zynq 보드에 Peta Linux 이미지 빌드

(시작하기 전에 peta linux 설정이 완료되어야 한다.)

http://cafe.naver.com/koittodaeng/53 를 참고!!

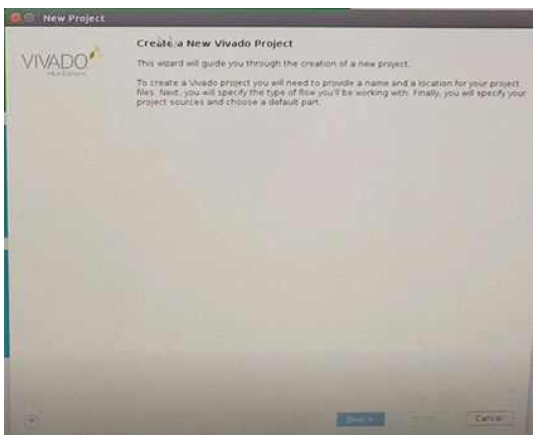
프로젝트 대상이되는 폴더를 생성 후 hardware 생성
(여기서는 lab7로 생성하였으므로 lab7로 진행하도록 한다.)

```
mkdir lab7

cd lab7

mkdir hardware

cd hardware
```

위의 디렉토리를 생성하였으면 우선 Vivado를 실행시키도록 한다.

1. Create Project 클릭

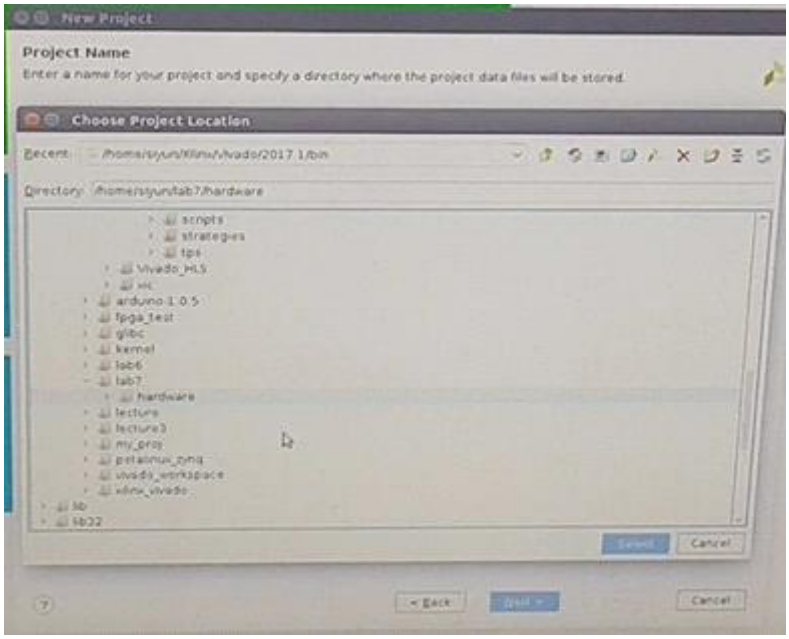
2. next 클릭



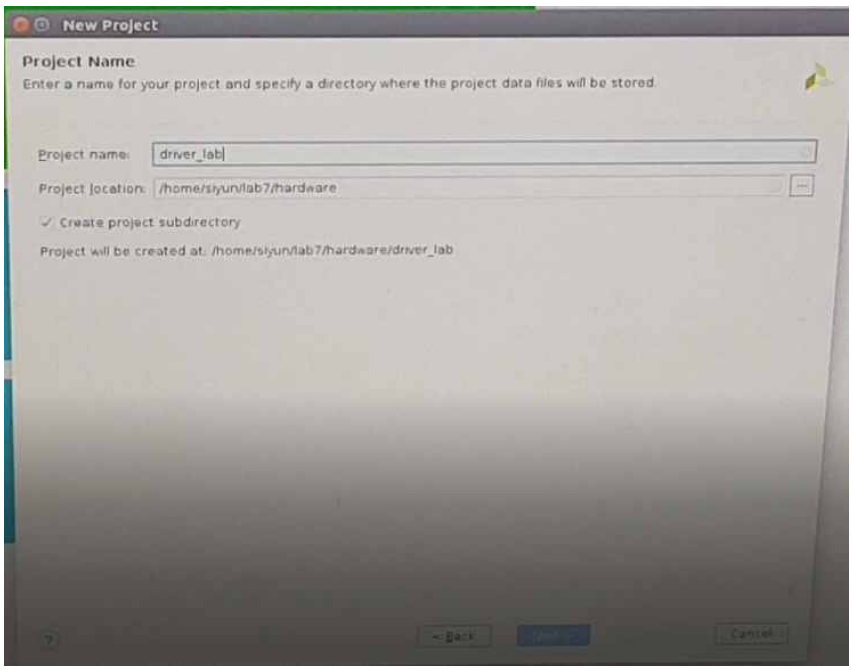
3. Project

경로는 작성자 컴퓨터 기준으로 한다.

/home/siyun/lab7/hardware/

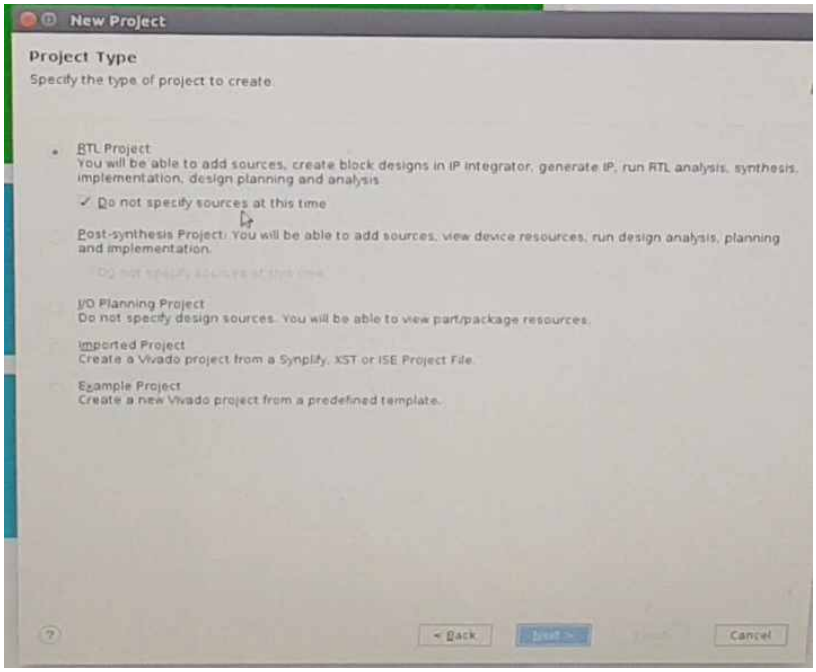


hardware 최종 경로 선택후 select



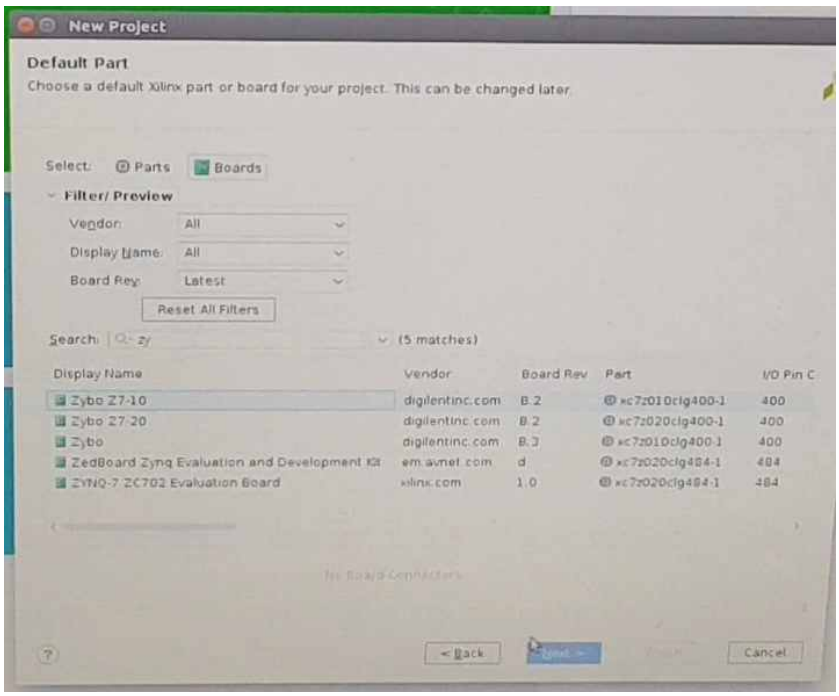
Project name을 driver_lab

Create project subdirectory를 체크 한 후 Next 클릭



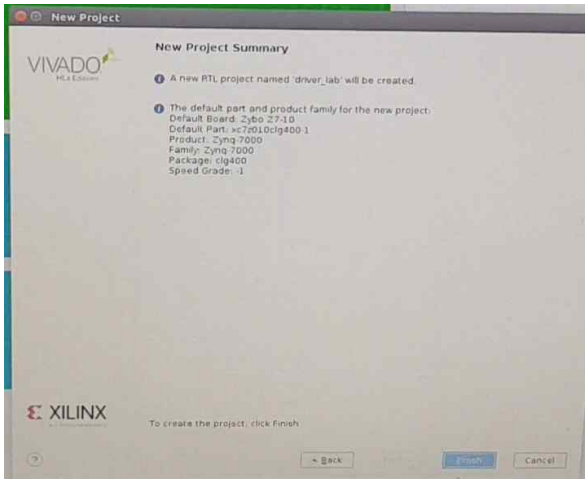
RTL Project 선택 후

Do not specity sources at this time 체크 후 Next



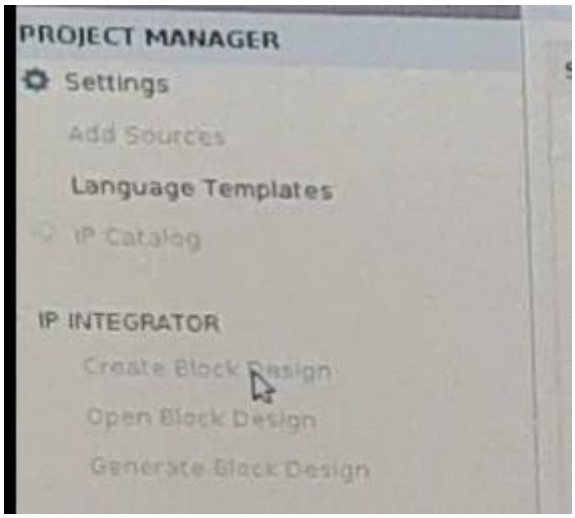
수업때는 Zybo를 선택하였지만, 우리가 사용하는 보드는 Zybo Z7-10이므로

Zybo Z7-10 선택 후 Next

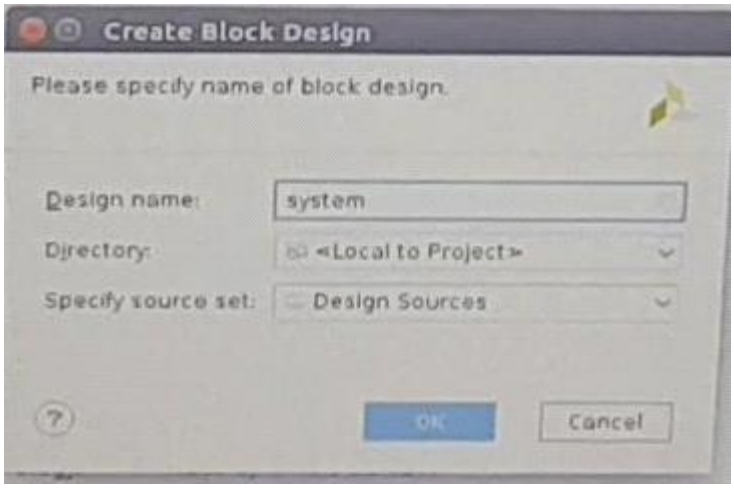


Finish 클릭후 프로젝트 생성

4.Create Block Design



Create Block Design 클릭



Create Block Design 클릭하면 디자인 이름과 기타 등등을 설정할 수 있다.

우리는 system 이라고 이름을 지정한다.
그 후 Ok 클릭

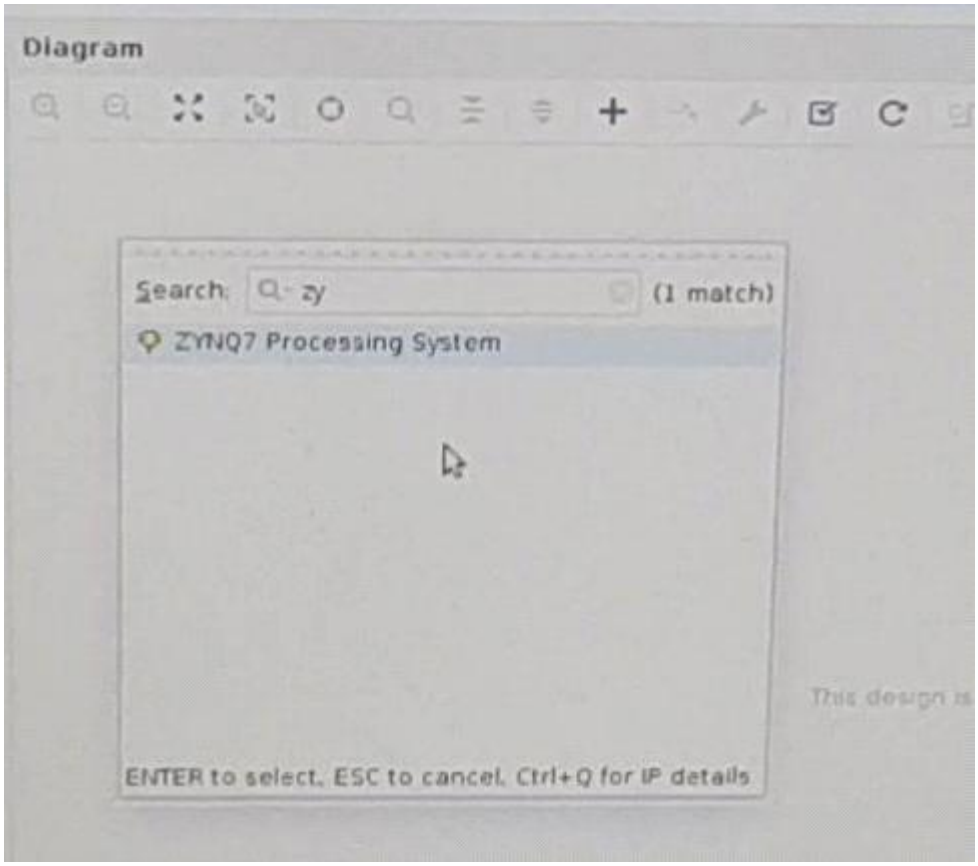
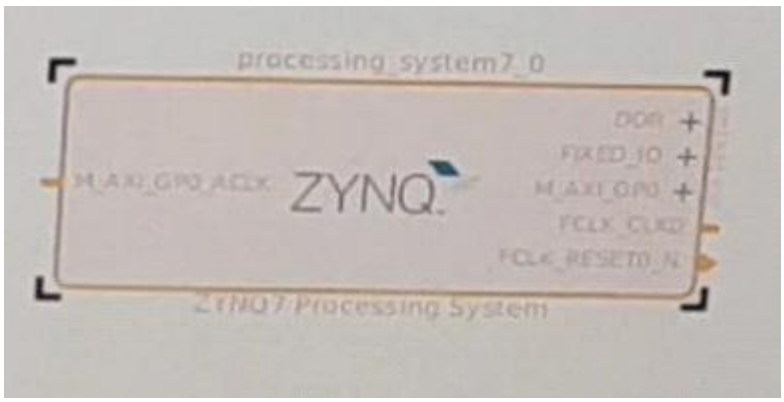
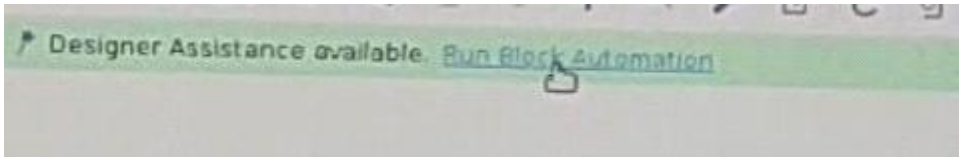
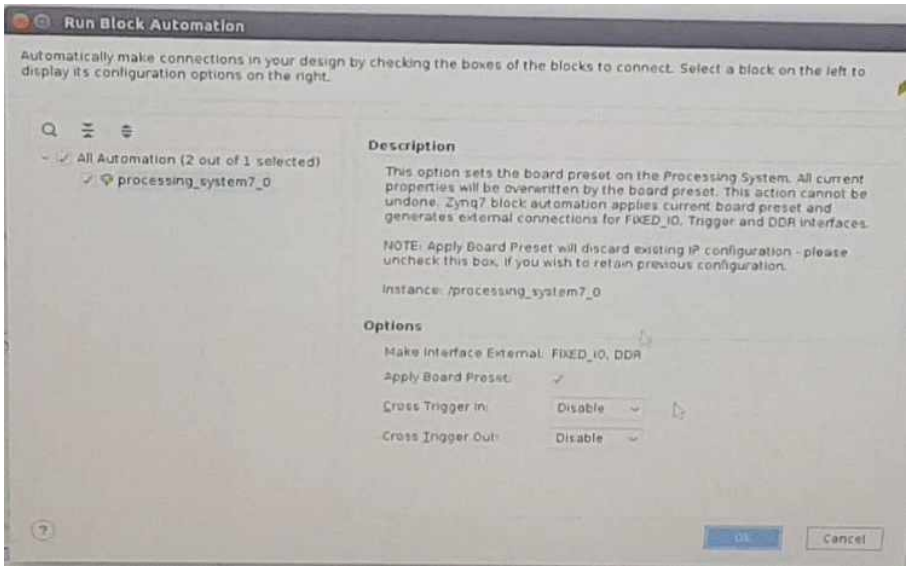


diagram 이 생성되면 +를 클릭한 후 ZZYNQ7 Processing System을 추가한다.



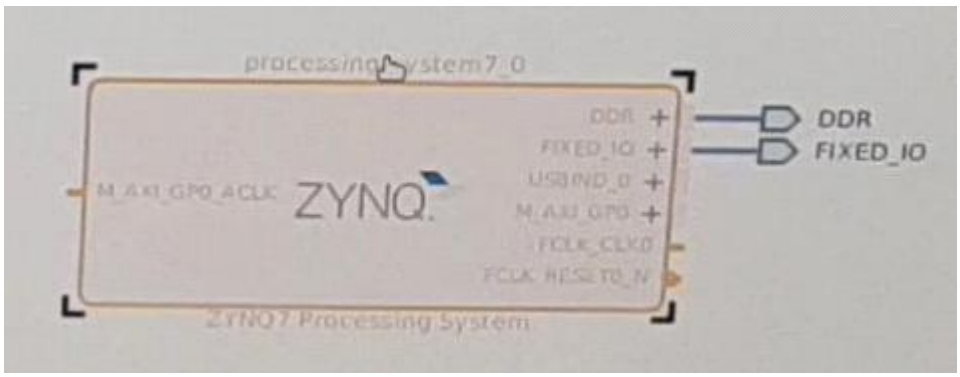
zynq block diagram을 불러오면 선택 후



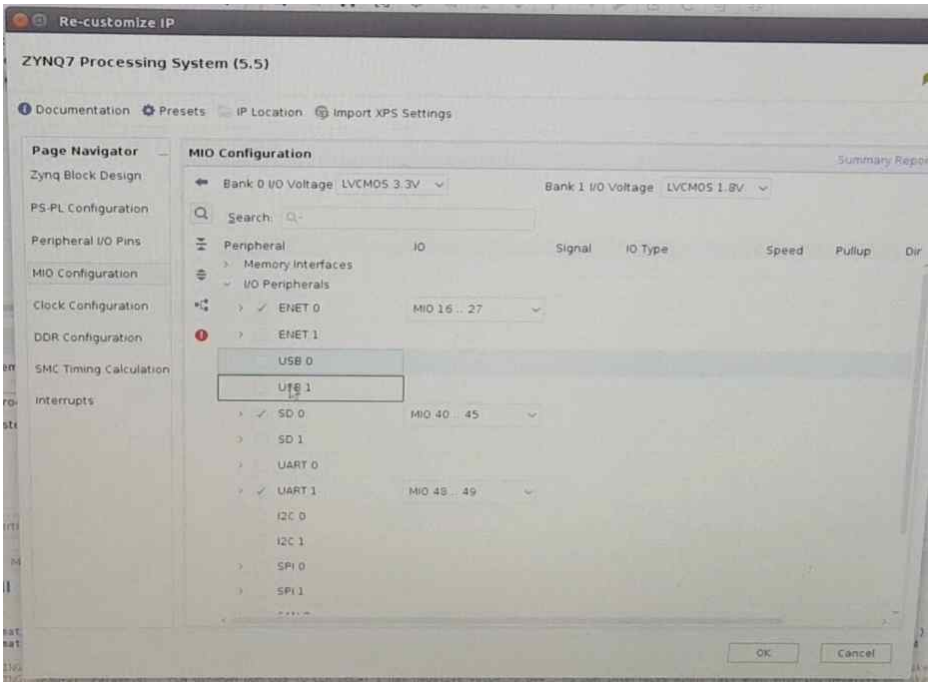


Run Block Automation을 클릭 후 Ok.
자동으로 핀 셋팅을 해주는 기능이다.

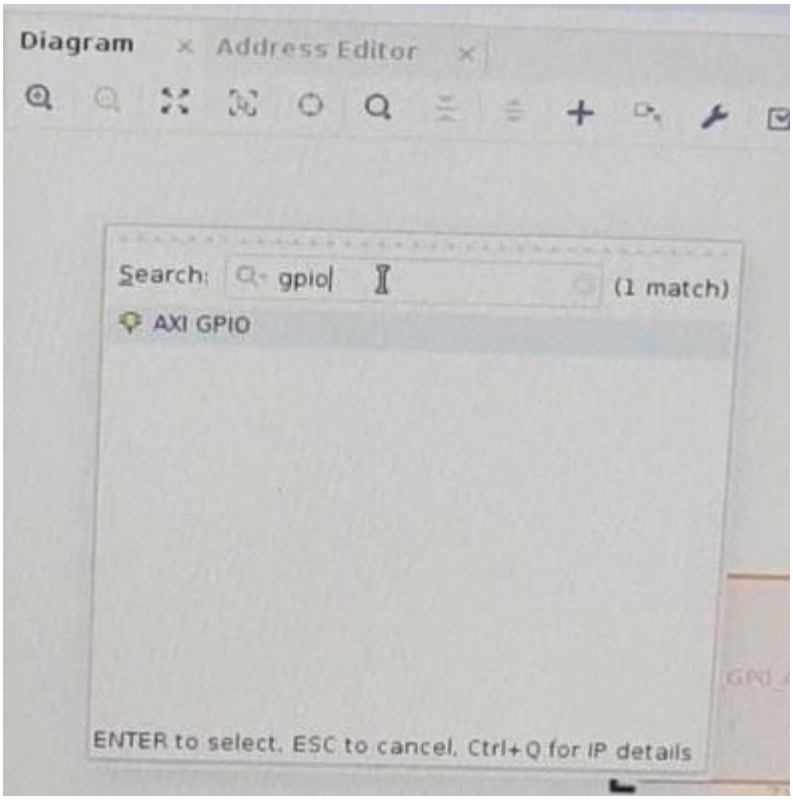
(가끔가다 컴퓨터 또는 노트북에 따라 자동 핀 설정이 누락되는 경우가 있다.
이 경우 어떤 짓을 해도 소용없고 구글링을 해도 자료가 그렇게 많지 않다.
하나하나 다 다른컴퓨터를 비교하며 수작업을 하거나 우리가 사용할 핀들을 하나하나 설정해주어야한다. 노가다 ..)



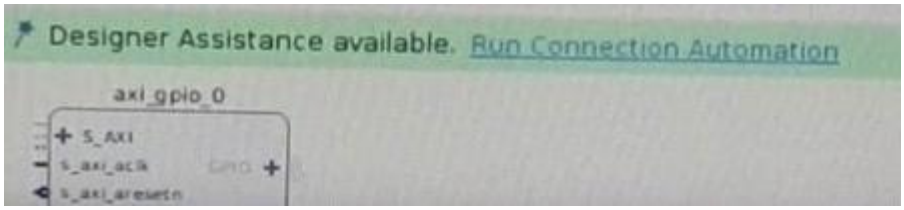
Auto set 완료 후 보드를 더블 클릭 한다.



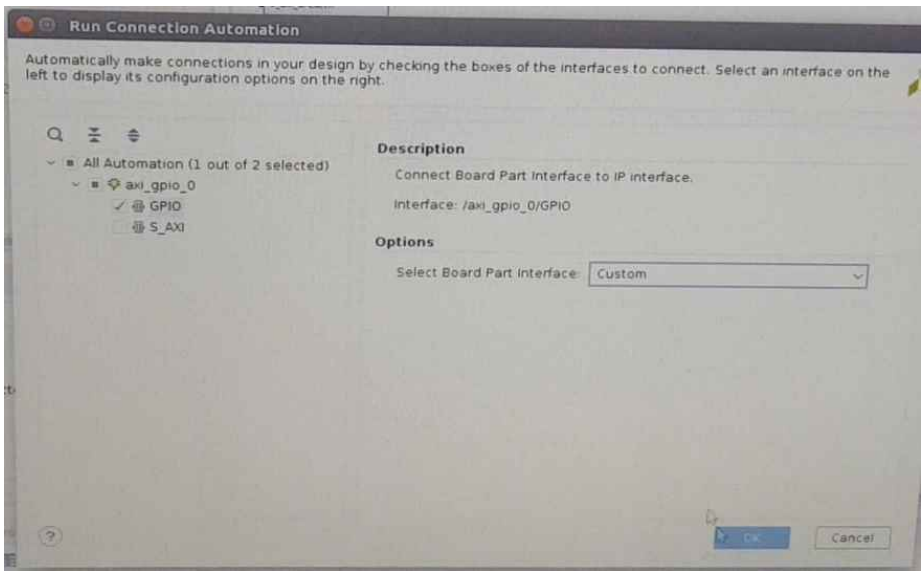
MIO Configulation -> I/O Peripharals -> USB0 체크 풀기 -> OK



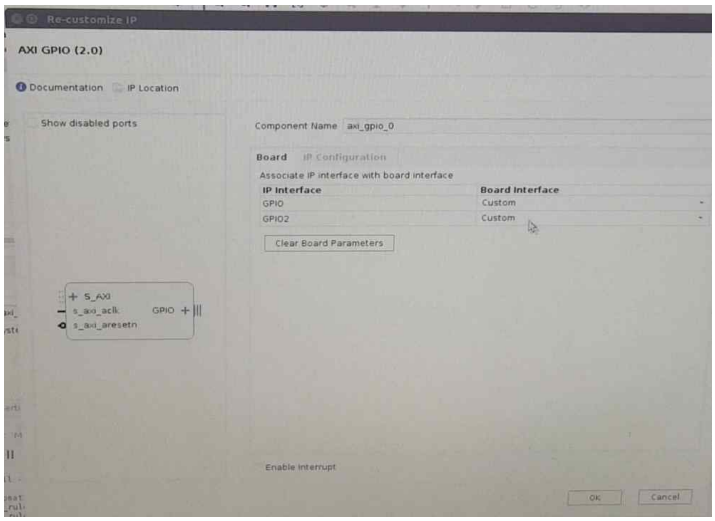
+ 클릭 후 gpio 블록다이어그램을 가져온다.



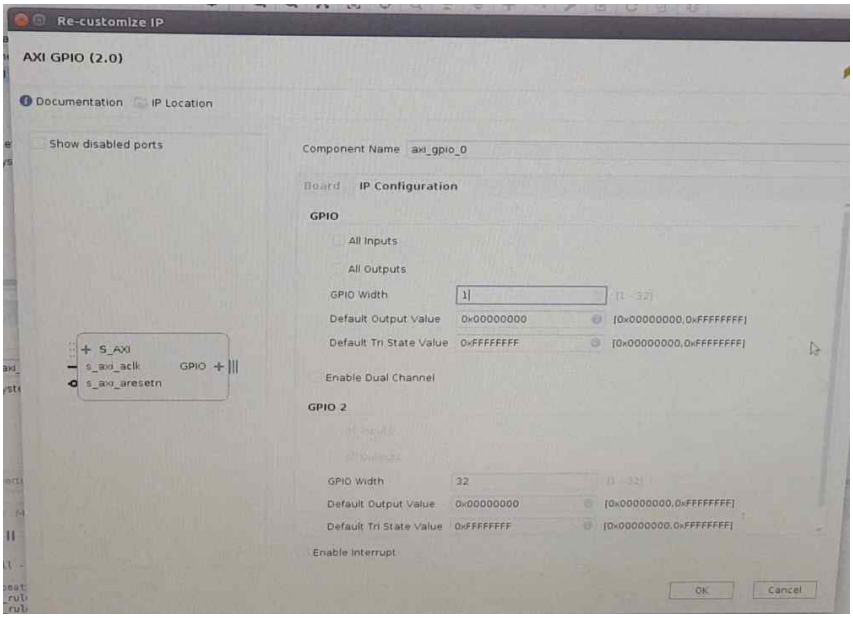
그 후 Run Connection Automation 클릭



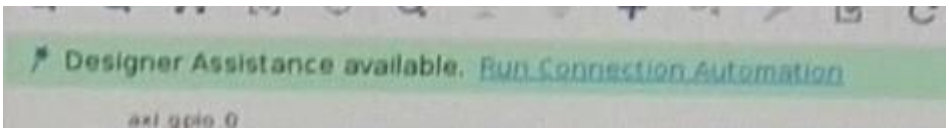
왼쪽에 GPIO만 체크 후 Select Board Part Interface를 Custom으로 설정 후 OK
그런 후 GPIO 블록다이어그램을 다시한번 더블클릭해서



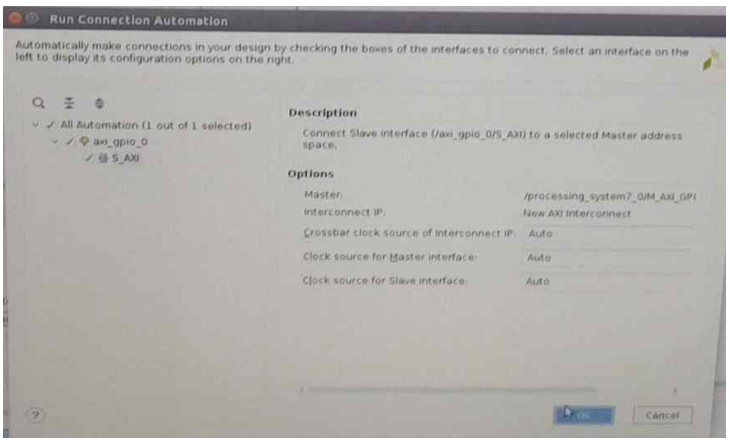
Board tab에서 커스텀으로 되어있는지 확인 후



IP Configuration 탭에서
GPIO Width를 1로 바꾼다 (1비트만 사용한다 라는 뜻)
그리고 OK 클릭

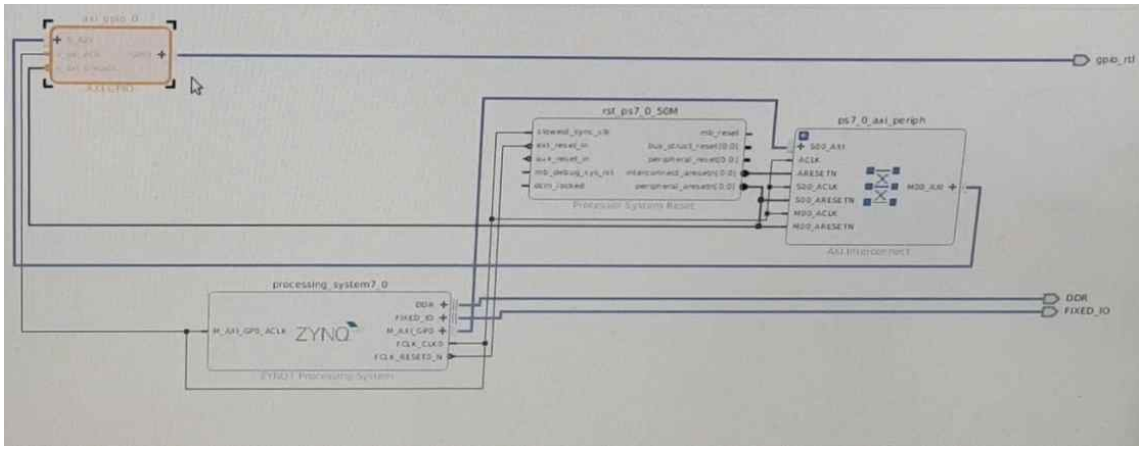


그리고 다시한번 Run Connection Automation



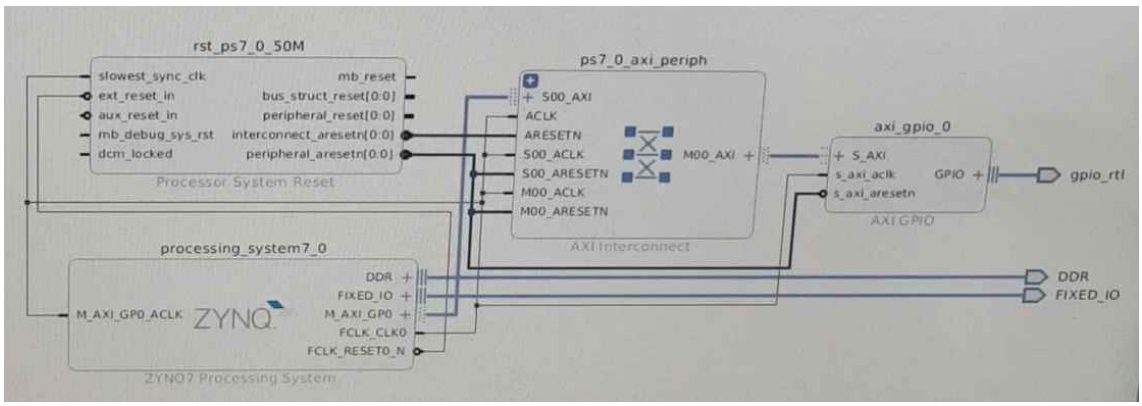
Run Connection Automation 해서 나온 창 바로 OK 클릭후 빠져나온다.

그러면 이제 Zynq 보드와 GPIO와 나머지 여러것들을 자동으로 연결해준다.



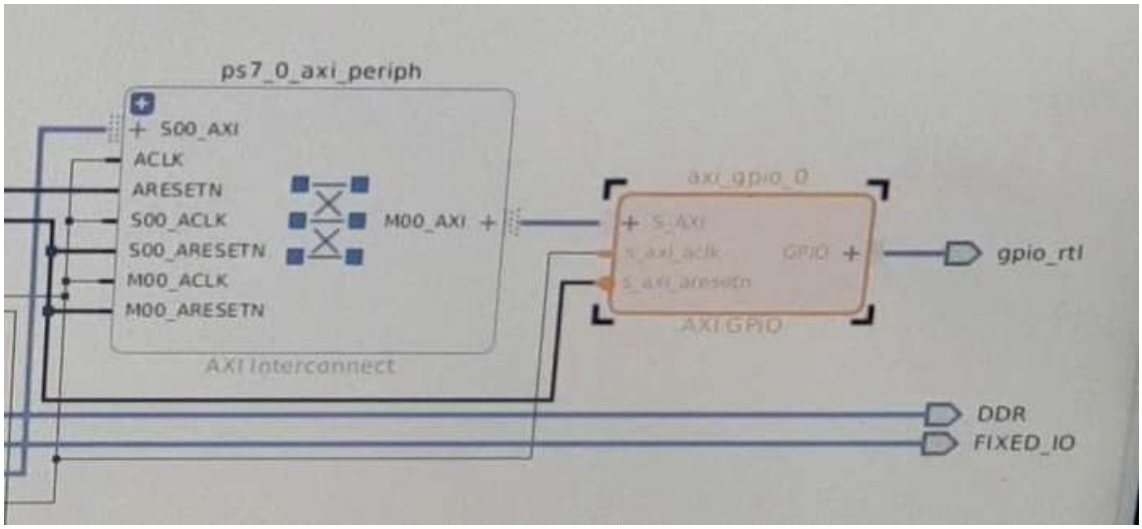
이상태가 되면 위에 새로고침 같은 버튼 (사이클 버튼)을 눌러 재배치 시켜준다.

 클릭

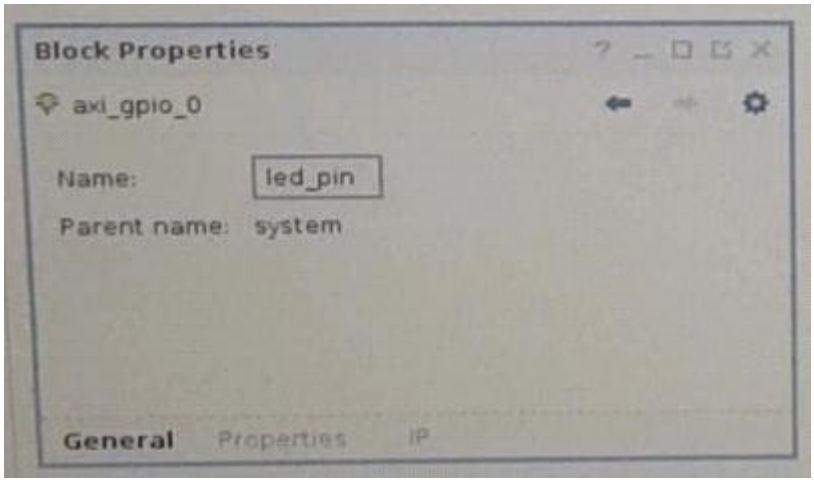


이제 우리가 알아보기 쉽게 하기위해 gpio 핀 이름을 바꿔줄 것이다.

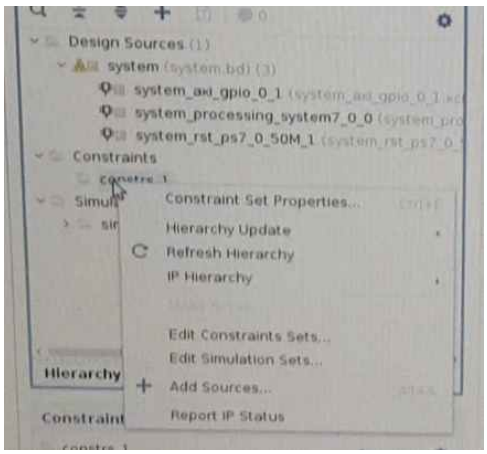
우선 gpio 블록을 클릭한다.



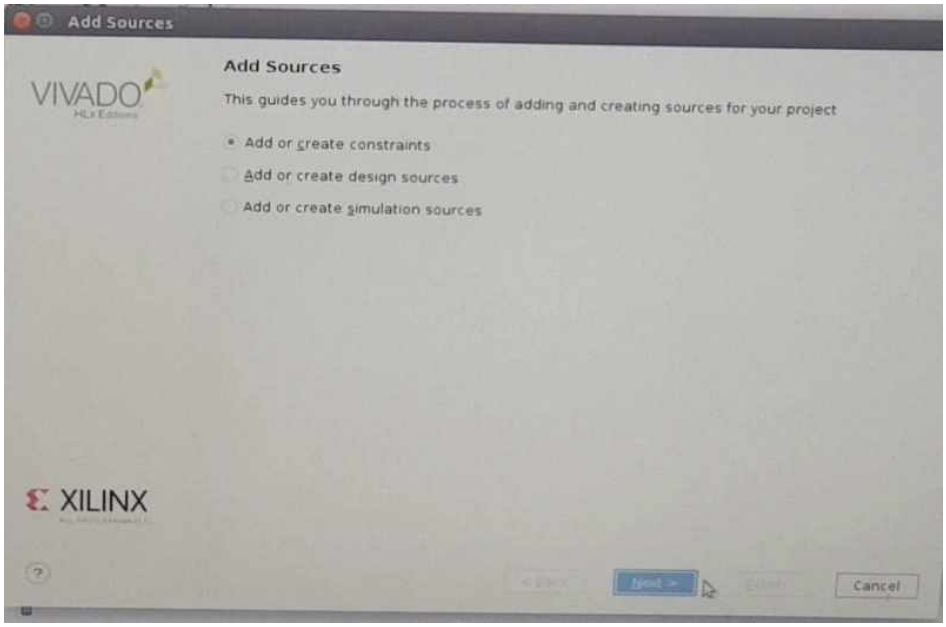
클릭후 왼쪽을보면 블록 속성(Block Properties)에서 Name을 수정한다.



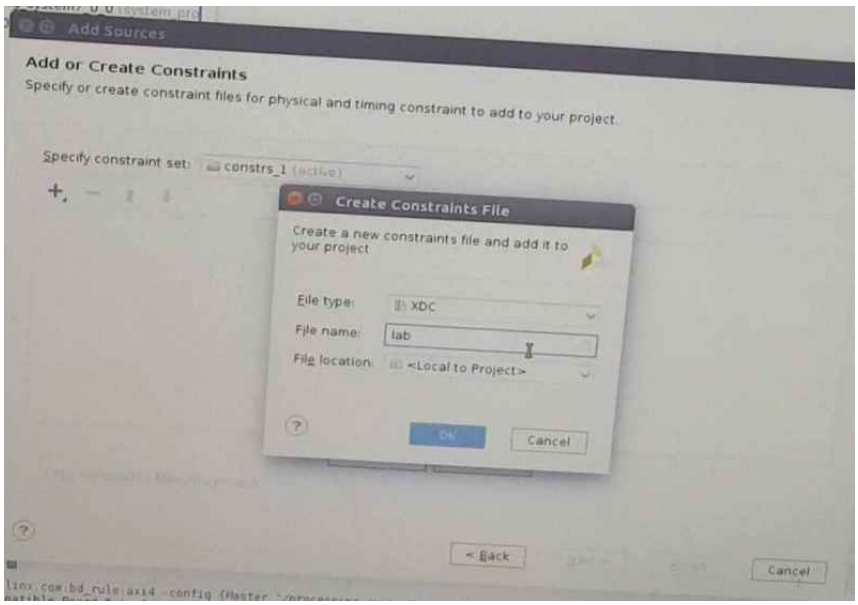
그후 Save(저장)



그 후 소스 탭에서 Constraints_1 마우스 우클릭 후 Add Sources 클릭



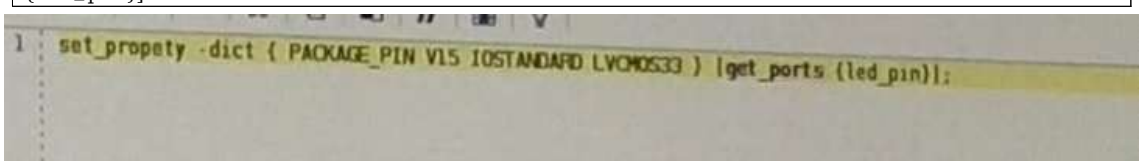
Next 클릭



create file 이름 마음데로해서 만들고 finish 클릭

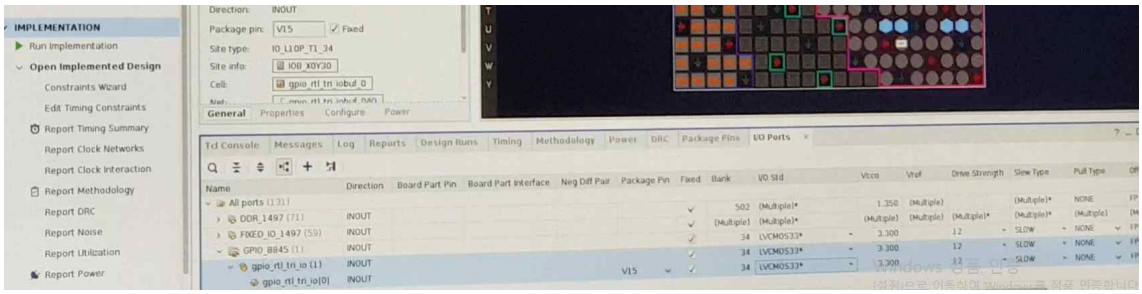
그다음 만든 파일 가서 코드작성 (핀 설정 코드)

```
set_property -dict { PACKAGE_PIN V15 IOSTANDARD LVCMOS33 } [get_ports {led_pin}];
```

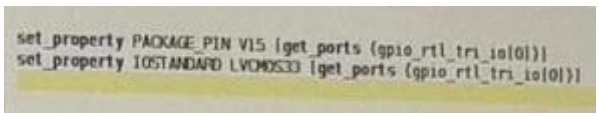


그 후 RUN Synthesis 한다.

완료 후
I/O Port에서 T19로 되어있는 핀을 V15로 수정
그리고 INOUT default 로 되있는 녀석들을 LVCMOS33으로 수정

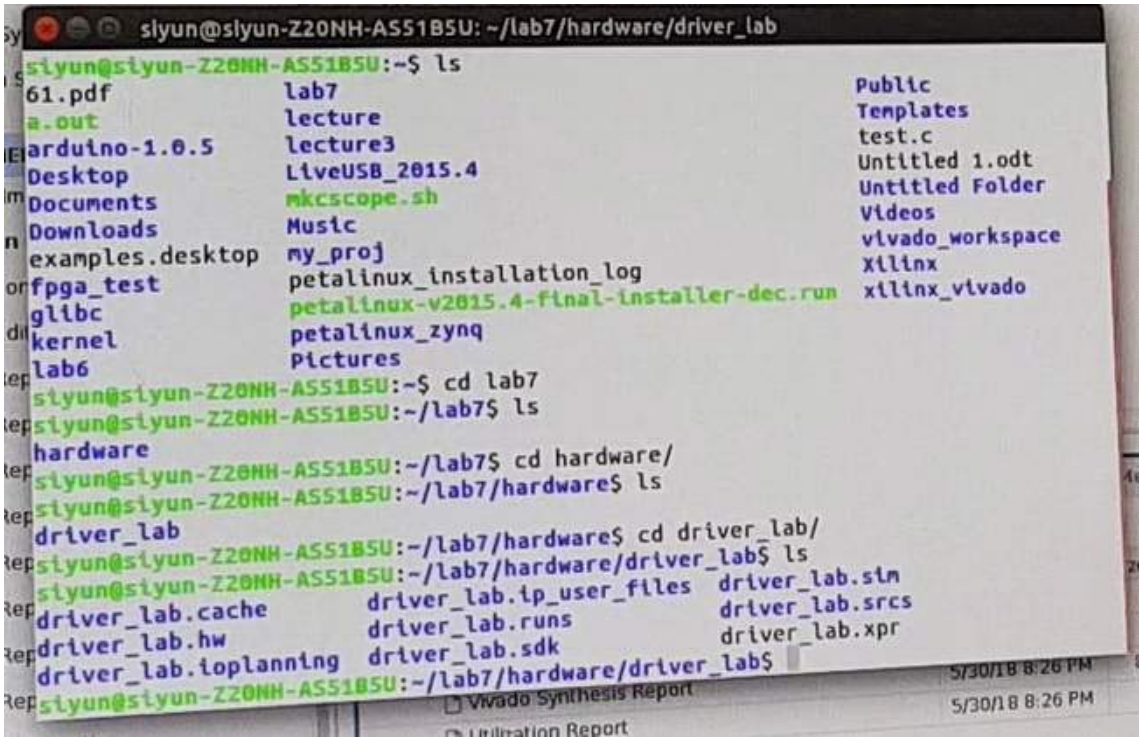


save out of design 하고 핀설정 소스코드가서 Reload 하면 소스코드가 오토셋 된다



그 후 Run Implementation 한다.
그리고 Generate Bitstream 하고 완료되면
파일 Export 클릭 후
include bit stream 클릭 후 ok

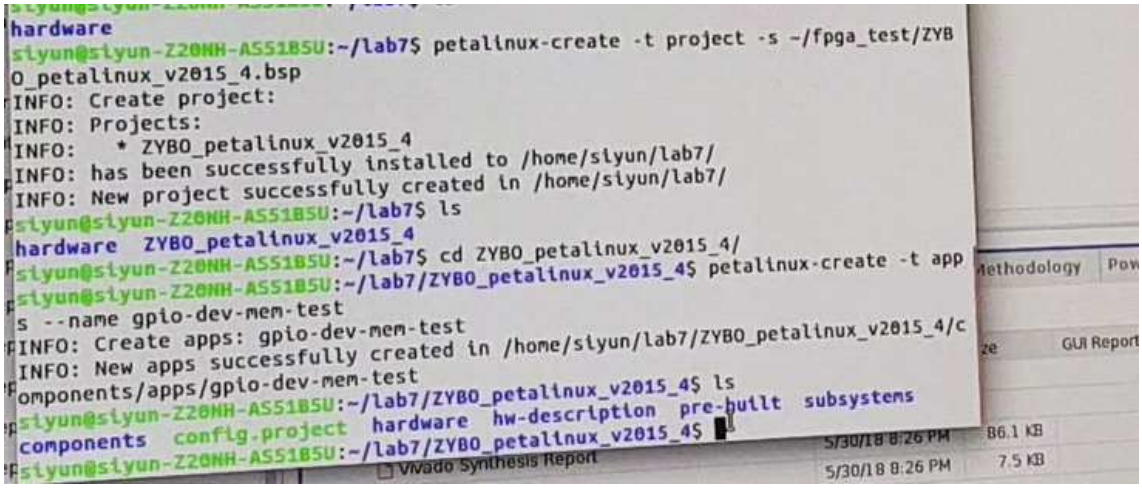
그러면 우리가 경로로 선택했던 hardware 폴더에 여러 가지들이 생성된다.



이제 peta linux를 만들어주면 된다.
peta linux 생성 할 때 맨처음 peta linux 설정할 때 만들어 주었던 ZYBO_petalinux_v2015_4.bsp가 필요하기 때문에 이 bsp 파일이 있는 경로를 확인하여 잘 작성하여야한다.

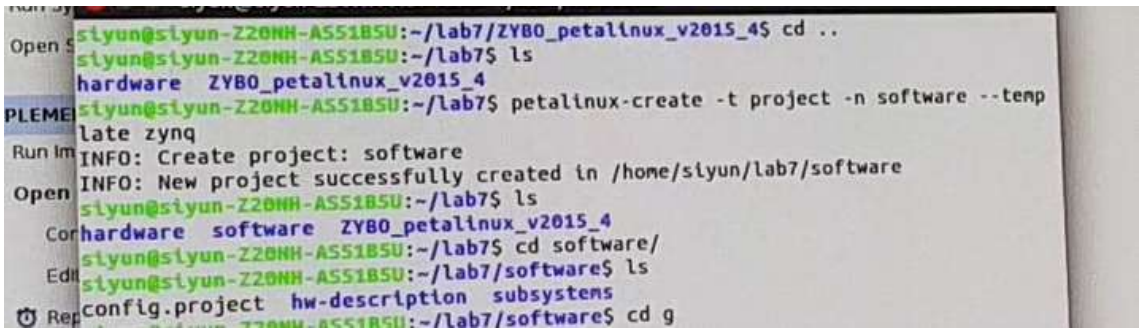
petalinux 는 생성한 hardware 폴더의 상위 폴도에 만들어주어야 한다.

```
cd ~/lab7

petalinux-create -t project -s ~/fpga_test/ZYBO_petalinux_v2015_4.bsp

cd ZYBO_petalinux_v2015_4/

petalinux-create -t apps --name gpio-dev-mem-test
```

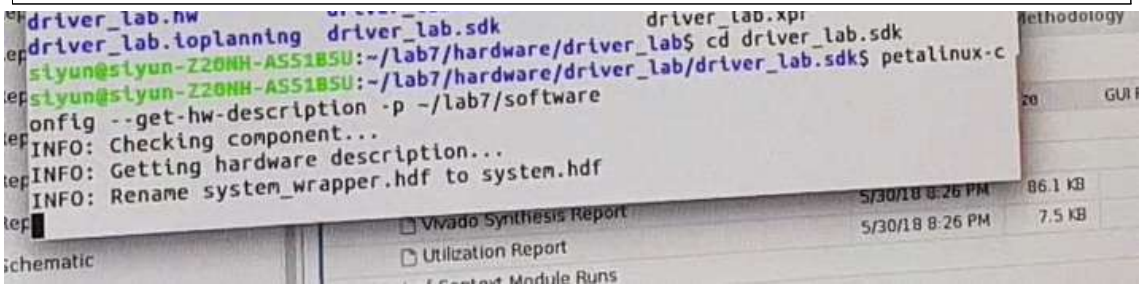


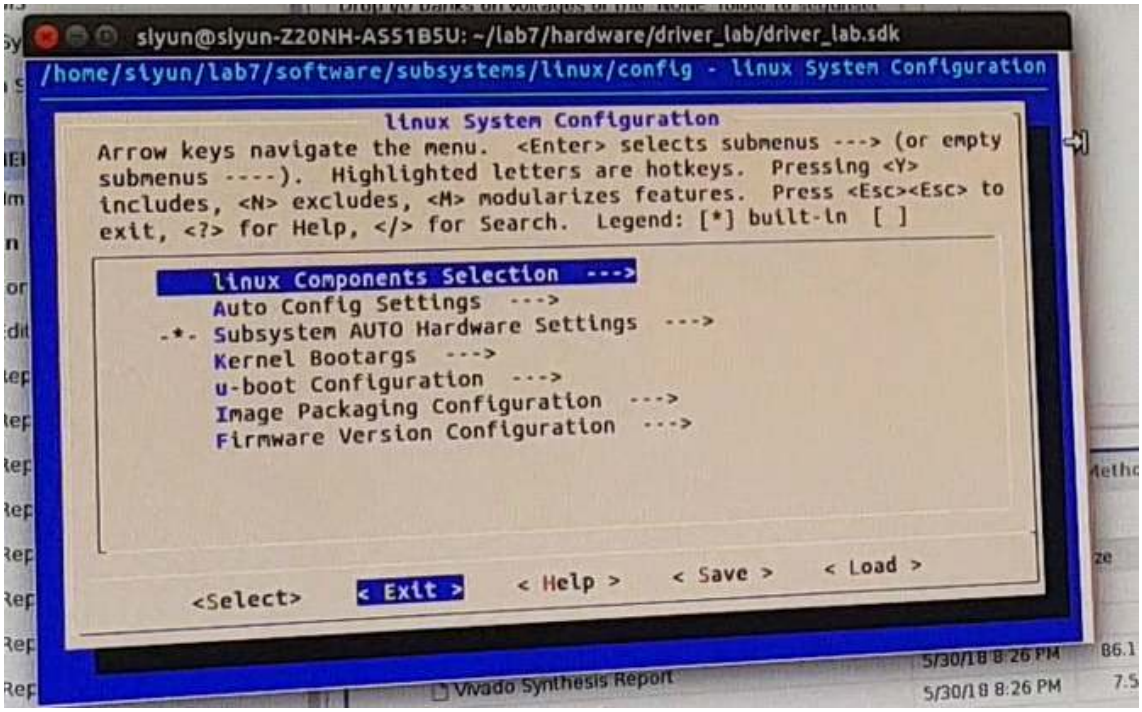
이제 소프트웨어도 생성해주어야한다.

```
cd ~/lab7

petalinux-create -t project -n software --template zynq
```

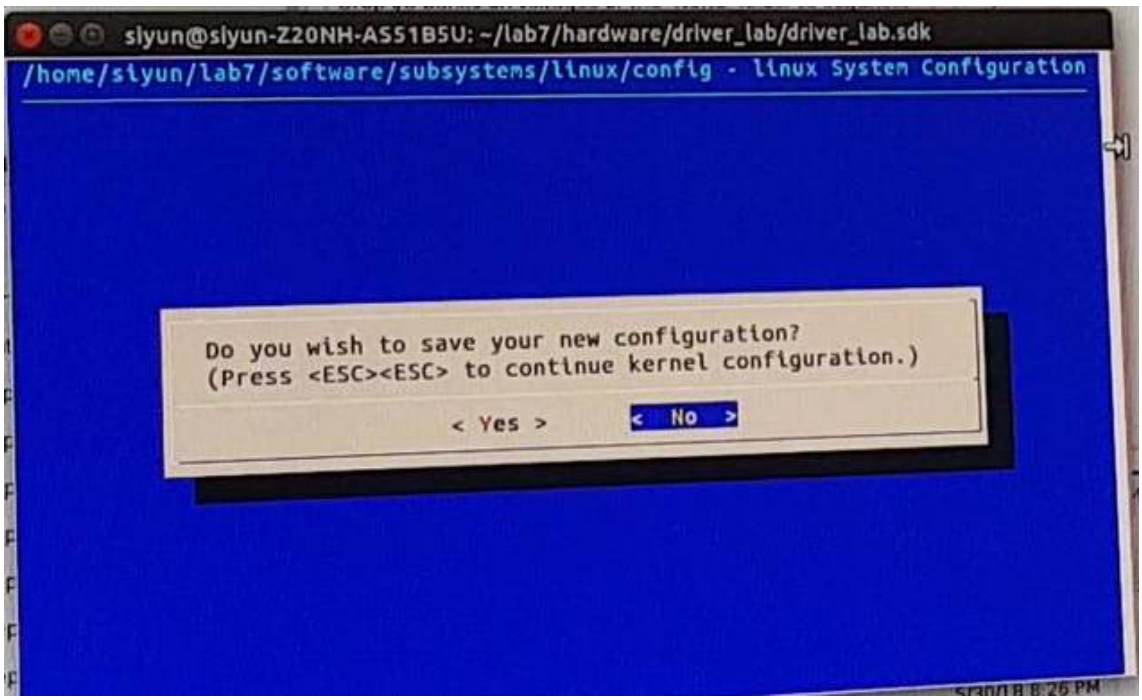



```
cd ~/lab7/hardware/drever_lab/driver_lab.sdk

petalinux-config --get-hw-description -p ~/lab7/software
```





Exit 하고 엔터



No 하고 엔터

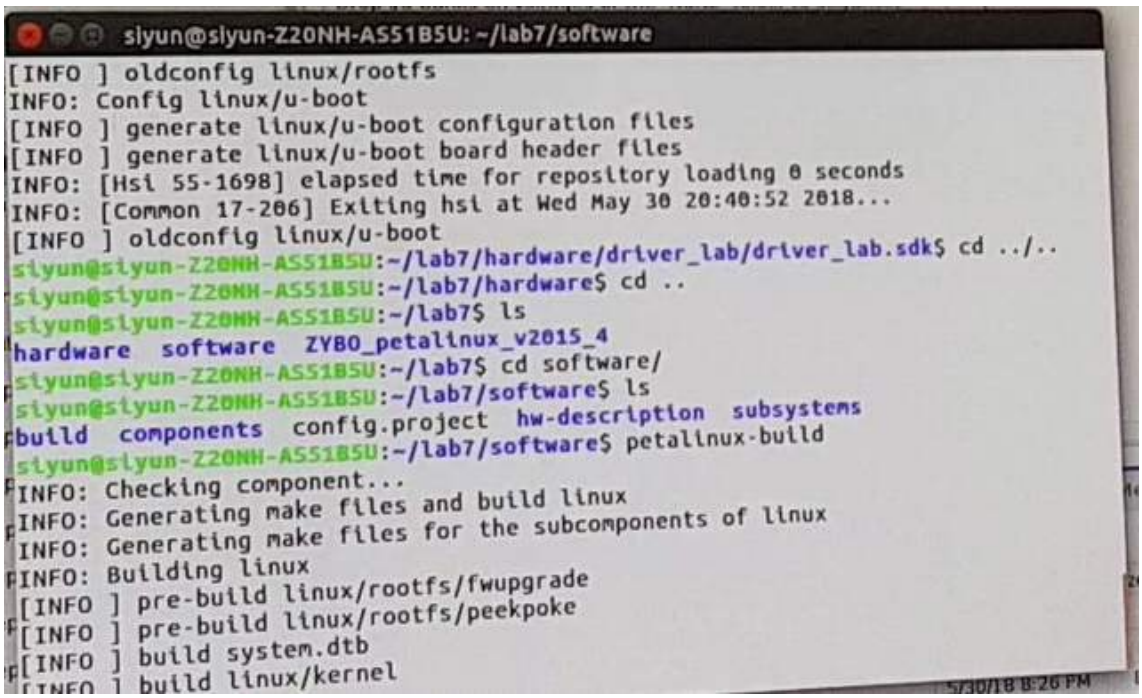
cd ~/lab7/software

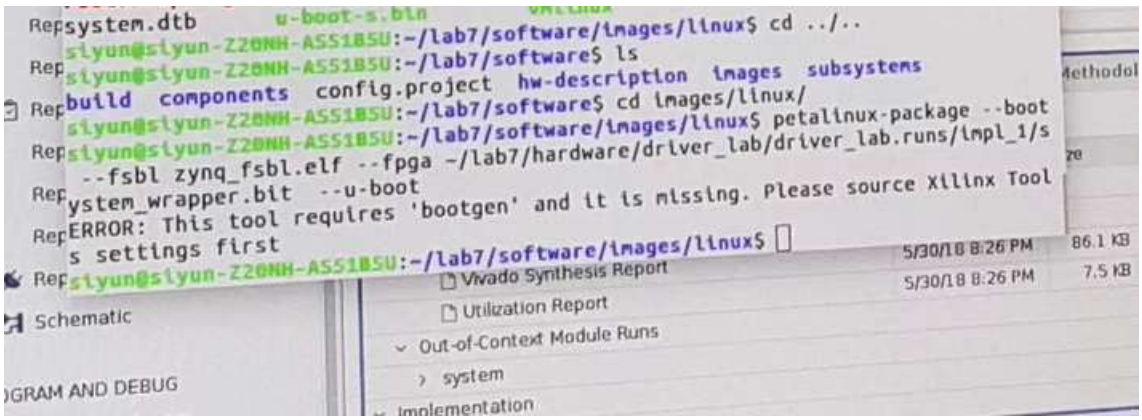
petalinux-build

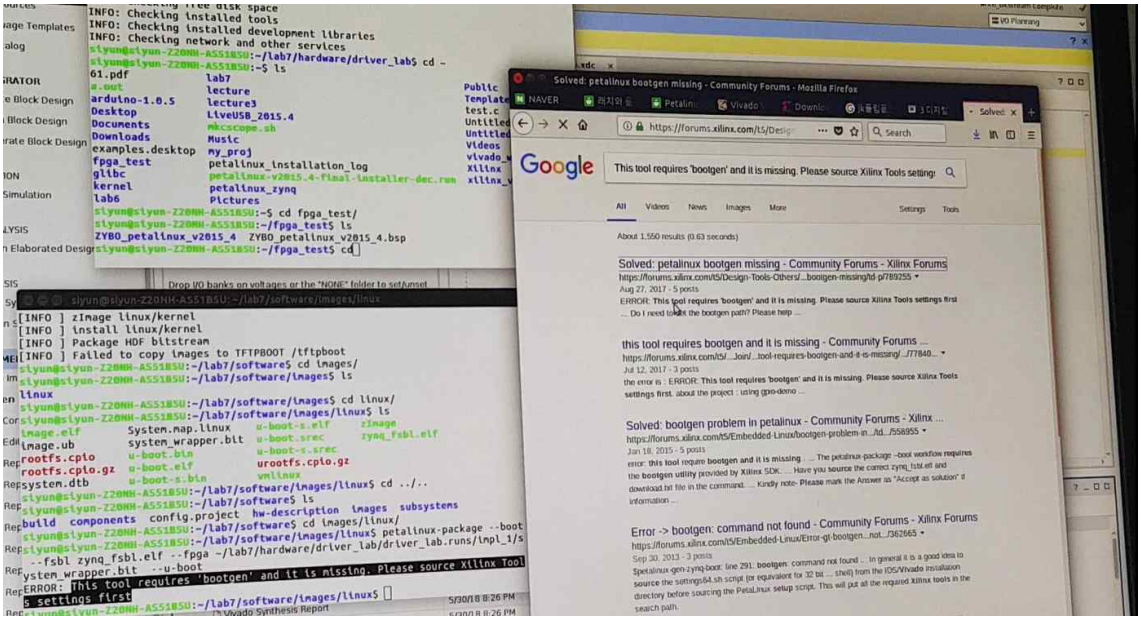
cd ~/lab7/software/images/linux

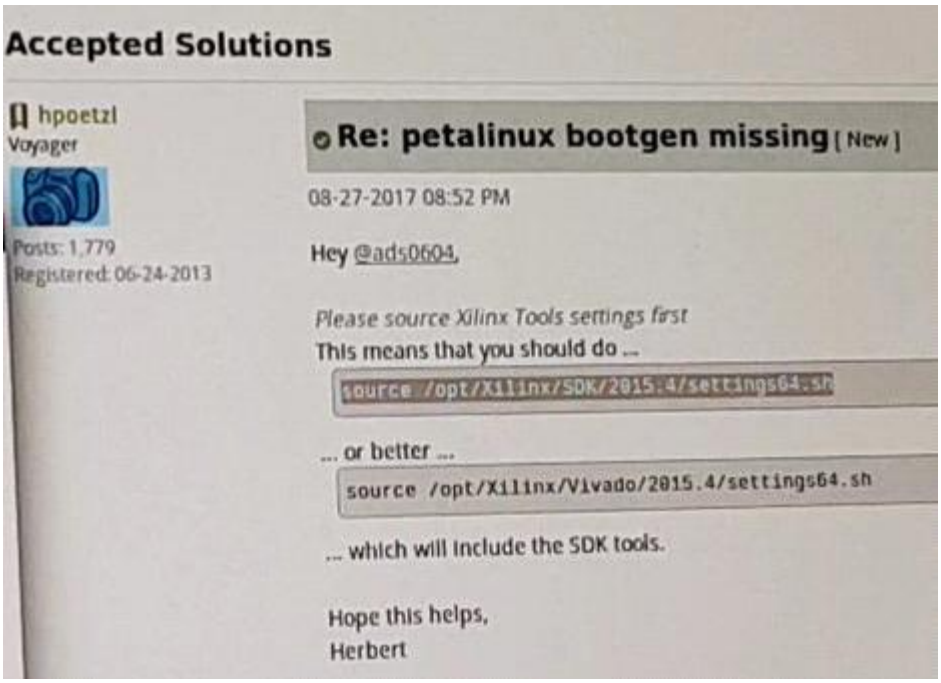
petalinux-package --boot --fsbl zynq_fsbl.elf --fpga ~/lab7/hardware/driver_lab.runs/impl_1/system_wrapper.bit --u-boot

이제 bootgen error 뜨면 구글에 검색해서 하면 끝!!!



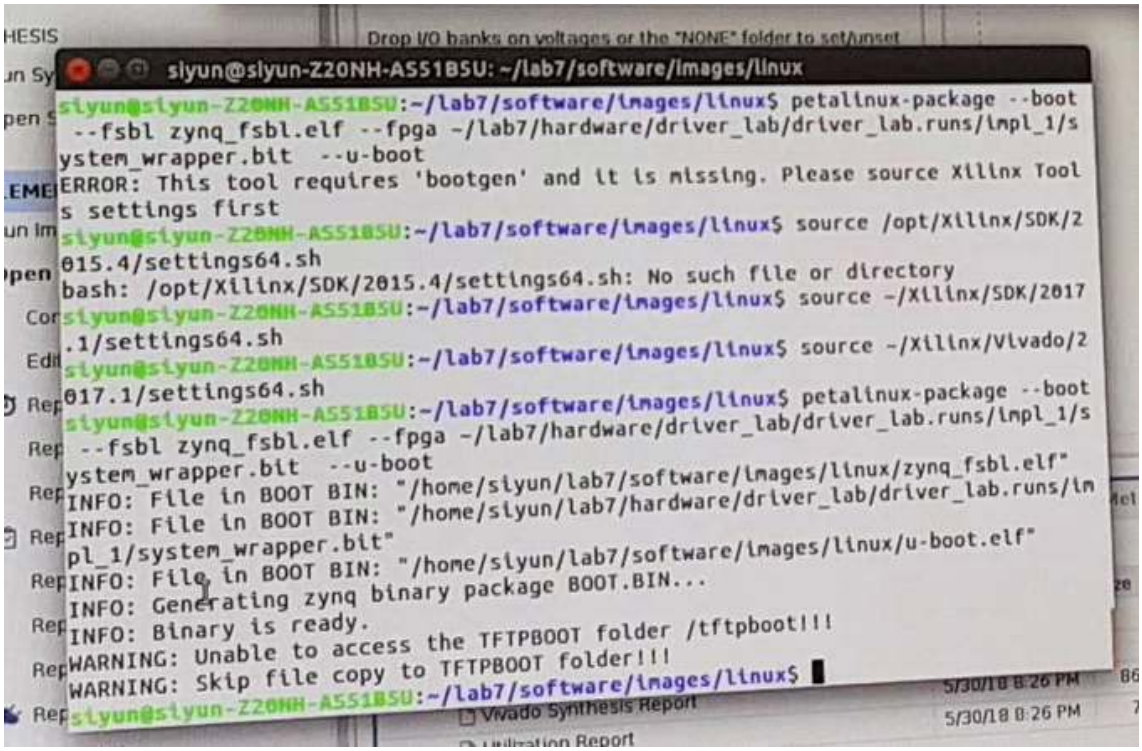






부트젠에러 해결할 때 경로설정 주의!!!

```
source ~/Xilinx/SDK/2017.1/settings64.sh

source ~/Xilinx/Vivado/2017.1/settings64.sh

petalinux-package --boot --fsbl zynq_fsbl.elf --fpga ~/lab7/hardware/driver_lab.runs/impl_1/system_wrapper.bit --u-boot
```


Drop I/O banks on voltages or the "NONE" folder to set/unset
siyun@siyun-Z20NH-AS51B5U: ~/lab7/software/images/linux
siyun@siyun-Z20NH-AS51B5U:~/lab7/software/images/linux$ petalinux-package --boot
--fsbl zynq_fsbl.elf --fpga ~/lab7/hardware/driver_lab/driver_lab.runs/impl_1/system_wrapper.bit --u-boot
ERROR: This tool requires 'bootgen' and it is missing. Please source Xilinx Tools settings first
siyun@siyun-Z20NH-AS51B5U:~/lab7/software/images/linux$ source /opt/Xilinx/SDK/2015.4/settings64.sh
bash: /opt/Xilinx/SDK/2015.4/settings64.sh: No such file or directory
siyun@siyun-Z20NH-AS51B5U:~/lab7/software/images/linux$ source ~/Xilinx/SDK/2017.1/settings64.sh
siyun@siyun-Z20NH-AS51B5U:~/lab7/software/images/linux$ source ~/Xilinx/Vivado/2017.1/settings64.sh
siyun@siyun-Z20NH-AS51B5U:~/lab7/software/images/linux$ petalinux-package --boot
--fsbl zynq_fsbl.elf --fpga ~/lab7/hardware/driver_lab/driver_lab.runs/impl_1/system_wrapper.bit --u-boot
INFO: File in BOOT BIN: "/home/siyun/lab7/software/images/linux/zynq_fsbl.elf"
INFO: File in BOOT BIN: "/home/siyun/lab7/hardware/driver_lab/driver_lab.runs/impl_1/system_wrapper.bit"
INFO: File in BOOT BIN: "/home/siyun/lab7/software/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!!!
siyun@siyun-Z20NH-AS51B5U:~/lab7/software/images/linux$
Vivado Synthesis Report
5/30/18 8:26 PM
Utilization Report
5/30/18 8:26 PM