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

PWM + ALL eQEP module
In FPGA zybo
based petalinux

강사 : Innova Lee(이 상훈)
학생 : 장 성환

지금까지 설계한 HW bitstream을 가지고 petalinux kernel의 apps에 해당 IP 를 제어할 수 있는 PS 를 추가하여 /dev/device_driver 앱을 생성하여 결과를 관측하였다.

소스 코드는 다음과 같다.

```
    printf("eCAP UIO Test\r\n");
    printf("Loopback Test!\r\n");
    uiod1 = argv[1];
    uiod2 = argv[2];
    uiod3 = argv[3];
    fd1 = open(uiod1, O_RDWR);

    if(fd1 < 1)
    {
        perror(argv[0]);
        printf("Invalid UIO Device File: %s\n", uiod1);
        return -1;
    }

    printf("%s open success!\r\n", uiod1);
    fd2 = open(uiod2, O_RDWR);

    if(fd2 < 1)
    {
        perror(argv[0]);
        printf("Invalid UIO Device File: %s\n", uiod2);
        return -1;
    }

    printf("%s open success!\r\n", uiod2);

    fd3 = open(uiod3, O_RDWR);


    if(fd3 < 1)
    {
        perror(argv[0]);
        printf("Invalid UIO Device File: %s\n", uiod3);
        return -1;
    }

    printf("%s open success!\r\n", uiod3);

    ptr1 = mmap(NULL, PWM_MAP_SIZE, PROT_READ|PROT_WRITE, MAP_SHARED, fd1, 0);
    ptr2 = mmap(NULL, PWM_MAP_SIZE, PROT_READ|PROT_WRITE, MAP_SHARED, fd2, 0);
    ptr3 = mmap(NULL, eQEP_MAP_SIZE, PROT_READ|PROT_WRITE, MAP_SHARED, fd3, 0);

    printf("ptr1 ptr2 memory allocation success!\r\n");
    printf("ptr1 : 0x%x\r\n",ptr1); // PWM0
    printf("ptr2 : 0x%x\r\n",ptr2); // PWM1
    printf("ptr3 : 0x%x\r\n",ptr3); // eQEP

    /*Infinit Loop*/
    while(1)
    {
        *((unsigned *)(ptr1 + PWM0_DATA_OFFSET)) = duty;
        *((unsigned *)(ptr2 + PWM1_DATA_OFFSET)) = duty;
        *((unsigned *)(ptr3 + eQEP_QUPRD_OFFSET)) = QUPRD;

        printf("RPM is %d\r\n", *((unsigned int*)ptr3 + eQEP_QPOSLAT_OFFSET));
    }

    return 0;
```

SDK와는 다르게 mmap을 사용하여 해당 IP의 레지스터 주소에 접근하여 값을 수정하고 값을 읽어 올 수 있다.
내가 설계한 IP이기 때문에 mmap으로 간단히 제어가 가능하다.

ioctl을 사용할 필요는 없다.
sci통신을 활용하여 putty를 확인한 결과 SDK로 확인한 결과가 같다. 결과는 다음과 같다.