

TI DSP, MCU, Xilinx Zynq FPGA 프로그래밍 전문가 과정

9주차 라젠카 발표

2018.09.05

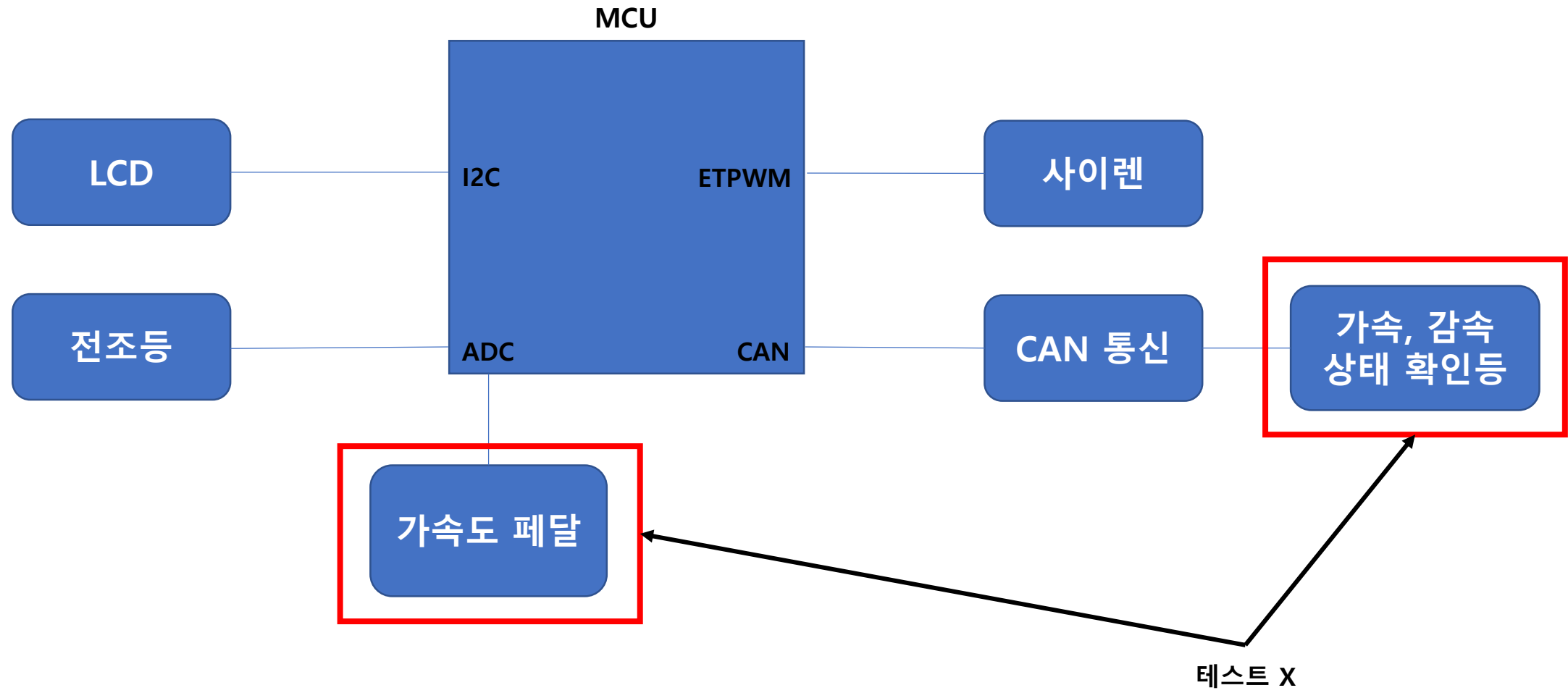
강사 – Innova Lee(이상훈)
gcccompil3r@gmail.com

01

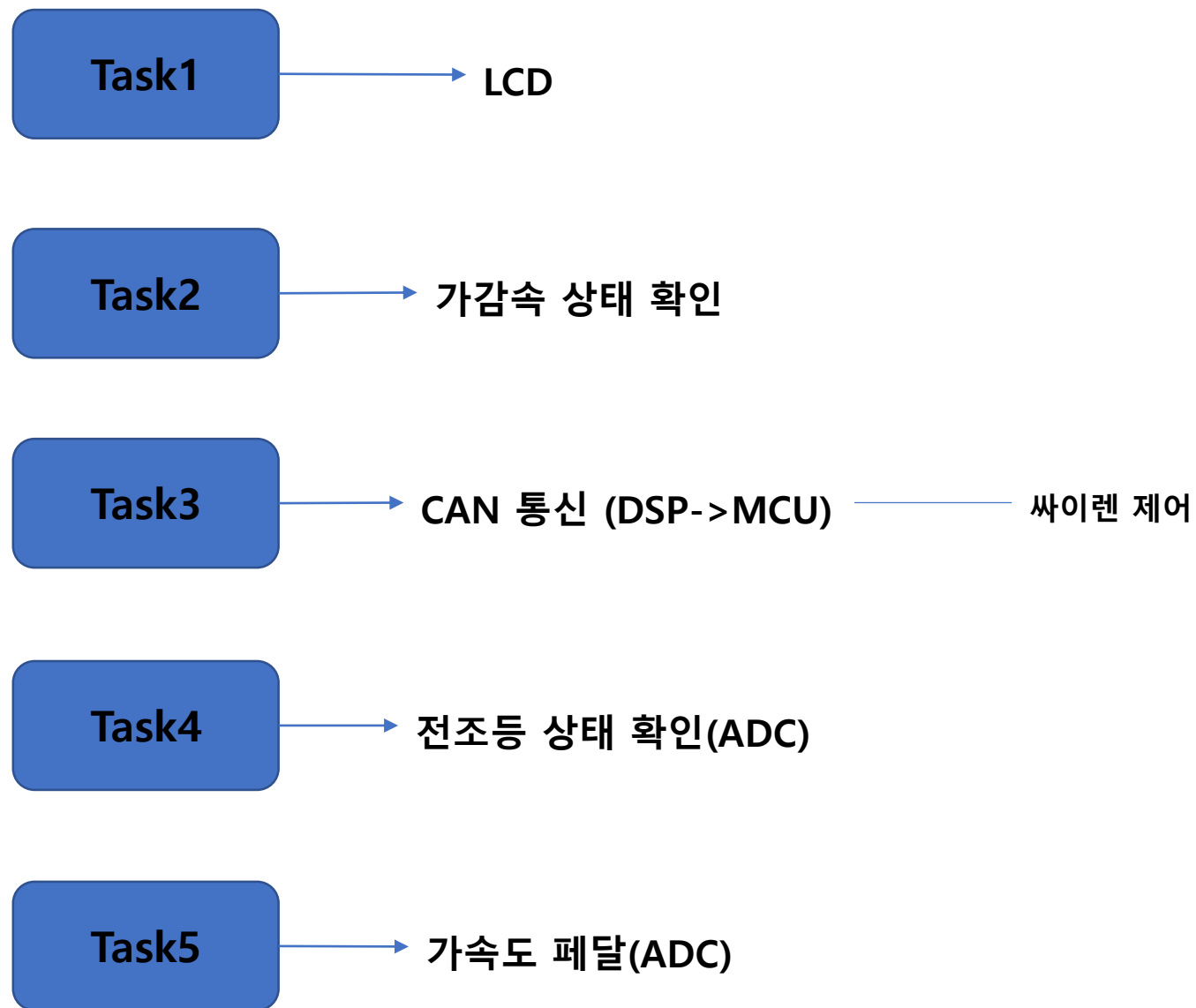
진행상황 및 문제점

안상재

* 통합 테스트 블록 다이어그램



* RTOS 통합 소스 코드 현황



* 현재 사용 중인 MCU Peripheral 및 핀 정리

* ADC

- ADC1 Group1 Pin0 : 조도센서 (전조등)
- ADC2 Group1 Pin0 : 압력센서 (가속도 페달)

* ETPWM

- ETPWM1A : 사이렌
- ETPWM2B : 차 뒷바퀴(BLDC) 제어

* GIO

- gioPORTB0 : ADC 트리거용 (ADC1, ADC2) - (전조등, 가속도 페달)
- gioPORTB1 : 조도센서 VCC 용
- gioPORTB2 : 브레이크 LED 제어용

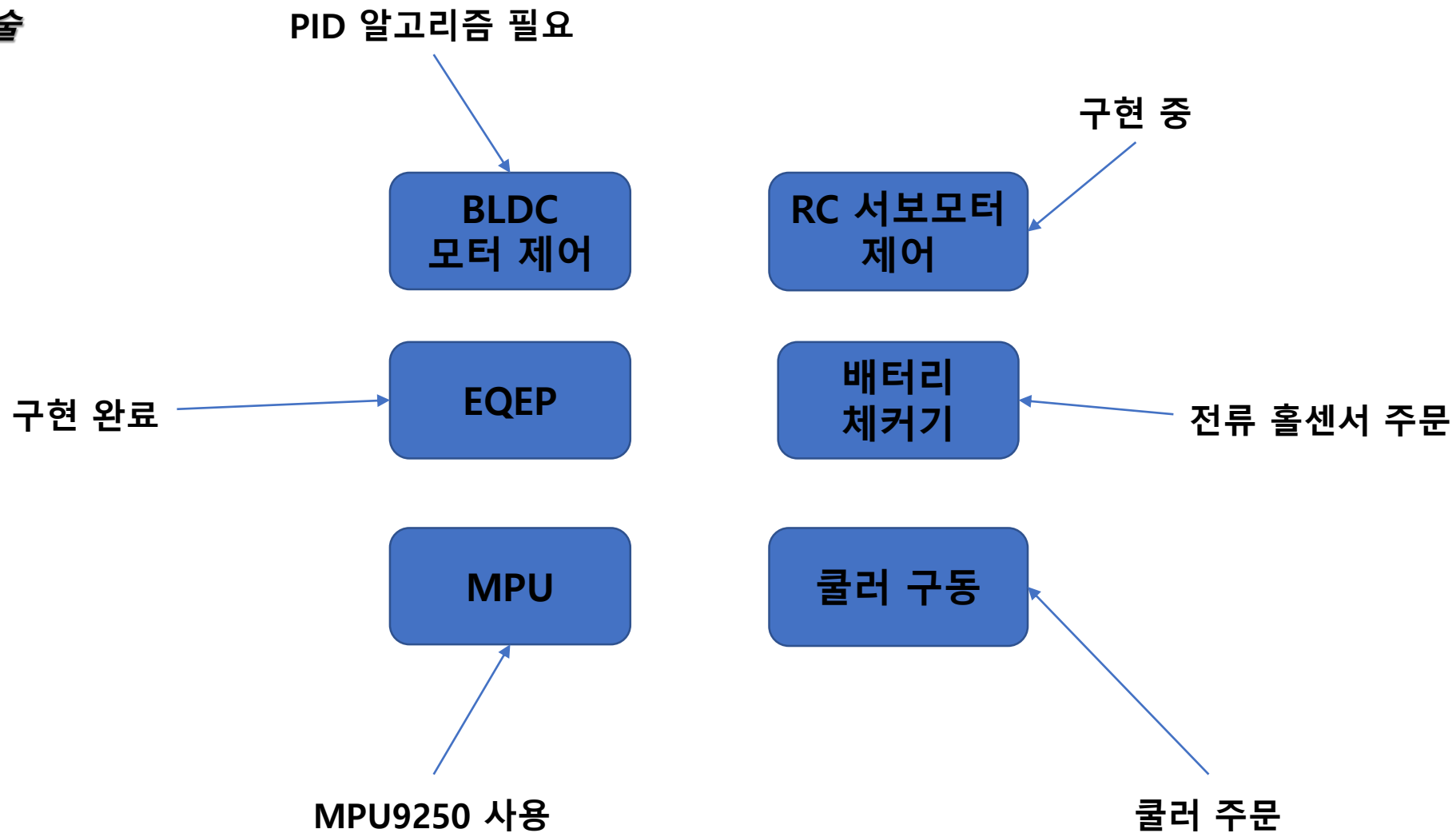
* CAN

- CAN1 : DSP->MCU 프로토콜

* I2C

- I2C2 : LCD 제어용

* 남은 기술



감사합니다

02 진행상황 및 문제점

황수정

안드로이드 앱 코드 이해

MCU Peripheral(Cortex-R5F)

- FreeRTOS Based Car light
- Buzzer(with CAN)
- Break Light(with CAN)
- Gas Pedal

03 진행사항

```
#include <HL_gio.h>
#include <HL_reg_gio.h>
#include <HL_can.h>
#include <HL_reg_can.h>
#include <HL_sci.h>
#include <HL_reg_sci.h>
#include <HL_adc.h>
#include <HL_reg_adc.h>
#include <FreeRTOS.h>
#include <FreeRTOSConfig.h>
#include <HL_hal_stdtypes.h>
#include <os_mpu_wrappers.h>
#include <os_projdefs.h>
#include <os_semphr.h>
#include <os_task.h>
#include <string.h>
#include <stdio.h>
```

```
QueueHandle_t mutex;
```

```
xTaskHandle xTask1Handle;
```

```
xTaskHandle xTask2Handle;
```

```
adcData_t counter;
```

```
uint8 msg[32] = { 0, };
```

```
uint32 value;
```

```
char num;
```

```
void vTask1(void* pvParameters);
```

```
void vTask2(void* pvParameters);
```

```
void led(int bri);
```

```
if (xTaskCreate(vTask2, "Task2", configMINIMAL_STACK_SIZE * 8, NULL, 1,
               &xTask1Handle) != pdTRUE)
```

```
{
    while (1)
    ;
}
```

```
vTaskStartScheduler();
```

```
while (1)
```

```
;
return 0;
```

```
}
```

```
void led(int bri)
```

```
{
```

```
if (bri <= 5)
```

```
{
```

```
gioSetBit(gioPORTB, 7, 1);
```

```
}
```

```
else
```

```
{
```

```
gioSetBit(gioPORTB, 7, 0);
```

```
}
```

```
}
```

```
void vTask1(void *pbParameters)
```

```
{
```

```
while (1)
```

```
{
```

```
gioSetBit(gioPORTB, 0, 1);
```

```
gioSetBit(gioPORTB, 4, 1);
```

```
while (adcIsConversionComplete(adcREG1, adcGROUP1) == 0)
```

```
;
```

```
adcGetData(adcREG1, adcGROUP1, &counter);
```

```
led(counter.value);
```

```
vTaskDelay(80);
```

```
gioSetBit(gioPORTB, 0, 0);
```

```
}
```

```
}
```

FreeRTOS Based Car light Code

```
void vTask2(void *pbParameters)
{
    while (1)
    {
        canTransmit(canREG1, canMESSAGE_BOX1, &num);
        vTaskDelay(500);

        canIsRxMessageArrived(canREG1, canMESSAGE_BOX2);
        vTaskDelay(500);
        canGetData(canREG1, canMESSAGE_BOX2, &num);

        switch (num)
        {
            //git의 선생님 can 조향 신호 기준. gio 신호는 조 회의로 커스텀
            {
                case 1: //좌회전
                    gioSetBit(gioPORTB, 1, 1);
                    vTaskDelay(500);
                    break;

                case 2: //우회전
                    gioSetBit(gioPORTB, 2, 1);
                    vTaskDelay(500);
                    break;

                case 4: //후진
                    gioSetBit(gioPORTB, 3, 1);
                    vTaskDelay(500);
                    break;

                case 5: //정지
                    gioSetBit(gioPORTB, 3, 1);
                    vTaskDelay(500);
                    break;

                case 7: //좌측깜박이
                    gioSetBit(gioPORTB, 1, 1);
                    vTaskDelay(500);
                    break;

                case 8: //우측깜박이
                    gioSetBit(gioPORTB, 2, 1);
                    vTaskDelay(500);
                    break;

                default:
                    gioSetPort(gioPORTB, 0);
                    vTaskDelay(500);
                    break;
            }
        }
    }
}
```

03 진행사항

```
#include <HL_can.h>
#include <HL_etpwm.h>
#include <HL_reg_can.h>
#include <HL_reg_etpwm.h>
#include <stdio.h>
```

```
char tr;
char re;
```

```
void delay(int time)
{
    int i;
    for (i = 0; i < time; i++)
        ;
}
```

```
int main()
{
    canInit();
    etpwmInit();
```

```
    while (1)
```

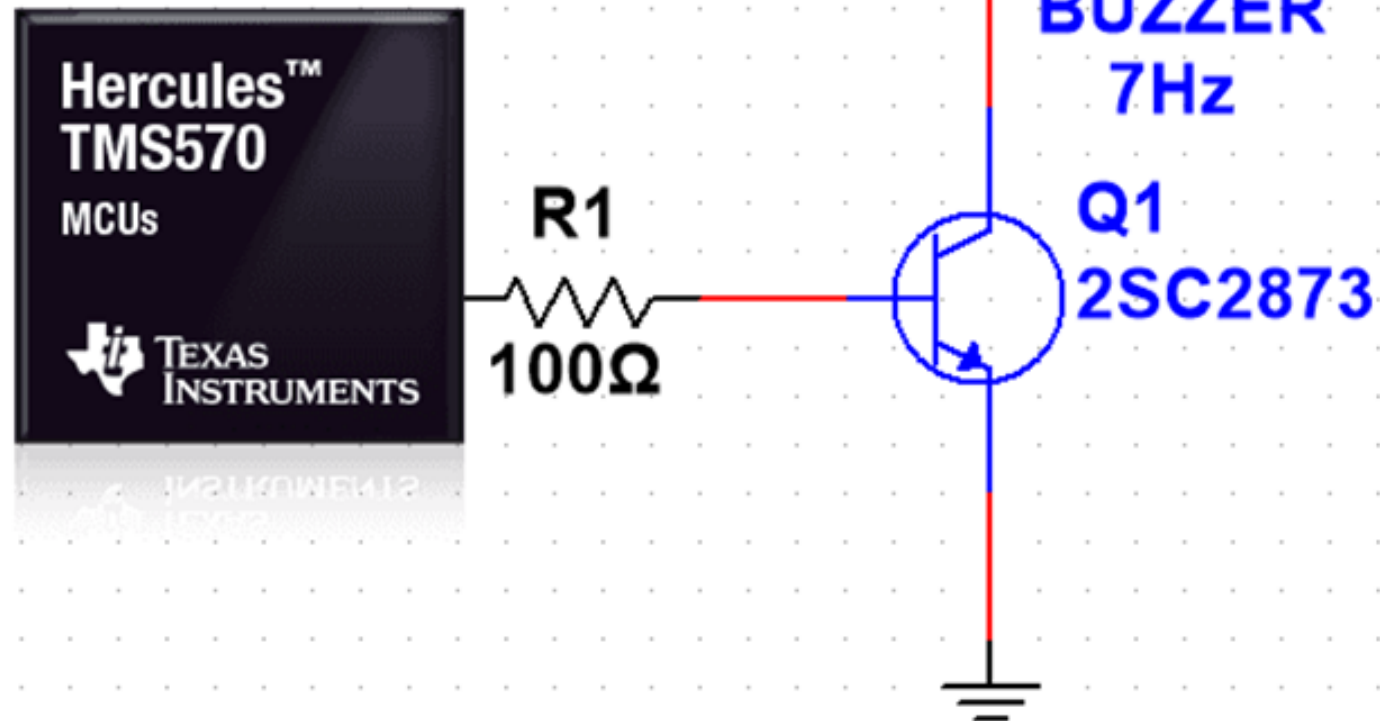
```
    {
        delay(1000000);
        canTransmit(canREG1, canMESSAGE_BOX1, &tr);
```

```
        case 5:
            etpwmStartTBCLK();
            etpwmREG1->CMPA = 2300;
            break;

        default:
            etpwmStopTBCLK();
            break;
    }
}
```

Buzzer_CAN CODE

Buzzer_CAN
Designer : 황수정, 김민호
2018.08.28
Project AI CAR





USB2CAN UI

Version 1.10 (2016. 9. 20)

UI Config.

USB(VCP) Devices

NTREX USB2CAN(FIFO) [NT2ASDZL]

Search

Connect

Disconnect

CAN Configuration

Bitrate 500K bps

Filter Identification 1 (Hexa)

Filter Mask 7F0 (Hexa)

Set

CAN Messages

Time	ID(Hex)	Flags	Data(Hex)
192794.327211	1	Ext	00 00 00 00 00 00 00 00 (8)
192793.925464	1	Ext	00 00 00 00 00 00 00 00 (8)
192793.521720	1	Ext	00 00 00 00 00 00 00 00 (8)
192793.117978	1	Ext	00 00 00 00 00 00 00 00 (8)
192792.714249	1	Ext	00 00 00 00 00 00 00 00 (8)
192792.311418	1	Ext	00 00 00 00 00 00 00 00 (8)
192791.907691	1	Ext	00 00 00 00 00 00 00 00 (8)
192791.503999	1	Ext	00 00 00 00 00 00 00 00 (8)
192791.101188	1	Ext	00 00 00 00 00 00 00 00 (8)
192790.697500	1	Ext	00 00 00 00 00 00 00 00 (8)
192789.566079	1	Self Ext	05 (1)
192789.276441	1	Ext	00 00 00 00 00 00 00 00 (8)
192788.873624	1	Ext	00 00 00 00 00 00 00 00 (8)
192788.469861	1	Ext	00 00 00 00 00 00 00 00 (8)
192788.067197	1	Ext	00 00 00 00 00 00 00 00 (8)
192787.664419	1	Ext	00 00 00 00 00 00 00 00 (8)
192787.260686	1	Ext	00 00 00 00 00 00 00 00 (8)

☒ Receive CAN Message

Clear

Transmit Message

ID(Hexa) 1 ☒ Ext ☐ RTR Data(Hexa) 5

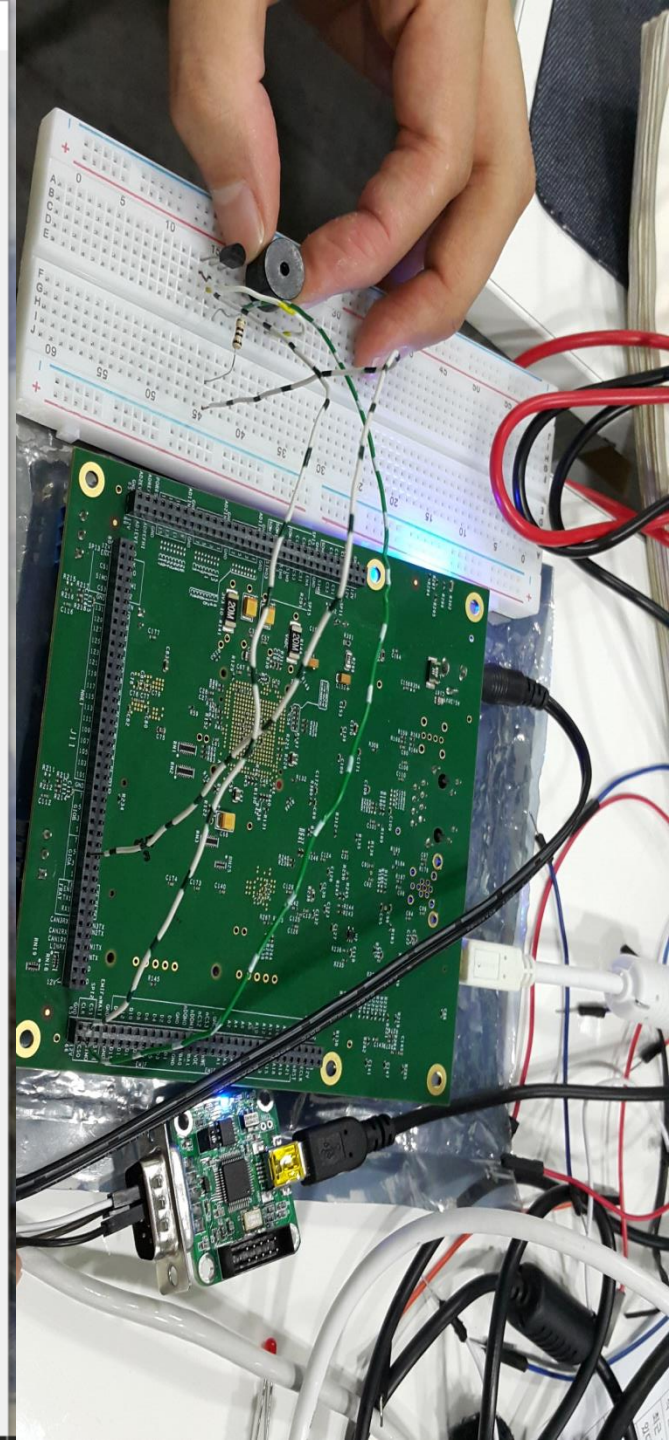
Send

ID(Hexa) 1 ☒ Ext ☐ RTR Data(Hexa) 1

Send

ID(Hexa) 0 ☐ Ext ☐ RTR Data(Hexa)

Send



```
#include <HL_can.h>
#include <HL_gio.h>
#include <HL_reg_can.h>
#include <HL_reg_gio.h>
#include <stdio.h>
#include <math.h>

int a, d, e, f, g, ex_data, cur_data;
char data_flag = 0;
char tr[5] = { 0, };
char re[5] = { 0, };

void delay(int time)
{
    int i;
    for (i = 0; i < time; i++)
        ;
}

int main()
{
    gioInit();
    canInit();
    while (1)
    {
        delay(1000000);
        for (a = 0; a < 5; a++){
            canTransmit(canREG1, canMESSAGE_BOX1, &tr[a]);
        }
        delay(1000000);
        if (canIsRxMessageArrived(canREG1, canMESSAGE_BOX2))
        {
            for (a = 0; a < 5; a++){
                canGetData(canREG1, canMESSAGE_BOX2, &re[a]);
            }
            if (data_flag == 0)
            {
                for (a = 1; a < 5; a++){
                    d = d + (re[a]) * pow(10, 4 - a);
                }
                ex_data = d;
                d = 0;
                printf("ex_data = %d\n", ex_data);
                data_flag = 1;
            }
        }
    }
}
```

```
else
{
    if (!ex_data == 0)
    {
        for (e = 1; e < 5; e++)
        {
            g = g + (re[e]) * pow(10, 4 - e);
            //printf("rx = %d\n", g);
        }
    }
    cur_data = g;
    g = 0;
    printf("cur_data = %d\n", cur_data);
    f = ex_data - cur_data;
    printf("f = %d\n", f);
    if (f < 0)
    {
        printf("bigger than ex_data\n");
        gioSetBit(gioPORTB, 0, 1);
    }
    else if(f > 0)
        gioSetBit(gioPORTB, 0, 0);
    f = 0;
    ex_data = cur_data;
    cur_data = 0;
}
```



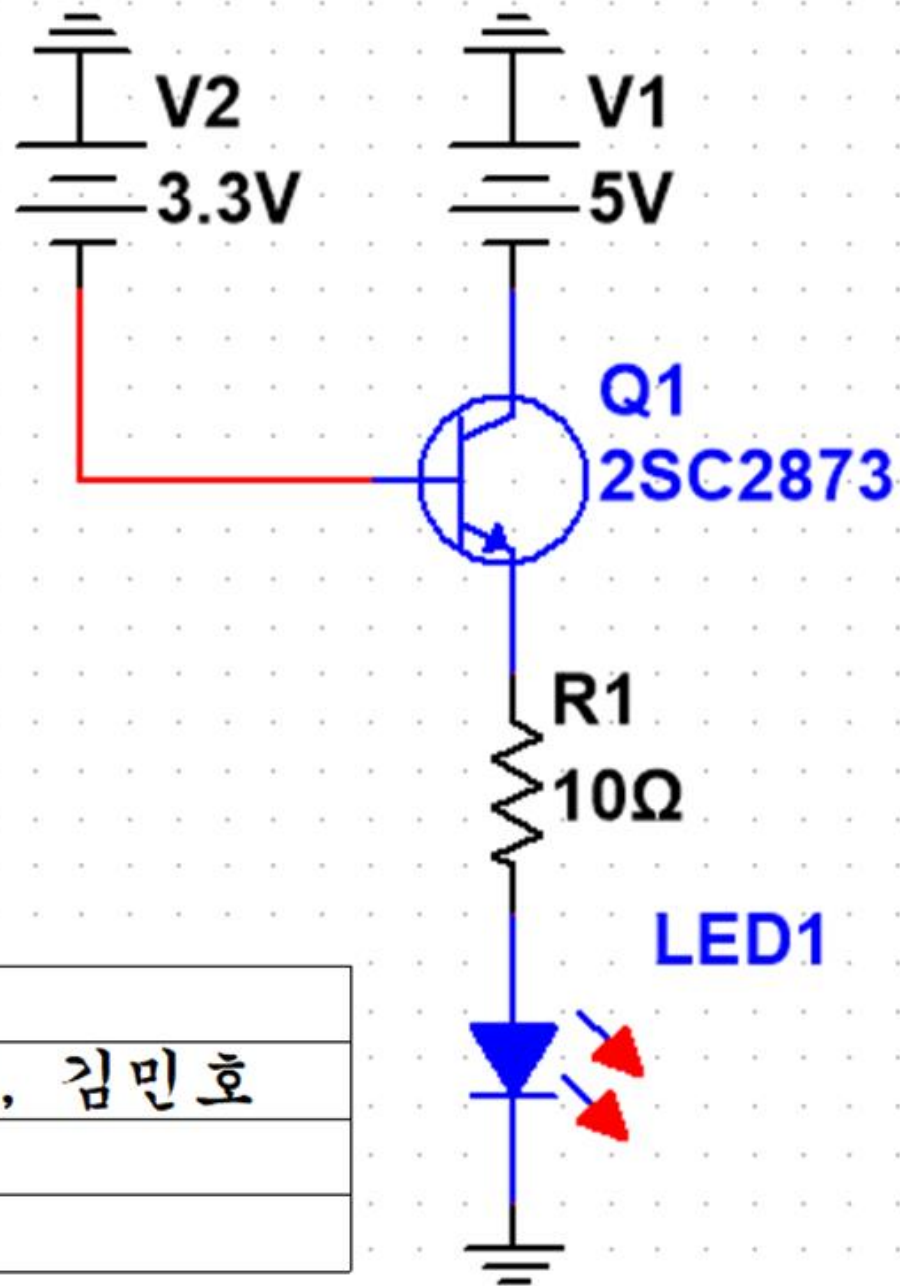
CAN

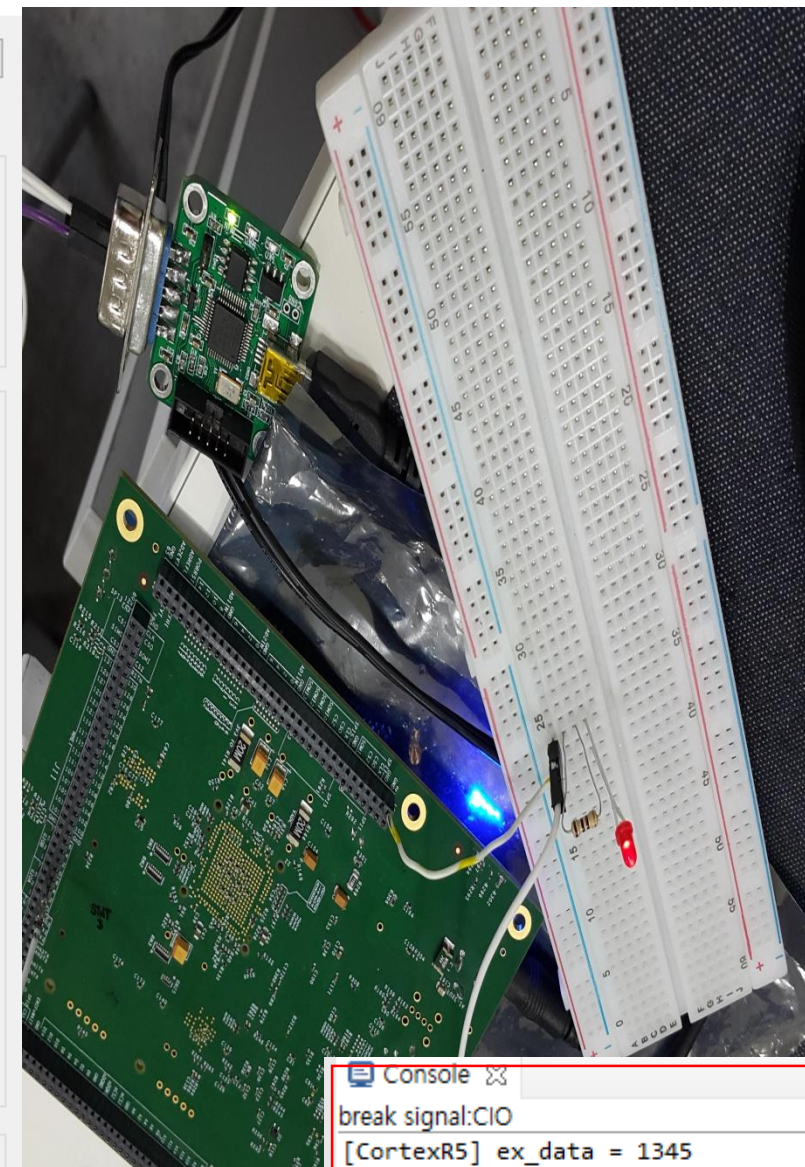
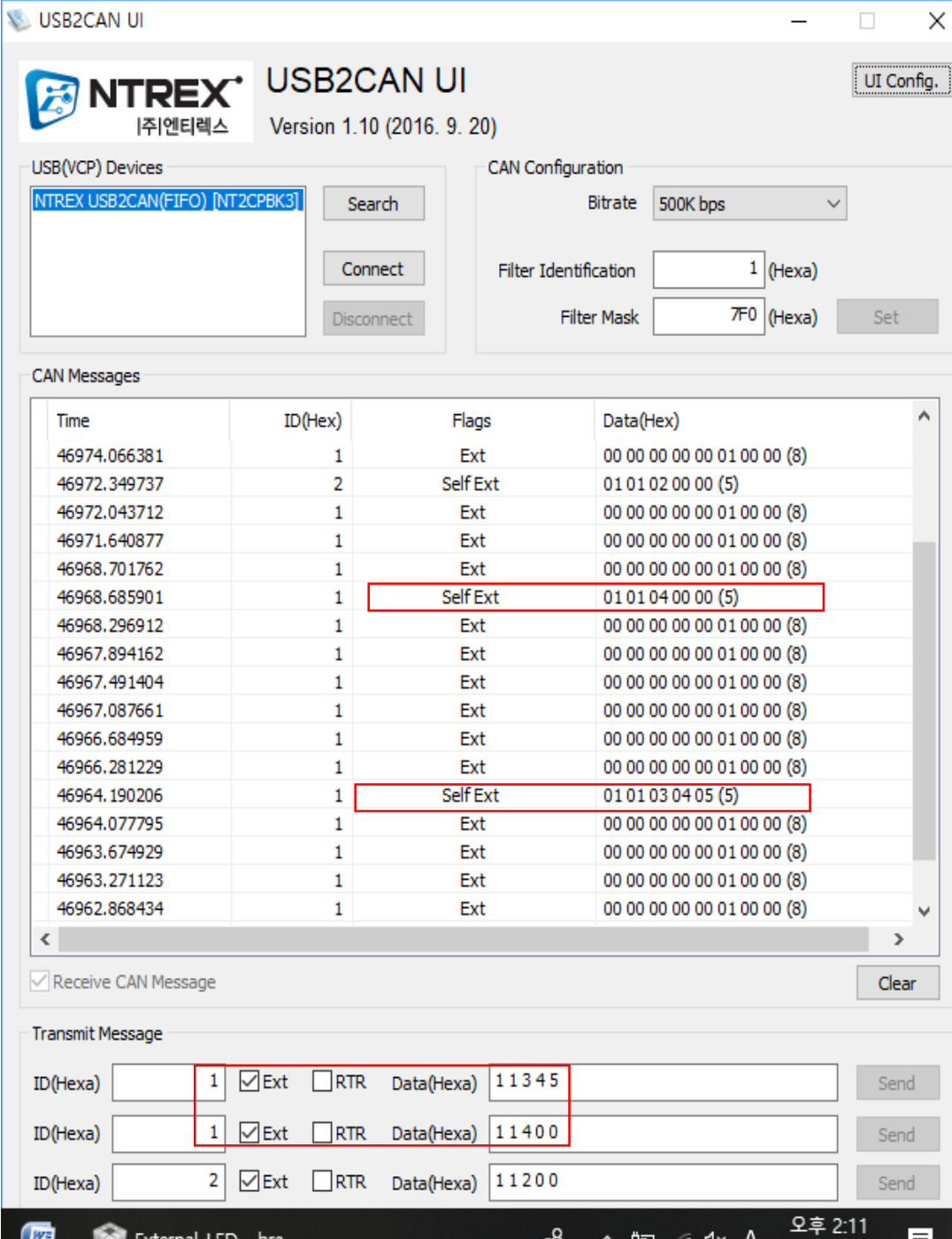
Break Signal_CAN

Designer : 황수정, 김민호

2018.08.29

Project AI CAR





```
Console
break signal:CIO
[CortexR5] ex_data = 1345
cur_data = 1400
f = -55
bigger than ex_data
cur_data = 1200
f = 200
cur_data = 1400
f = -200
```

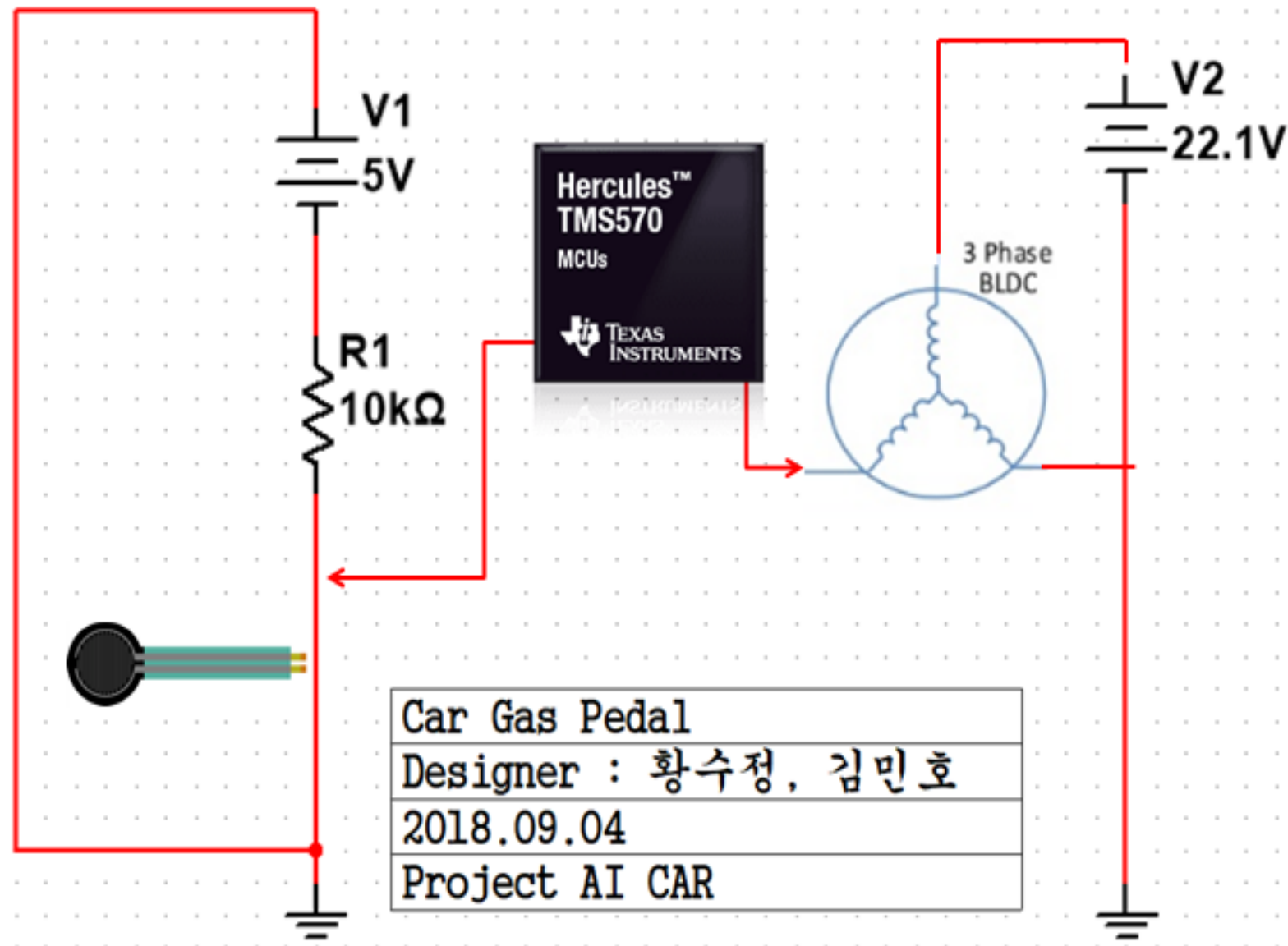
```
#include <HL_adc.h>
#include <HL_gio.h>
#include <HL_hal_stdtypes.h>
#include <HL_reg_adc.h>
#include <HL_reg_gio.h>
#include <HL_reg_sci.h>
#include <HL_sci.h>
#include <HL_etpwm.h>
#include <HL_reg_etpwm.h>
void send_data(sciBASE_t *sci, uint8 *msg, uint32 length)
{
    int i;
    for (i = 0; i < length; i++)
        sciSendByte(sciREG1, msg[i]);
}
void mortor(int rpm)
{
    if (0 <= rpm && rpm <= 500)
    {
        etpwmREG1->CMPA = 1540 * 1.25;
    }
    else if (501 <= rpm && rpm <= 1000)
    {
        etpwmREG1->CMPA = 1550 * 1.25;
    }
    else if (1001 <= rpm && rpm <= 1500)
    {
        etpwmREG1->CMPA = 1560 * 1.25;
    }
    else if (1501 <= rpm && rpm <= 2000)
    {
        etpwmREG1->CMPA = 1570 * 1.25;
    }
    else if (2001 <= rpm && rpm <= 2500)
    {
        etpwmREG1->CMPA = 1580 * 1.25;
    }
    else if (2501 <= rpm && rpm <= 3000)
    {
        etpwmREG1->CMPA = 1590 * 1.25;
    }
    else if (3001 <= rpm && rpm <= 4000)
    {
        etpwmREG1->CMPA = 1600 * 1.25;
    }
}
```

```
void delay(int EE)
{
    int i;
    for (i = 0; i < EE; i++)
    {
        ;
    }
}
```

```
void main(void)
{
    adcData_t counter;
    uint8 msg[32] = { 0, };
    uint32 value = 0;
    etpwmInit();
    etpwmStartTBCLK();
    sciInit();
    gioInit();
    adclInit();
    adcStartConversion(adcREG1, adcGROUP1);
    gioSetBit(gioPORTB, 0, 0);
    etpwmREG1->CMPA = 1500 * 1.25;
    while (1)
    {
        gioSetBit(gioPORTB, 0, 1);
        while (adclConversionComplete(adcREG1, adcGROUP1) == 0)
            ;
        adcGetData(adcREG1, adcGROUP1, &counter);
        sprintf(msg, "value = %d\r\n", counter.value);
        send_data(sciREG1, msg, strlen(msg));
        mortor(counter.value);
        delay(9000000);
        gioSetBit(gioPORTB, 0, 0);
    }
}
```

Car Gas Pedal Setting

03 진행사항





COM4 - PuTTY

```
value = 3607
value = 3534
value = 3505
value = 3479
value = 3489
value = 3288
value = 3219
value = 3520
value = 3578
value = 3460
value = 3432
value = 3275
value = 3069
value = 3349
value = 3358
value = 3365
value = 3422
value = 3396
value = 3393
value = 3313
value = 3319
value = 3347
value = 3478
value = 3544
value = 283
value = 139
value = 1062
value = 1215
value = 2140
value = 3160
value = 3109
value = 1713
value = 83
value = 1165
value = 1296
value = 1171
value = 1296
value = 1663
value = 1791
value = 2346
value = 2668
value = 3224
value = 3287
value = 522
value = 122
value = 101
value = 924
value = 1029
value = 1513
value = 1919
value = 2385
value = 2689
value = 2917
value = 3195
value = 3370
value = 50
value = 86
value = 93
value = 94
value = 86
value = 98
value = 102
value = 135
value = 103
value = 94
value = 96
```

사각형

감사합니다

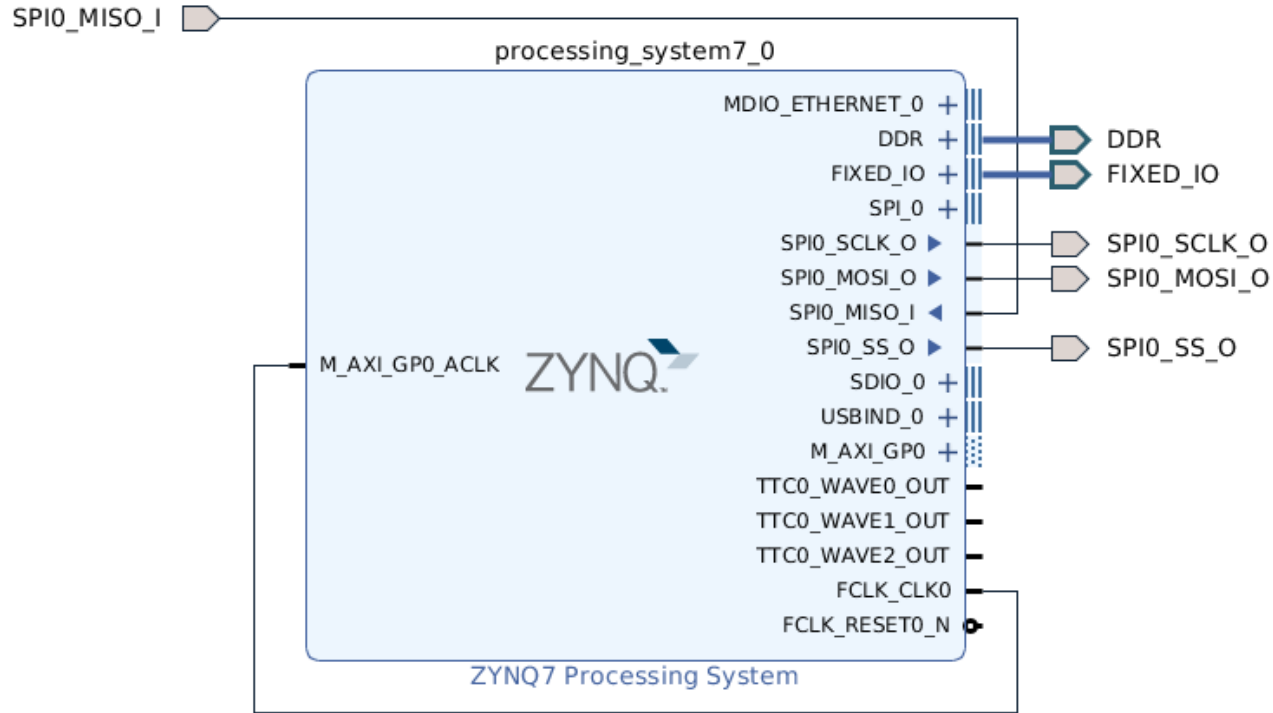


03

진행상황 및 문제점

김시윤

FPGA SPI Controller H/W



SPI 프로토콜은 ZYNQ에서 지원한다.

SPI를 사용하는 모드에 맞게 SPI핀을 Create Port해주면 된다.

들어오는 클럭이 없으므로 FCLK를 동작 트리거로 설정해주고 PMOD Can 모듈과 통신 할 계획이므로 SPI는 Master 모드에 맞게 구성한다.

FPGA SPI Controller Device Tree

```
1 #dts-v1/;
2 /include/ "system-conf.dtsi"
3 / {
4 };
5
6
7 &spi0 {
8     is-decoded-cs = <0>;
9     num-cs = <1>;
10    status = "okay";
11    spidev@0x00 {
12        compatible = "spidev";
13        spi-max-frequency = <50000000>;
14        reg = <0>;
15    };
16 };
```

Petalinux-config -c kernel -> device driver -> spi support -> user 로 시작된걸 활성화 시키고

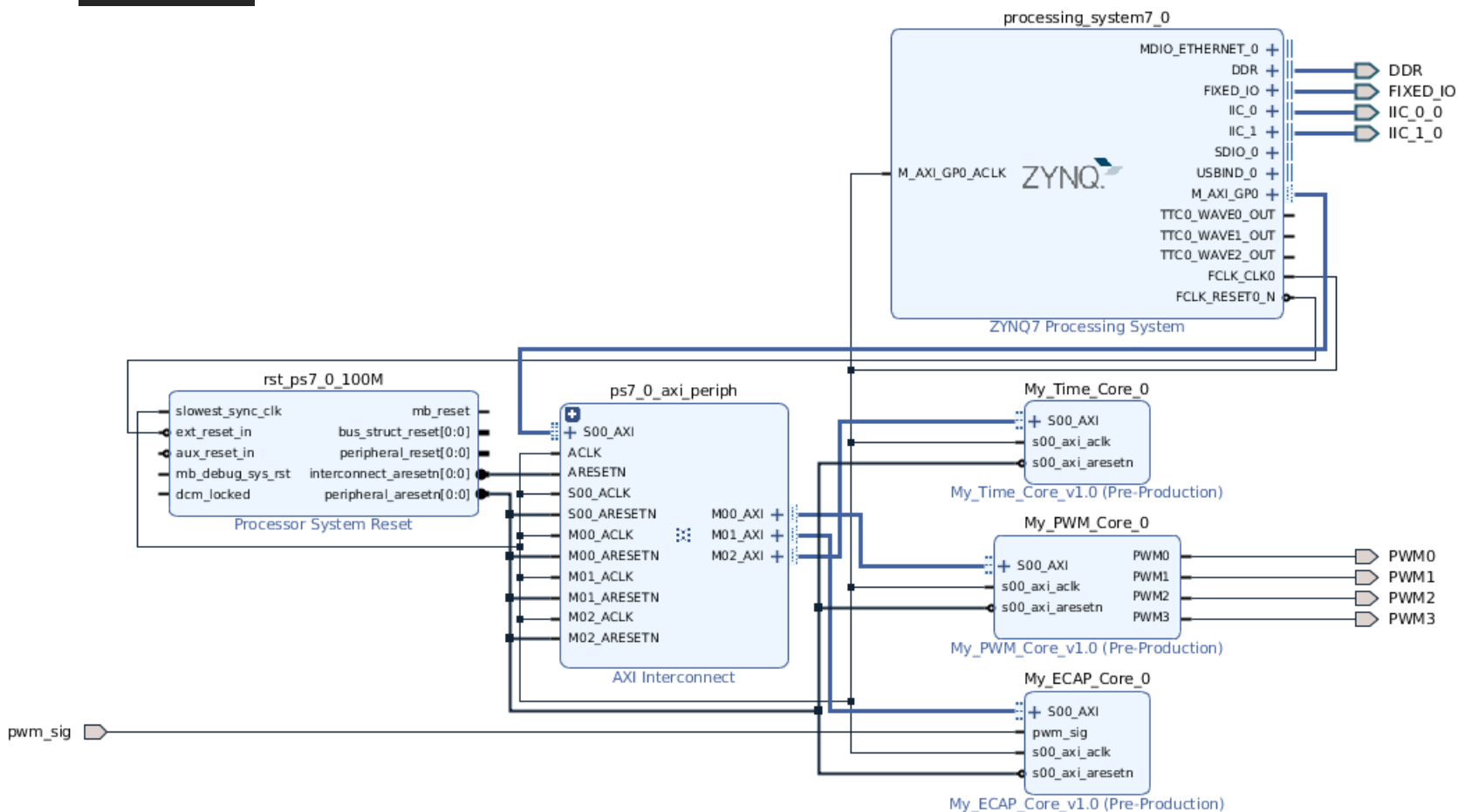
Petalinux-config -c u-boot 에서 spi zynq를 활성화 시킨 뒤

SPI 장치파일을 생성하기 위해 device tree를 작성한다.

그후 petalinux-build

Petalinux-package 를 진행 한 후 장치파일이 생긴것을 확인하였다.

FPGA 통합테스트 H/W 설계



현재까지 만든 Custom IP 와 I2C 핀 2 개 를 활성화 시킨 후 연결한다.
Custom IP 는 github를 확인한다.

I2c-0은 mpu9250을 위한 핀, i2c-1은 라이더를 위한 핀으로 사용하도록 한다.

Time_Core 는 mpu9250의 데이터 컨버팅에 필요한 회로이며 PWM과 ECAP는 모터에 필요하다.

FPGA 통합테스트 H/W 설계

Diagram x Address Editor x					
Q ≡ ⚙ 🗑					
Cell	Slave Interface	Base Name	Offset Address	Range	High Address
▼ 📌 processing_system7_0					
▼ 🗃 Data (32 address bits : 0x40000000 [1G])					
My_ECAP_Core_0	S00_AXI	S00_AXI_reg	0x43C1_0000	64K ▼	0x43C1_FFFF
My_PWM_Core_0	S00_AXI	S00_AXI_reg	0x43C0_0000	64K ▼	0x43C0_FFFF
My_Time_Core_0	S00_AXI	S00_AXI_reg	0x43C2_0000	64K ▼	0x43C2_FFFF

Address Editor 에 Custom IP의 Base address가 나와있다.
원래 Base address의 역순으로 장치파일인 uio2,uio1,uio0이 생겼지만 이본엔 이상하게 순서대로 생겼다.

```
/*uio0 = ECAP, uio1 = PWM*/  
/*uio2 = timer*/
```

FPGA 통합테스트

```
Built with PetaLinux v2015.4 (Yocto 1.8) software /dev/ttyPS0
software login: root
Password:
login[887]: root login on 'ttyPS0'
root@software:~# cd /dev/
root@software:/dev# ls
console          ram1             tty21            tty51
cpu_dma_latency ram10            tty22            tty52
flash            ram11            tty23            tty53
full             ram12            tty24            tty54
i2c-0            ram13            tty25            tty55
i2c-1            ram14            tty26            tty56
iio:device0      ram15            tty27            tty57
initctl          ram2             tty28            tty58
input            ram3             tty29            tty59
kmsg             ram4             tty3              tty6
loop-control     ram5             tty30            tty60
loop0            ram6             tty31            tty61
loop1            ram7             tty32            tty62
loop2            ram8             tty33            tty63
loop3            ram9             tty34            tty7
loop4            random           tty35            tty8
loop5            shm              tty36            tty9
loop6            snd              tty37            ttyPS0
loop7            tty              tty38            uio0
mem              tty0             tty39            uio1
memory_bandwidth tty1             tty4              uio2
mmcblk0          tty10            tty40            urandom
mmcblk0p1        tty11            tty41            vcs
mmcblk0p2        tty12            tty42            vcs1
mtab             tty13            tty43            vcsa
network_latency  tty14            tty44            vcsa1
network_throughput tty15           tty45            vga_arbiter
null             tty16            tty46            watchdog
port            tty17            tty47            watchdog0
psaux           tty18            tty48            xdevcfg
ptmx            tty19            tty49            zero
pts             tty2             tty5
ram0            tty20            tty50
root@software:/dev#
```

소스코드를 짜기전에 장치파일이 잘 생겼는지 먼저 확인하였다.

I2c 장치파일은 petalinux-config -c u-boot 에서 i2c 를 모두 enable 시키면 된다.

Uio 는 Custom IP 를 system-top.dts 에다 generic-uio를 추가해 주면 된다.

FPGA 통합테스트

```
1 ifndef PETALINUX
2 $(error "Error: PETALINUX environment variable not set.  Change to the root of your PetaLinux install, and source the settings.sh file")
3 endif
4
5 include apps.common.mk
6
7 APP = device_driver
8 LDLIBS += -lm
9
10 # Add any other object files to this list below
11 APP_OBJS = device_driver.o
12
13 all: build install
14
15 build: $(APP)
16
17 $(APP): $(APP_OBJS)
18 $(CC) $(LD_FLAGS) -o $@ $(APP_OBJS) $(LDLIBS)
19
20 clean:
21 -rm -f $(APP) *.elf *.gdb *.o
22
23 .PHONY: install image
24
25 install: $(APP)
26 $(TARGETINST) -d $(APP) /bin/$(APP)
27 $(TARGETINST) -d -p 0755 device_driver /etc/init.d/device_driver
28 $(TARGETINST) -s /etc/init.d/device_driver /etc/rc5.d/S99device_driver
29 %.o: %.c
30 $(CC) -c $(CFLAGS) -o $@ $<
31
32 help:
33 @echo ""
34 @echo "Quick reference for various supported build targets for $(INSTANCE)."
```

만든 user code 는 math.h를 사용하기 때문에 LDLIBS += -lm 을 통해 lm옵션을 추가하였다.

Boot 하자마자 바로 apps 가 실행될수 있도록 install 부분에 2줄을 추가해 주었다.

FPGA 통합테스트

```
pwm : 174200, period - 2000000, duty = 171601
pwm : 174300, period - 2000000, duty = 171601
pwm : 174400, period - 2000000, duty = 171601
pwm : 174500, period - 2000000, duty = 171601
pwm : 174600, period - 2000000, duty = 171601
pwm : 174700, period - 2000000, duty = 171601
ax = 66.833496  ay = -191.223145  az = 1008.361816 mg
gx = -0.205994  gy = 0.099182  gz = -0.083923 mg
mx =5  my =280  mz =104 mG
q0 =0.993559  qx =-0.096078  qy =-0.026797  qz =-0.053764  Yaw, Pitch, Roll:2.381877  -3.645280  -10.860470
rate = 4397.021484 Hz
pwm : 174800, period - 2000000, duty = 174701
pwm : 174900, period - 2000000, duty = 174701
pwm : 175000, period - 2000000, duty = 174701
pwm : 175100, period - 2000000, duty = 174701
pwm : 175200, period - 2000000, duty = 174701
pwm : 175300, period - 2000000, duty = 174701
pwm : 175400, period - 2000000, duty = 174701
```

실행결과 mpu9250 과 pwm이 정상적으로 작동한다.
Mpu9250은 침대에서 확인했기때문에 roll pitch 값이 0이 아니다.

Mpu9250 코드에 delay하는 동작이 많기 때문에 fork를 사용하여 여러 개의 프로세스에 나눠 소스코드를 구현하였다.

문제점 및 개선사항

1. Custom IP 장치파일인 uio 의 생성 번호

- IP 주소기준 역순으로 생길 때도 있고 순서대로 생길 때도 있다. 직접 값을 넣어보며 확인해야 할 것 같다.

2. ZYBO JC 포트 핀 한 개의 접촉 불량.

- JC 포트의 핀 한 개가 접촉 불량이라 이상한 값이 나오는 현상 하지만 ZYBO가 단종되어 수리 불가능.

3. LIDAR 코드 미구현

- 코드 작성을 늦게 시작하여 구현하지 못했다.

4. SPI ioctl 의문

- 리눅스 커널 드라이빙과 SPI 프로토콜에 대한 개념정리가 매우 필요하다고 생각된다.

감사합니다

04 진행상황 및 문제점

이유성

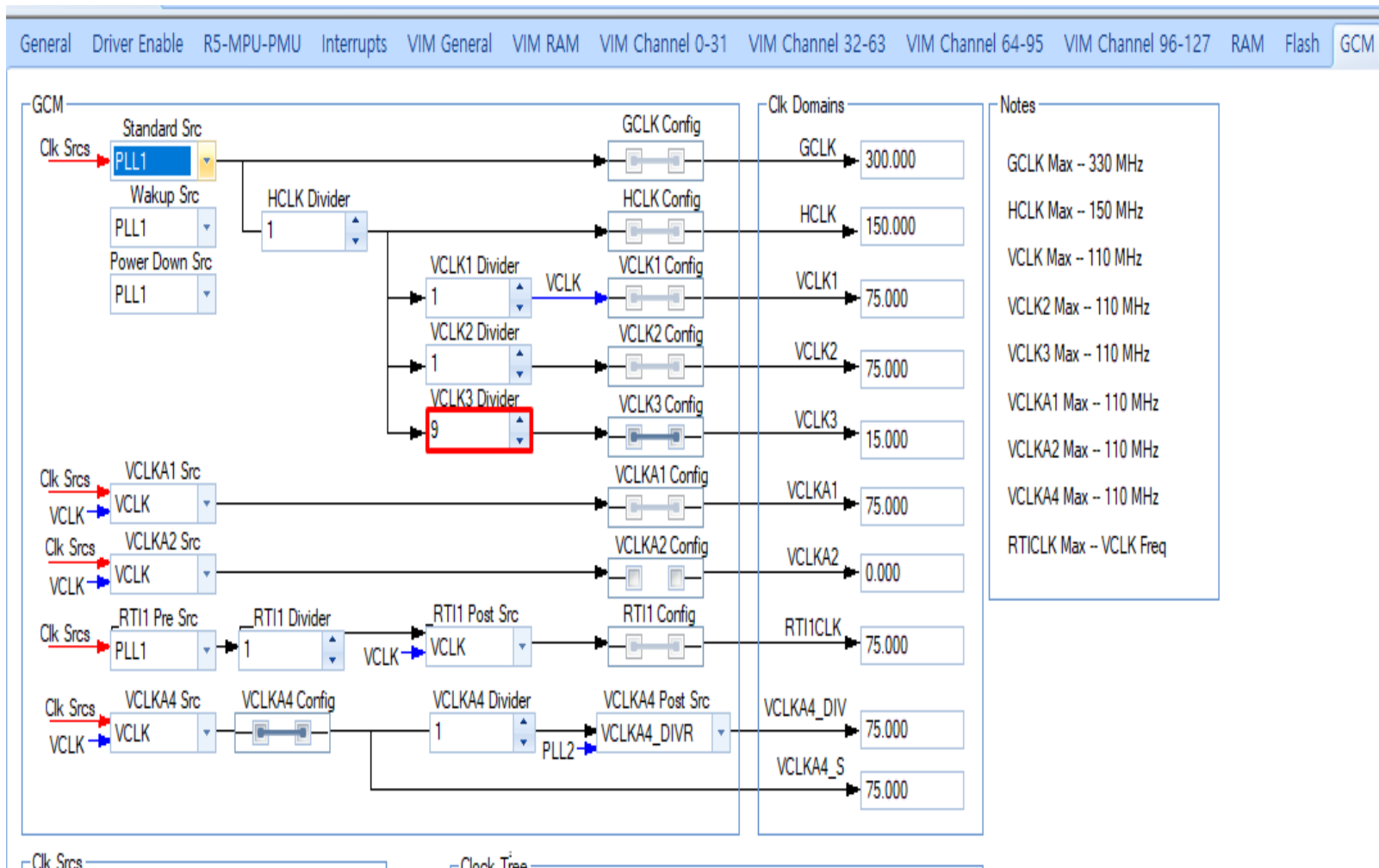
모터 eQEP

엔코더에서 나오는 펄스를 통해 eQEP로 각속도,방향을 구할 수 있다.

모터 eQEP HCG 설정

- ☐ Enable LIN drivers
 - ☐ Enable LIN1 driver ** / ☒ Enable SCI1 driver **
 - ☐ Enable LIN2 driver ** / ☐ Enable SCI2 driver **
- ☐ Enable MIBSPI drivers
 - ☐ Enable MIBSPI1 driver ** ☐ Enable SPI1 driver **
 - ☐ Enable MIBSPI2 driver ** ☐ Enable SPI2 driver **
 - ☐ Enable MIBSPI3 driver ** ☐ Enable SPI3 driver **
 - ☐ Enable MIBSPI4 driver ** ☐ Enable SPI4 driver **
 - ☐ Enable MIBSPI5 driver ** ☐ Enable SPI5 driver **
- ☐ Enable CAN drivers
 - ☐ Enable CAN1 driver
 - ☐ Enable CAN2 driver
 - ☐ Enable CAN3 driver
 - ☐ Enable CAN4 driver **
- ☐ Enable ADC drivers
 - ☐ Enable ADC1 driver **
 - ☐ Enable ADC2 driver **
- ☐ Enable HET drivers
 - ☐ Enable HET1 driver **
 - ☐ Enable HET2 driver **
- ☐ Enable I2C driver **
 - ☐ Enable I2C1 driver **
 - ☐ Enable I2C2 driver **
- ☐ Enable EMAC driver **
- ☐ Enable DCC driver
- ☐ Enable EMIF driver **
- ☐ Enable POM driver
- ☐ Enable CRC driver
 - ☐ Enable CRC1 driver
 - ☐ Enable CRC2 driver
- ☒ Enable EQEP driver
 - ☒ Enable EQEP1 driver **
 - ☐ Enable EQEP2 driver **
- ☒ Enable ETPWM driver

모터 eQEP HCG 설정



모터 eQEP HCG 설정

TMS570LC4357ZWT PINMUX RTI GIO ESM SCI1 SCI2 SCI3 SCI4 LIN1 LIN2 MIBSPI1 MIBSPI2 MIBSPI3 MIBSPI4 MIBSPI5 SPI1

Pin Muxing Input Pin Muxing Special Pin Muxing

Enable / Disable Peripherals

☐ HET1
 ☐ HET2
 ☐ EMIF
 ☐ ETPWM
 ☐ GIOA
 ☐ GIOB
 ☒ EQEP
 ☐ ECAP
 ☐ MIBSPI2
 ☐ MIBSPI4
 ☐ AD1EVT
 ☐ AD2EVT
 ☐ MIBSPI1
 ☐ MIBSPI3
 ☐ MIBSPI5
 ☐ I2C1
 ☐ SCI3
 ☐ SCI4
 ☐ LIN2/SCI2
 ☐ I2C2
 ☐ RMI
 ☐ MII
 ☐ CAN4

Note

GIO pins are mapped to two terminals. The checkboxes enable both the default and alternate terminals. Remove the unwanted terminal to avoid conflicts

MII have dedicated pins. Alternate terminals are enabled using the MII checkbox. RMII and MII checkboxes does not set the functional mode. Enable them in Special Pinmuxing tab

Ball	Default Mux	Mux Option 1	Mux Option 2	Mux Option 3	Mux Option 4	Mux Option 5	Conflict?
A4	N2HET1[16]	NONE	NONE	ETPWM1SYNCl	NONE	ETPWM1SYNCO	
A13	N2HET1[17]	EMIF_nOE	SCI4RX	NONE	NONE	NONE	
A14	N2HET1[26]	NONE	MII_RXD[1]	RMII_RXD[1]	NONE	NONE	
B2	MIBSPI3NCS[2]	I2C1_SDA	NONE	N2HET1[27]	NONE	nTZ1_2	
B3	N2HET1[22]	EMIF_nDQM[3]	NONE	NONE	NONE	NONE	
B4	N2HET1[12]	MIBSPI4NCS[5]	MII_CRS	RMII_CRS_DV	NONE	NONE	
B5	GIOA[5]	NONE	NONE	EXTCLKIN	NONE	eTPWM1A	
	MIBSPI1NCS[1]	DMM_DATA[0]	NONE	NONE	NONE	NONE	

모터 eQEP HCG 설정

TMS570LC4357ZWT PINMUX RTI GIO ESM SCI1 SCI2 SCI3 SCI4 LIN1 LIN2 MIBSPI1 MIBSPI2 MIBSPI3 MIBSPI4 MIBSPI5 SPI1

Pin Muxing Input Pin Muxing Special Pin Muxing

Enable / Disable Peripherals

☐ HET1
 ☐ HET2
 ☐ EMIF
 ☐ ETPWM
 ☐ GIOA
 ☐ GIOB
 ☒ EQEP
 ☐ ECAP
 ☐ MIBSPI2
 ☐ MIBSPI4
 ☐ AD1EVT
 ☐ AD2EVT
 ☐ MIBSPI1
 ☐ MIBSPI3
 ☐ MIBSPI5
 ☐ I2C1
 ☐ SCI3
 ☐ SCI4
 ☐ LIN2/SCI2
 ☐ I2C2
 ☐ RMI
 ☐ MII
 ☐ CAN4

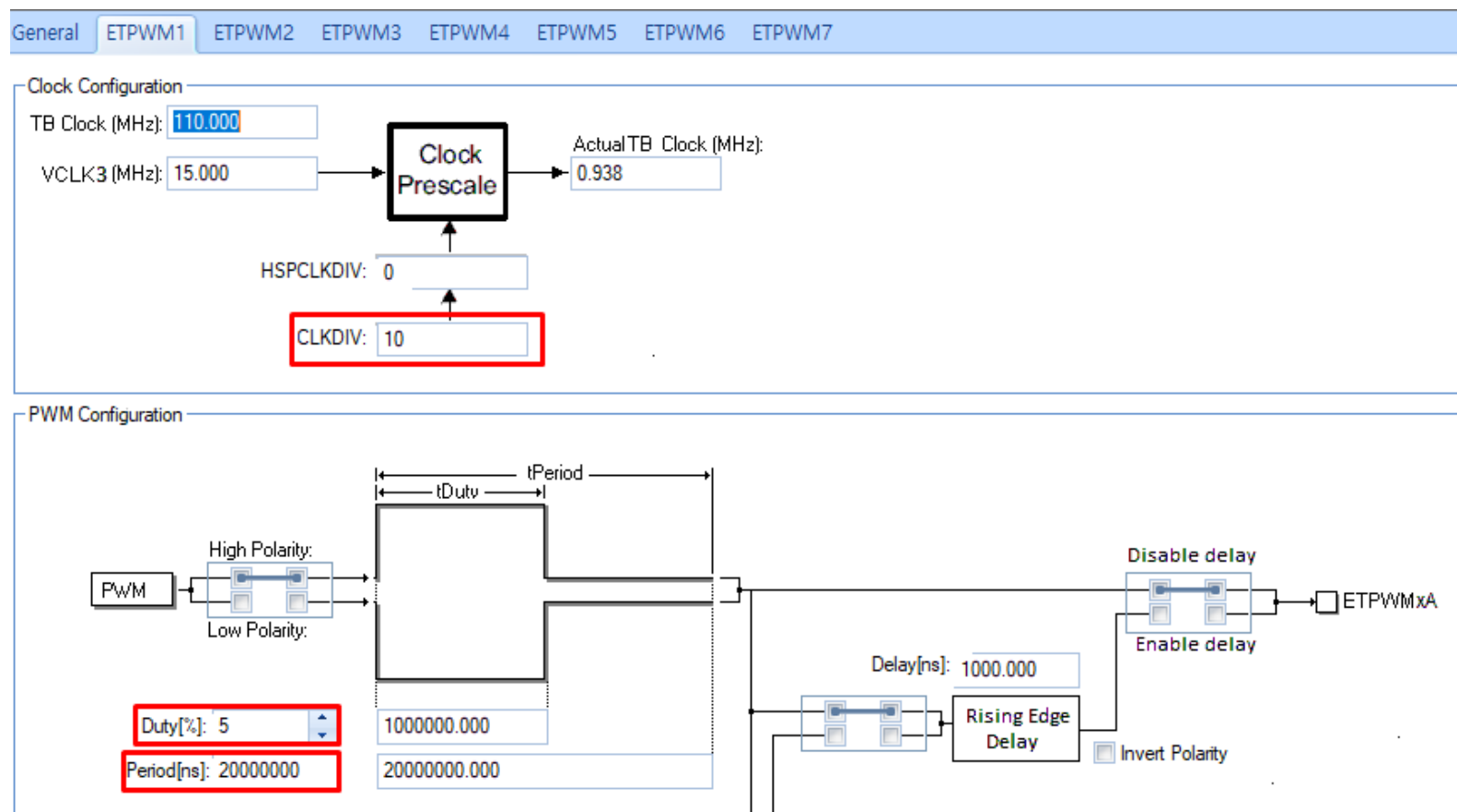
Note

GIO pins are mapped to two terminals. The checkboxes enable both the default and alternate terminals. Remove the unwanted terminal to avoid conflicts

MII have dedicated pins. Alternate terminals are enabled using the MII checkbox. RMII and MII checkboxes does not set the functional mode. Enable them in Special Pinmuxing tab

Ball	Default Mux	Mux Option 1	Mux Option 2	Mux Option 3	Mux Option 4	Mux Option 5	Conflict?
A4	N2HET1[16]	NONE	NONE	ETPWM1SYNCl	NONE	ETPWM1SYNCO	
A13	N2HET1[17]	EMIF_nOE	SCI4RX	NONE	NONE	NONE	
A14	N2HET1[26]	NONE	MII_RXD[1]	RMII_RXD[1]	NONE	NONE	
B2	MIBSPI3NCS[2]	I2C1_SDA	NONE	N2HET1[27]	NONE	nTZ1_2	
B3	N2HET1[22]	EMIF_nDQM[3]	NONE	NONE	NONE	NONE	
B4	N2HET1[12]	MIBSPI4NCS[5]	MII_CRS	RMII_CRS_DV	NONE	NONE	
B5	GIOA[5]	NONE	NONE	EXTCLKIN	NONE	eTPWM1A	
	MIBSPI5NCS[1]	DMM_DATA[0]	NONE	NONE	NONE	NONE	

모터 eQEP HCG 설정



모터 eQEP HCG 설정

EQEP1
EQEP2

General Configuration

Position Counter Mode:

QUADRATURE_COUNT

External clock rate:

RESOLUTION_1x

Select QDIR:

CLOCKWISE

☐ Invert QEPxA Polarity

☐ Invert QEPxB Polarity

☐ Invert QEPxI Polarity

☐ Invert QEPxS Polarity

☐ Gate Index Pin with Strobe

☐ Swap Quadrature Clock Input

Compare Output Configurations

Sync Output Pin Select:

INDEX_PIN

Shadow Load Mode:

QPOSCNT_EQ_QPSCMP

Sync Output Polarity:

ACTIVE_HIGH

☐ Enable Sync Output

☐ Enable Position Compare Shad

Compare Value:

0x00000000

Sync Pulse Width:

0x000

x 4 VCLK4

Position Counter Configuration

Counter Init Index Event:

RISING_EDGE

Counter Init Strobe Event:

DIRECTON_DEPENDENT

Position Counter Reset On:

MAX_POSITION

Counter Latch Index Event:

RISING_EDGE

Counter Latch Strobe Event:

RISING_EDGE

Max Position Count:

0x0000FFFF

☐ Init Counter on Index Event

☐ Init Counter on Strobe Event

☐ Enable SW Initialization

Init Position Count to

0x00000000

Capture Configuration

Capture Timer Prescaler:

PS_1

Unit Pos Event Prescaler:

PS_1

Cap Timer Pos Mode:

ON_UNIT_TIMEOUT_EVENT

☐ Init Counter on Strobe Event

Unit Init Period:

0x0000000F

Interrupt Configuration

☐ Position counter error Interrupt

☐ Quadrature phase error Interrupt

☐ Quadrature direction change Interrupt

☐ Watchdog time out Interrupt

☐ Position counter underflow Interrupt

☐ Position counter overflow Interrupt

☐ Position-compare ready Interrupt

☐ Position-compare match Interrupt

☐ Strobe event latch Interrupt

☐ Index event latch Interrupt

☐ Unit time out interrupt

Watchdog Configuration

Watchdog Timer Value:

0x0000

모터 CCS 코드

```
#include "HL_etpwm.h"
#include "HL_sci.h"
#include "HL_eqep.h"
#include "HL_sys_common.h"
#include "HL_system.h"
#include "math.h"
#include "stdio.h"
#include "HL_eqep.h"
#include "HL_esm.h"
#include "HL_sys_core.h"
#include "stdlib.h"
#include "string.h"

#define UNIT_TIME_OUT 0.5
#define UNIT_POSITION_ANGLE 180.0 // 180도당 unit position 발생, 펄스당 각도 2 P/R 르 360을 나눔
#define circle_angle 360.0

void delay(uint32 time);
void pwm_set(uint32 value);

uint8 input;
char msg[256] = {0,};

// HalCoGen: 주기 20ms, VCLK3: 15Mhz를 CLKDIV를 10분주해서 Actual TB Clock은 0.938Mhz
int main(void)
{
    int QCLK_Value = 0 ;
    double QCLK_Value_Prescale = 0 ;
    double gear_ratio = 74.84835 ;
    double real_velocity = 0 ;
    double angle_velocity ;
    uint8 i, num = 0;
    int value = 0;
    double real_wheel_revolution = 0;
    long double distance_per_revolution = 0.0007172249 ; // 지름 : 22.83cm -> 22.83 * 10^-5 (km단위) * 파이 (3.14159) = 0.0007172249
    double BLDC_MOTOR_revolution_per_sec = 0 ;

    QEPIInit();
    etpwmInit();
    sciInit();
    etpwmStartTBCLK();

    _enable_IRQ_interrupt();

    delay(1000);

    eqepEnableUnitTimer(eqepREG1); // eQEP unit timer is enabled
    eqepEnableCounter(eqepREG1); // eQEP position counter is enabled.
    eqepEnableCapture(eqepREG1); //eQEP capture unit is enabled

    eqepREG1->QEPCNTL |= 0x3000; //13-12 3h : Position counter reset on a unit time event.

    eqepREG1->QUPRD = 0x7270E0 ; // 15Mhz -> 1초, 7.5Mhz -> 0.5초 7500000 == 0x7270E0
```


모터 CCS 코드

```
while(1)
{
    if(sciIsRxReady(sciREG1)){
        sciSend(sciREG1, strlen("Put a Value for PWM Duty(0-1876): "), "Put a Value for PWM Duty(0-1876): ");

        for(i=0;i<4;i++)
        {
            input = sciReceiveByte(sciREG1);
            sciSendByte(sciREG1, input);
            num = input - 48;
            value += num * pow(10,3-i);
            if(num+48 == '\r' || num+48 == '\n'){
                input = 0 ;
                break;
            }
        }
        delay(1000);

        if(value>=0 && value<1877) // 1876 -> 0938 -> 1407 -> 원하는 값. (1500이면 적당히 돌아감) 1440 -> 0.59바퀴 , 1450 -> 1.5바퀴
        {
            sprintf(msg, "\r\nPWM Duty : %d\r\n", value);
            sciSend(sciREG1, strlen(msg), msg);
            pwm_set(value);
        }

        else
            sciSend(sciREG1, strlen("\r\nWrong Value, Range 0 to 1876\r\n"), "\r\nWrong Value, Range 0 to 1876\r\n");

        value = 0;
    }

    QCLK_Value = (eqepREG1->QPOS1AT);
    printf("QCLK_Value = %d\n", QCLK_Value ); // 0.5초 동안 QCLK_Value 만큼 QCLK가 발생한다.
    QCLK_Value_Prescale = QCLK_Value / 4 ;
    angle_velocity = (QCLK_Value_Prescale * UNIT_POSITION_ANGLE) / UNIT_TIME_OUT ; //초당 angle_velocity도 만큼 회전한다.

    printf("각 속도 : %f\n", angle_velocity);

    BLDC_MOTOR_revolution_per_sec = angle_velocity/circle_angle;
    printf("BLDC MOTOR 초당 회전 수 : %f\n", BLDC_MOTOR_revolution_per_sec);

    real_wheel_revolution = BLDC_MOTOR_revolution_per_sec / gear_ratio ;

    printf("실제 바퀴 초당 회전 수 : %f\n", real_wheel_revolution);

    real_velocity = 3600 * real_wheel_revolution * distance_per_revolution ; // 시간 * 회전 수 * 거리

    printf("속도 km/h : %f\n", real_velocity) ;

}
}
```

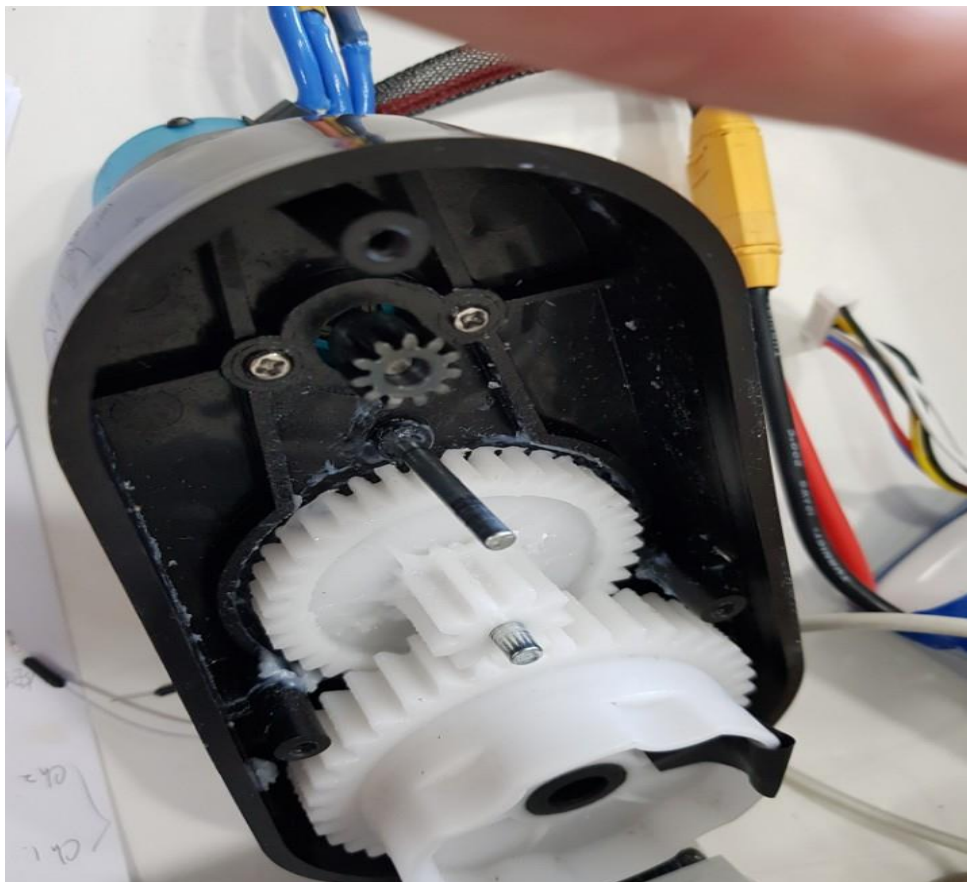
모터 CCS 코드

```
void pwm_set(uint32 value)
{
    etpwmREG1->CMPA = value;
}

void delay(uint32 time)
{
    int i;
    for(i=0; i<time ; i++)
        ;
}
```

원하는 듀티를 주면서 속도를 계속 체크할 수 있는 코드

기어비 구하기



기어비 =
11 : 40 -> 1회전 당 3.63636회전
8 : 38 -> 1회전 당 4.75회전
9 : 39 -> 1회전 당 4.333333회전
총 모터 74.8483524회전 당 바퀴 1회전

기어비 구하기



기어비 =
11 : 40 -> 1회전 당 3.63636회전
8 : 38 -> 1회전 당 4.75회전
9 : 39 -> 1회전 당 4.333333회전
총 모터 74.8483524회전 당 바퀴 1회전

모터

바퀴 지름 : 22.83cm ,

바퀴 원주 : $2 \cdot R \cdot \text{파이}(3.141592) = 71.72254536(\text{CM})$

=> 0.0007172254536(KM)

지름 : 22.83cm -> $22.83 \cdot 10^{-5} (\text{km단위}) \cdot \text{파이} (3.14159) = 0.0007172249$

기어비 =

11: 40 -> 1회전 당 3.63636회전

8 : 38 -> 1회전 당 4.75회전

9 :39 -> 1회전 당 4.333333회전

총 모터 74.8483524회전 당 바퀴 1회전

즉, BLDC모터가 74.8483524회전 해야 0.0007172254536km를 움직인다.

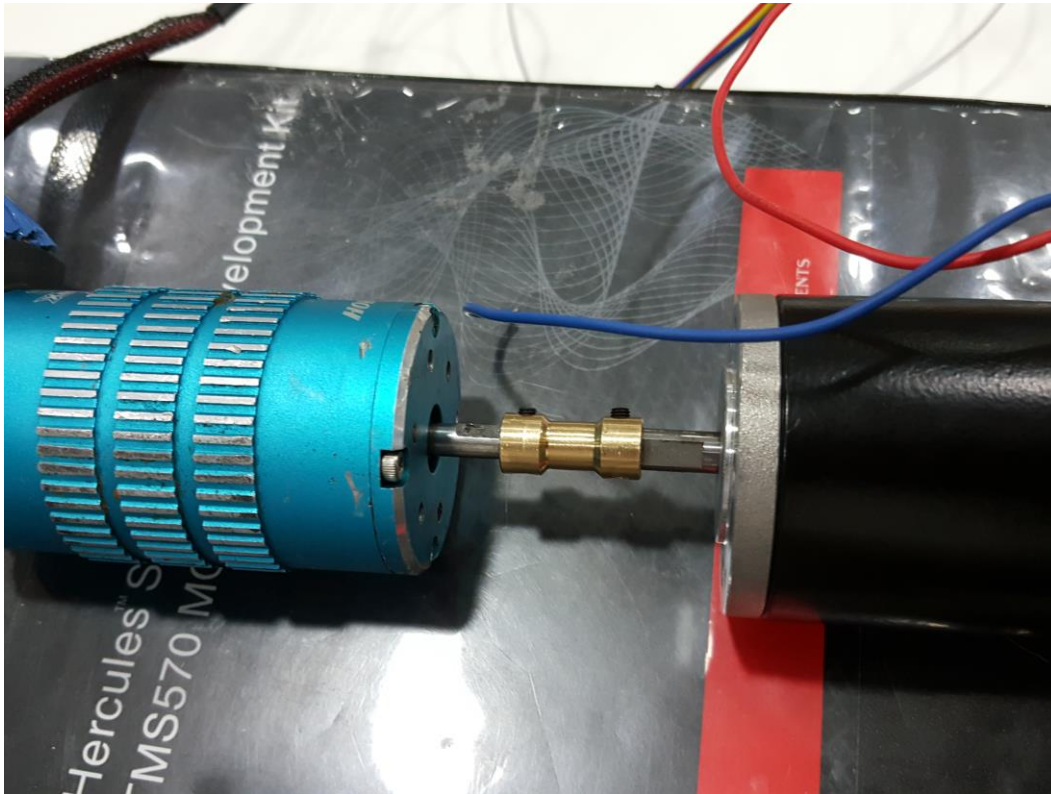
초당 각속도를 구해서 초당 얼마나 나가는지 거리를 구해보자.

문제점

문제점 발생 : Sensored 모터에서 얼마의 펄스가 나오는지 모른다

해결 -> 커플링 커넥터로 펄스를 알아냄 !

문제점



CH 1 .(DC모터 432P/R)

1번째:9Khz

2번째:4khz

CH 2 .(BLDC모터 ?P/R)

1번째:41.5Hz

2번째:18Hz

$$4000 : 18 = 432 : X \rightarrow X = 1.994$$

$$9000 : 41.5 = 432 : X \rightarrow X = 1.992$$

BLDC모터의 펄스가 회전당 2번
나오는 것을 확인.

조향제어 제어 구현 중

1. 기어 박스를 서보모터에 맞게 만들었음. 구현할 코드도 있음
2. 서보모터만 오면 됨.

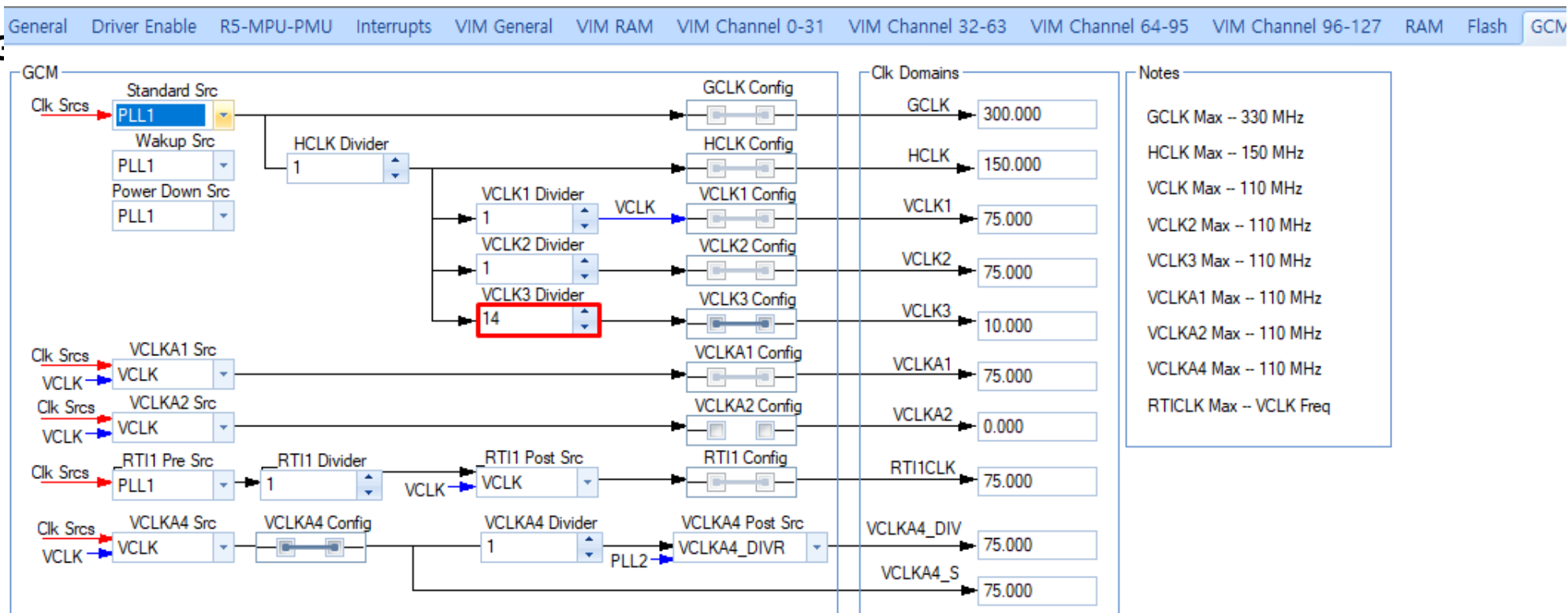
문제점 : 서보모터가 알 수 없는 고장으로 2번이나 교체해야하는 상황이 있어서
구현이 늦어졌다.

조향 제어 HCG

- ☐ Enable LIN drivers
 - ☐ Enable LIN1 driver ** / ☒ Enable SCI1 driver **
 - ☐ Enable LIN2 driver ** / ☐ Enable SCI2 driver **
- ☐ Enable MIBSPI drivers
 - ☐ Enable MIBSPI1 driver ** ☐ Enable SPI1 driver **
 - ☐ Enable MIBSPI2 driver ** ☐ Enable SPI2 driver **
 - ☐ Enable MIBSPI3 driver ** ☐ Enable SPI3 driver **
 - ☐ Enable MIBSPI4 driver ** ☐ Enable SPI4 driver **
 - ☐ Enable MIBSPI5 driver ** ☐ Enable SPI5 driver **
- ☐ Enable CAN drivers
 - ☐ Enable CAN1 driver
 - ☐ Enable CAN2 driver
 - ☐ Enable CAN3 driver
 - ☐ Enable CAN4 driver **
- ☐ Enable ADC drivers
 - ☐ Enable ADC1 driver **
 - ☐ Enable ADC2 driver **
- ☐ Enable HET drivers
 - ☐ Enable HET1 driver **
 - ☐ Enable HET2 driver **
- ☐ Enable I2C driver **
 - ☐ Enable I2C1 driver **
 - ☐ Enable I2C2 driver **
- ☐ Enable EMAC driver **
- ☐ Enable DCC driver
- ☐ Enable EMIF driver **
- ☐ Enable POM driver
- ☐ Enable CRC driver
 - ☐ Enable CRC1 driver
 - ☐ Enable CRC2 driver
- ☐ Enable EQEP driver
 - ☐ Enable EQEP1 driver **
 - ☐ Enable EQEP2 driver **
- ☒ Enable ETPWM driver

조향 제어할 HCG 설정

조향 제어 HCC



조향 제어 HCG

General ETPWM1 ETPWM2 ETPWM3

Enable ETPWM modules

☒ Enable ETPWM1

☐ Enable ETPWM2

☐ Enable ETPWM3

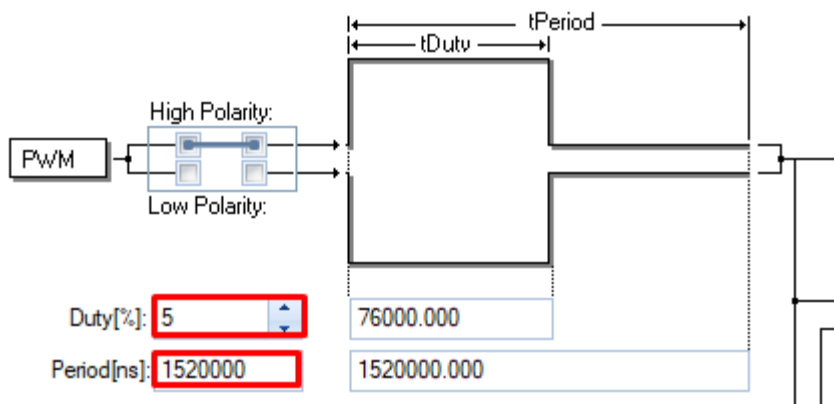
☐ Enable ETPWM4

☐ Enable ETPWM5

☐ Enable ETPWM6

☐ Enable ETPWM7

조향 제어 HCG



조향 제어 CCS 코드

```
#include "HL_etpwm.h"
#include "HL_sci.h"
#include "HL_sys_common.h"
#include "HL_system.h"
#include "math.h"
#include "stdio.h"

void delay(uint32 time);
void pwm_set(uint32 value);

uint8 input;
int test_value = 0;
char msg[256] = {0,};
int count = 0 ;
int main(void)
{
    uint32 pr_pulse_cnt = 0;
    double angle_velocity = 0;
    uint8 i, num = 0;
    int value = 0 ;

    etpwmInit();
    sciInit();
    etpwmStartTBCLK();

    delay(1000);

    while(1) // 3029U 1519U
    {
        sciSend(sciREG1, strlen("Put a Value for PWM Duty(0-1876): "), "Put a Value for PWM Duty(0-3000): ");

        for(i=0;i<4;i++)
        {
            input = sciReceiveByte(sciREG1);
            sciSendByte(sciREG1, input);
            num = input - 48;
            value += num * pow(10,3-i);
            if(num+48 == '\r' || num+48 == '\n')
                break;
        }
        delay(1000);

        if(value>=0 && value<3001)
        {
            sprintf(msg, "\r\nPWM Duty : %d\r\n", value);
            sciSend(sciREG1, strlen(msg), msg);
            pwm_set(value);
        }

        else
            sciSend(sciREG1, strlen("\r\nWrong Value, Range 0 to 3000\r\n"), "\r\nWrong Value, Range 0 to 3000\r\n");

        value = 0;
    }
}
```

```
}

void pwm_set(uint32 value)
{
    etpwmREG1->CMPA = value;
}

void delay(uint32 time)
{
    int i;
    for(i=0; i<time ; i++)
        ;
}
```

조향 제어할 소스코드

목표

시간이 얼마 안남았기 때문에 빨리 구현을 해서 주행 테스트를 하자!

감사합니다