

Xilinx Zynq FPGA, TI DSP, MCU 기반 의 회로 설계 및 임베디드 전문가 과정

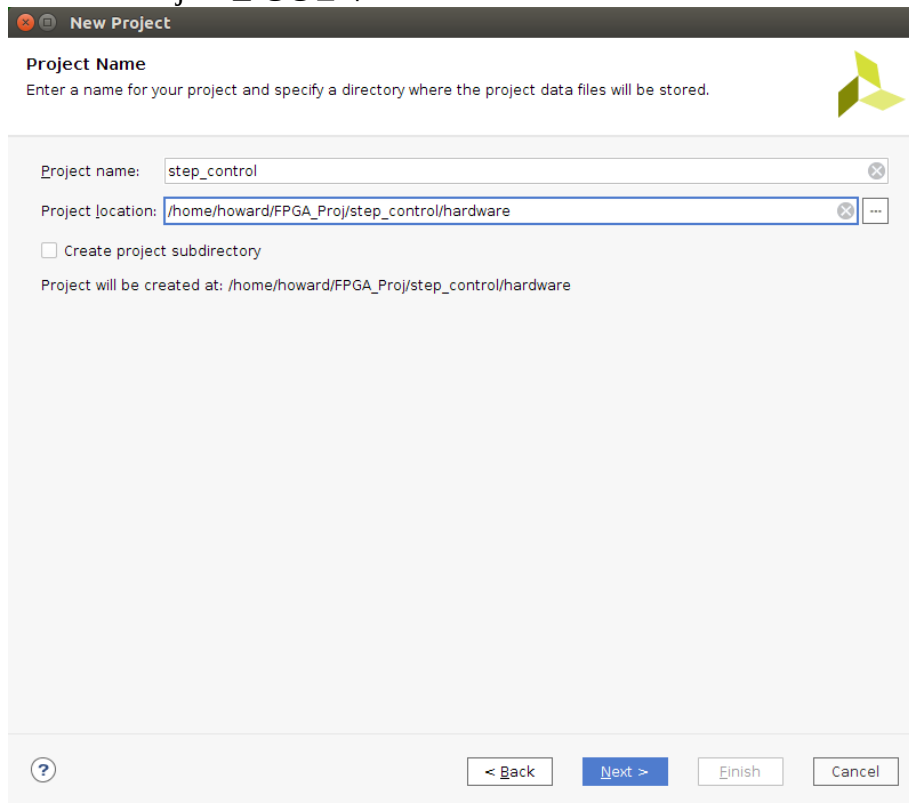
강사 - Innova Lee(이상훈)

gcccompil3r@gmail.com

학생 - Hyungjoo Kim(김형주)

mihaelkel@naver.com

1. Vivado Project 를 생성한다.



New Project

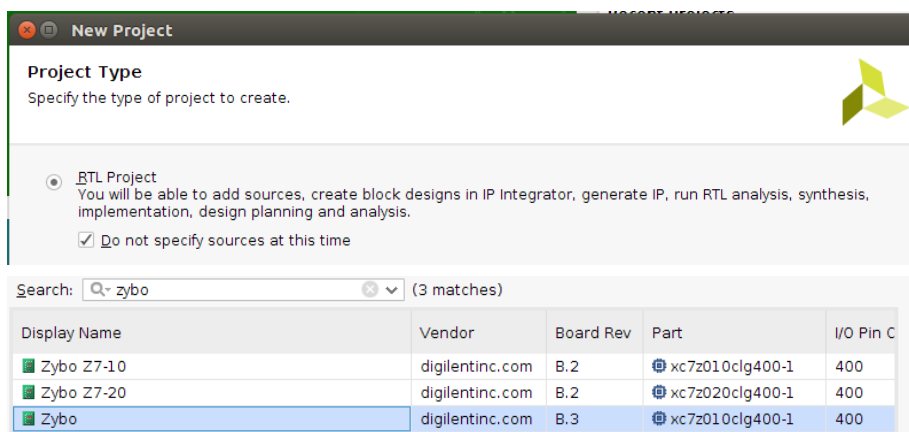
Project Name
Enter a name for your project and specify a directory where the project data files will be stored.

Project name:

Project location:

☐ Create project subdirectory

Project will be created at: /home/howard/FPGA_Proj/step_control/hardware



New Project

Project Type
Specify the type of project to create.

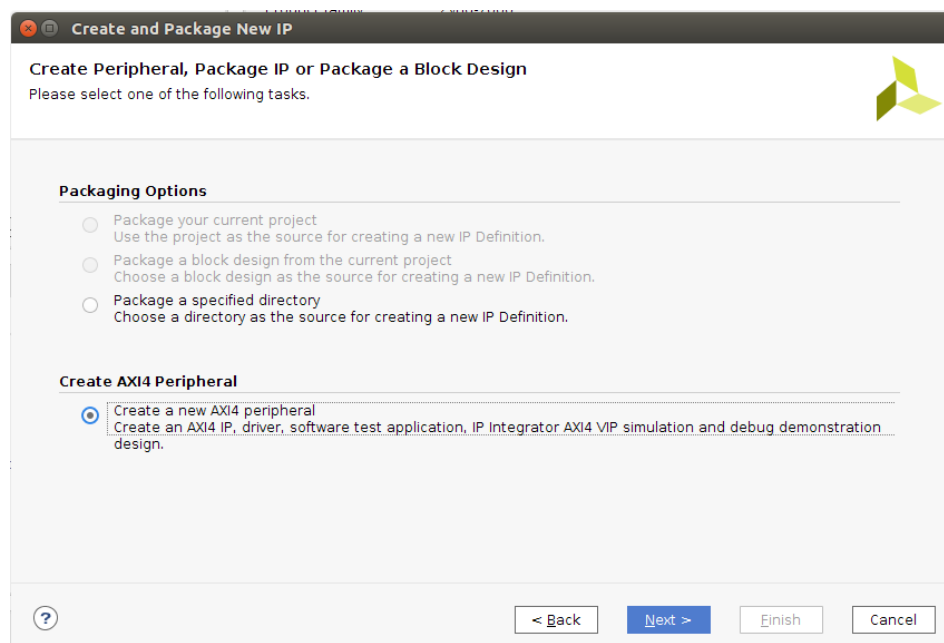
☒ **RTL Project**
You will be able to add sources, create block designs in IP Integrator, generate IP, run RTL analysis, synthesis, implementation, design planning and analysis.

☒ Do not specify sources at this time

Search: (3 matches)

Display Name	Vendor	Board Rev	Part	I/O Pin C
Zybo Z7-10	digilentinc.com	B.2	xc7z010clg400-1	400
Zybo Z7-20	digilentinc.com	B.2	xc7z020clg400-1	400
Zybo	digilentinc.com	B.3	xc7z010clg400-1	400

2. IP 를 설계한다. Tools → Create and Package New IP



Create and Package New IP

Create Peripheral, Package IP or Package a Block Design
Please select one of the following tasks.

Packaging Options

- ☐ Package your current project
Use the project as the source for creating a new IP Definition.
- ☐ Package a block design from the current project
Choose a block design as the source for creating a new IP Definition.
- ☐ Package a specified directory
Choose a directory as the source for creating a new IP Definition.

Create AXI4 Peripheral

- ☒ Create a new AXI4 peripheral
Create an AXI4 IP, driver, software test application, IP Integrator AXI4 VIP simulation and debug demonstration design.

Create and Package New IP

Peripheral Details

Specify name, version and description for the new peripheral

Name:

step_motor

Version:

1.0

Display name:

step_motor_v1.0

Description:

My new AXI IP

IP location:

/home/howard/FPGA_Proj/step_control/MY_PWM_Core_1.0

☒ Overwrite existing

Create and Package New IP

Add Interfaces

Add AXI4 interfaces supported by your peripheral

☐ Enable Interrupt Support

+

–

Interfaces

S00_AXI

S00_AXI

step_motor_v1.0

Name

S00_AXI

Interface Type

Lite

Interface Mode

Slave

Data Width (Bits)

32

Memory Size (Bytes)

64

Number of Registers

4

[4..512]

?

< Back

Next >

Finish

Cancel

3.하위 모듈에 출력 포트 4 개를 추가하고, pwm 로직을 설계한다.

Project Summary

Package IP - step_motor

step_motor_v1_0_S00_AXI.v

/home/howard/FPGA_Proj/step_control/MY_PWM_Core_1.0/step_motor_1.0/hdl/step_motor_v1_0_0_

Q

←

→

✂

//

💡

```

1
2  `timescale 1 ns / 1 ps
3
4  module step_motor_v1_0_S00_AXI #
5  (
6      // Users to add parameters here
7
8      // User parameters ends
9      // Do not modify the parameters beyond this line
10
11      // Width of S_AXI data bus
12      parameter integer C_S_AXI_DATA_WIDTH    = 32,
13      // Width of S_AXI address bus
14      parameter integer C_S_AXI_ADDR_WIDTH    = 4
15  )
16  (
17      // Users to add ports here
18      output wire pwm1,
19      output wire pwm2,
20      output wire pwm3,
21      output wire pwm4,

```

```

403 // Add user logic here
404 reg [31:0] counter;
405
406 always @( posedge S_AXI_ACLK)
407 begin
408     if(counter > slv_reg0 - 32'd1)
409         counter <= 32'd0;
410     else
411         counter <= counter + 32'd1;
412     end
413
414 assign pwm1 = counter > slv_reg0 / 5 && counter < slv_reg0 * 3 / 5 ? 1'b0 : 1'b1;
415 assign pwm2 = counter > slv_reg0 / 5 && counter < slv_reg0 * 3 / 5 ? 1'b1 : 1'b0;
416 assign pwm3 = counter > slv_reg0 * 2 / 5 && counter < slv_reg0 * 4 / 5 ? 1'b0 : 1'b1;
417 assign pwm4 = counter > slv_reg0 * 2 / 5 && counter < slv_reg0 * 4 / 5 ? 1'b1 : 1'b0;
418 // User logic ends
419
420 endmodule
421

```

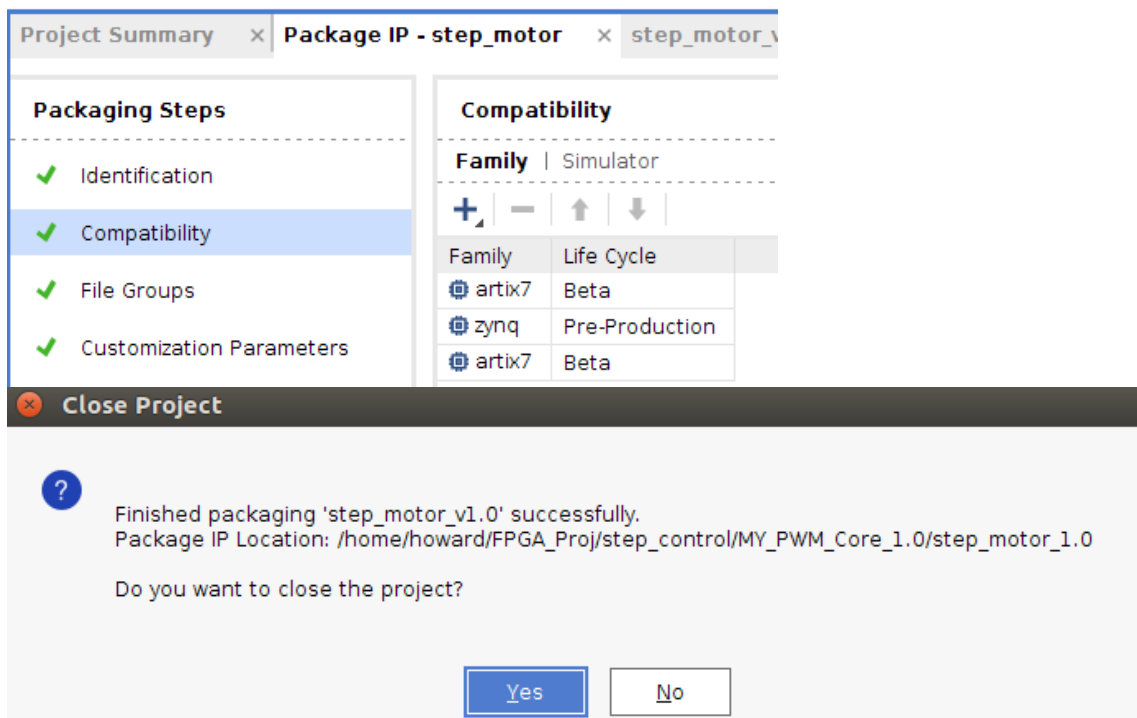
4.상위 모듈에 출력 포트 4 개를 추가하고, 하위 모듈과 연결한다.

```

17 // Users to add ports here
18 output wire pwm1,
19 output wire pwm2,
20 output wire pwm3,
21 output wire pwm4,
22 // User ports ends
23 // Do not modify the ports beyond this line
24
49 // Instantiation of Axi Bus Interface S00_AXI
50 step_motor_v1_0_S00_AXI # (
51     .C_S_AXI_DATA_WIDTH(C_S00_AXI_DATA_WIDTH),
52     .C_S_AXI_ADDR_WIDTH(C_S00_AXI_ADDR_WIDTH)
53 ) step_motor_v1_0_S00_AXI_inst (
54     .pwm1(pwm1),
55     .pwm2(pwm2),
56     .pwm3(pwm3),
57     .pwm4(pwm4),
58     .S_AXI_ACLK(s00_axi_aclk),

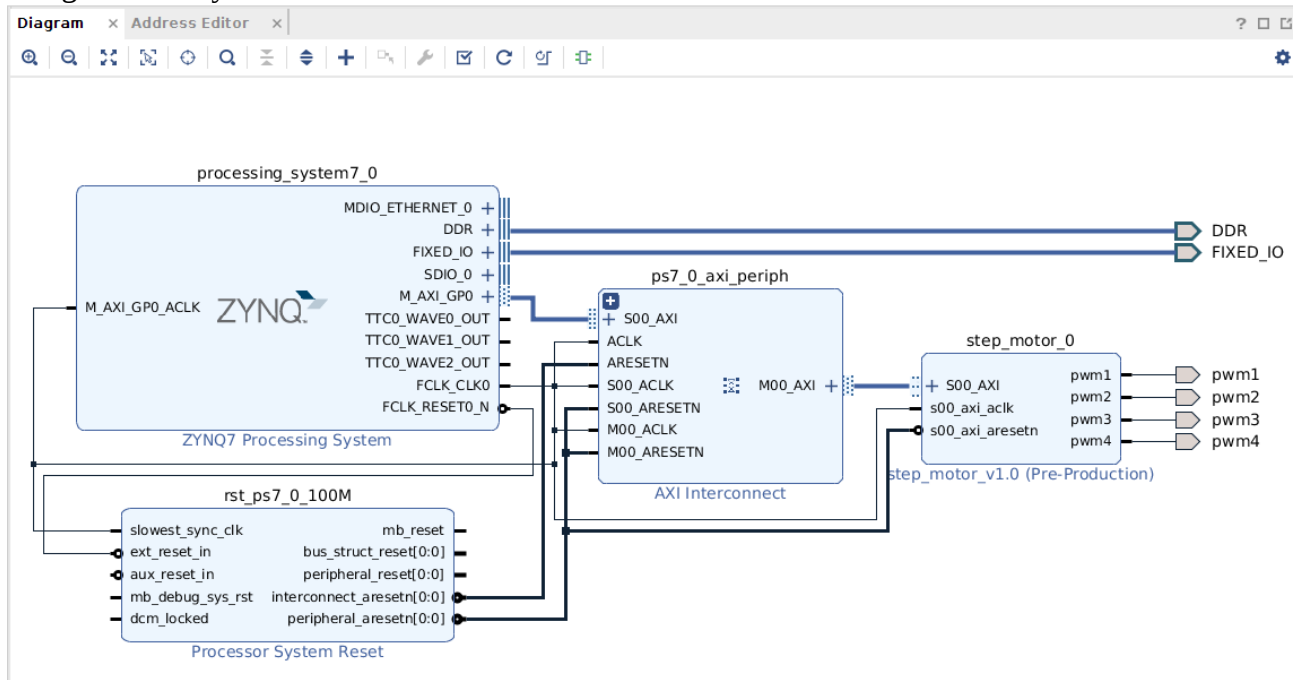
```

5.Package IP 에서 호환성 부분에 artix7 을 추가하고, ip 를 생성한다.

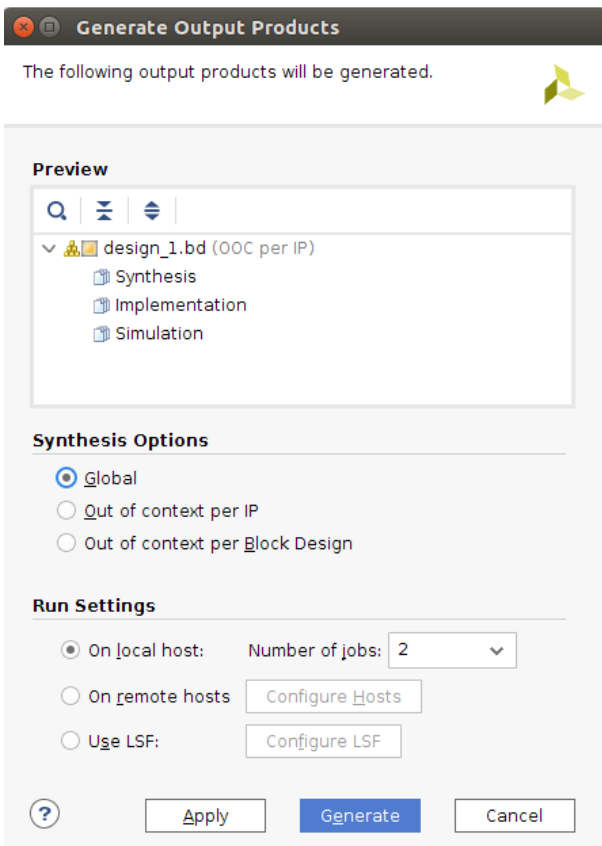


6.블록 다이어그램을 만들어 아래와 같이 구성한다.

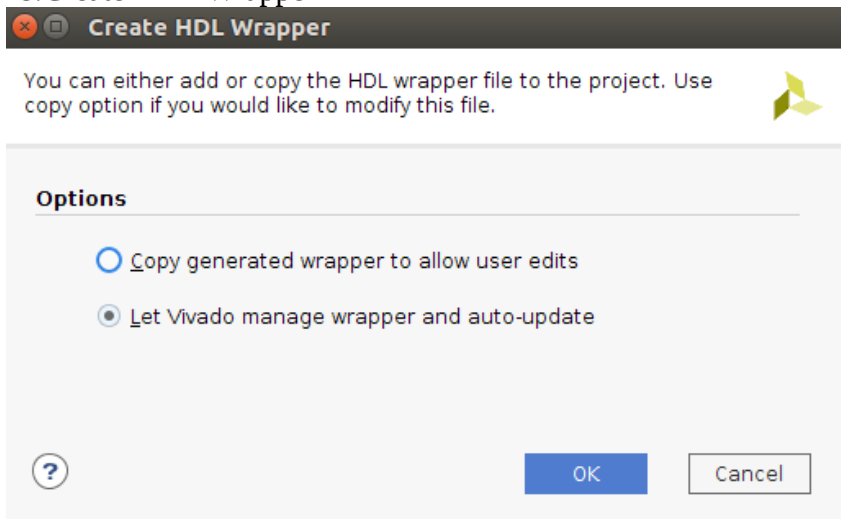
- zynq 보드 추가
- 더블클릭 후 USB0 제거
- step_motor IP 추가
- 각 포트에 create port
- Run connection Automation
- Validate design
- Regenerate Layout



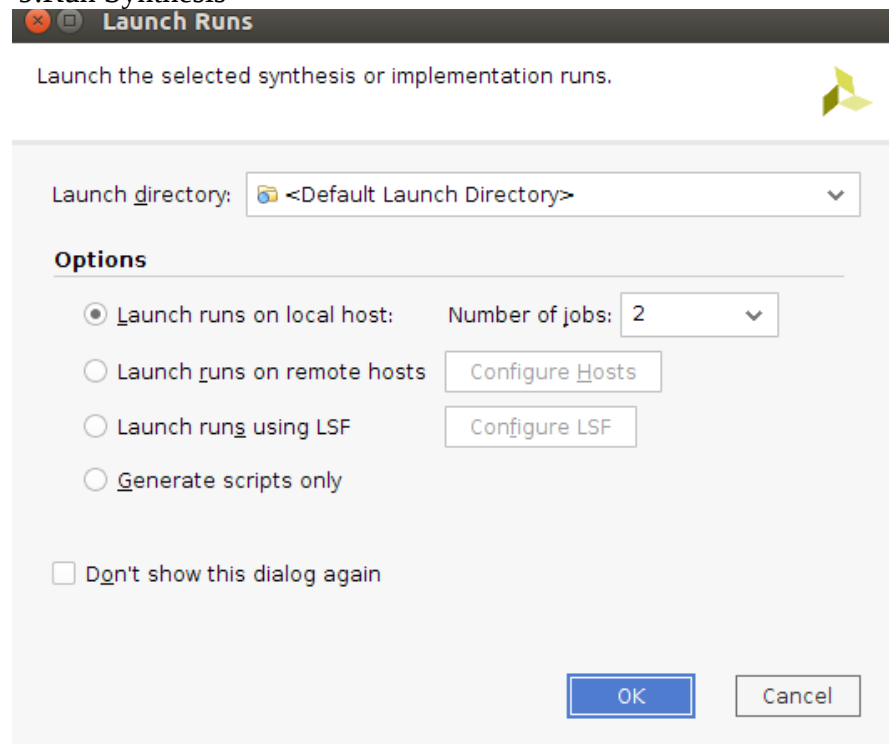
7.Generate Output Products



8. Create HDL Wrapper



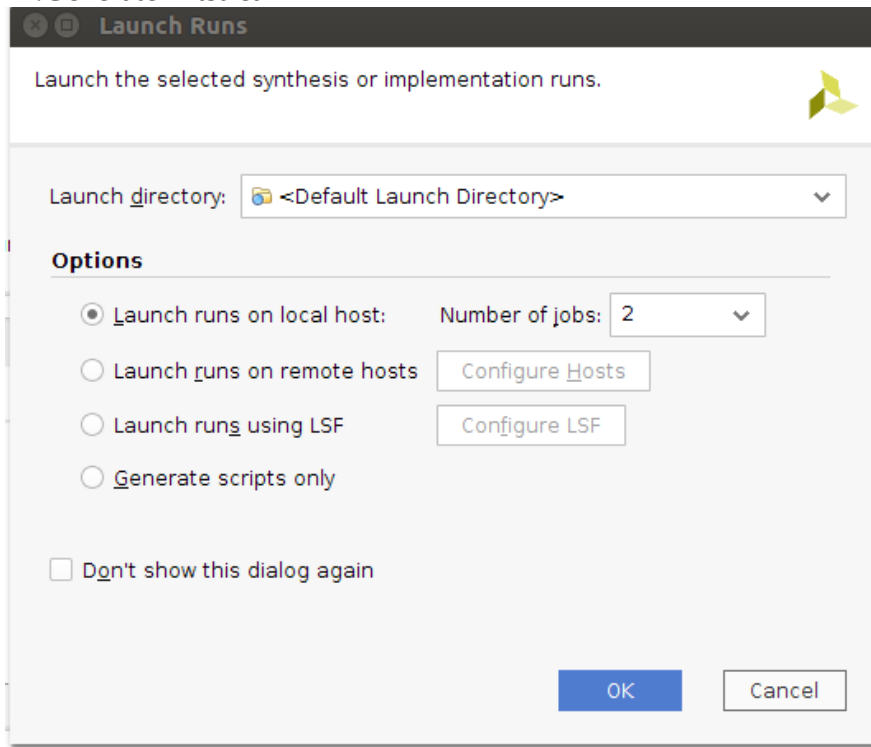
9. Run Synthesis



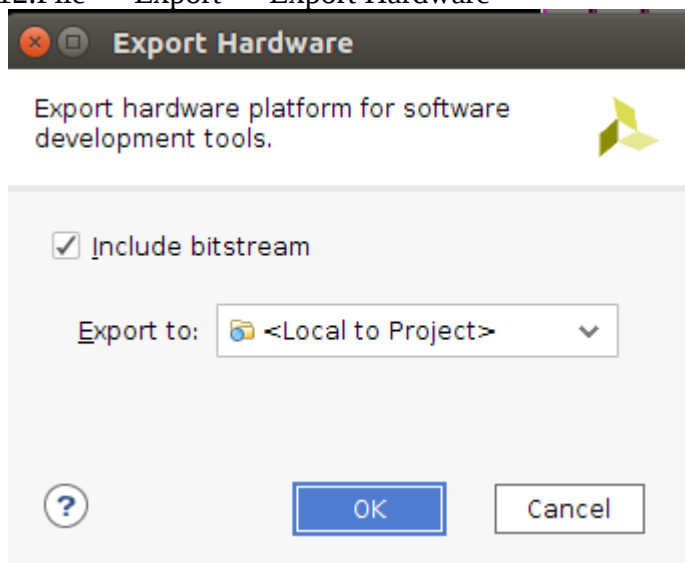
10. I/O Ports 설정

Tcl Console Messages Log Reports Design Runs IP Status Package Pins I/O Ports x									
Name	Direction	Board Part Pin	Board Part Interface	Neg Diff Pair	Package Pin	Fixed	Bank	I/O Std	
> DDR_54576 (71)	INOUT					✓	502	(Multiple)*	
> FIXED_IO_54576 (59)	INOUT					✓	(Multiple)	(Multiple)*	
▼ Scalar ports (4)									
✓ pwm1	OUT				Y14	✓	34	LVCMOS33*	
✓ pwm2	OUT				W14	✓	34	LVCMOS33*	
✓ pwm3	OUT				U12	✓	34	LVCMOS33*	
✓ pwm4	OUT				T12	✓	34	LVCMOS33*	

11. Generate Bitstream

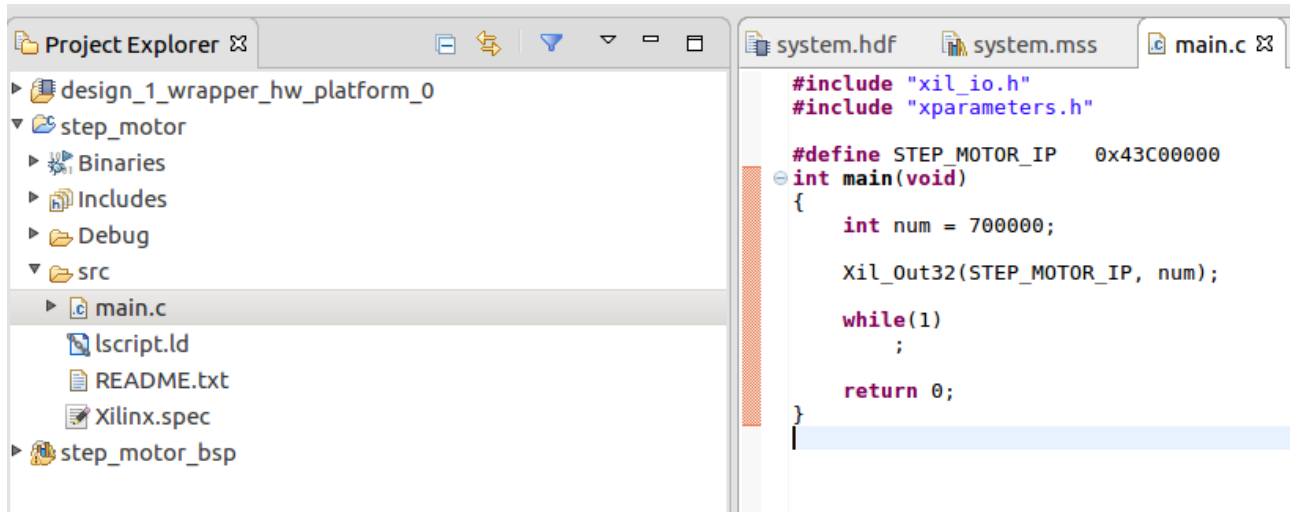


12. File → Export → Export Hardware



13. File → Launch SDK

14.main.c 에 아래와 같이 작성한다.

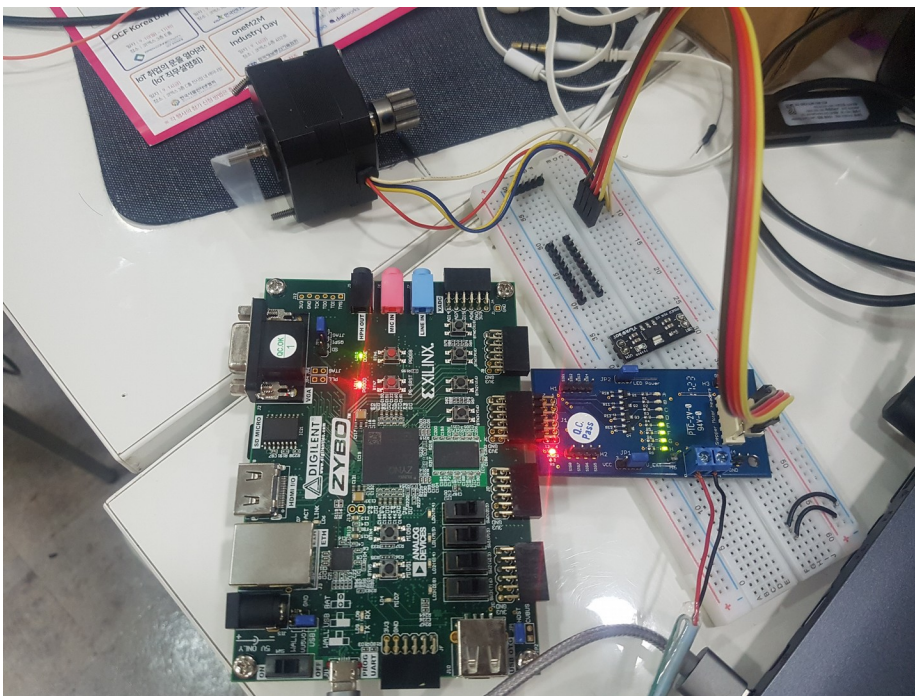


15.Program FPGA 후 Run as → Launch On Hardware(GDB)

16.결선은 아래와 같이 한다.

JC 에 연결하고, 스텝모터는 위쪽 순서대로 A 상,A-상,B 상,B-상이다.

외부 전원은 왼쪽이 +, 오른쪽이 -.



사용 모터 : 4017-875

스텝모터 드라이버 : Pmod Stepmotor driver

외부전원 : 5V1A(노트북 USB)