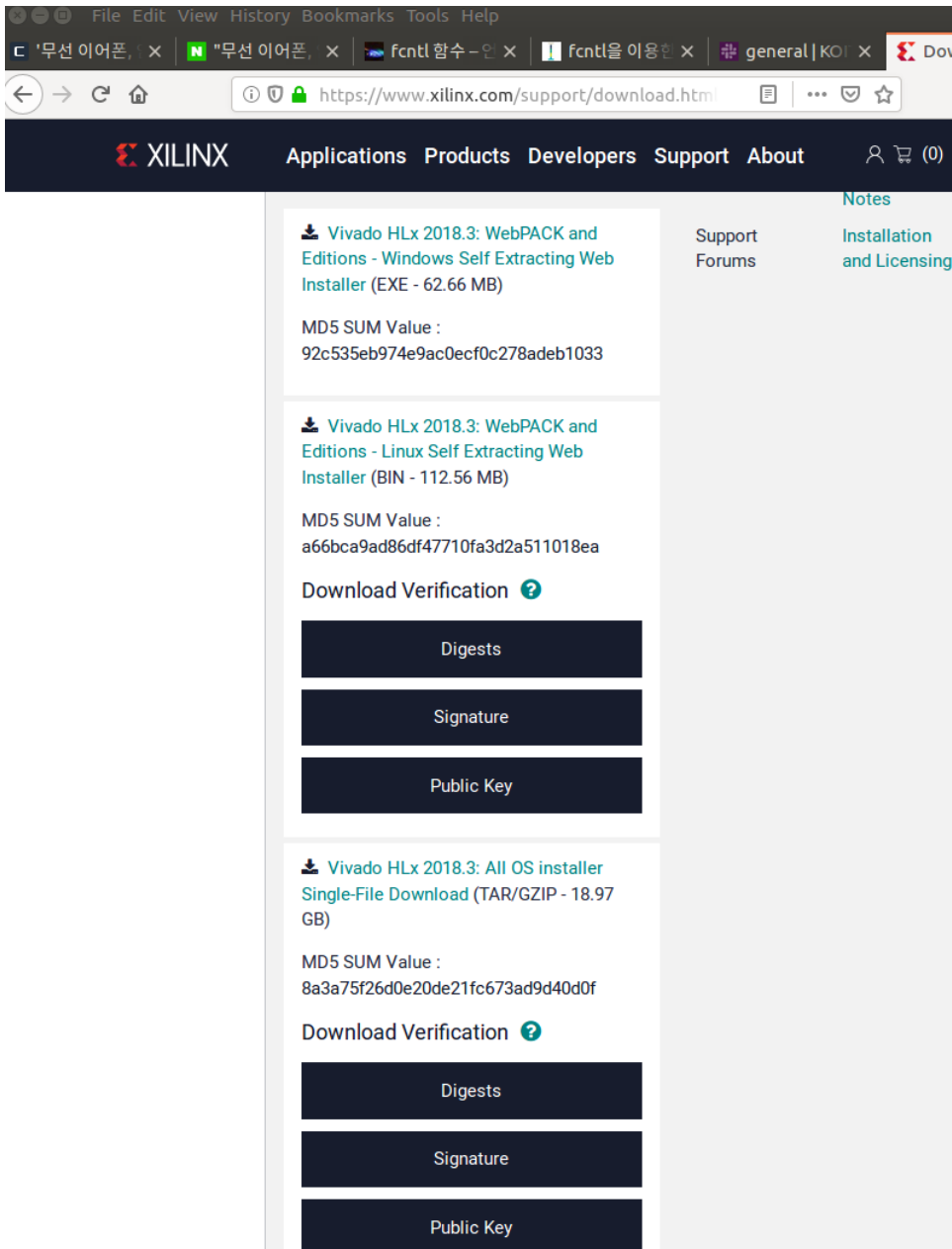


[1 일차 : 사용 환경 구성(VIVADO 설치)]

[다운로드 및 설치]

Xilinx.com → support → download



다운 1 : Vivado - all installer ?

다운 2 : SDSoc - 너무 최신 버전이 아닌 애로 다운로드. Sdx 2018.2.SFD

다운 3 : Embedded Development – 2017.4(old) PetaLinux

다운 4 : 위의 PetaLinux – sstate cache 29GB (압축해제 해놓으면, build 빨리 됨)

SDSoc 다운 및 설치

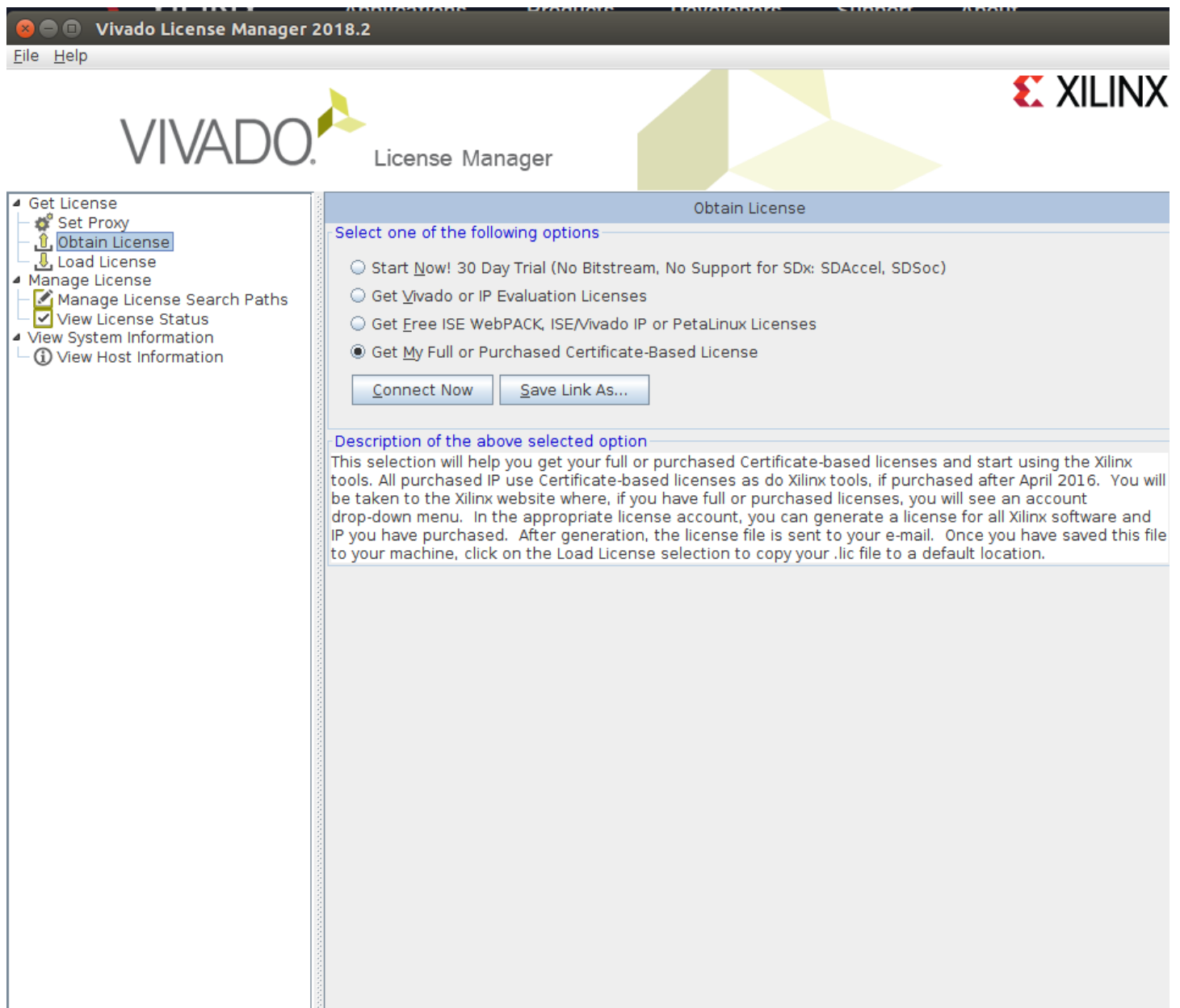
설치 :

1. 압축해제. `tar -zxvf ./파일명.tar.gz`
2. 설치실행. 폴더안으로 들어가서, `sudo ./xsetup` 하면 설치창 뜬다.
3. 용량되면, 옵션 모두 선택해 설치한다.

<라이선스 설정 주의사항 : 설치 시 라이선스하지말고, 나중에 실행해서 Load license 해야, view license 에 보인다. >

라이선스 설정→ 설치창 다음에 나오는데

obtain license - connect now 클릭한다. 온라인 안 가지면, save link as 하고 실행해서 웹페이지 이동한다.



No charge 다 클릭해서 license file 만든다. Mac address 확인하고 맞는 애로 선택한다.

```
koitt@koitt:~$ ifconfig
enp0s31f6 Link encap:Ethernet  HWaddr a4:4c:c8:49:44:57
        inet addr:192.168.0.101  Bcast:192.168.0.255  Mask:255.255.255.0
        inet6 addr: fe80::d8b:ab5:27cb:7ddd/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:20288099 errors:0 dropped:0 overruns:0 frame:0
        TX packets:10704829 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:30530272085 (30.5 GB)  TX bytes:792389346 (792.3 MB)
        Interrupt:16 Memory:ef200000-ef220000
```

Create New Licenses

Manage Licenses

Host Name	Host Type	Host ID	License Type	OS	Created By	Created Date
koitt	Ethernet MAC	a44cc8494457	Node	Linux 64-bit	GODRIC HONG	19 MAR 2019

Congratulations

Your new license file has been successfully generated and e-mailed to boris11036@gmail.com. You can also view the license file under the Manage Licenses tab.

Please add this sender (xilinx.notification@entitlenow.com) to your address book.

License File Details

Node License

Host ID: a44cc8494457

Products

Xilinx MicroBlaze/All Programmable SoC Software Development Kit – Standalone (No Charge): 1 seats

ISE WebPACK License (No Charge): 1 seats

Vivado Design Suite: HL WebPACK 2015 and Earlier License (No Charge): 1 seats

ISE Embedded Edition License (No Charge): 1 seats

Comments

Description

Activated Seats

1

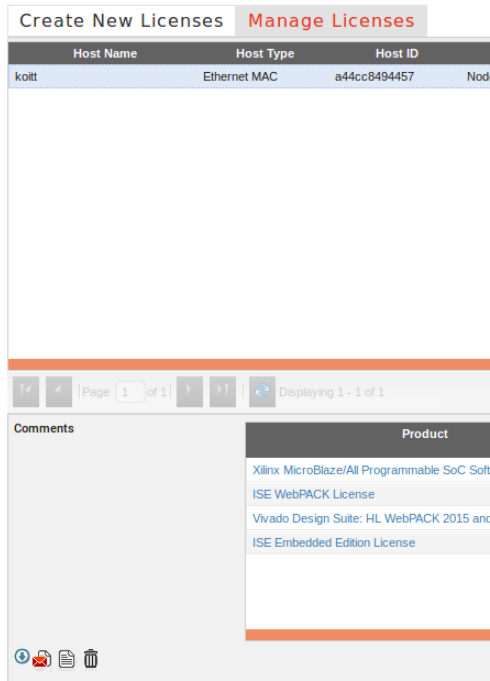
1

1

1

Modify License

라이선스 다운받는다.



다운받은 라이선스를 이제 VIVADO License Manager – Load License 통해 로딩한다.

확인 : View License Status

<vivado 타이핑만으로 실행되게 만들기>

1. 설치경로 확인 : /opt/Xilinx/Vivado/2018.2/

2. ~/.bashrc 들어가기 : vi ~/.bashrc

3. 밑에 라인 추가

```
# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi

# Enable Xilinx Vivado 2018.2
PATH=$PATH:/opt/Xilinx/Vivado/2018.2/bin
export PATH
```

4. 터미널 바탕화면 우클릭으로 실행, vivado 치면 자동 실행된다. (안될경우 터미널 다시 켜서 해 보자)

주의사항 : /opt 경로 권한 문제로 잘 안 될 경우 ?

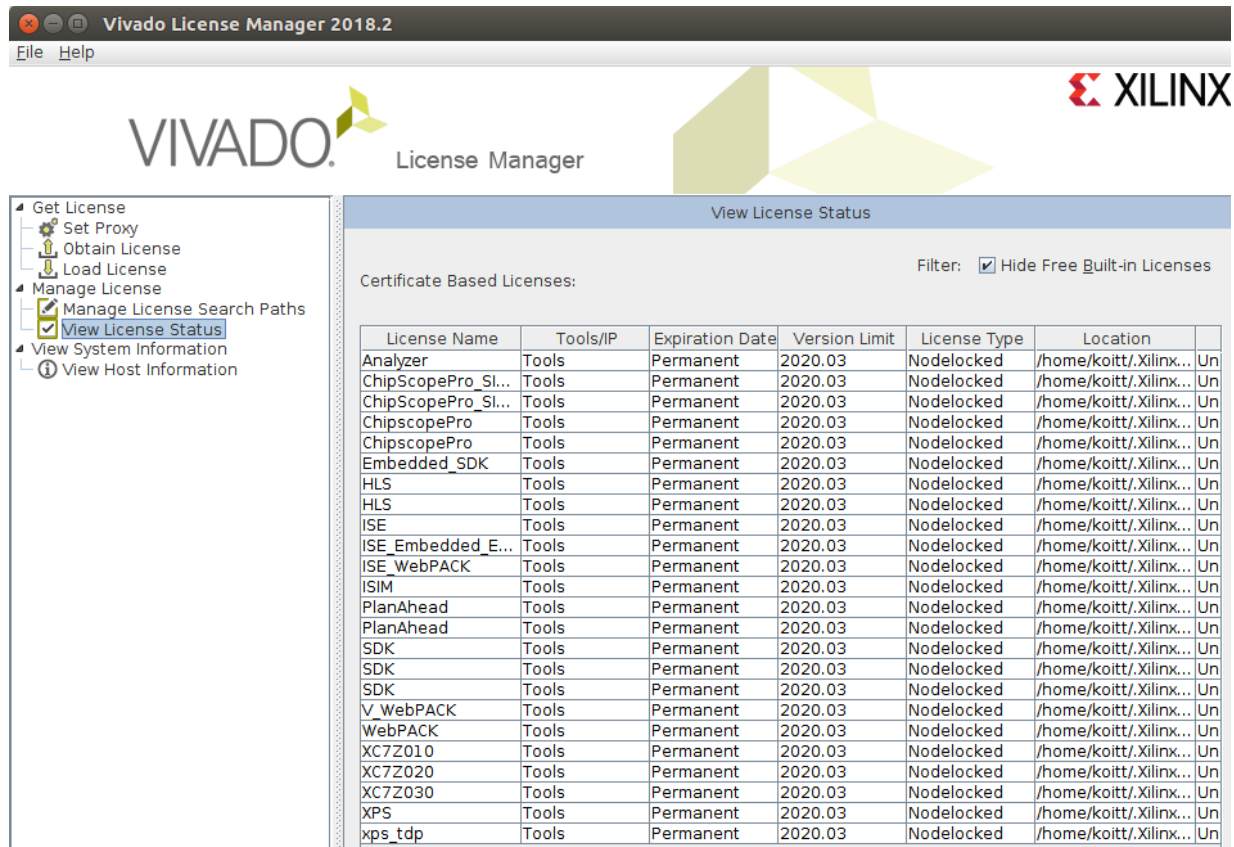
sudo chmod -R 755 /opt

sudo chgrp -R 유저이름 /opt

sudo chown -R 유저이름 /opt 그다음에 위 순서대로 다시 시도 해 본다.

실행되면, help – manage license 들어가 라이선스 Load 한다.

View License 해서 아래처럼 보여야한다.



<참고>

PETA LINUX 있고 거기에

SSTATE 쓰면, 컴파일 속도가 없는것에비해 10 배빨라짐.

<참고> ip 관련 사항 때문에, 그것 수천에서 억까지 됨 이거 돈 안내려고

Verilog

vhdl 국책 벌루ㅏㅏㅏ

[2 일차]

- * LED 켜기
- * 페타리눅스 깔기 install petalinux

[LED 켜기] : zybo_z7 ip 받기+경로에 풀기, LED ON,OFF

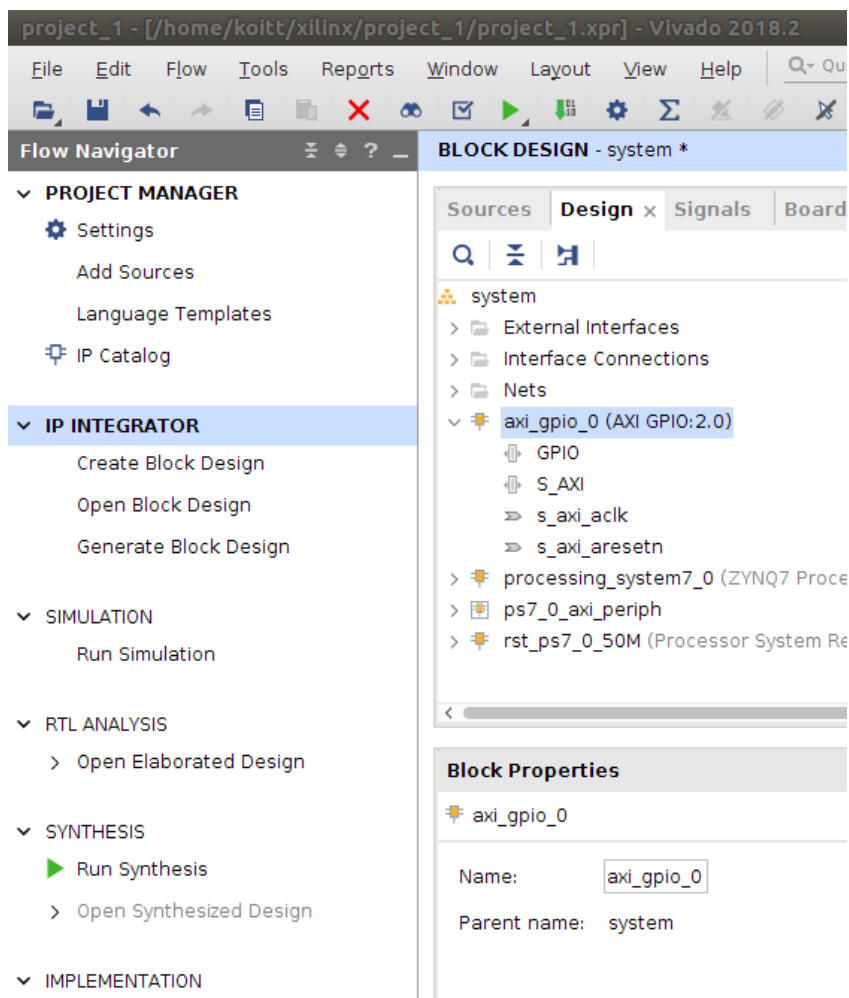
다운 1: vivado-boards-master 받고 압축 해제.

다운 2: Petalinux-Zybo-Z7-10-2017.4-1.bsp

vivado - new project – next next 해서 constraints 에 추가(zybo_z7) 하고 다음에 board 에서 zybo_z7 있어야한다. 없으면, vivado-boards-master 의 new 에 board_files 안의 여러 폴더를 설치경로 – 2018.2/data/boards/board_files 에 추가해준다.
sudo cp -r ./new/board_files/* /opt/Xilinx/Vivado/2018.2/data/boards/board_files

만일 권한 문제 생길경우 - sudo chmod -R 755 /opt

다시 vivado 켜서 위의 newfile 부터 진행한다.



< led 용 new project >

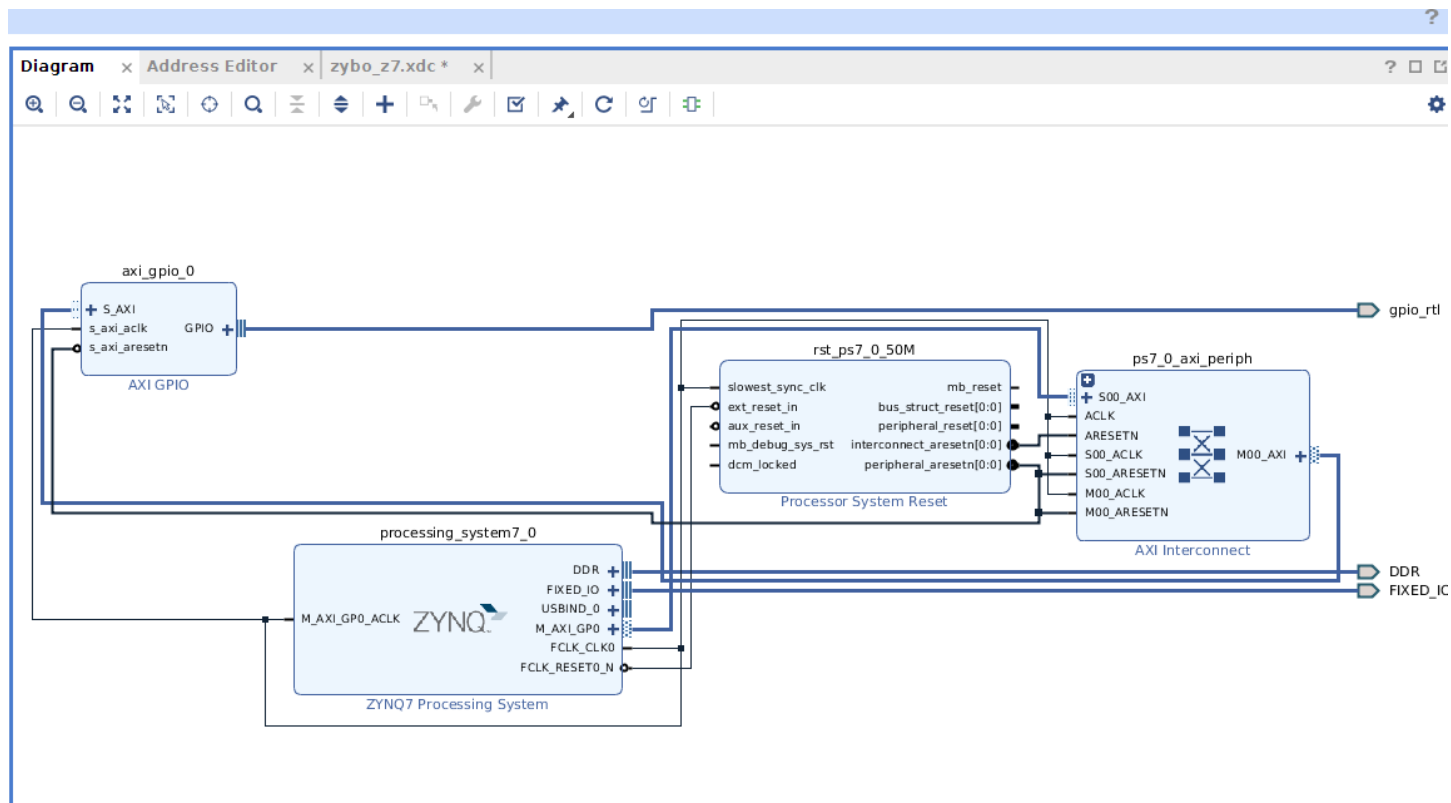
들어와서, create block design – system 이라 친다.

Diagram 창에 + 누르고, zy 치면 하나 나온다. 클릭.

그리고 add 해서 gpio 누른다.

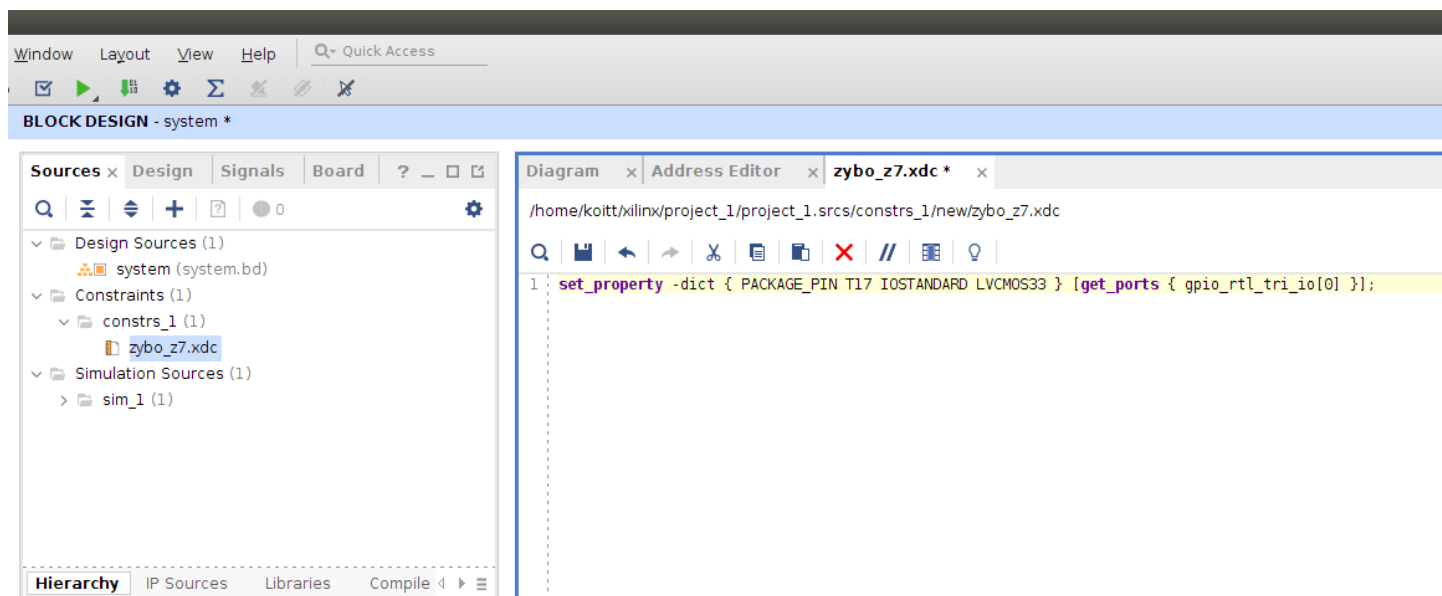
그리고 위에 auto ~~ 눌러서 모두 선택 후 gpio 는 옵션에 custom 으로 하고 enter 한다.

마지막으로 gpio 블록 더블클릭 – ip config – gpio width → 32 를 1 로 바꿔준다.



Refresh 한다.

< constraints add > (new project 에서도 추가할 수 있음)
만약 없으면, constraints – add 해서 추가할 수 있다.



```
set_property -dict { PACKAGE_PIN T17 IOSTANDARD LVCMOS33 } [get_ports { gpio_rtl_tri_io[0] }];
```

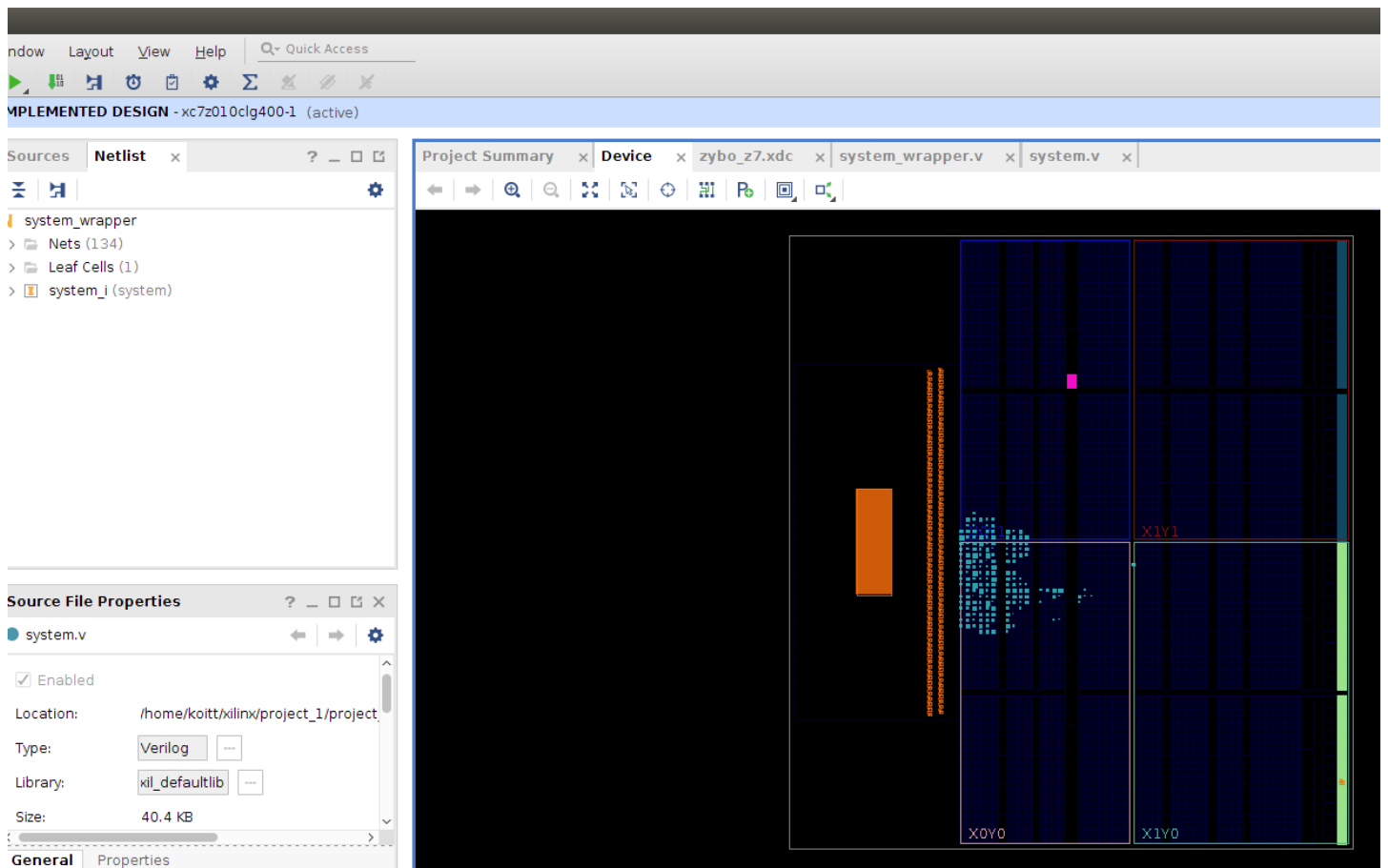
pin 17 을 io standard 로 3.3v 로 gpio 0 으로 쓰겠다 는 의미.

design source - 우클릭 - new wrapper - ok -ok

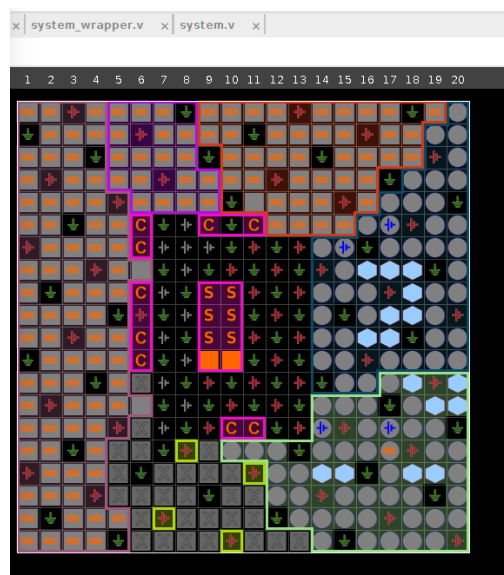
왼쪽에 Run Synthesis(합성) ok ok → 오른쪽 위 작게 동글뱅이 다 끝나고 다른 작업해야함 안그러면꼬임.

끝나면, 또 ok - ok 한다. 오른쪽 위 작게 동글뱅이 다 끝나고 다른 작업해야함 안그러면꼬임.

끝나면, 원래는 바로 bit stream 해도되는데, 일단 한번 보는게 좋으니까, 그대로 open implemented design



오른쪽 창에 - I/o planning 들어간다.



Tcl Console	Messages	Log	Reports	Design Runs	Timing	Power	Methodology	Package Pins	I/O Ports	x
Name	Direction	Board Part Pin	Board Part Interface	Neg Diff Pair	Package Pin	Fixed	Bank	I/O Std		
▼ All ports (131)										
> DDR_12642 (71)	INOUT					✓	502	(Multiple)*		
> FIXED_IO_12642 (59)	INOUT					✓	(Multiple)	(Multiple)*		
▼ gpio_rtl_12642 (1)	INOUT					✓	34	LVC MOS33*		
▼ gpio_rtl_tri_io (1)	INOUT					✓	34	LVC MOS33*		
gpio_rtl_tri_io[0]	INOUT				T17	✓	34	LVC MOS33*		
Scalar ports (n)										

T17, LVCMOS33 확인. 안되어있는데 계속하면 보드 고장날 수 있음.

그다음, 왼쪽 창에 generate – bitstream 누른다. 오른쪽 위 작게 동글뱅이 기다린다.

끝나면, ok 누르지말고 창 닫는다.

그리고 꼭! 해야될일이있음. File – export- export hardware – include bitstream 하고 ok 한다.

그다음, file – launch sdk

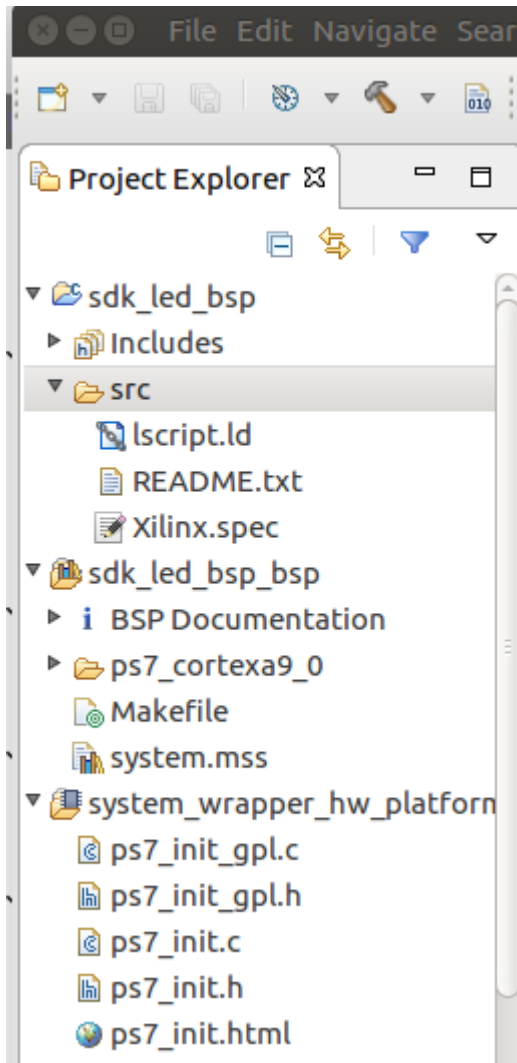
새 창이 뜬다.

The screenshot shows the Vivado IDE interface. On the left, the Project Explorer displays the file structure for 'system_wrapper_hw_platform_0', including files like ps7_init_gpl.c, ps7_init.c, ps7_init.h, ps7_init.html, ps7_init.tcl, system_wrapper.bit, and system.hdf. The main window shows the 'system.hdf' file, which contains the 'system_wrapper_hw_platform_0 Hardware Platform Specification'. Under 'Design Information', it lists the Target FPGA Device as 7z010, Part as xc7z010clg400-1, Created With as Vivado 2018.2, and Created On as Wed Mar 20 11:15:56 2019. Below this, the 'Address Map for processor ps7_cortexa9_[0-1]' is shown as a table.

Cell	Base Addr	High Addr	Slave I/f
ps7_intc_dist_0	0xf8f01000	0xf8f01fff	
ps7_gpio_0	0xe000a000	0xe000afff	
ps7_scutimer_0	0xf8f00600	0xf8f0061f	
ps7_slcr_0	0xf8000000	0xf8000fff	
axi_gpio_0	0x41200000	0x4120ffff	S_AXI
ps7_scuwdt_0	0xf8f00620	0xf8f006ff	

Part 가 위와 동일한지확인한다. 우리쓰는 Fpga Cpu 명임.

File – new project(이름 sdk_led 치면 됨) – empty application



위 그림의 src 에서 우클릭 – new – sourcefile main.c 만들고, 아래 코드 친다.

```
#include <stdio.h>
#include <xgpio.h>
#include "xparameters.h"
#include "sleep.h"

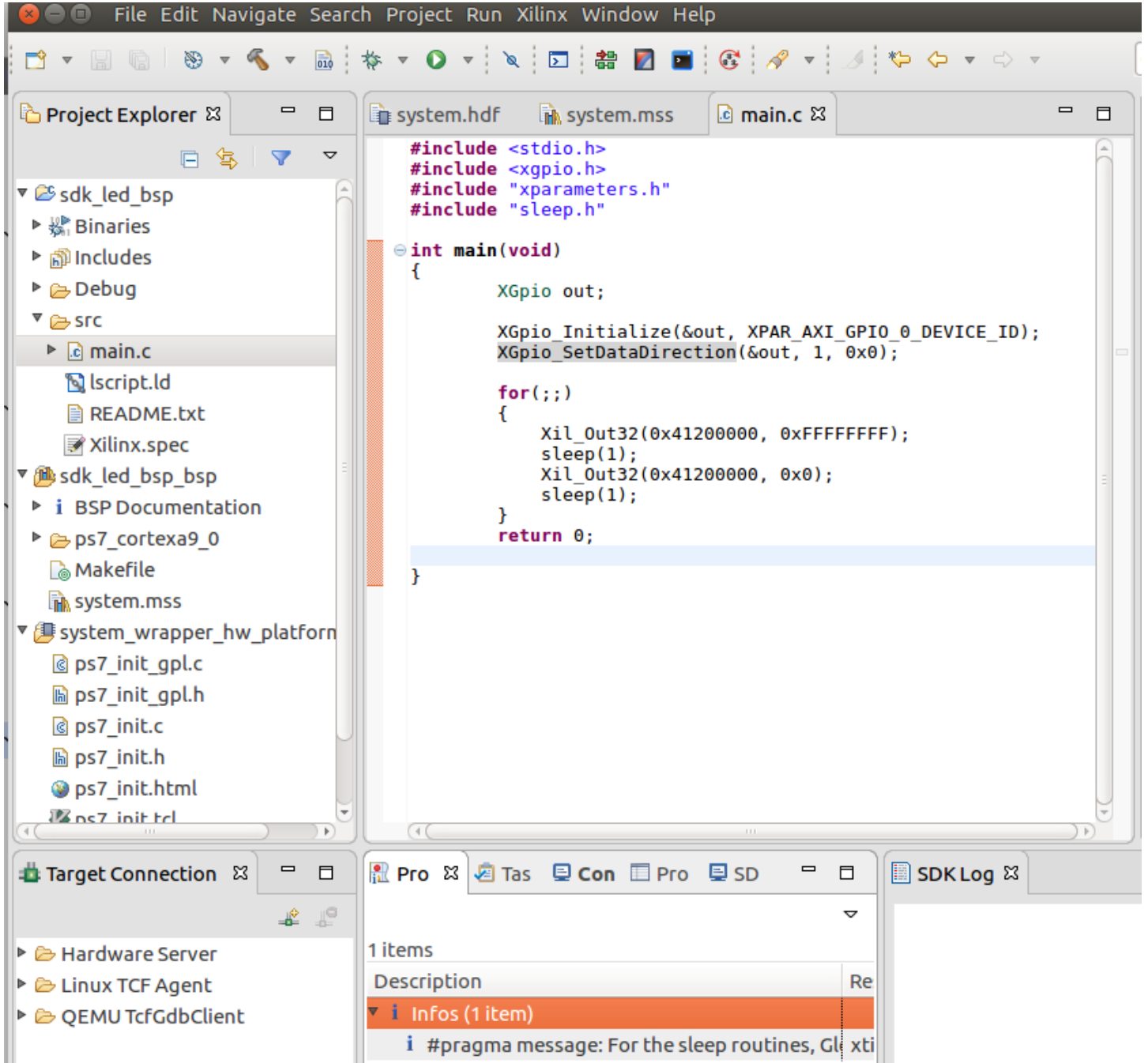
int main(void)
{
    XGpio out;
```

```

XGpio_Initialize(&out, XPAR_AXI_GPIO_0_DEVICE_ID);
XGpio_SetDataDirection(&out, 1, 0x0);

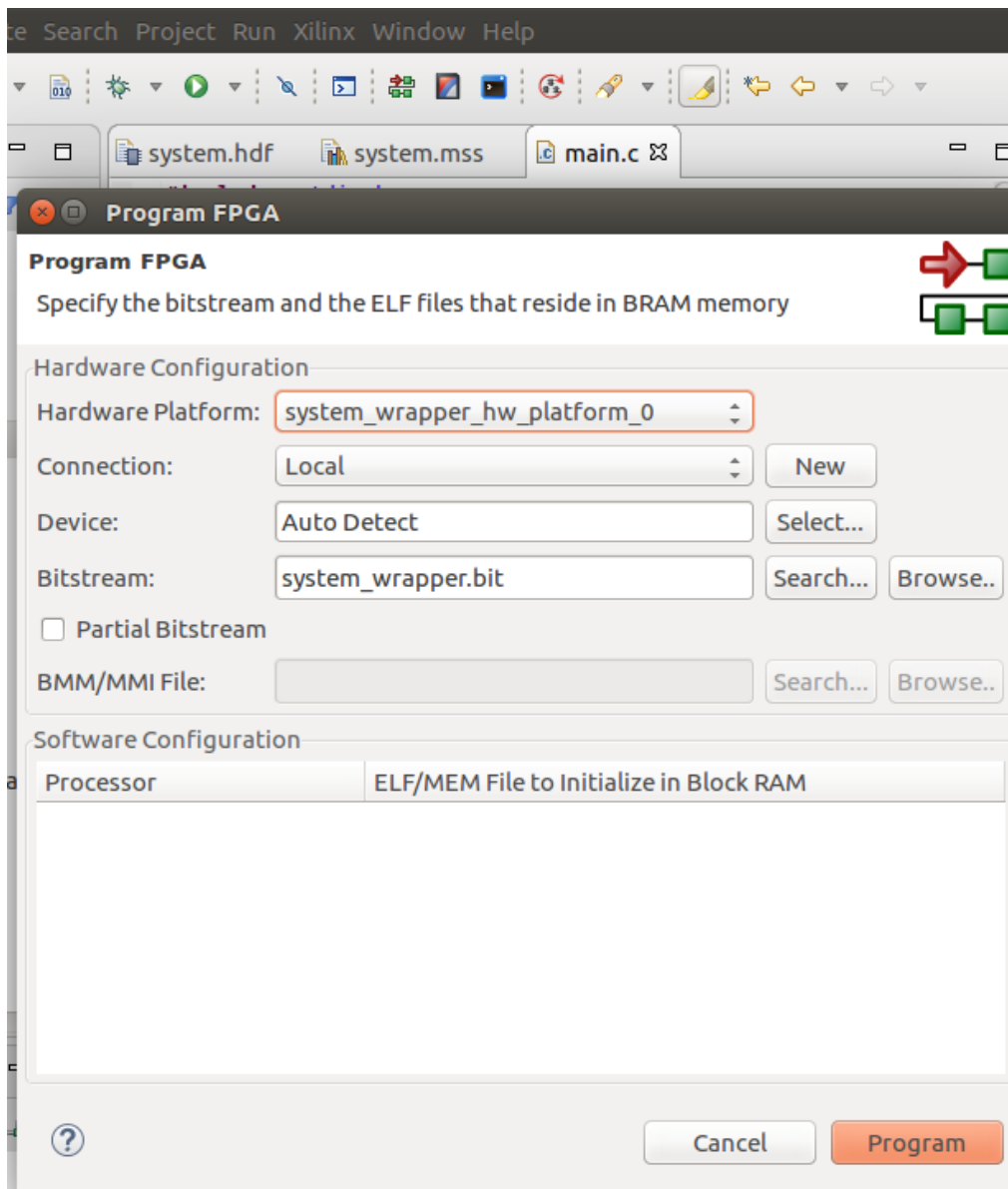
for(;;)
{
    Xil_Out32(0x41200000, 0xFFFFFFFF);
    sleep(1);
    Xil_Out32(0x41200000, 0x0);
    sleep(1);
}
return 0;
}

```



망치모양(build)하기전에 저장. 해야 에러 안남

program fpga



program 누르면, 초록불 올라온다. (초록불 없어지면안됨)

run – launch on hardware 클릭

그다음에 옆에 log successful 어쩌고 뜨면, sdk 끈다. vivado 끈다.

전원 버튼 off 로 내린다.

이제 led 연결 할 것이다.

<https://reference.digilentinc.com/reference/programmable-logic/zybo-z7/reference-manual>

(these are described in the “Power supplies” section).

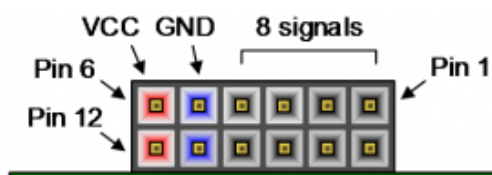


Figure 16.1. Pmod port

Digilent produces a large collection of Pmod accessory boards that can attach to the Pmod ports to add ready-made functions like A/D's, D/A's, motor drivers, sensors, and other functions. See www.digilentinc.com for more information. The vivado-library repository on the [Digilent Github](#) contains pre-made IP cores for many of these Pmods that greatly reduces the work of integrating them into your project. See the “Using Pmod IP” tutorial on the Zybo Z7 Resource Center for help using them.

The Zybo Z7 has six Pmod ports, some of which behave differently than others. Each Pmod port falls into one of four categories: standard, MIO connected, XADC, or high-speed. Table 16.1 specifies which category each Pmod port falls into, and also lists the Zynq pins they are connected to. The sections that follow describe the different types of Pmods.

	Pmod JA	Pmod JB*	Pmod JC	Pmod JD	Pmod JE	Pmod JF
Pmod Type	XADC	High-Speed	High-Speed	High-Speed	Standard	MIO
Pin 1	N15	V8	V15	T14	V12	MIO-13
Pin 2	L14	W8	W15	T15	W16	MIO-10
Pin 3	K16	U7	T11	P14	J15	MIO-11
Pin 4	K14	V7	T10	R14	H15	MIO-12
Pin 7	N16	Y7	W14	U14	V13	MIO-0
Pin 8	L15	Y6	Y14	U15	U17	MIO-9
Pin 9	J16	V6	T12	V17	T17	MIO-14
Pin 10	J14	W6	U12	V18	Y17	MIO-15

T17 → Pmod JE 의 Pin 9 임을 확인했다.

이제 fpga 에 hw 연결한다.

1. vivado 실행
2. sdk 실행
3. program FPGA
4. Run

이건 RAM 에 올라가서 프로그램 키고 넣어야한다.

<참고>

보드구매 <https://www.inipro.net/>

zybo z7 10

zybo z7-20

<참고>

오늘 진행 한 LED 키는 과정만 외국에서는 \$1200 과정으로 돌리고있음.

< zynq 7010 development board zybo z7 전원 관리방법 >

방법 1 : dc 잭 은 5V 2.5A 이고 어댑터 기본제공이 아니라 없다.

방법 2 : 아래 사진의 micro usb 포트로 전원 공급한다. 지금은 단순 LED 니까 전류많이요구하지않는다. 동작 가능.

아래 보이는 스위치로 전원 ON, OFF 한다. (주의 : 코드 동작하다가 다 끄고싶을때, 프로그램끄고 스위치 내린다. 반대로 하면 꼬이면서 맛이 갈 수 있다.)

