

zynq 프로젝트

이름 : 임세진

목차

1. 프로젝트 소개
2. 개발 보드
3. FPGA 소개
4. FPGA PS와 PL
5. vivadoHLS
6. vivado 로직설계
7. 로직테스트
8. Device driver
9. 어려웠던 점

1. 프로젝트 소개

- 차량에 부착된 센서, 카메라를 활용해 전방의 사람을 감지하고 수집된 정보로 모터의 속도와 방향을 조절한 충돌 방지 시스템
- 사용기술
 - DSP를 이용하여 Face Detection
 - ZYBO(FPGA)를 이용한 임베디드 리눅스
 - Cortex-R5F(MCU)를 이용한 BLDC모터 제어
 - 통신(CAN)

2.개발 보드



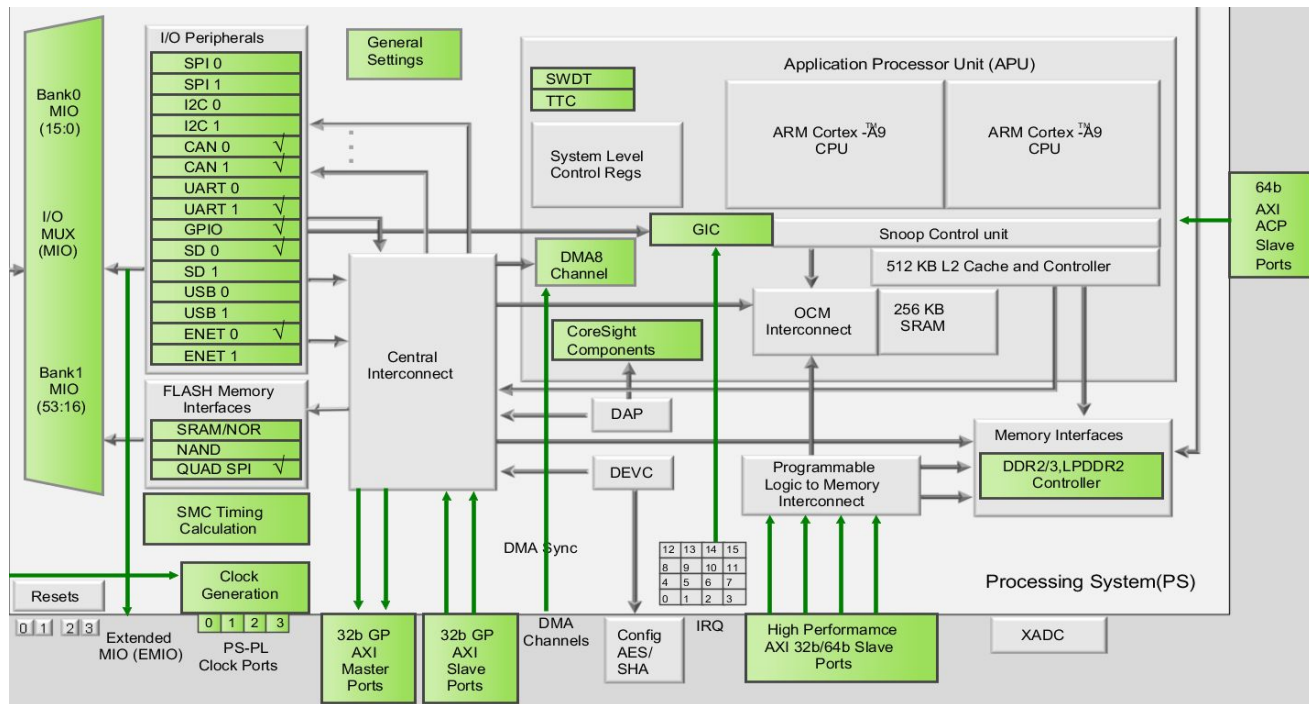
Figure 1. ZYBO Zynq-7000 development board.

- ZYNQ XC7Z010-1CLG400C
- 512MB x32 DDR3 w/ 1050Mbps bandwidth
- Dual-role (Source/Sink) HDMI port
- 16-bits per pixel VGA source port
- Trimode (1Gbit/100Mbit/10Mbit) Ethernet PHY
- MicroSD slot (supports Linux file system)
- OTG USB 2.0 PHY (supports host and device)
- External EEPROM (programmed with 48-bit globally unique EUI-48/64™ compatible identifier)
- Audio codec with headphone out, microphone and line in jacks
- 128Mb Serial Flash w/ QSPI interface
- On-board JTAG programming and UART to USB converter
- GPIO: 6 pushbuttons, 4 slide switches, 5 LEDs
- Six Pmod connectors (1 processor-dedicated, 1 dual analog/digital, 3 high-speed differential, 1 logic-dedicated)

3. FPGA소개

1. FPGA(field programmable gate array)는 설계 가능 논리 소자와 프로그래밍 가능 내부선이 포함된 반도체 소자
2. zynq는 ARM-v7 와
3. PS와 PL로 구분
 - a. PS(Processing System)은 Cortex-A9의 SOC를 사용
 - b. PL(Programmable Logic)은 프로그램이 가능한 논리 제어기
4. vivado downloads
 - a. <http://www.xilinx.com/> -> support -> downloads & licensing -> vivado
5. peta linux
 - a. 4-a에서 Embedded Development에서 2015.4 메뉴

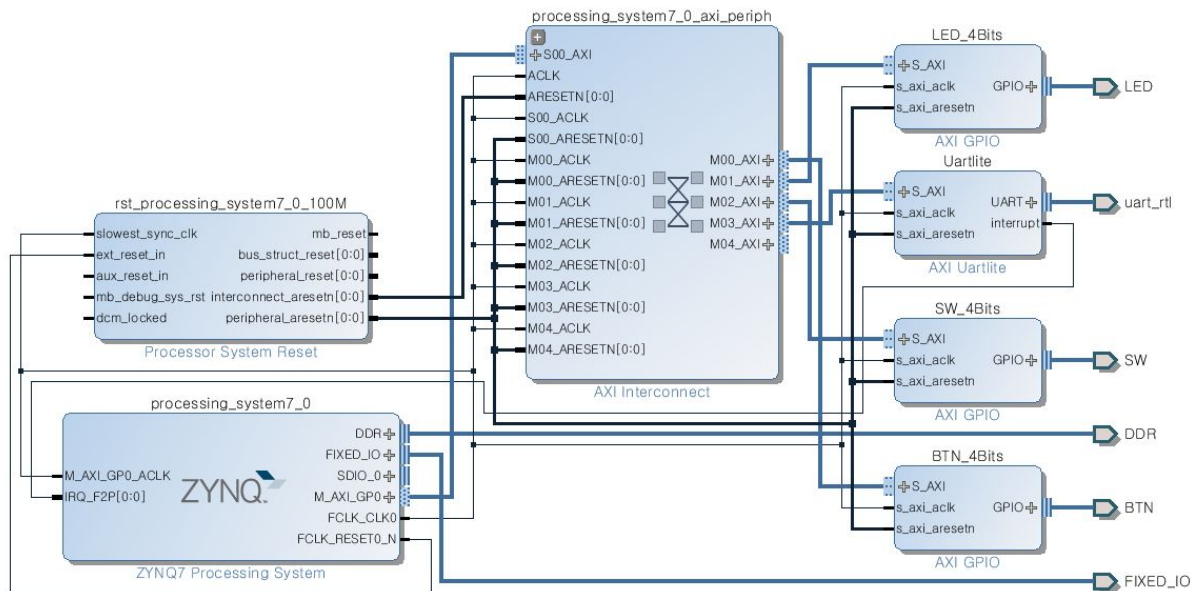
4. FPGA PS와 PL



FPGA는 PS과 PL
로 구성되었으며

PS는 Processing
System의 약자로
cortex-A9의 SOC칩
이다

4. FPGA PS와 PL



PL은
Programmable
Logic 의 약자로 논
리 소자를 이용 프
로그램이 가능한 영
역을 말한다.

5. vivadoHLS - 소개

vivadoHLS는 C/C++을 이용 로직을 설계할 경우 VHDL/Verilog로 변환
zynq보드의 PL영역에 프로그래밍이 가능하도록 한다.

5. vivadoHLS

```
void polar(int x, int y, double *r, double *degree)
{
#pragma HLS INTERFACE s_axilite port=return bundle=CRTL_BUS
#pragma HLS INTERFACE s_axilite port=x bundle=CRTL_BUS
#pragma HLS INTERFACE s_axilite port=y bundle=CRTL_BUS
#pragma HLS INTERFACE s_axilite port=r bundle=CRTL_BUS
#pragma HLS INTERFACE s_axilite port=degree bundle=CRTL_BUS

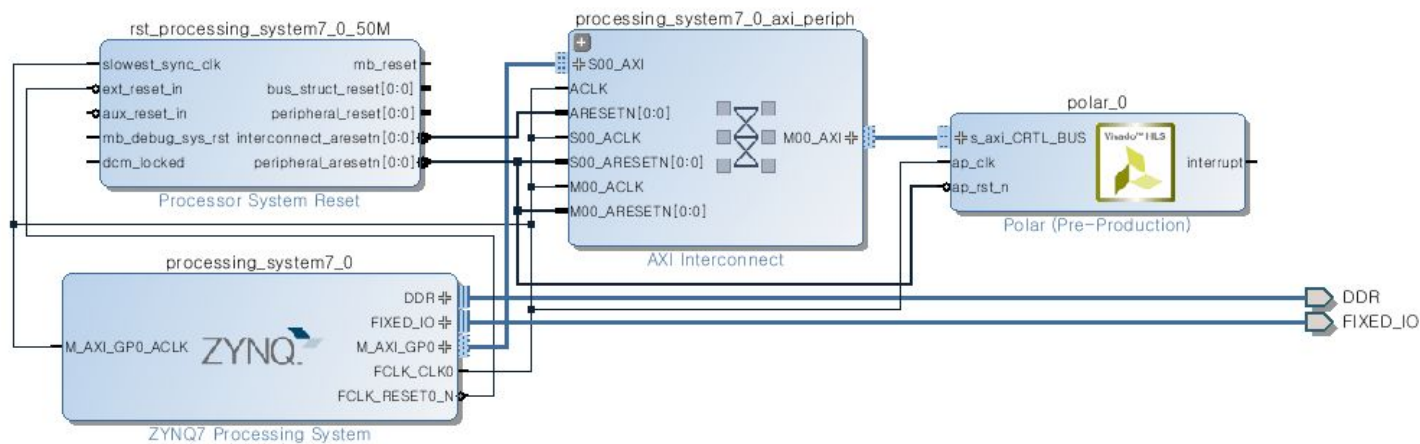
    int x1, y1;

    x1 = x-320;
    y1 = y-240;

    *r = sqrt(pow(x1,2)+pow(y1,2));
    *degree = atan2(y1,x1)*180/M_PI;
}
```

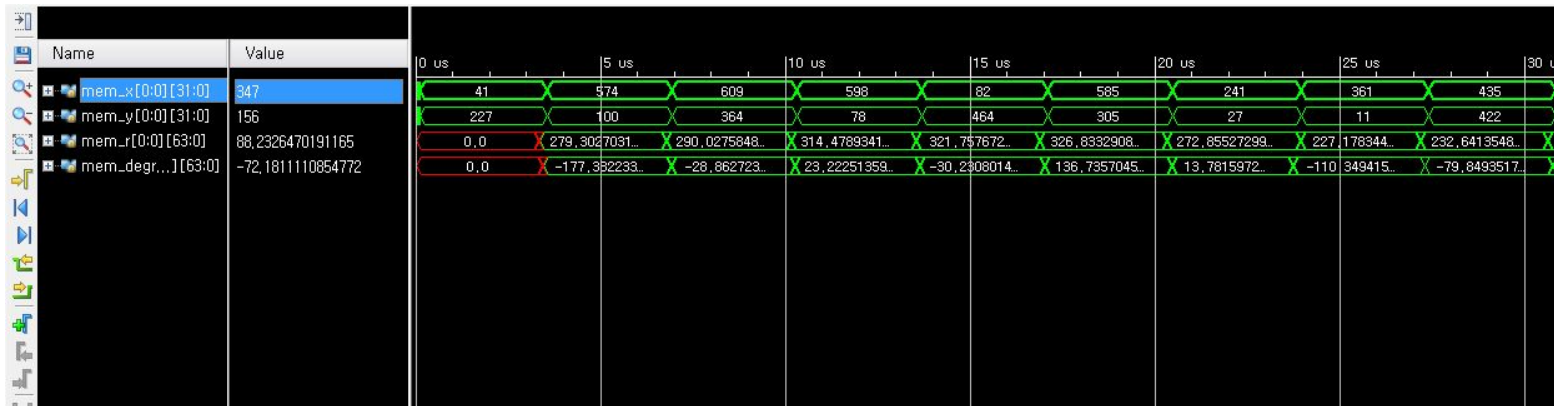
DSP에서 탐지된 좌표는 직교좌표이며 이것은 이동차량이 zynq보드에서 편하게 탐지할 수있게 직교좌표를 극좌표로 변경하는 코드

6. vivado 로직설계



FPGA보드의
전체적 하드웨
어 구성된

7. 로직테스트



HLS에서 작성한코드를 테스트하여 타이밍차트에 표시된 그래프

8. Device driver

```
polar_0: polar@43c00000 {  
    compatible = "xlnx,polar-1.0";  
    interrupt-parent = <&intc>;  
    interrupts = <0 31 4>;  
    reg = <0x43c00000 0x10000>;  
    xlnx,s-axi-crtl-bus-addr-width = <0x6>;  
    xlnx,s-axi-crtl-bus-data-width = <0x20>;  
};
```

polar를 구성하는 Device Tree 입니다.

8. Device driver

```
root@zybo:~# insmod /lib/modules/4.0.0-xilinx/extra/polar2.ko
polar2 module world.
polar2 43c00000.polar: Device Tree Probing
regX : 0xe09a0010
regY : 0xe09a0018
regR : 0xe09a0020
regDEG : 0xe09a002c
regCtl : 0xe09a0000
polar2 43c00000.polar: polar2 at 0x43c00000 mapped to 0xe09a0000, irq=167
root@zybo:~# po_test
write
read
R[279.302703]
DEG[-177.332234]
root@zybo:~#
```

Device Tree를 이용하여 임베디드 리눅스에서 Device driver module를 올렸습니다.

9. 어려웠던 점

1. 커널 영역에서 원칙적으로 소수점연산을 할 수 없다는 것
 - a. `double`을 `char`로 형변환해 바이트단위로 결과데이터를 확인했습니다.
2. `uart`통신에서의 데이터 **Read** 문제
 - a. `uart Devices driver`코드를 분석해보며 이해했습니다.