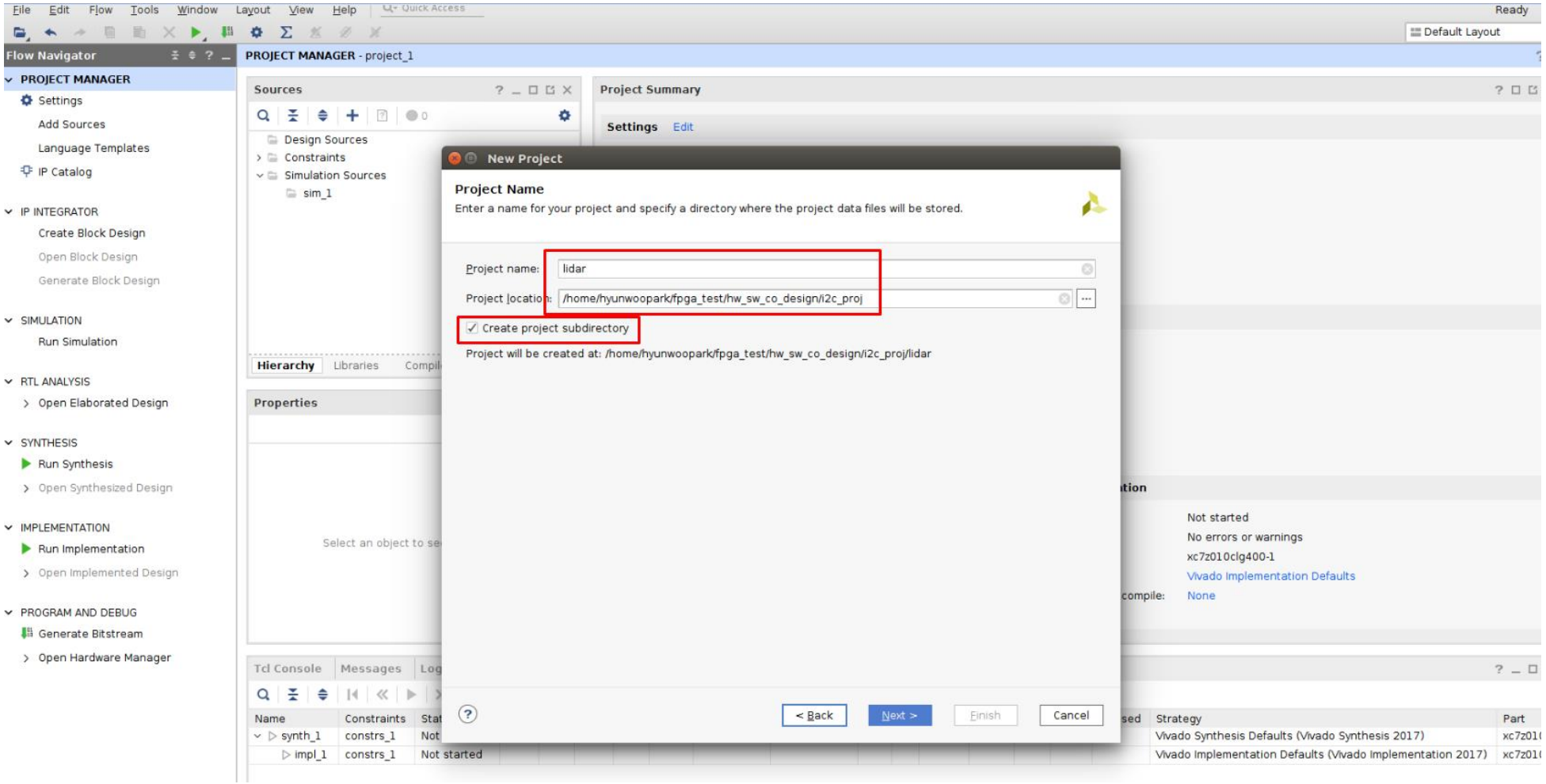
TI DSP, MCU 및 Xilinx Zynq FPGA 프로그래밍 전문가 과정

강사 – Innova Lee(이상훈)
gcccompil3r@gmail.com
학생 – GJ (박현우)
uc820@naver.com

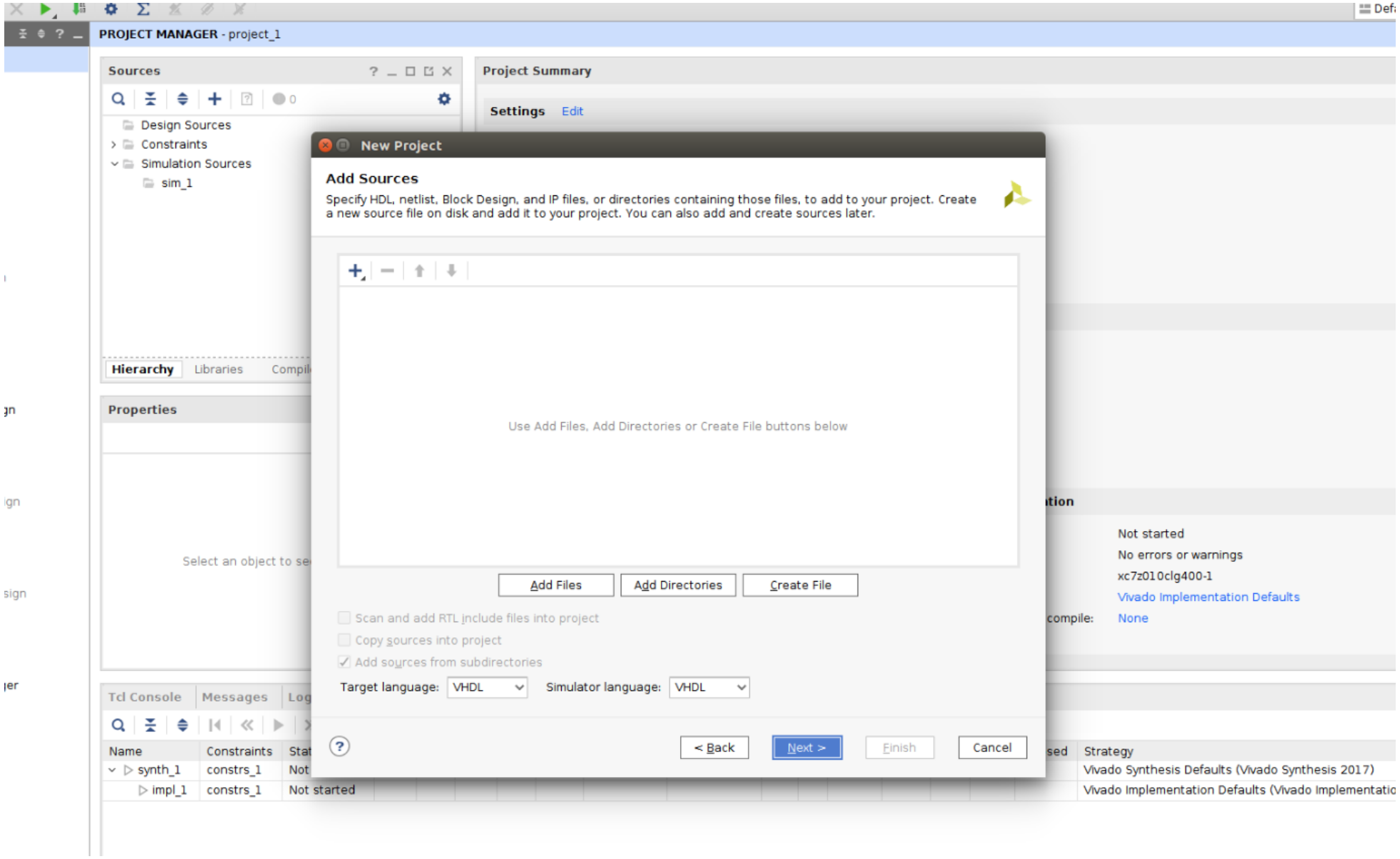
목차

Zynq FPGA on Petalinux for PWM

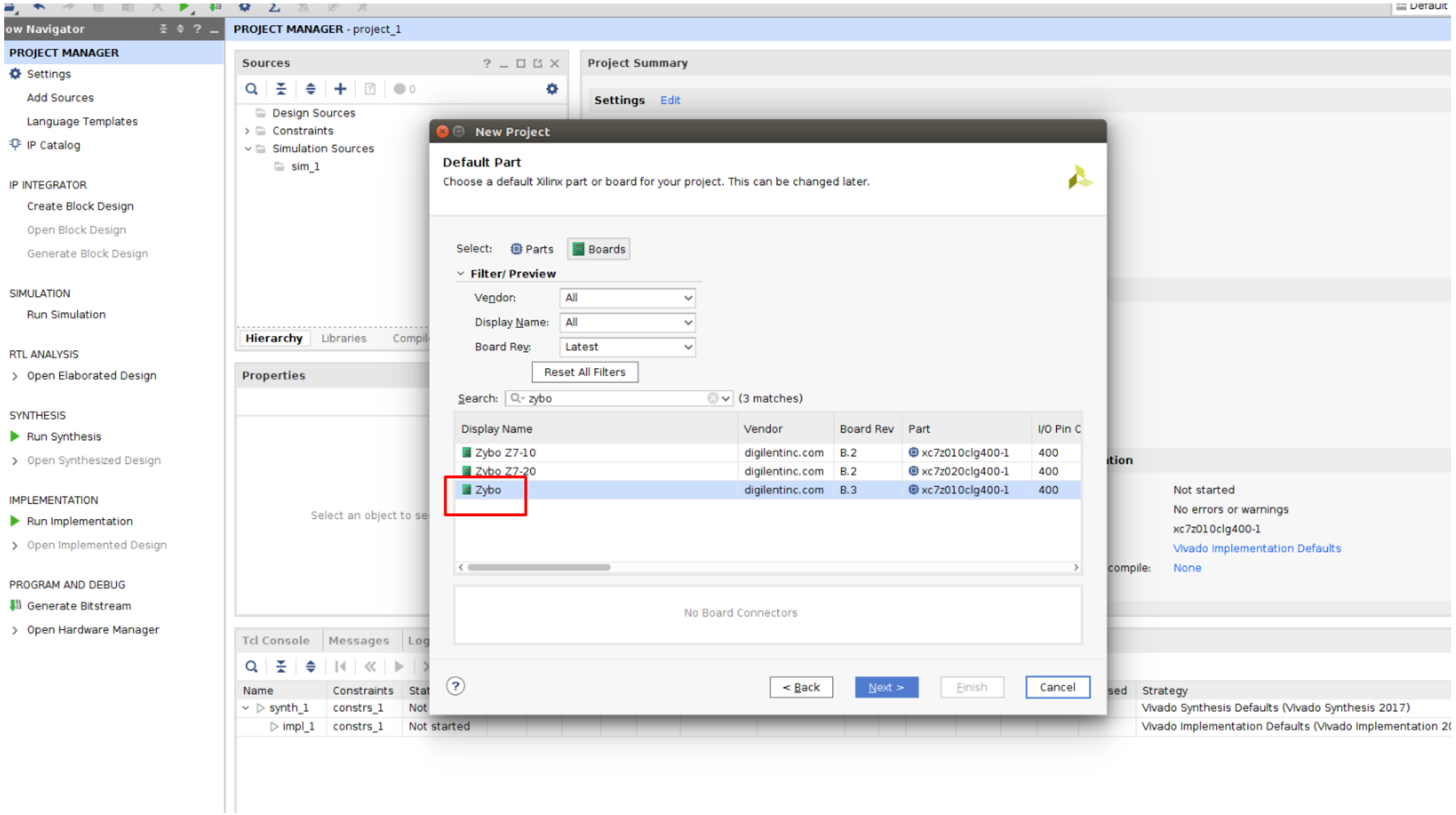
1) Vivado I2C 설계 1



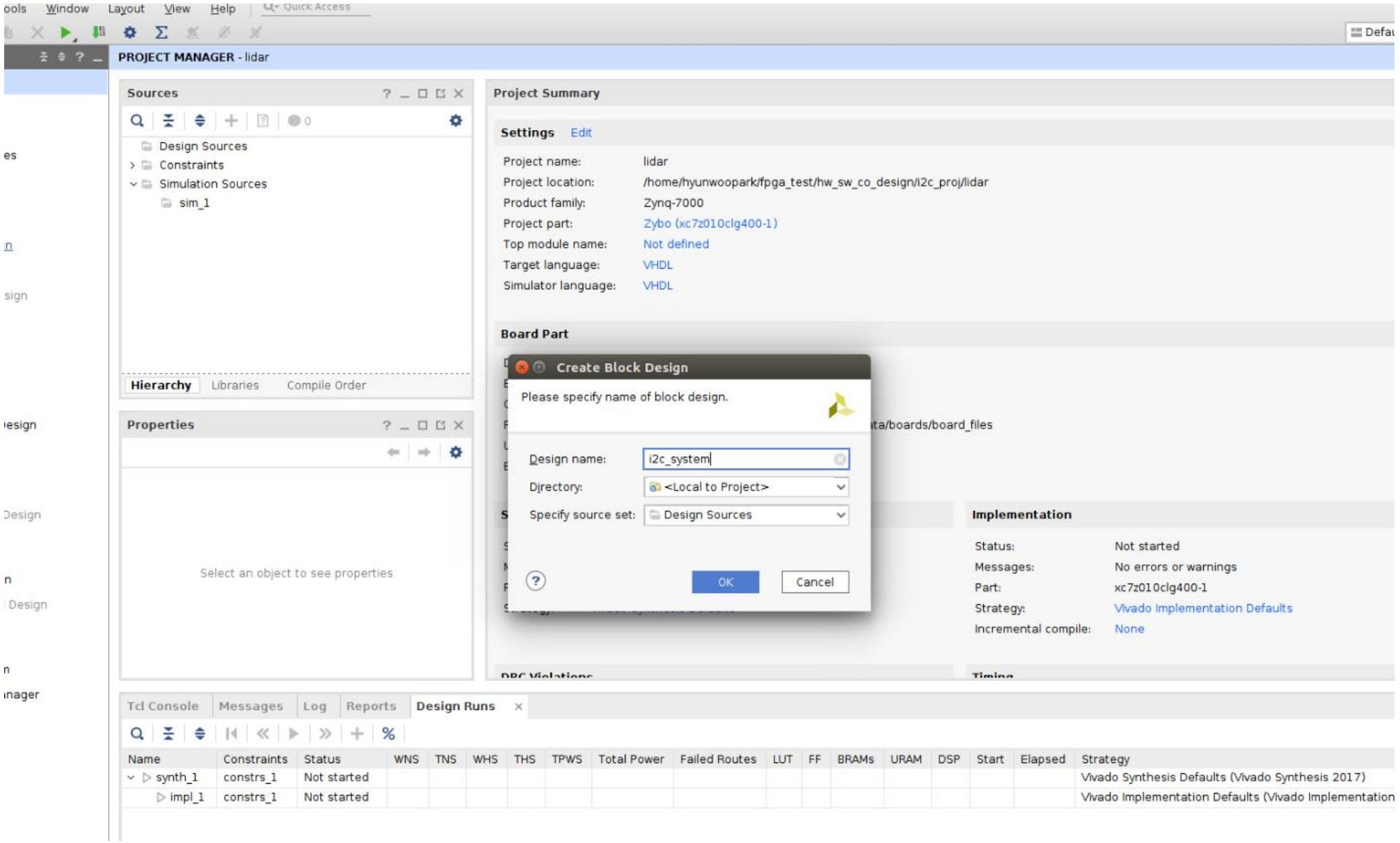
1) Vivado I2C 설계 2



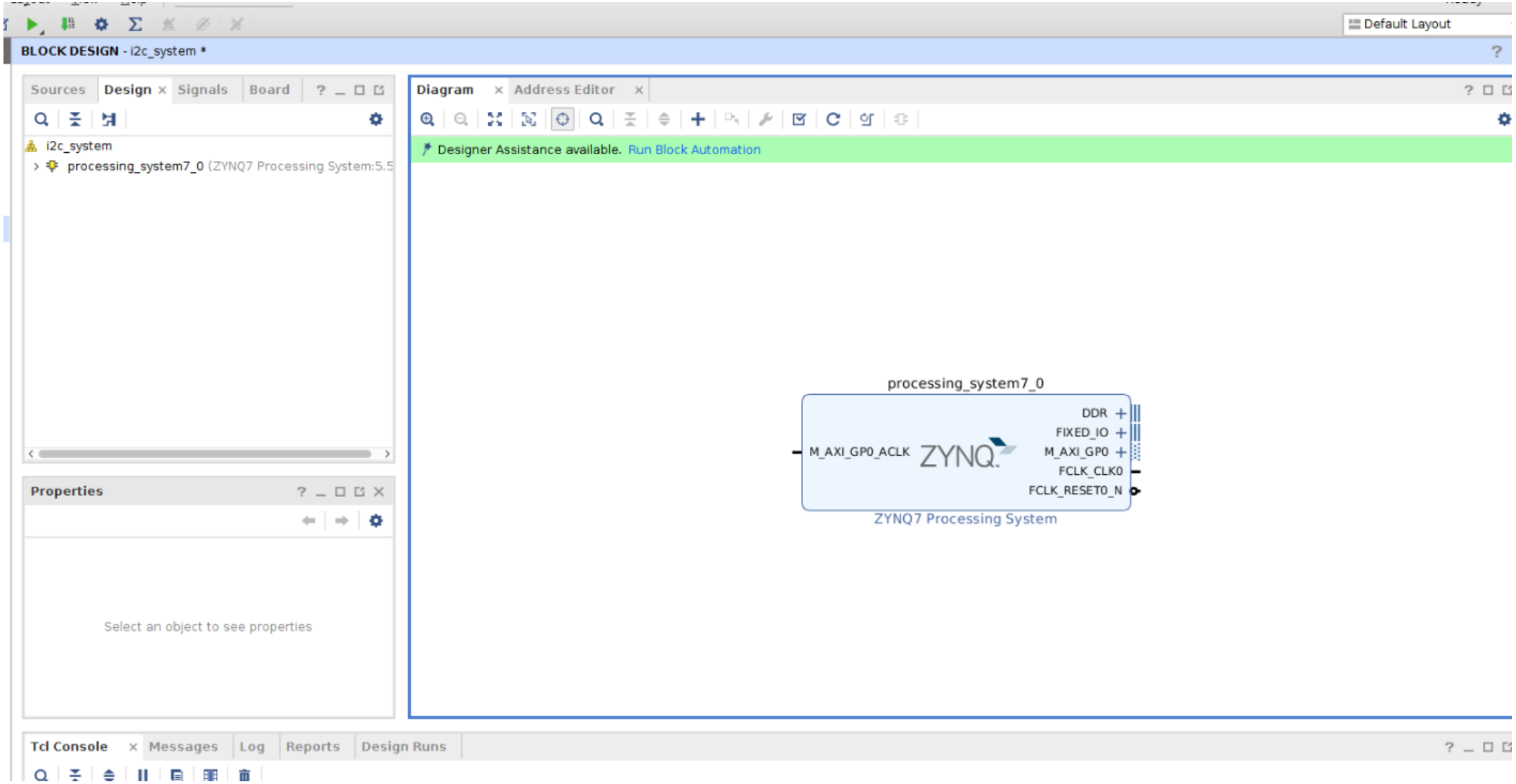
1) Vivado I2C 설계 3



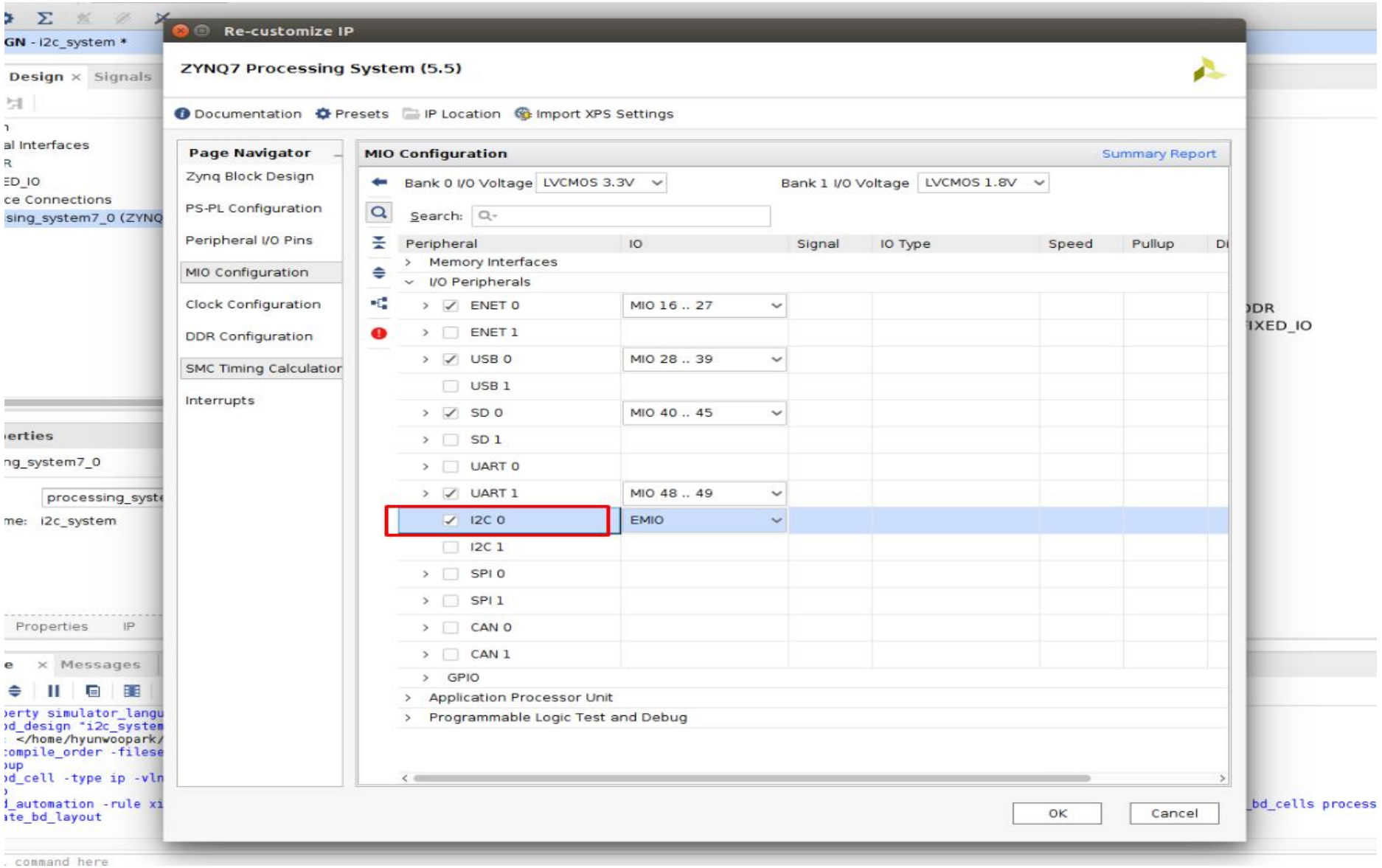
1) Vivado I2C 설계 4



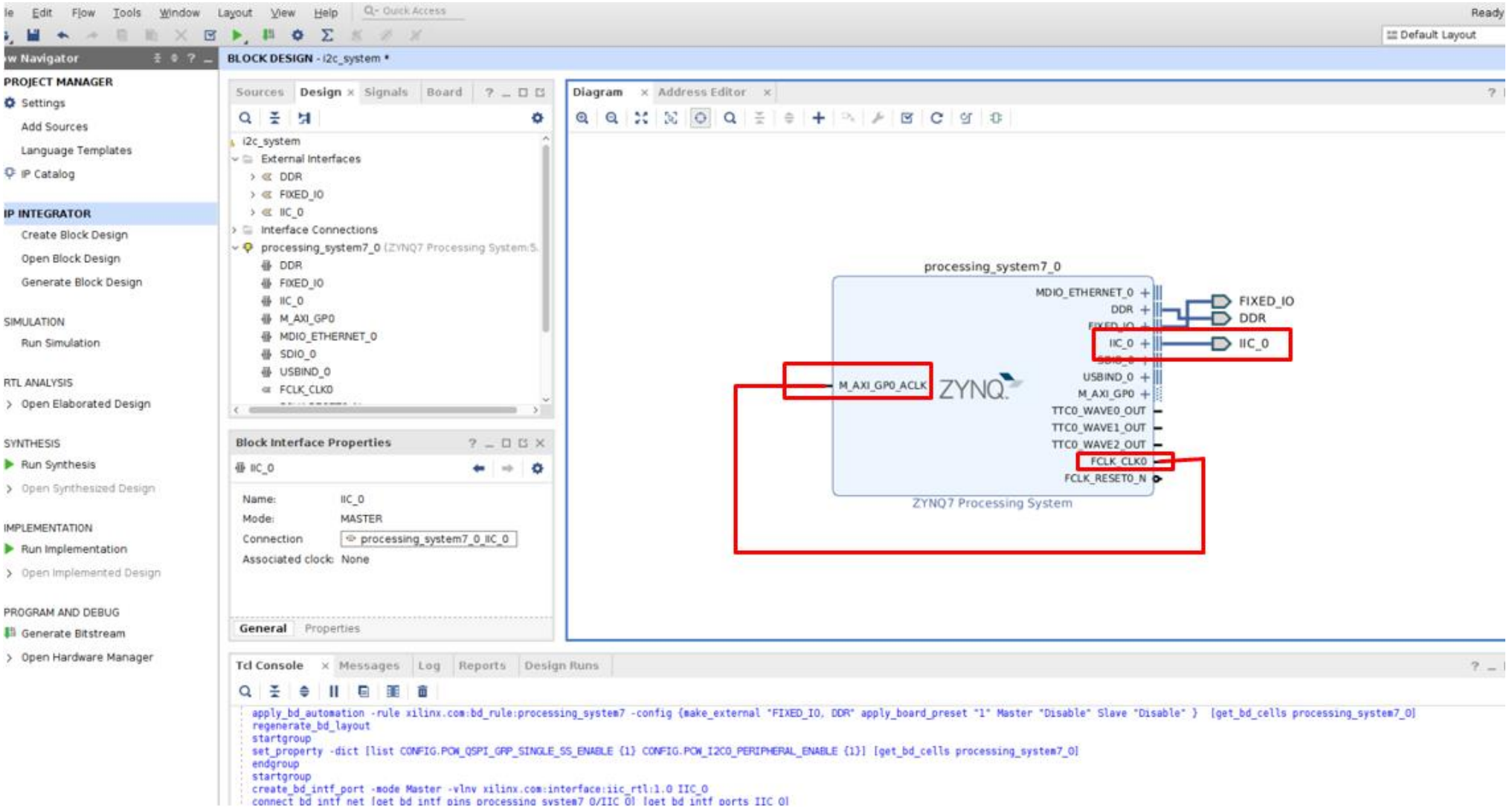
1) Vivado I2C 설계 5



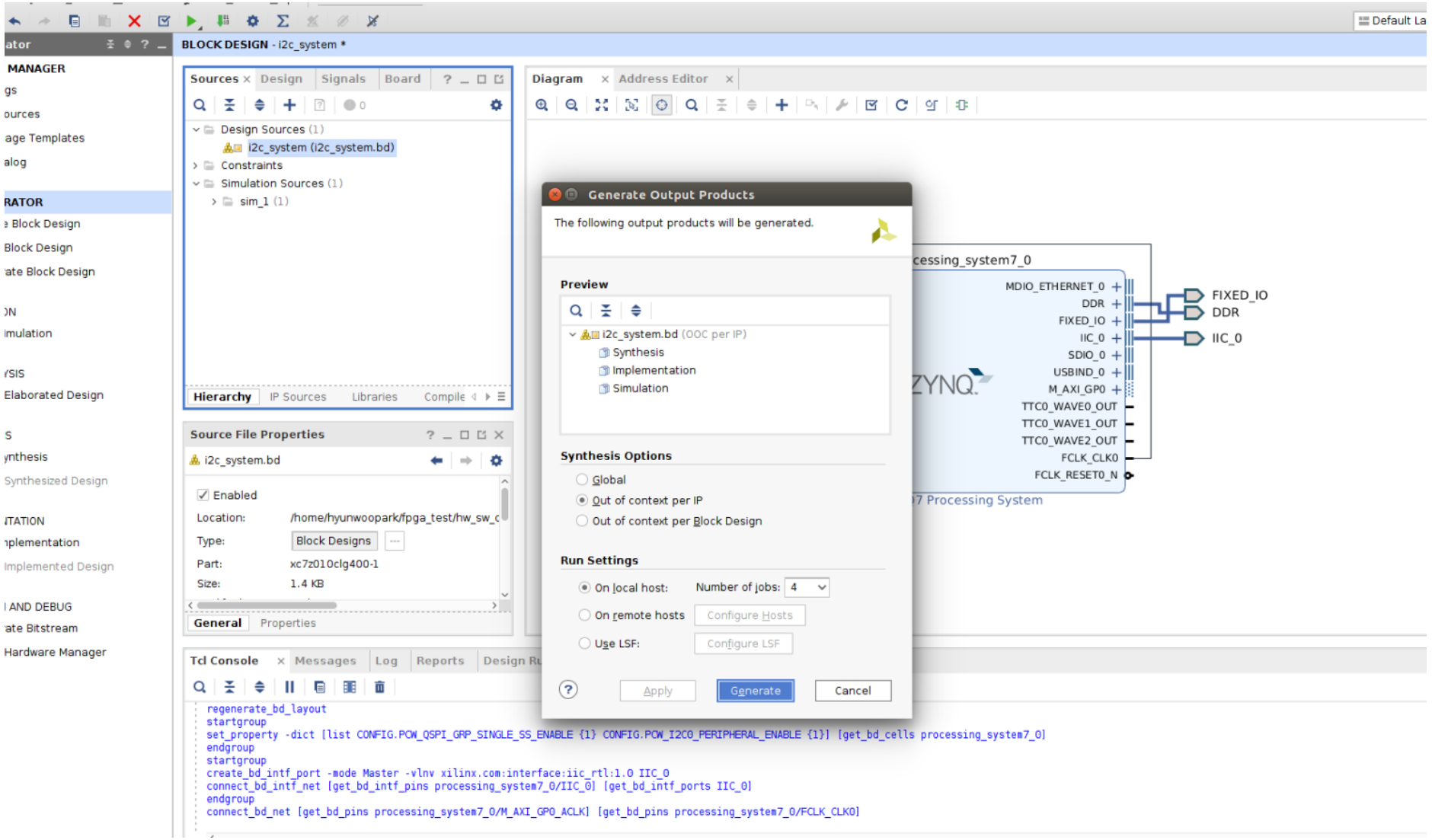
1) Vivado I2C 설계 6



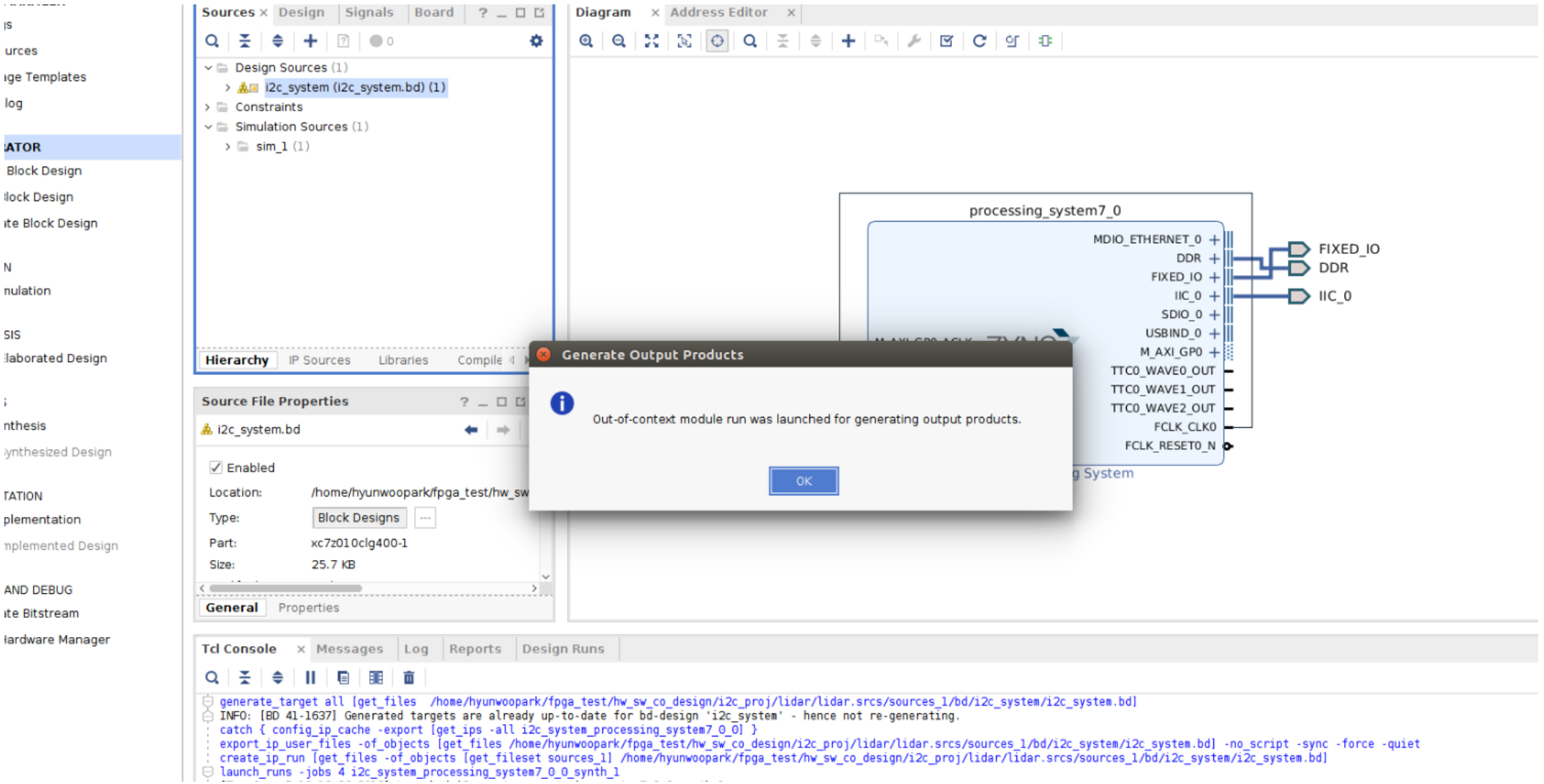
1) Vivado I2C 설계 7



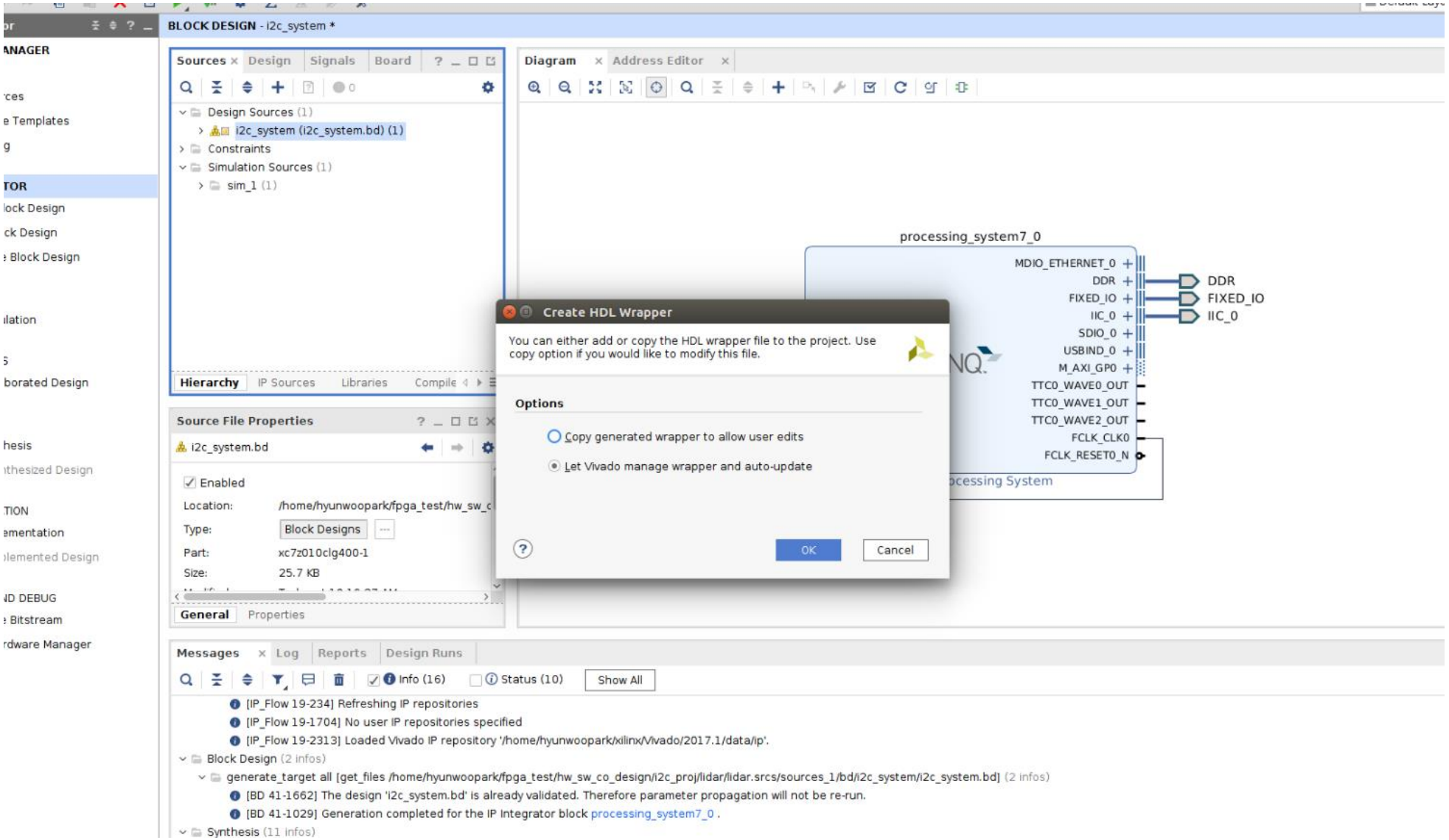
1) Vivado I2C 설계 8



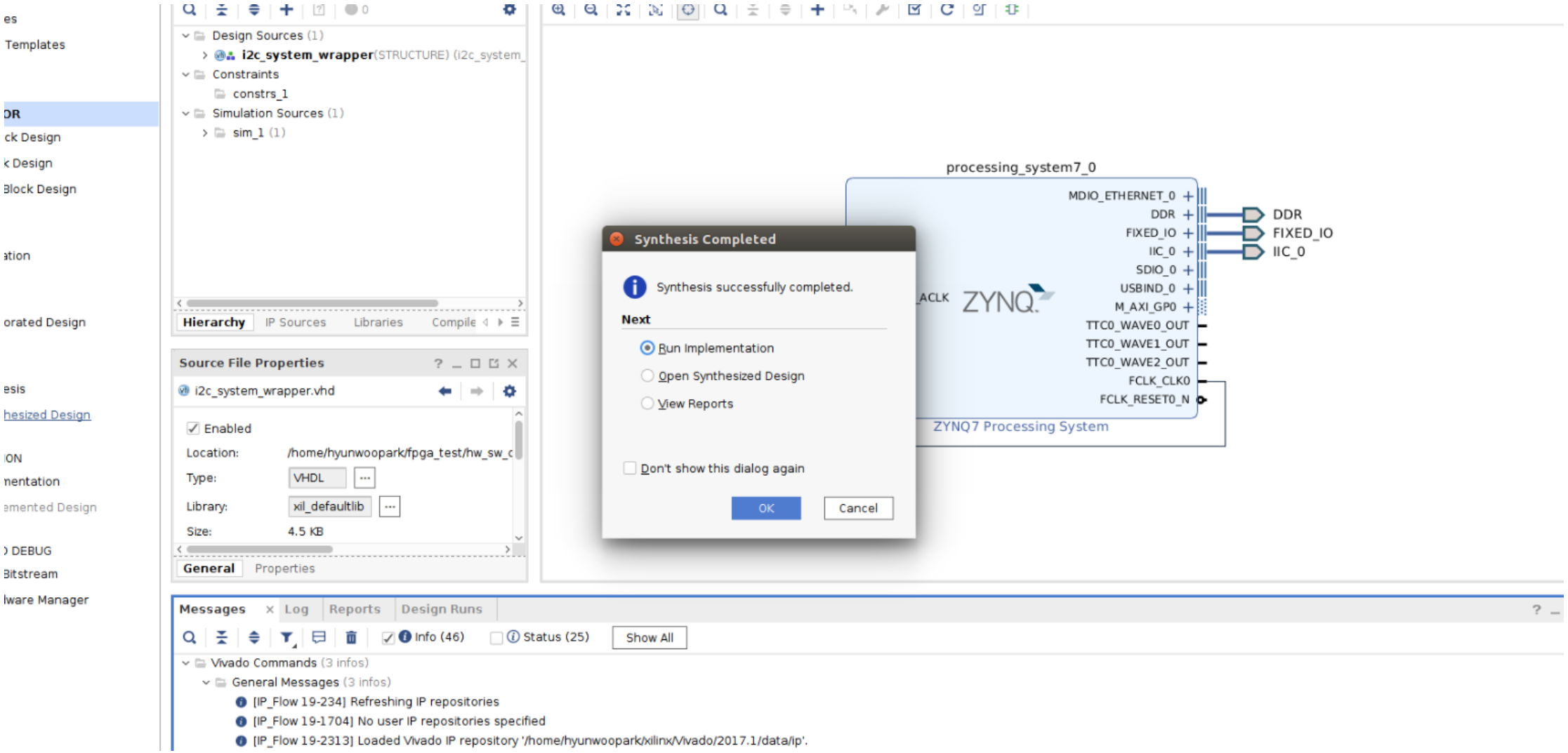
1) Vivado I2C 설계 9



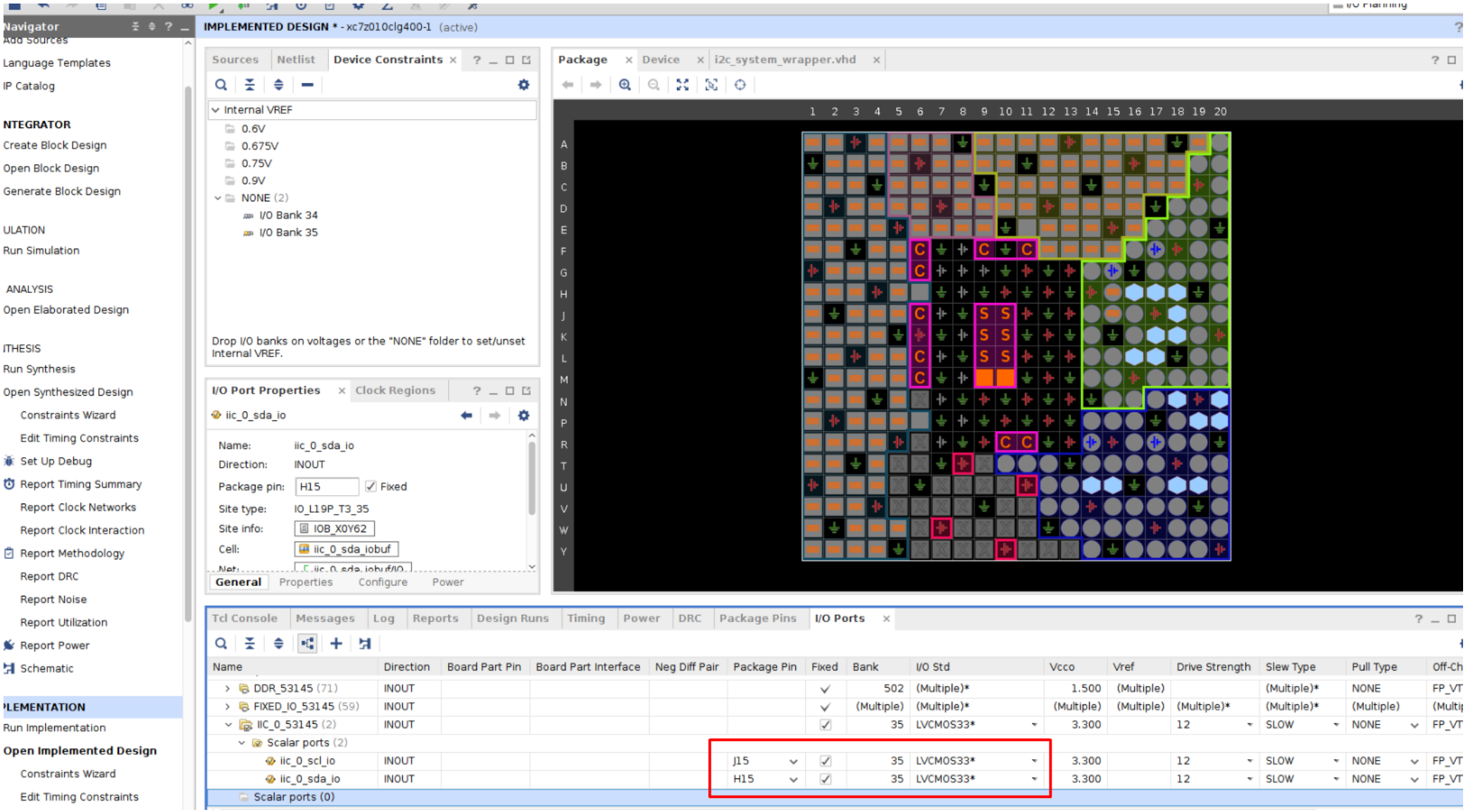
1) Vivado I2C 설계 10



1) Vivado I2C 설계 11



1) Vivado I2C 설계 12



1) Vivado I2C 설계 13

```
hyunwoopark@hyunwoopark-P65-P67SG:~/fpga_test/hw_sw_co_design/i2c_proj$ ls
lidar
hyunwoopark@hyunwoopark-P65-P67SG:~/fpga_test/hw_sw_co_design/i2c_proj$ mkdir PETALINUX
hyunwoopark@hyunwoopark-P65-P67SG:~/fpga_test/hw_sw_co_design/i2c_proj$ ls
lidar  PETALINUX
hyunwoopark@hyunwoopark-P65-P67SG:~/fpga_test/hw_sw_co_design/i2c_proj$ cd PETALINUX/
hyunwoopark@hyunwoopark-P65-P67SG:~/fpga_test/hw_sw_co_design/i2c_proj/PETALINUX$ ls
hyunwoopark@hyunwoopark-P65-P67SG:~/fpga_test/hw_sw_co_design/i2c_proj/PETALINUX$
```

먼저, petalinux-create project create 해야함

```
INFO: Create project: i2c_lidar
INFO: New project successfully created in /home/hyunwoopark/fpga_test/hw_sw_co_design/i2c_proj/PETALINUX/i2c_lidar
hyunwoopark@hyunwoopark-P65-P67SG:~/fpga_test/hw_sw_co_design/i2c_proj/PETALINUX$ ls
i2c_lidar
hyunwoopark@hyunwoopark-P65-P67SG:~/fpga_test/hw_sw_co_design/i2c_proj/PETALINUX$ cd i2c_lidar/
hyunwoopark@hyunwoopark-P65-P67SG:~/fpga_test/hw_sw_co_design/i2c_proj/PETALINUX/i2c_lidar$ ls
config.project  hw-description  subsystems
hyunwoopark@hyunwoopark-P65-P67SG:~/fpga_test/hw_sw_co_design/i2c_proj/PETALINUX/i2c_lidar$ petalinux-config --get-hw-description=~/fpga_test
/hw_sw_co_design/i2c_proj/lidar/lidar.sdk
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/hyunwoopark/fpga_test/hw_sw_co_design/i2c_proj/PETALINUX/i2c_lidar/build/linux/hw-description/hw-description.tcl -notrace
INFO: [Common 17-206] Exiting hsi at Tue Jun  5 10:49:46 2018...
INFO: Config linux
[INFO ] config linux
configuration written to /home/hyunwoopark/fpga_test/hw_sw_co_design/i2c_proj/PETALINUX/i2c_lidar/subsystems/linux/config

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

[INFO ] generate DTS to /home/hyunwoopark/fpga_test/hw_sw_co_design/i2c_proj/PETALINUX/i2c_lidar/subsystems/linux/configs/device-tree
INFO: [Hsi 55-1698] elapsed time for repository loading 5 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 10:50:04 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 10:50:17 2018...
```

1) Vivado I2C 설계 14

Petalinux 개조를 하려면 여기서 bootcode를 건들이면 됨.

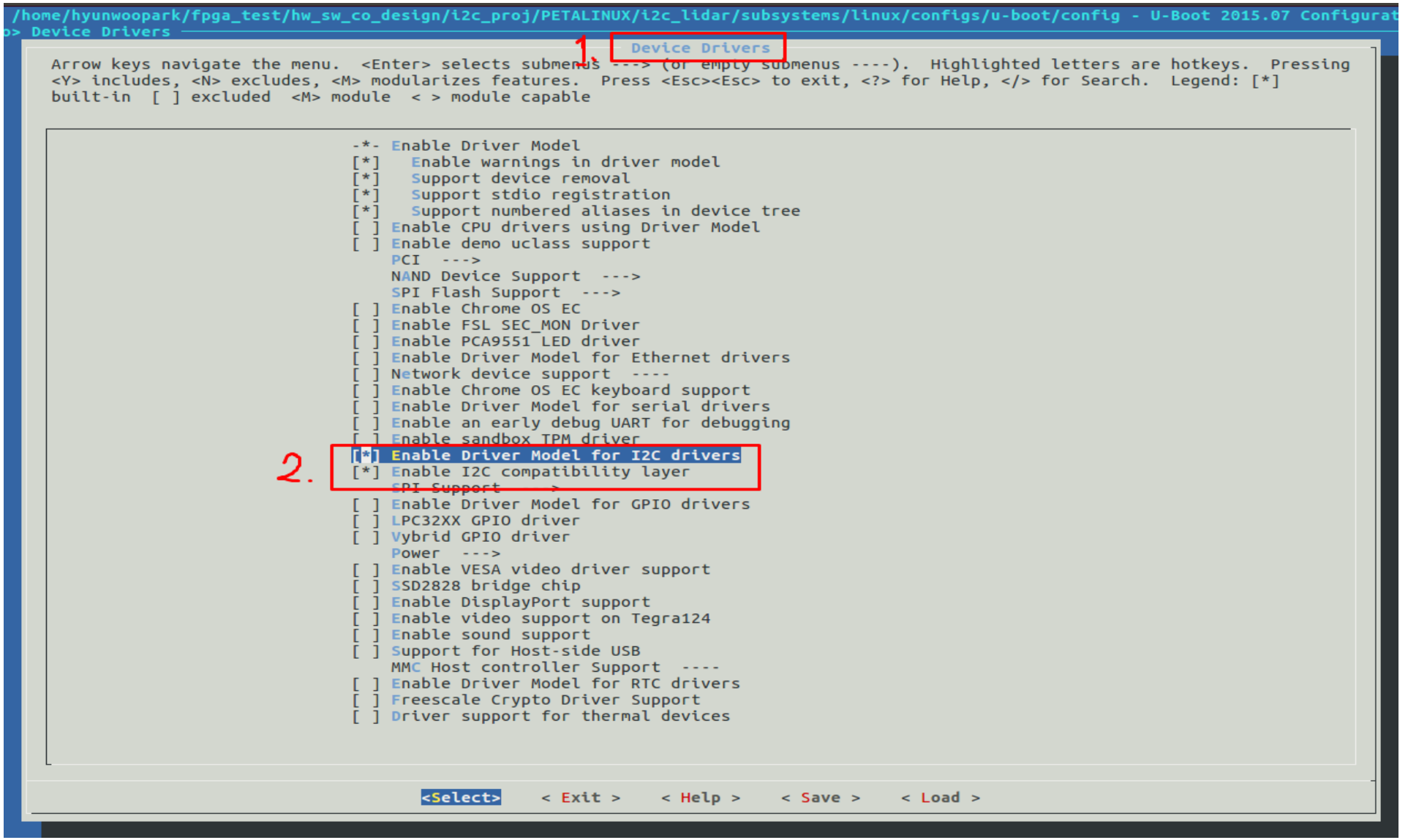
```
hyunwoopark@hyunwoopark-P65-P67SG:~/fpga_test/hw_sw_co_design/i2c_proj/PETALINUX/i2c_lidar$ ls
build  components  config.project  hw-description  subsystems
hyunwoopark@hyunwoopark-P65-P67SG:~/fpga_test/hw_sw_co_design/i2c_proj/PETALINUX/i2c_lidar$ cd components/
hyunwoopark@hyunwoopark-P65-P67SG:~/fpga_test/hw_sw_co_design/i2c_proj/PETALINUX/i2c_lidar/components$ ls
bootloader
hyunwoopark@hyunwoopark-P65-P67SG:~/fpga_test/hw_sw_co_design/i2c_proj/PETALINUX/i2c_lidar/components$ cd bootloader/
hyunwoopark@hyunwoopark-P65-P67SG:~/fpga_test/hw_sw_co_design/i2c_proj/PETALINUX/i2c_lidar/components/bootloader$ ls
zynq_fsbl
hyunwoopark@hyunwoopark-P65-P67SG:~/fpga_test/hw_sw_co_design/i2c_proj/PETALINUX/i2c_lidar/components/bootloader$ cd zynq_fsbl/
hyunwoopark@hyunwoopark-P65-P67SG:~/fpga_test/hw_sw_co_design/i2c_proj/PETALINUX/i2c_lidar/components/bootloader/zynq_fsbl$ ls
fsbl_debug.h    fsbl_hooks.c   image_mover.h  Makefile  nand.c  nor.h   ps7_init.c          qspi.c  rsa.h  zynq_fsbl_bsp
fsbl.h          fsbl_hooks.h   lscript.ld     md5.c     nand.h  pcap.c  ps7_init.h          qspi.h  sd.c
fsbl_handoff.S  image_mover.c  main.c         md5.h     nor.c   pcap.h  ps7_parameters.xml  rsa.c   sd.h
hyunwoopark@hyunwoopark-P65-P67SG:~/fpga_test/hw_sw_co_design/i2c_proj/PETALINUX/i2c_lidar/components/bootloader/zynq_fsbl$
```

```
hyunwoopark@hyunwoopark-P65-P67SG:~/fpga_test/hw_sw_co_design/i2c_proj/PETALINUX/i2c_lidar/components/bootloader/zynq_fsbl$ vi main.c
hyunwoopark@hyunwoopark-P65-P67SG:~/fpga_test/hw_sw_co_design/i2c_proj/PETALINUX/i2c_lidar/components/bootloader/zynq_fsbl$ 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'.

hyunwoopark@hyunwoopark-P65-P67SG:~/fpga_test/hw_sw_co_design/i2c_proj/PETALINUX/i2c_lidar/components/bootloader/zynq_fsbl$
```

1) Vivado I2C 설계 15



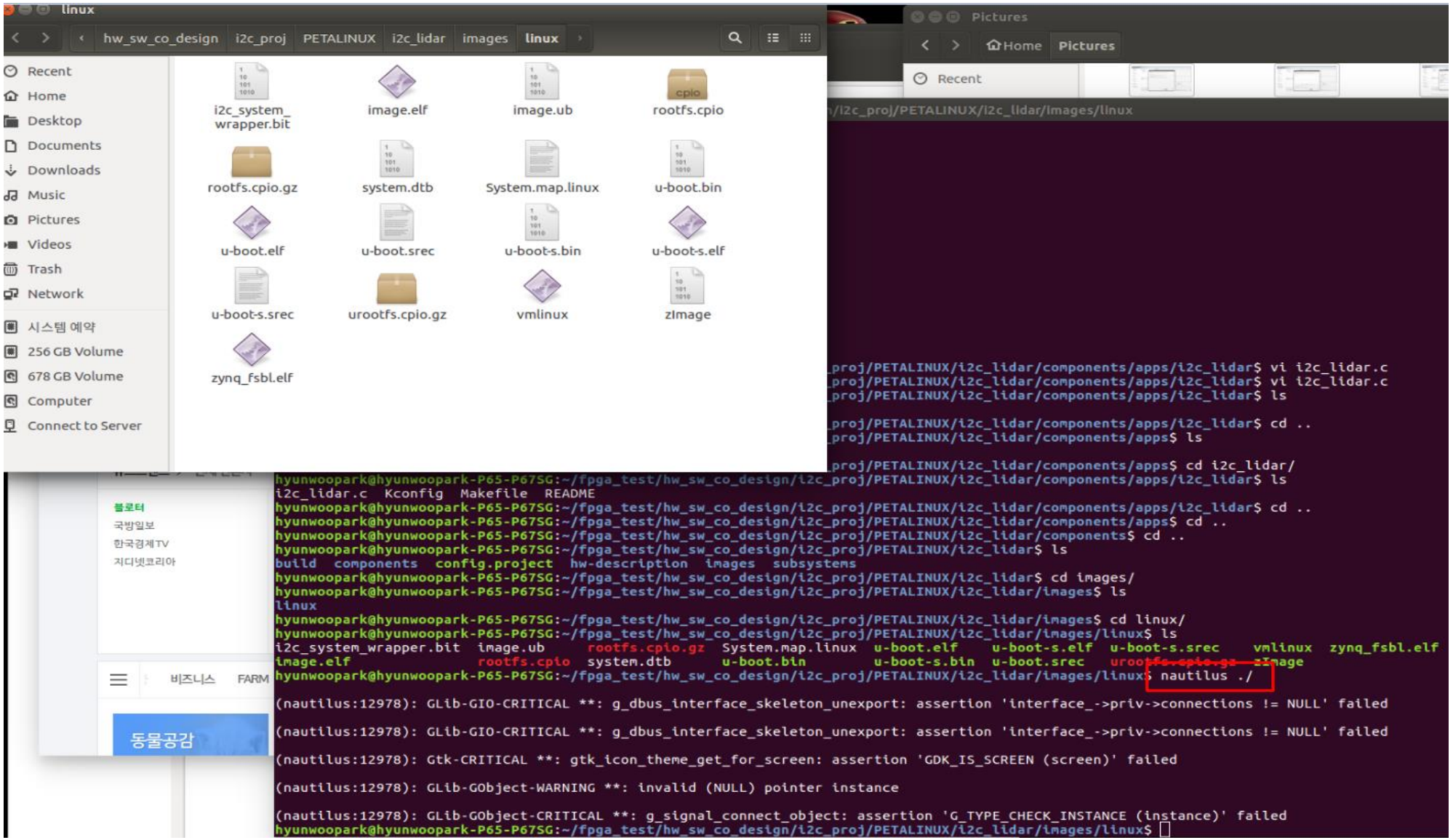
1) Vivado I2C 설계 16

```
hyunwoopark@hyunwoopark-P65-P67SG:~/fpga_test/hw_sw_co_design/i2c_proj/PETALINUX/i2c_lidar/components/bootloader/zynq_fsbl$ petalinux-build
INFO: Checking component...
INFO: Generating make files and build linux
INFO: Generating make files for the subcomponents of linux
INFO: Building linux
[INFO ] pre-build linux/rootfs/fwupgrade
[INFO ] pre-build linux/rootfs/peekpoke
[INFO ] build system.dtb
[INFO ] build linux/kernel
```

```
hyunwoopark@hyunwoopark-P65-P67SG:~/fpga_test/hw_sw_co_design/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/hyunwoopark/fpga_test/hw_sw_co_design/i2c_proj/PETALINUX/i2c_lidar/components/apps/i2c_lidar
INFO: Enabling created component...
INFO: It has been enabled to linux/rootfs
```

```
hyunwoopark@hyunwoopark-P65-P67SG:~/fpga_test/hw_sw_co_design/i2c_proj/PETALINUX/i2c_lidar$ cd components/apps/
hyunwoopark@hyunwoopark-P65-P67SG:~/fpga_test/hw_sw_co_design/i2c_proj/PETALINUX/i2c_lidar/components/apps$ ls
i2c_lidar
hyunwoopark@hyunwoopark-P65-P67SG:~/fpga_test/hw_sw_co_design/i2c_proj/PETALINUX/i2c_lidar/components/apps$ cd i2c_lidar/
hyunwoopark@hyunwoopark-P65-P67SG:~/fpga_test/hw_sw_co_design/i2c_proj/PETALINUX/i2c_lidar/components/apps/i2c_lidar$ ls
i2c_lidar.c  Kconfig  Makefile  README
hyunwoopark@hyunwoopark-P65-P67SG:~/fpga_test/hw_sw_co_design/i2c_proj/PETALINUX/i2c_lidar/components/apps/i2c_lidar$ vi i2c_lidar.c
```

1) Vivado I2C 설계 17



1) Vivado I2C 설계 18

```
(nautilus:12978): GLib-GObject-CRITICAL **: g_signal_connect_object: assertion 'G_TYPE_CHECK_INSTANCE (instance)' failed
hyunwoopark@hyunwoopark-P65-P67SG:~/fpga_test/hw_sw_co_design/i2c_proj/PETALINUX/i2c_lidar/images/linux$ petalinux-build
INFO: Checking component...
INFO: Generating make files and build linux
INFO: Generating make files for the subcomponents of linux
INFO: Building linux
[INFO ] pre-build linux/rootfs/fwupgrade
[INFO ] pre-build linux/rootfs/i2c_lidar
[INFO ] pre-build linux/rootfs/peekpoke
[INFO ] build linux/kernel
[INFO ] generate linux/u-boot configuration files
[INFO ] update linux/u-boot source
[INFO ] build linux/u-boot
[INFO ] build zynq-fsbl
```

2) LIDAR 센서 제어 C code 1

```
#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 READ_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_OPTION>₩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, 0x08);
    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;
}
```

2) LIDAR 센서 제어 C code 2

```
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 OUTT !₩n");
            return ;
        }
    }
    if(!busy_flag){
        if(write(fd, buf, 1) != 1){
            perror("---WRITE REGISTER ERROR ");
            return ;
        }

        if(read(fd, receives, read_size) != read_size){
            perror("---WRITE REGISTER ERROR ");
            return ;
        }
    }
}
```

```
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 ;
    }
    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(READ_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");
```

3) LIDAR 제어 결과물