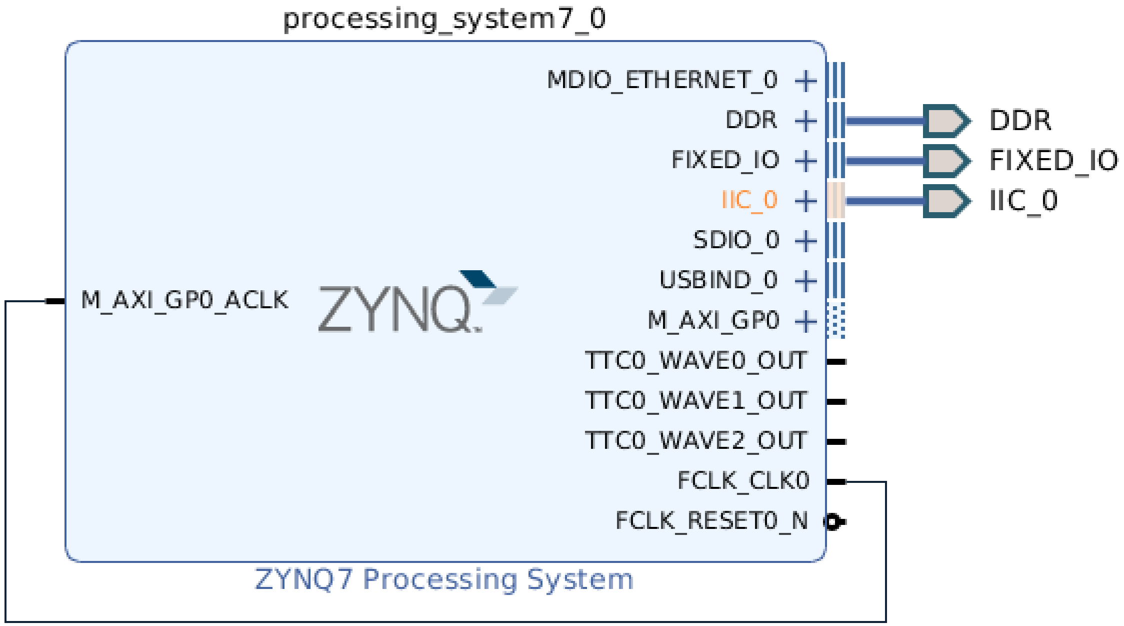
gcccompil3r@gmail.com

학생 – hoseong Lee(이호성)

hslee00001@naver.com

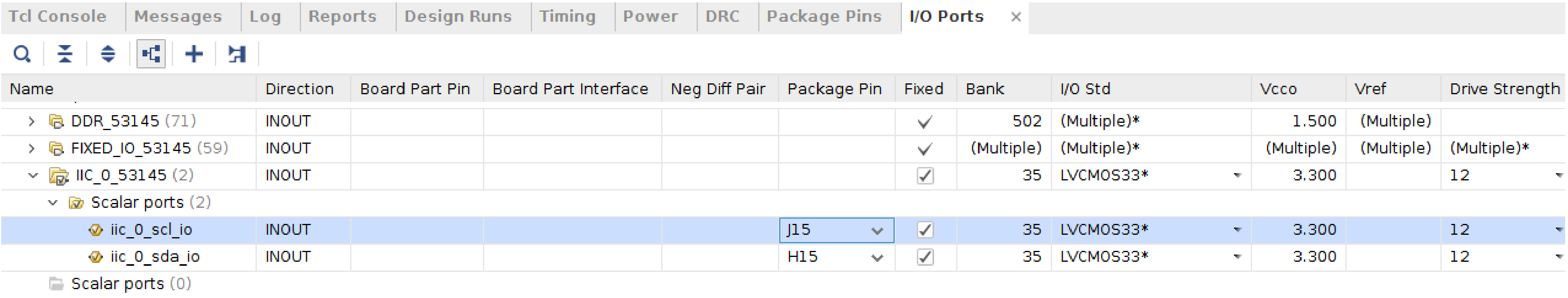
i2c_system 설정

*i2c_proj → lidar (vivado) 생성 → i2c_system (create block design) → zynq 오토메이션후 → re-customize (MIO configuration) → ENET0, USB0, SD0, UART1, I2C0 select 확인 후 나와서 블록에서 → IIC_0 포트 추가 → 그림처럼 ACLK 과 CLK 연결



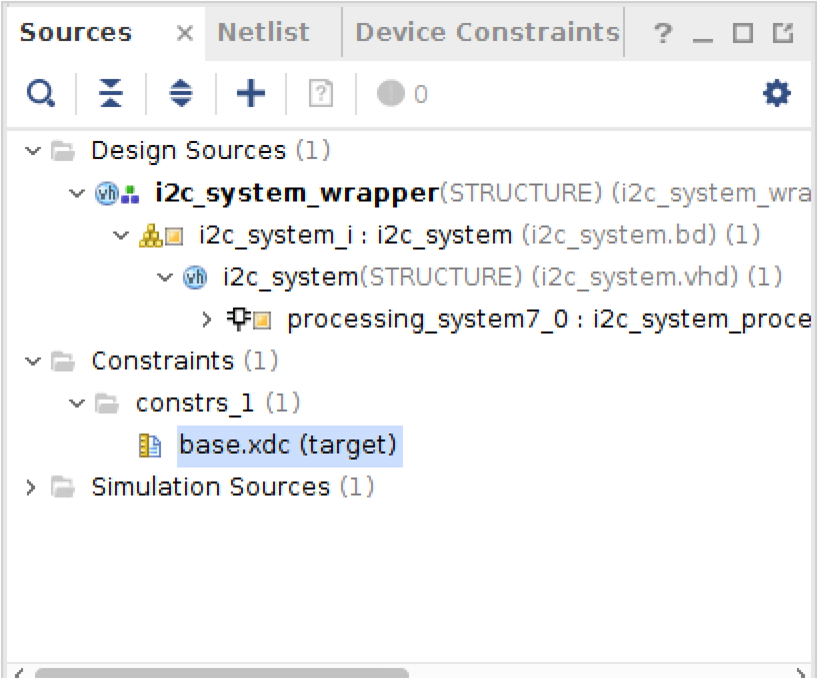
*source → (i2c_system_i) generate output product 눌러 제너레이트 시키고, create HDL wrapper 해서 방금 만든 블럭을 VHDL로 만들어준다.

*Run Synthesized → Run Implementation → I/0 Ports 설정

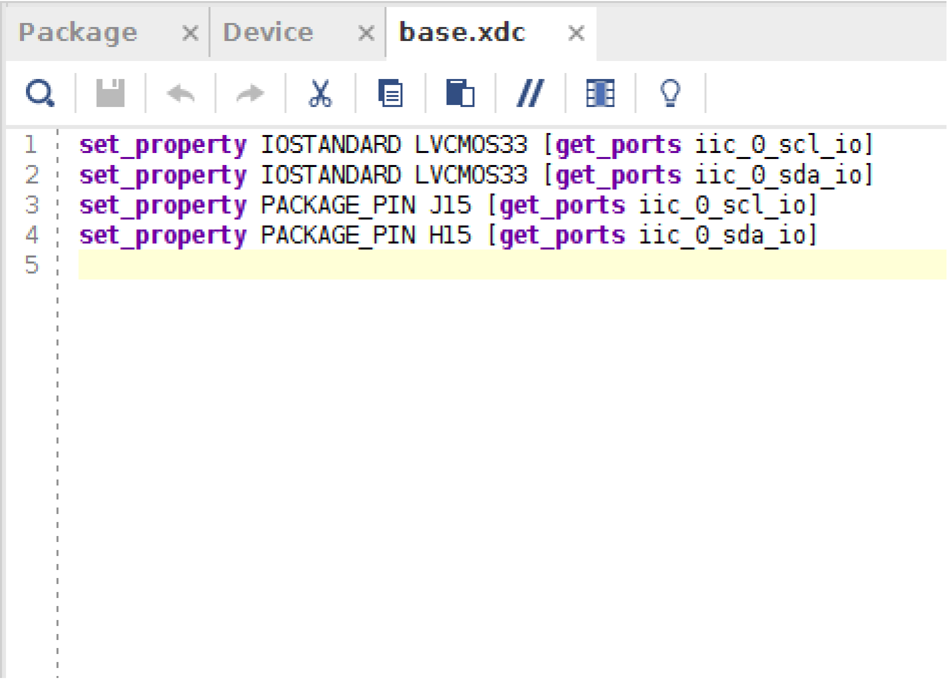


Name	Direction	Board Part Pin	Board Part Interface	Neg Diff Pair	Package Pin	Fixed	Bank	I/O Std	Vcco	Vref	Drive Strength
DDR_53145 (71)	INOUT					✓	502	(Multiple)*	1.500	(Multiple)	
FIXED_IO_53145 (59)	INOUT					✓	(Multiple)	(Multiple)*	(Multiple)	(Multiple)	(Multiple)*
IIC_0_53145 (2)	INOUT					☑	35	LVCMOS33*	3.300		12
Scalar ports (2)											
iic_0_scl_io	INOUT				J15	☑	35	LVCMOS33*	3.300		12
iic_0_sda_io	INOUT				H15	☑	35	LVCMOS33*	3.300		12
Scalar ports (0)											

*Constraints → Add or create constraints → base.xdc → I/O Ports 로 가서 control+s 로 저장하면 base.xdc에 자동으로 핀설정이 완료됨

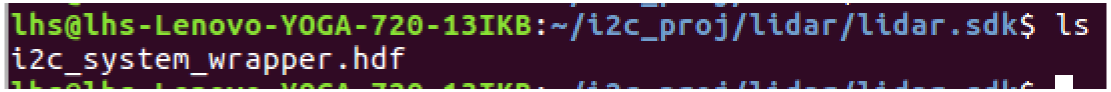


Design Sources (1)
i2c_system_wrapper(STRUCTURE) (i2c_system_wra
i2c_system_i : i2c_system (i2c_system.bd) (1)
i2c_system(STRUCTURE) (i2c_system.vhd) (1)
processing_system7_0 : i2c_system_proce
Constraints (1)
constrs_1 (1)
base.xdc (target)
Simulation Sources (1)



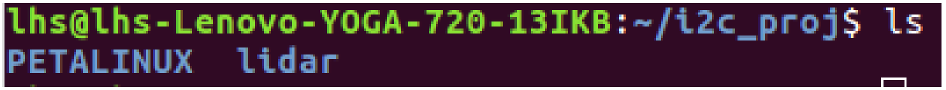
```
set_property IOSTANDARD LVCMOS33 [get_ports iic_0_scl_io]
set_property IOSTANDARD LVCMOS33 [get_ports iic_0_sda_io]
set_property PACKAGE_PIN J15 [get_ports iic_0_scl_io]
set_property PACKAGE_PIN H15 [get_ports iic_0_sda_io]

```

* Generate Bitstream → File → Export → Export Hardware → include bitstream 체크하고 확인 → 그럼 sdk 파일에 hdf 파일이 생성된 것을 볼 수 있다.
(vivado 에서 hw 설계한 내용에 대한 정보들이 이 파일안에 있다.)



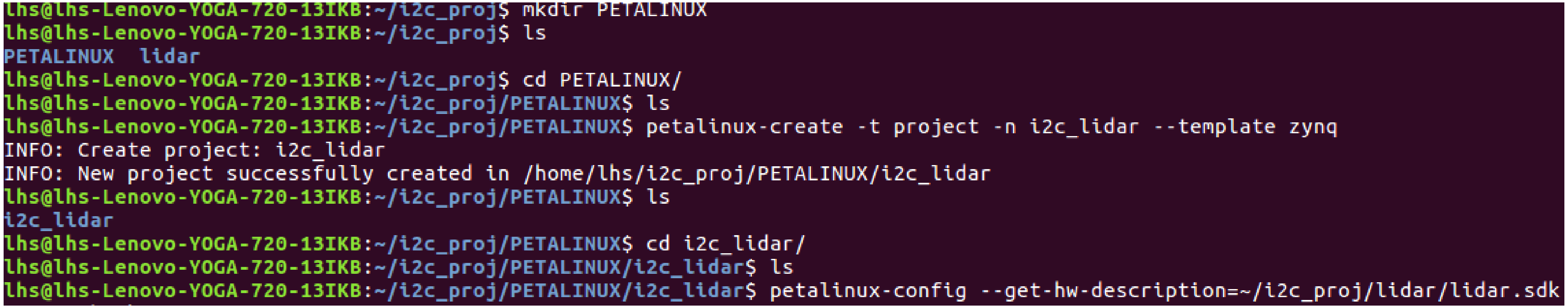
```
lhs@lhs-Lenovo-YOGA-720-13IKB:~/i2c_proj/lidar/lidar.sdk$ ls
i2c_system_wrapper.hdf
lhs@lhs-Lenovo-YOGA-720-13IKB:~/i2c_proj/lidar/lidar.sdk$
```

* 프로젝트만든 상위 디렉터리로 가서 PETALINUX 폴더를 만들어준다.



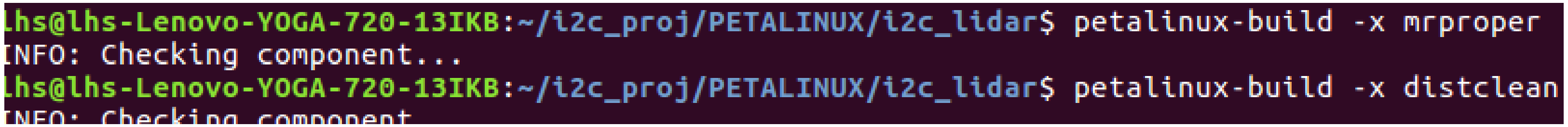
```
lhs@lhs-Lenovo-YOGA-720-13IKB:~/i2c_proj$ ls
PETALINUX  lidar
```

* 다음을 따라한다.



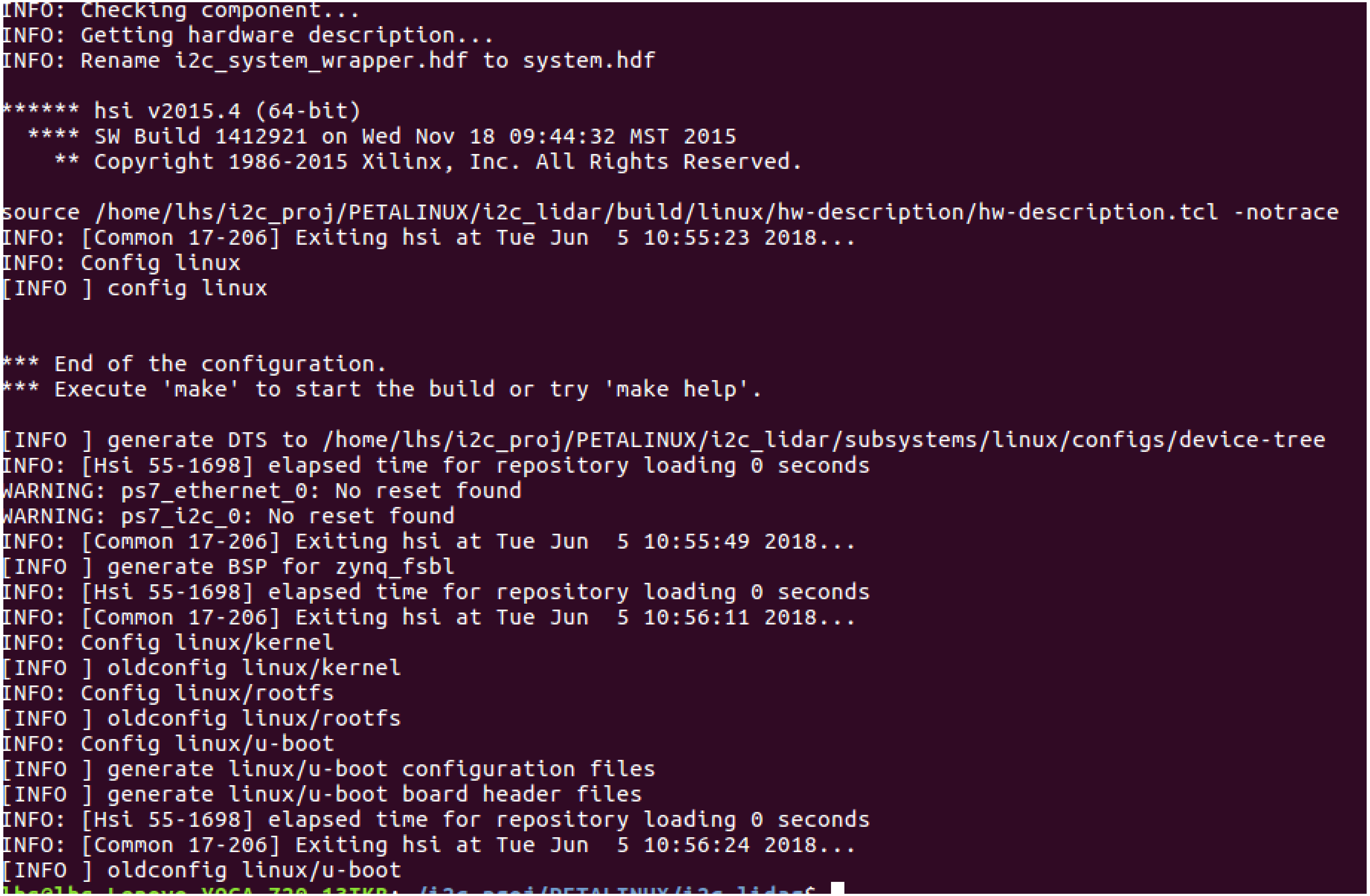
```
lhs@lhs-Lenovo-YOGA-720-13IKB:~/i2c_proj$ mkdir PETALINUX
lhs@lhs-Lenovo-YOGA-720-13IKB:~/i2c_proj$ ls
PETALINUX  lidar
lhs@lhs-Lenovo-YOGA-720-13IKB:~/i2c_proj$ cd PETALINUX/
lhs@lhs-Lenovo-YOGA-720-13IKB:~/i2c_proj/PETALINUX$ ls
lhs@lhs-Lenovo-YOGA-720-13IKB:~/i2c_proj/PETALINUX$ petalinux-create -t project -n i2c_lidar --template zynq
INFO: Create project: i2c_lidar
INFO: New project successfully created in /home/lhs/i2c_proj/PETALINUX/i2c_lidar
lhs@lhs-Lenovo-YOGA-720-13IKB:~/i2c_proj/PETALINUX$ ls
i2c_lidar
lhs@lhs-Lenovo-YOGA-720-13IKB:~/i2c_proj/PETALINUX$ cd i2c_lidar/
lhs@lhs-Lenovo-YOGA-720-13IKB:~/i2c_proj/PETALINUX/i2c_lidar$ ls
lhs@lhs-Lenovo-YOGA-720-13IKB:~/i2c_proj/PETALINUX/i2c_lidar$ petalinux-config --get-hw-description=~/i2c_proj/lidar/lidar.sdk
```

* 파란화면이 뜨는데, 설정을 save → 확인 하고 , exit로 나가면 된다. 만약 설정을 하지 않고 exit 로 그냥 나갔다면



```
lhs@lhs-Lenovo-YOGA-720-13IKB:~/i2c_proj/PETALINUX/i2c_lidar$ petalinux-build -x mrproper
INFO: Checking component...
lhs@lhs-Lenovo-YOGA-720-13IKB:~/i2c_proj/PETALINUX/i2c_lidar$ petalinux-build -x distclean
INFO: Checking component
```

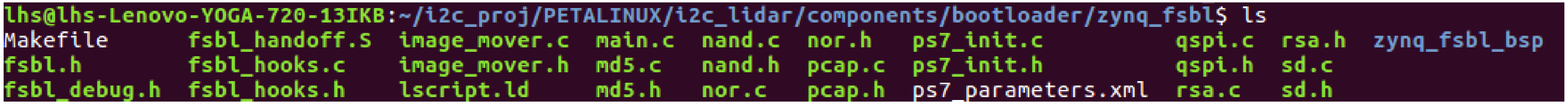
다시 설정해주기 위해 위에 명령어로 기존것을 지우고 다시 하면 된다. Petalinux-config --~~~~~ 하면된다.



```
INFO: Checking component...
INFO: Getting hardware description...
INFO: Rename i2c_system_wrapper.hdf to system.hdf

****** hsi v2015.4 (64-bit)
  **** SW Build 1412921 on Wed Nov 18 09:44:32 MST 2015
    ** Copyright 1986-2015 Xilinx, Inc. All Rights Reserved.

source /home/lhs/i2c_proj/PETALINUX/i2c_lidar/build/linux/hw-description/hw-description.tcl -notrace
INFO: [Common 17-206] Exiting hsi at Tue Jun  5 10:55:23 2018...
INFO: Config linux
[INFO ] config linux


*** End of the configuration.
*** Execute 'make' to start the build or try 'make help'.

[INFO ] generate DTS to /home/lhs/i2c_proj/PETALINUX/i2c_lidar/subsystems/linux/configs/device-tree
INFO: [Hsi 55-1698] elapsed time for repository loading 0 seconds
WARNING: ps7_ethernet_0: No reset found
WARNING: ps7_i2c_0: No reset found
INFO: [Common 17-206] Exiting hsi at Tue Jun  5 10:55:49 2018...
[INFO ] generate BSP for zynq_fsbl
INFO: [Hsi 55-1698] elapsed time for repository loading 0 seconds
INFO: [Common 17-206] Exiting hsi at Tue Jun  5 10:56:11 2018...
INFO: Config linux/kernel
[INFO ] oldconfig linux/kernel
INFO: Config linux/rootfs
[INFO ] oldconfig linux/rootfs
INFO: Config linux/u-boot
[INFO ] generate linux/u-boot configuration files
[INFO ] generate linux/u-boot board header files
INFO: [Hsi 55-1698] elapsed time for repository loading 0 seconds
INFO: [Common 17-206] Exiting hsi at Tue Jun  5 10:56:24 2018...
[INFO ] oldconfig linux/u-boot
lhs@lhs-Lenovo-YOGA-720-13IKB:~/i2c_proj/PETALINUX/i2c_lidar$ 
```

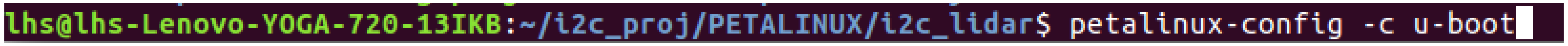
* 설정 완료 후 u-boot, root파일 시스템이 만들어 진다.

* 다음 디렉터리에 cortex-A9의 부트코드를 확인할 수 있다. 필요시 해석하면 된다...고 한다...



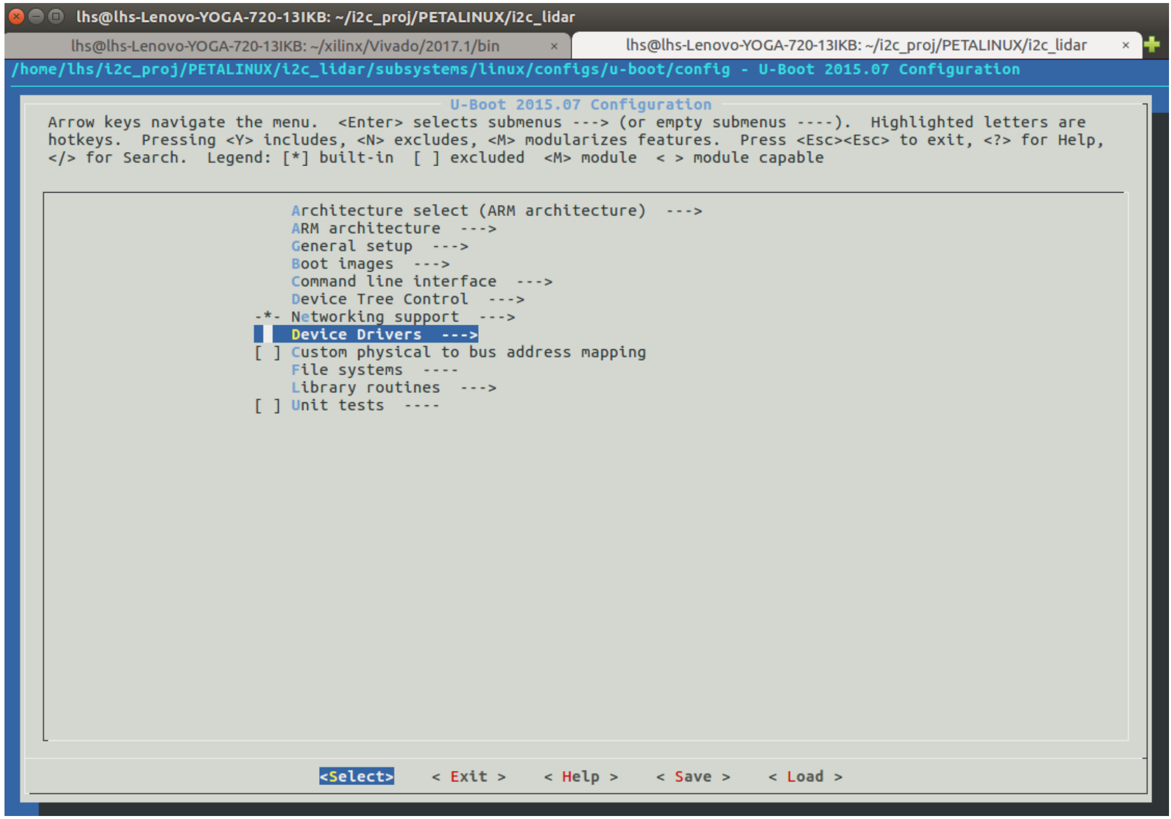
```
lhs@lhs-Lenovo-YOGA-720-13IKB:~/i2c_proj/PETALINUX/i2c_lidar/components/bootloader/zynq_fsbl$ ls
Makefile      fsbl_handoff.S  image_mover.c  main.c  nand.c  nor.h   ps7_init.c           qspi.c  rsa.h  zynq_fsbl_bsp
fsbl.h        fsbl_hooks.c    image_mover.h  md5.c   nand.h  pcap.c  ps7_init.h           qspi.h  sd.c
fsbl_debug.h  fsbl_hooks.h    lscript.ld     md5.h   nor.c   pcap.h  ps7_parameters.xml   rsa.c   sd.h
```

* 다음 디렉터리에서 명령어(petalinux-config -c u-boot) 를 입력한다



```
lhs@lhs-Lenovo-YOGA-720-13IKB:~/i2c_proj/PETALINUX/i2c_lidar$ petalinux-config -c u-boot
```

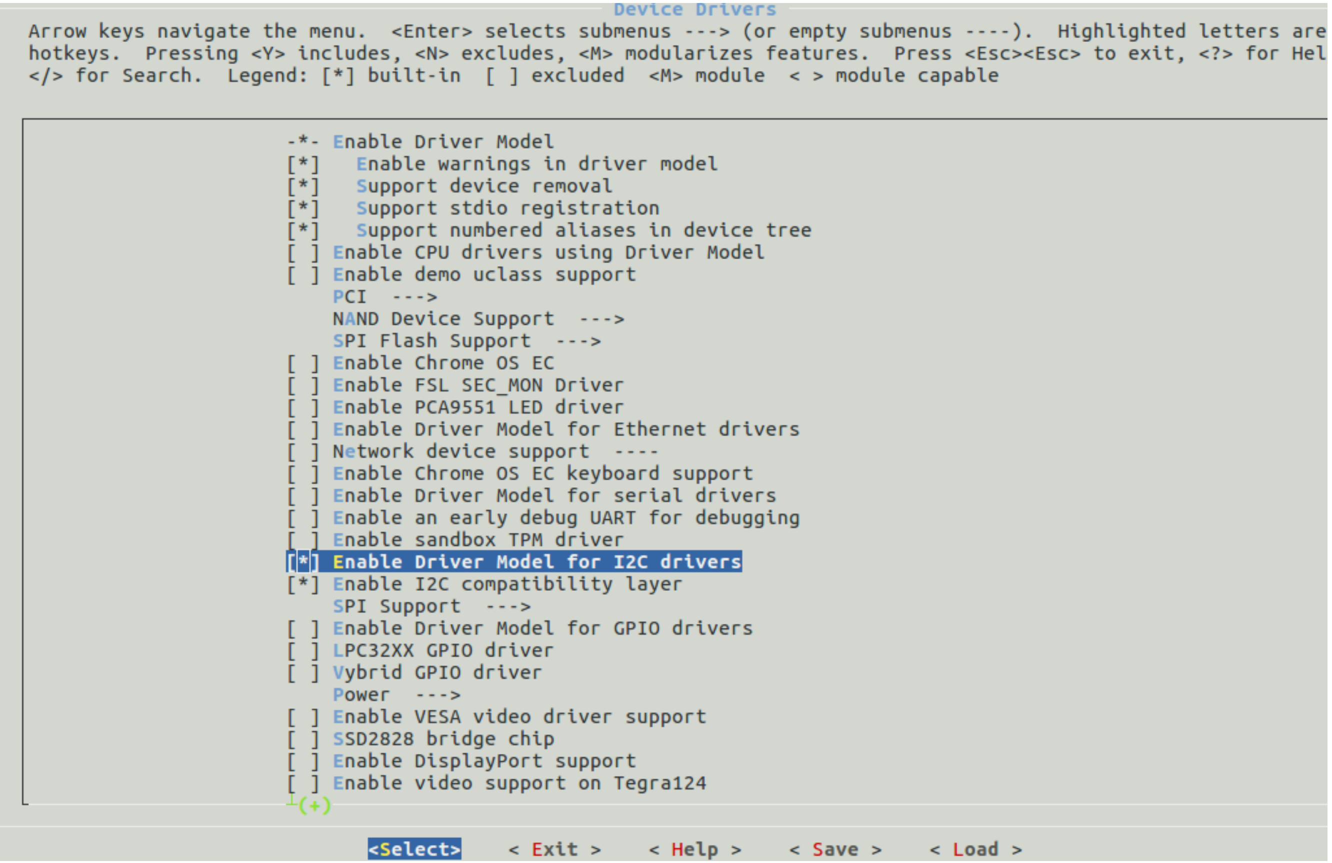
* 설정창이 나오는데 Device Drivers 선택한다.



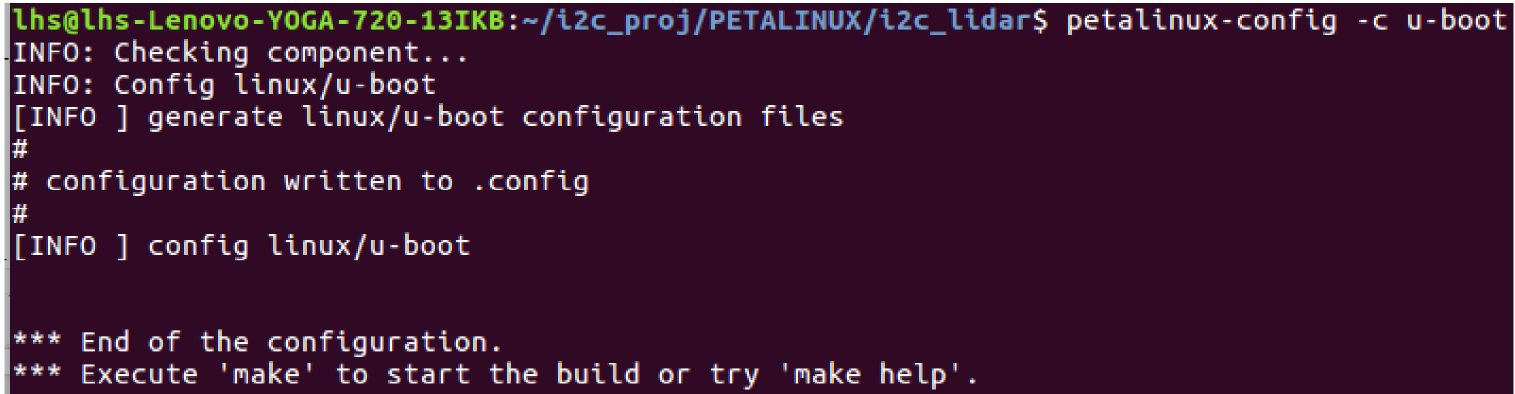
```
lhs@lhs-Lenovo-YOGA-720-13IKB: ~/i2c_proj/PETALINUX/i2c_lidar
lhs@lhs-Lenovo-YOGA-720-13IKB: ~/xilinx/Vivado/2017.1/bin   ×   lhs@lhs-Lenovo-YOGA-720-13IKB: ~/i2c_proj/PETALINUX/i2c_lidar   ×
/home/lhs/i2c_proj/PETALINUX/i2c_lidar/subsystems/linux/configs/u-boot/config - U-Boot 2015.07 Configuration
                              U-Boot 2015.07 Configuration
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus ----).  Highlighted letters are
  hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes features.  Press <Esc><Esc> to exit, <?> for Help,
  </> for Search.  Legend: [*] built-in  [ ] excluded  <M> module  < > module capable

                          Architecture select (ARM architecture)  --->
                          ARM architecture  --->
                          General setup  --->
                          Boot images  --->
                          Command line interface  --->
                          Device Tree Control  --->
                      -*- Networking support  --->
                          Device Drivers  --->
                      [ ] Custom physical to bus address mapping
                          File systems  ----
                          Library routines  --->
                      [ ] Unit tests  ----

                  <Select>    < Exit >    < Help >    < Save >    < Load >
```

드

* 다음 i2c driverd와 i2c compatibility layer을 y 로 체크후 **save** 를 한 후 exit 하여 나간다. (save 필수)



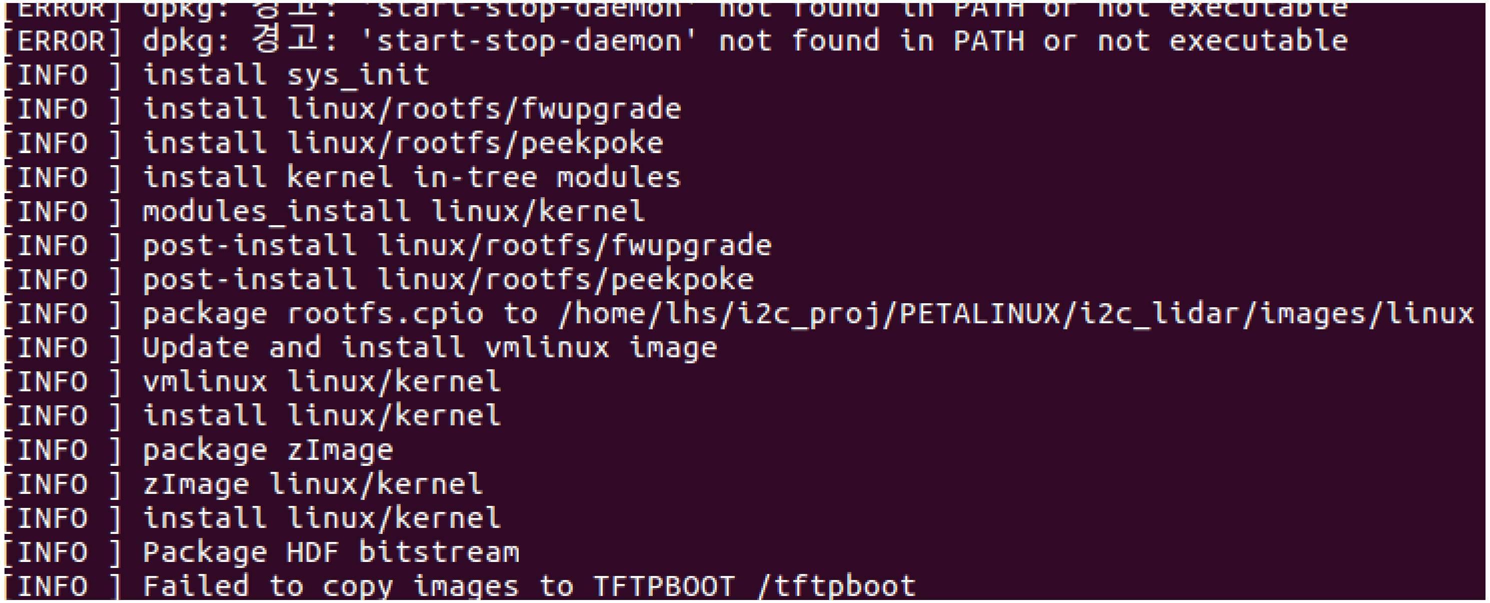
```
                                  Device Drivers
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus ----).  Highlighted letters are
  hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes features.  Press <Esc><Esc> to exit, <?> for Hel
  </> for Search.  Legend: [*] built-in  [ ] excluded  <M> module  < > module capable

                        -*- Enable Driver Model
                        [*]   Enable warnings in driver model
                        [*]   Support device removal
                        [*]   Support stdio registration
                        [*]   Support numbered aliases in device tree
                        [ ] Enable CPU drivers using Driver Model
                        [ ] Enable demo uclass support
                            PCI  --->
                            NAND Device Support  --->
                            SPI Flash Support  --->
                        [ ] Enable Chrome OS EC
                        [ ] Enable FSL SEC_MON Driver
                        [ ] Enable PCA9551 LED driver
                        [ ] Enable Driver Model for Ethernet drivers
                        [ ] Network device support  ----
                        [ ] Enable Chrome OS EC keyboard support
                        [ ] Enable Driver Model for serial drivers
                        [ ] Enable an early debug UART for debugging
                        [ ] Enable sandbox TPM driver
                        [*] Enable Driver Model for I2C drivers
                        [*] Enable I2C compatibility layer
                            SPI Support  --->
                        [ ] Enable Driver Model for GPIO drivers
                        [ ] LPC32XX GPIO driver
                        [ ] Vybrid GPIO driver
                            Power  --->
                        [ ] Enable VESA video driver support
                        [ ] SSD2828 bridge chip
                        [ ] Enable DisplayPort support
                        [ ] Enable video support on Tegra124
                        └(+)

                    <Select>    < Exit >    < Help >    < Save >    < Load >
```



```
lhs@lhs-Lenovo-YOGA-720-13IKB:~/i2c_proj/PETALINUX/i2c_lidar$ petalinux-config -c u-boot
INFO: Checking component...
INFO: Config linux/u-boot
[INFO ] generate linux/u-boot configuration files
#
# configuration written to .config
#
[INFO ] config linux/u-boot


*** End of the configuration.
*** Execute 'make' to start the build or try 'make help'.
```

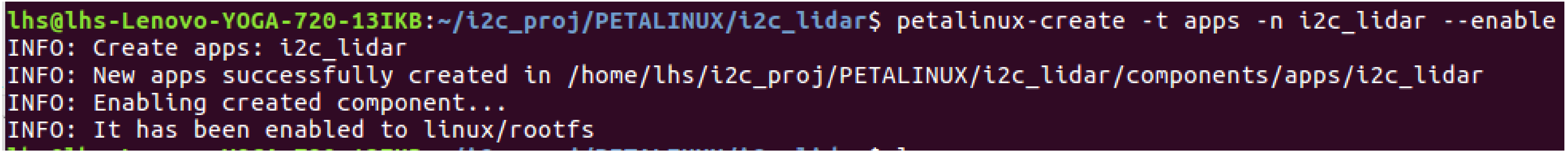
* petalinux-config -c u-boot 명령어를 친다. → configuration 완료

* petalinux-build 명령어를 친다. Build 완료 (dpkg: 경고 → 고치려고 하지마라.. 리눅스 날라감..)



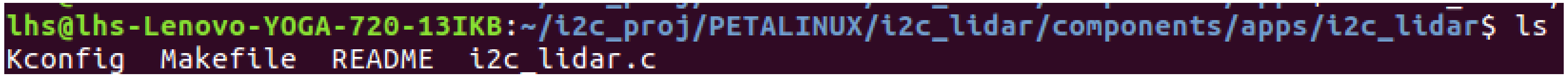
```
[ERROR] dpkg: 경고: 'start-stop-daemon' not found in PATH or not executable
[ERROR] dpkg: 경고: 'start-stop-daemon' not found in PATH or not executable
[INFO ] install sys_init
[INFO ] install linux/rootfs/fwupgrade
[INFO ] install linux/rootfs/peekpoke
[INFO ] install kernel in-tree modules
[INFO ] modules_install linux/kernel
[INFO ] post-install linux/rootfs/fwupgrade
[INFO ] post-install linux/rootfs/peekpoke
[INFO ] package rootfs.cpio to /home/lhs/i2c_proj/PETALINUX/i2c_lidar/images/linux
[INFO ] Update and install vmlinux image
[INFO ] vmlinux linux/kernel
[INFO ] install linux/kernel
[INFO ] package zImage
[INFO ] zImage linux/kernel
[INFO ] install linux/kernel
[INFO ] Package HDF bitstream
[INFO ] Failed to copy images to TFTPBOOT /tftpboot
```

* petalinux-create -t apps -n i2c_lidar –enable 명령어를 친다. (HW 를 제어하기 위한 SW 코드인 Device Driver 코드를 작성할 수 있다.)



```
lhs@lhs-Lenovo-YOGA-720-13IKB:~/i2c_proj/PETALINUX/i2c_lidar$ petalinux-create -t apps -n i2c_lidar --enable
INFO: Create apps: i2c_lidar
INFO: New apps successfully created in /home/lhs/i2c_proj/PETALINUX/i2c_lidar/components/apps/i2c_lidar
INFO: Enabling created component...
INFO: It has been enabled to linux/rootfs
```

* vi i2c_lidar.c



```
lhs@lhs-Lenovo-YOGA-720-13IKB:~/i2c_proj/PETALINUX/i2c_lidar/components/apps/i2c_lidar$ ls
Kconfig  Makefile  README  i2c_lidar.c
```

* 다음 코드 작성

```
/*
 * Placeholder PetaLinux user application.
 *
 * Replace this with your application code
 */
#include <stdio.h>
#include <linux/i2c.h>
#include <linux/i2c-dev.h>
#include <sys/ioctl.h>
#include <fcntl.h>
#include <unistd.h>
#include <errno.h>
#include <string.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/stat.h>

#define I2C_FILE_NAME_0         "/dev/i2c-0"
#define I2C_FILE_NAME_1         "/dev/i2c-1"
#define LIDAR_SLAVE_ADDR        0x62

#define ACQ_COMMAND     0x00
#define STATUS          0x01
#define SIG_COUNT_VAL   0x02
#define ACQ_CONFIG_REG          0x04
#define THRESHOLD_BYPASS        0x1c
#define RAED_FROM       0x89

#define NO_CORRECTION 0
#define CORRECTION      1
```

```
#define AR_VELOCITY      0
#define AR_PEAK_CORR   1
#define AR_NOISE_PEAK  2
#define AR_SIGNAL_STRENGTH 3
#define AR_FULL_DELAY_HIGH 4
#define AR_FULL_DELAY_LOW  5

#define OUTPUT_OF_ALL  0
#define DISTANCE_ONLY  1
#define DISTANCE_WITH_VELO 2
#define VELOCITY_ONLY  3

#define USAGE         "i2c_LiDAR_fn <OUTPUT_OPTIONS> <I2C_DEVICE_NUMBER>\n"\
        "<OUTPUT_OPTIONS>\n"\
"0 : output of all\n"\
"1 : distance only\n"\
"2 : distance with velocity\n"\
"3 : velocity only\n"\

unsigned get_status();
void i_read(unsigned char, unsigned, unsigned char*);
void i_write(unsigned char reg, unsigned char value);
void measurement(unsigned char, unsigned char, unsigned char*);
void display(unsigned char, unsigned char*);

int fd = 0;

int main(int argc, char *argv[])
{
        unsigned char receives[8] = {AR_VELOCITY, 0, 0, AR_PEAK_CORR, AR_NOISE_PEAK, AR_SIGNAL_STRENGTH,
AR_FULL_DELAY_HIGH, AR_FULL_DELAY_LOW};
        unsigned char i, options;
```

```
char* file_name = NULL;

if(argc < 2) printf("%s\n",USAGE);
else if(argc > 2 && atoi(argv[2])) file_name = I2C_FILE_NAME_1;
else file_name = I2C_FILE_NAME_0;

options = atoi(argv[1]);

if((fd = open(file_name, O_RDWR)) < 0)
{
        perror("--OPEN DEVICE ERROR ");
        return -1;
}
if(ioctl(fd, I2C_SLAVE, LIDAR_SLAVE_ADDR) < 0)
{
        perror("---SLAVE ADDR CONNECT ERROR ");
        return -1;
}
i_write(SIG_COUNT_VAL, 0x80);
i_write(ACQ_CONFIG_REG, 0x68);
i_write(THRESHOLD_BYPASS, 0x00);

while(1)
{
        measurement(CORRECTION, options ,receives);
        for(i=0; i<99; i++)
                measurement(NO_CORRECTION, options, receives);
}

close(fd);

return 0;
```

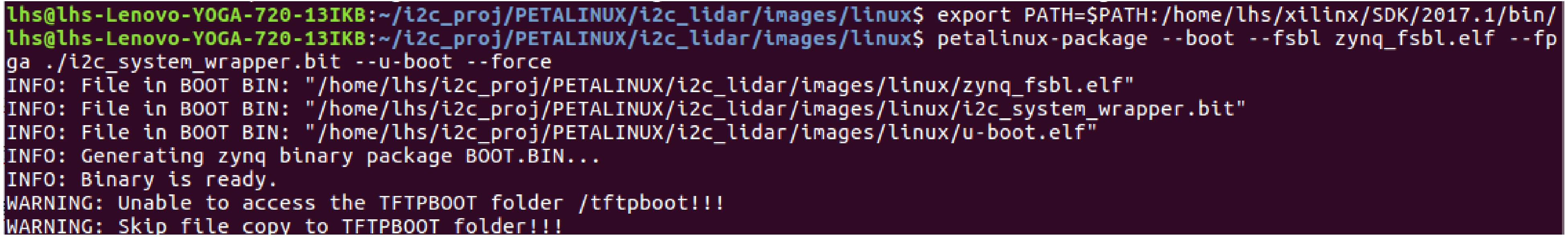
```
}

unsigned get_status()
{
    unsigned char buf[1] = {STATUS};

    if(write(fd, buf, 1) != 1)
    {
        perror("---WRITE REGISTER ERROR ");
        return -1;
    }
    if(read(fd, buf, 1) != 1)
    {
        perror("---WRITE REGISTER ERROR ");
        return -1;
    }
    return buf[0] & 0x01;
}

void i_read(unsigned char reg, unsigned read_size, unsigned char *receives)
{
    unsigned char buf[1] = {reg};
    unsigned busy_flag = 1, busy_counter = 0;

    while(busy_flag)
    {
        busy_flag = get_status();
        busy_counter++;
        if(busy_counter > 9999)
        {
            printf("BUSY COUNT TIME OUT !\n");
            return ;
```

```
                }
        }
        if(!busy_flag)
        {
                if(write(fd, buf, 1) != 1)
                {
                        perror("---WRITE REGISTER ERROR ");
                        return -1;
                }
                if(read(fd, receives, read_size) != read_size)
                {
                        perror("---WRITE REGISTER ERROR ");
                        return -1;
                }
        }
}
void i_write(unsigned char reg, unsigned char value)
{
        unsigned char buf[2] = {reg, value};

        if(write(fd, buf, 2) != 2)
        {
                perror("---WRITE REGISTER ERROR ");
                return -1;
        }
        usleep(1000);
}

void measurement(unsigned char is_correction, unsigned char options, unsigned char *buf)
{
        unsigned char i;
        if(is_correction)
        i_write(ACQ_COMMAND, 0x04);
    else
        i_write(ACQ_COMMAND, 0x03);

    i_read(RAED_FROM, 8, buf);

    for(i=1; i<6; i++)
        buf[i] = buf[i + 2];

    display(options, buf);
}

void display(unsigned char options, unsigned char *buf)
{
    unsigned char i;
    char* strings[5] = {"Velocity", "Peak value in correlation record", "Correlation record noise floor", "Received signal strength", "Distance"};
    buf[AR_FULL_DELAY_HIGH] = buf[AR_FULL_DELAY_HIGH] << 8 | buf[AR_FULL_DELAY_LOW];

    switch(options)
    {
        case OUTPUT_OF_ALL :
            for(i=0;i<5;i++)
                printf("%s \t\t\t\t = %d\n",strings[i], buf[i]);
            break;
        case DISTANCE_ONLY :
            printf("%s \t\t\t\t = %d\n",strings[4], buf[AR_FULL_DELAY_HIGH]);
            break;
        case DISTANCE_WITH_VELO :
            printf("%s \t\t\t\t = %d\n",strings[0], buf[AR_VELOCITY]);
            printf("%s \t\t\t\t = %d\n",strings[4], buf[AR_FULL_DELAY_HIGH]);
            break;
        case VELOCITY_ONLY :
```

```
                        printf("%s \t\t\t\t = %d\n",strings[0], buf[AR_VELOCITY]);
                        break;
        }
        printf("\n");
}
```

* Cd ../../.. → i2c_lider/image/linux 디렉터리로 가서 명령어; nautilus ./ 로 현재 폴더를 열어 준다.
* petalinux-build

* export PATH=$PATH:/home/lhs/xilinx/SDK/2017.1/bin/
* petalinux-package --boot --fsbl zynq_fsbl.elf --fpga ./i2c_system_wrapper.bit --u-boot -–force



```
lhs@lhs-Lenovo-YOGA-720-13IKB:~/i2c_proj/PETALINUX/i2c_lidar/images/linux$ export PATH=$PATH:/home/lhs/xilinx/SDK/2017.1/bin/
lhs@lhs-Lenovo-YOGA-720-13IKB:~/i2c_proj/PETALINUX/i2c_lidar/images/linux$ petalinux-package --boot --fsbl zynq_fsbl.elf --fp
ga ./i2c_system_wrapper.bit --u-boot --force
INFO: File in BOOT BIN: "/home/lhs/i2c_proj/PETALINUX/i2c_lidar/images/linux/zynq_fsbl.elf"
INFO: File in BOOT BIN: "/home/lhs/i2c_proj/PETALINUX/i2c_lidar/images/linux/i2c_system_wrapper.bit"
INFO: File in BOOT BIN: "/home/lhs/i2c_proj/PETALINUX/i2c_lidar/images/linux/u-boot.elf"
INFO: Generating zynq binary package BOOT.BIN...
INFO: Binary is ready.
WARNING: Unable to access the TFTPBOOT folder /tftpboot!!!
WARNING: Skip file copy to TFTPBOOT folder!!!
```

→ Vivado 에서 설계한 HW 와 관련한 비트스트림 정보가 여기에 있다.
→ 예로 비트스트림 파일명이 test_wrapper.bit 라면 ./비트스트림은 ./test_wrapper.bit 로 변경되어야 한다.

* sd card에 BOOT.BIN, image.ub 를 넣는다.(http://cafe.naver.com/hestit 선생님 사이트 sd카드 설정방법)

* 여기서 보드를 컴퓨터와 연결한다. 이 때 board에 전원을 usb모드로, sd 카드 모드로 설정한다.

* 터미널창에 dmesg | grep ttyUSB 명령어를 쳐서 맨마지막에 나온 usb 인식 번호를 확인한다.

* putty 실행. TtyUSB1 , 속도 115200

* 부팅 완료후 로그인 : 아이디: root 비밀번호: root

* i2c_lidar 1 명령어를 친다.

*lidar 사용시 주의사항 : 반드시 아두이노 5V 로 전원을 넣고 콘덴서 680uF 연결위치 확인, Lidar 배선주의 ,사람눈에 절대 쏘지말것.

– c update -

software/component/apps 폴더 application c 업로드.
lab/software/image/linux 에서

```
petalinux-build -x distclean
```

```
petalinux-build -x mrproper
```

```
petalinux-build
```

cd /images/linux 에서

```
petalinux-build
```

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