Xilinx Zynq FPGA, TI DSP, MCU 기반의 프로그래밍 및 회로 설계 전문가 과정

8 월 23 일 진행상황 발표

FPGA – Device Driver 요약

1. Vivado 에서의 작업은 오직 하드웨어 H/W 설계이다 .

2. Vivado 에서 설계한 H/W 또는 이미 설계되어있는 H/W 의 인터페이스에 맞게 코딩 한 것이 디바이스 드라버이 다 .

3. 리눅스에서는 많이 사용하는 인터페이스에 대한 디바이스 드라이버를 제공한다 .

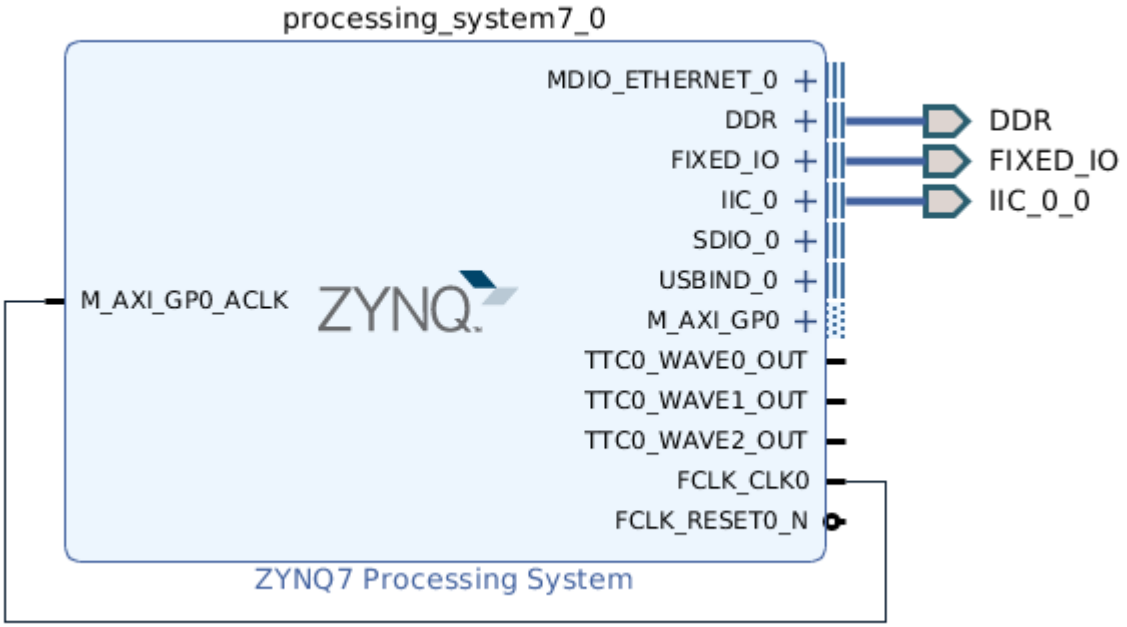
4. petalinux 는 사용할 드라이버를 설정하면 Makefile 을 생성해주어 드라이버를 사용가능하게 해준다 .

5. open 을 통해 장치파일을 열면 내부에 등록되어있는 디바이스 드라이버로 연동되고 , User 영역에서 ioctl read wirte 를 하게 되면 사용하는 인터페이스에 맞게 함수가 바뀐다 .

MPU9250 i2C

- 리눅스에서는 i2c 인터페이스에 맞는 디바이스 드라이버를 제공한다 .
- linux/i2c.h 와 linux/i2c-dev.h 를 추가하여 open(“/dev/i2c-0”,O_RDWR); 을 하게 되면 i2c 인터페이스에 맞게 read,write,ioctl 등 함수가 디바이스드라이버에 wrapping 된다 .
- Ioctl 함수로 디바이스의 입출력을 설정한다 . 여기서 MPU9250 은 Slave 로 사용하기 때문에 아래와 같이 Ioctl 을 선언한다 .
- ioctl(fd,I2C_SLAVE_FORCE,MPU9250_ADDRESS);

Block design



Source code

```
void writeByte(int fd, uint8_t regAddr, uint8_t data)
{
	int8_t buf[2] = {regAddr,data};

	if(write(fd, buf ,sizeof(buf)) != sizeof(buf))
	{
		printf("write register error - writeByte\n");
	}
}
```

```
uint8_t readByte(int fd, uint8_t regAddr)
{
uint8_t buf[1] = {regAddr};
uint8_t data[1] = {0};
if(write(fd,buf,1) != 1)
{
perror("read register error -w \n");
return -1;
}
if(read(fd, data, 1) != 1)
{
perror("read register error -r \n");
return -1;
}

return data[0];

}
```

```
  void readBytes(int fd, uint8_t regAddr, int length, uint8_t *data)
  {
      uint8_t buf[1] = {regAddr};
      if(write(fd, buf, 1) != 1)
      {
          perror("read register error - readBytes\n");
      }
      if(read(fd, data, length) != length)
      {
          printf("recieve data error - readBytes\n");
      }
  }
void ioctl_mpu9250(int fd)
{
            if(ioctl(fd,I2C_SLAVE_FORCE, MPU9250_ADDRESS) < 0)
            {
                  perror("slave address connect error - ioctl_mpu9250\n");
            }
}

void ioctl_ak8963(int fd)
{
            if(ioctl(fd, I2C_SLAVE_FORCE, AK8963_ADDRESS) < 0)
            {
                  perror("slave address connect error - ioctl_ak8963\n");
            }
}
```

```c
int main(void)
{

                if( (fd = open(I2C_FILE_NAME, O_RDWR)) <0)
                {
                                perror("Open Device Error! \n");
                                return -1;
                }

                ioctl_mpu9250(fd);

                usleep(1000000);

                uint8_t c = readByte(fd, WHO_AM_I_MPU9250 );
                printf("I AM = %x \n\r", c);
```

문제점

- 디버깅 부족 .

  - 회로 구성이 정확하지 않아 동작하지 않은걸 MCU 에서 확인함

  - 커스텀 함수의 오타가 있었음 , 데이터 버퍼를 선언하고 사용하지 않아 값을 받는데 오류 발생 .

- MAG 값

  MAG 값을 알아볼수 있게 계산