# TI DSP, MCU 및 Xilinx Zynq FPGA
# 프로그래밍 전문가 과정

강사 – Innova Lee(이상훈)

gcccompil3r@gmail.com

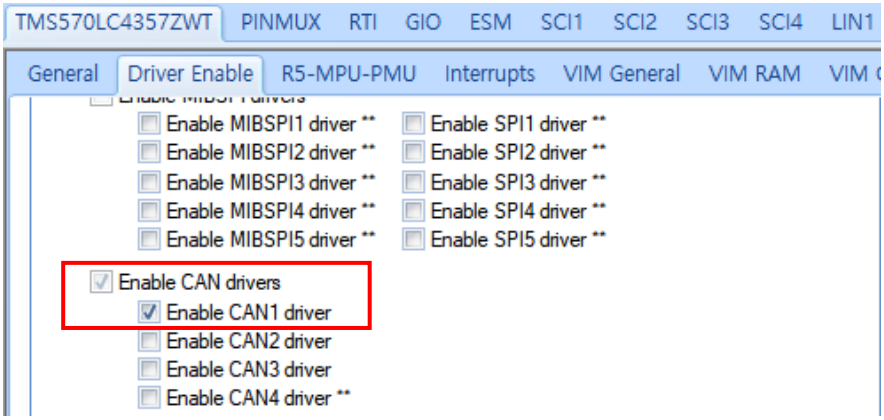
학생 – 문한나

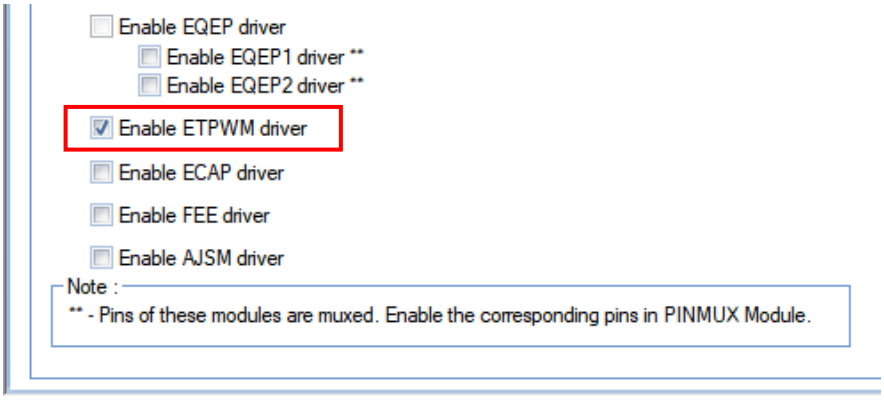
mhn97@naver.com

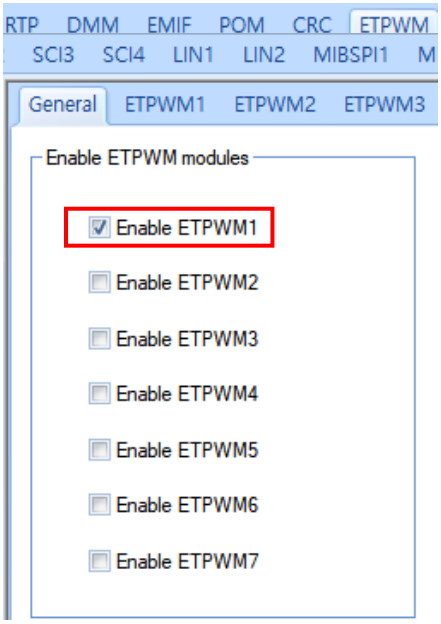
# CAN통신과 PWM신호를 이용한 모터제어

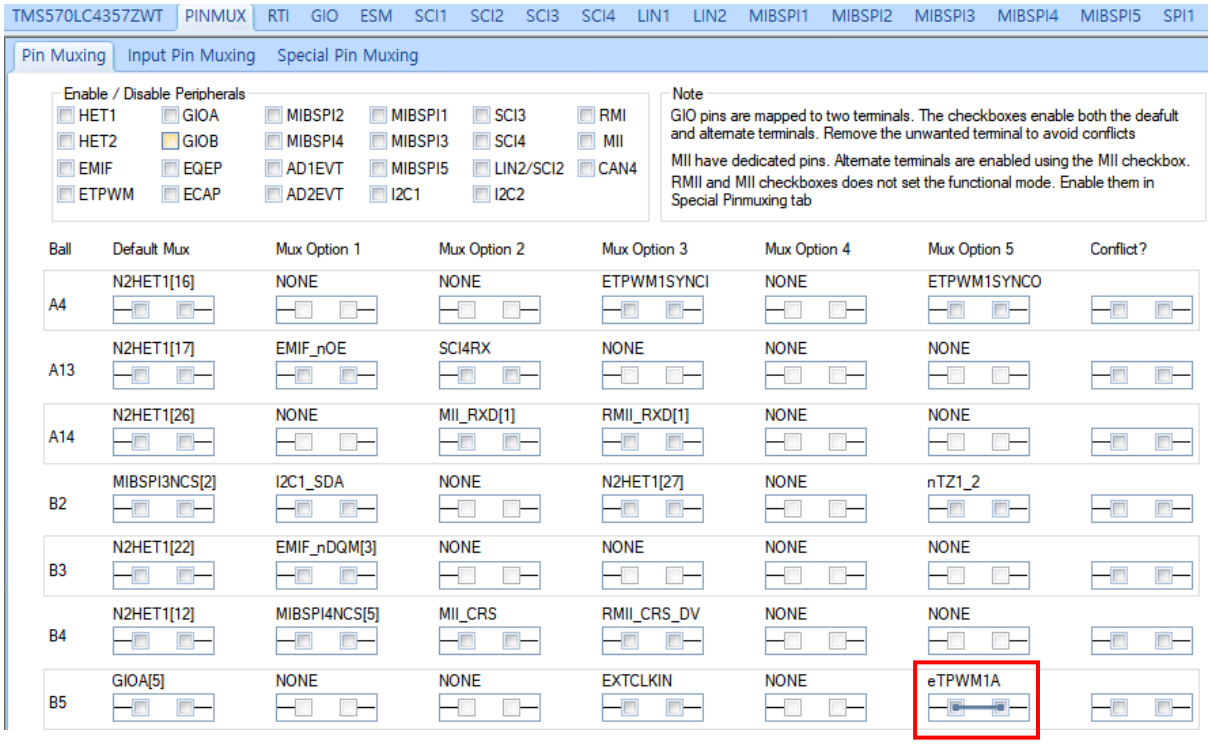
can통신으로 방향과 각도를 전달하고, mcu가 값을 받아 모터를 제어한다.

## *HGC 설정*



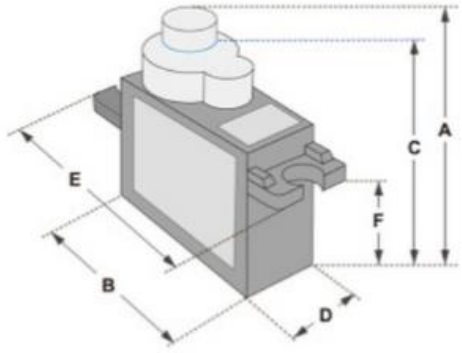





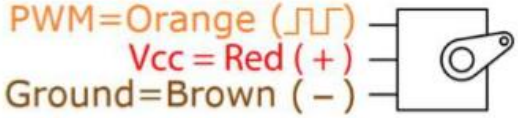

GIOA[5]를 PWM 으로 쓴다

# SERVO MOTOR SG90



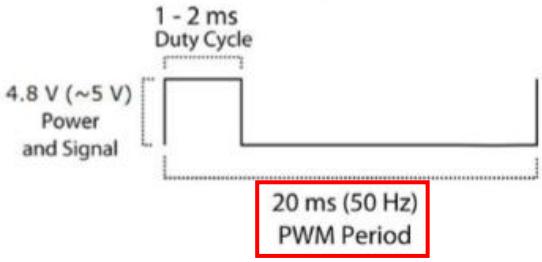

| Dimensions & Specifications |
|---|
| A (mm) : 32 |
| B (mm) : 23 |
| C (mm) : 28.5 |
| D (mm) : 12 |
| E (mm) : 32 |
| F (mm) : 19.5 |
| Speed (sec) : 0.1 |
| Torque (kg-cm) : 2.5 |
| Weight (g) : 14.7 |
| Voltage : 4.8 - 6 |

Position "0" (1.5 ms pulse) is middle, "90" (~2ms pulse) is middle, is all the way to the right, "-90" (~1ms pulse) is all the way to the left.

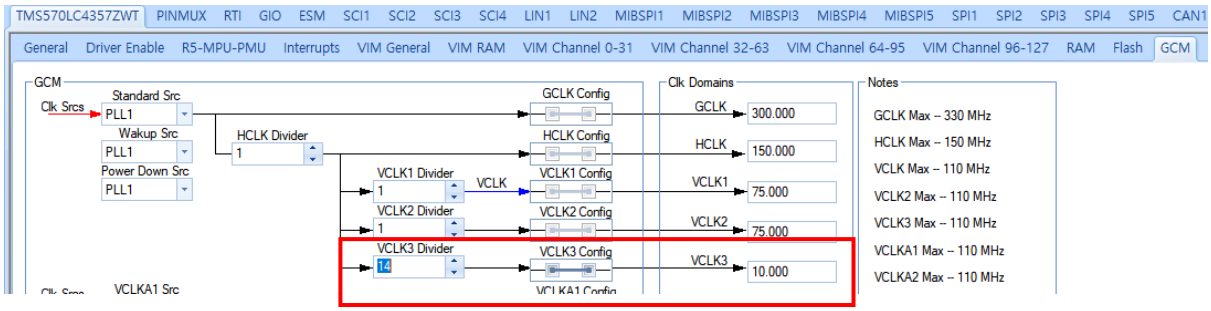



모터 주기 : 20ms

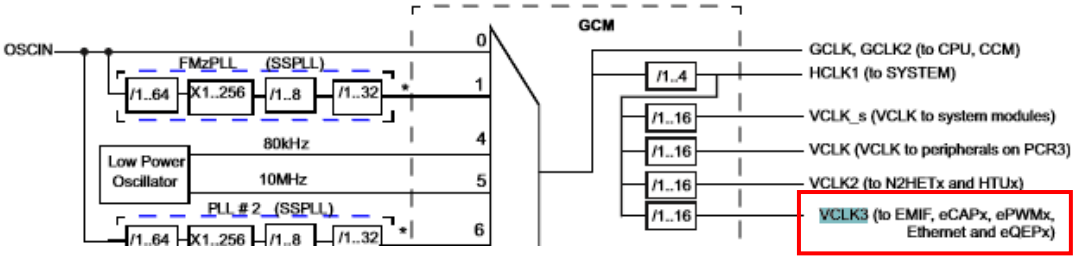
동작주파수 : 50Hz

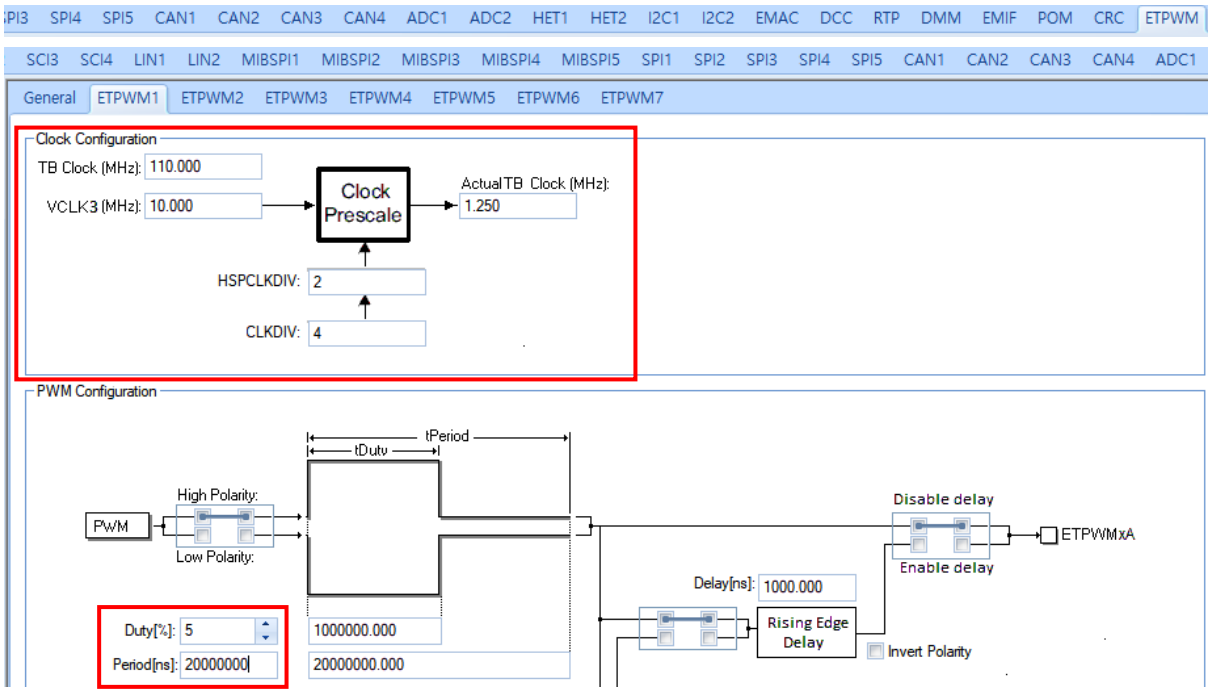



PLL 제어를 통하여 동작주파수 50hz 를 맞춰주자




VCLK3 을 10Mhz 로 분주한다.




주파수 설정 중에 주의해야할 것은 TBPRD레지스터의 임계값이 65536(2 ^ 16)이라는 것이다. 이 값을 넘지 않게 조정해준다.

### 35.4.1.4 Time-Base Period Register (TBPRD)

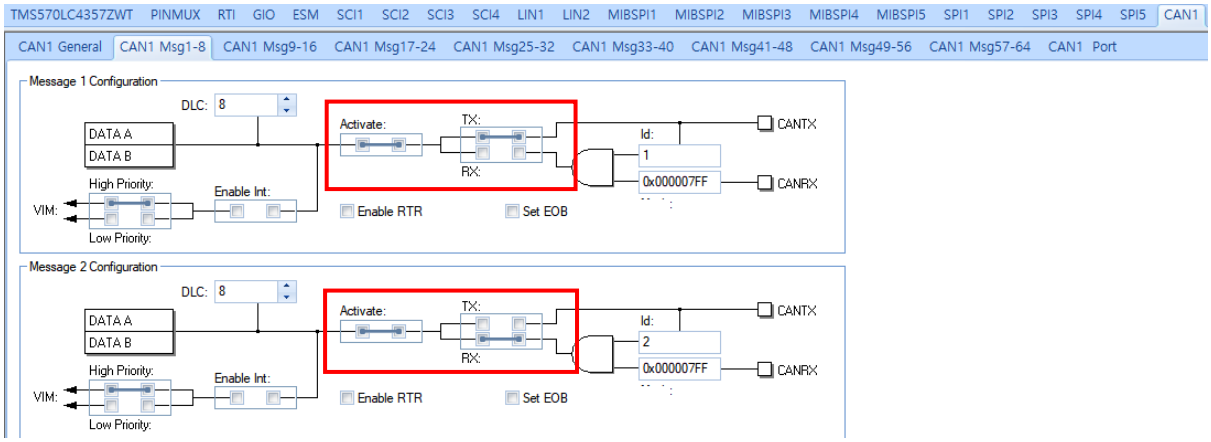
**Figure 35-66. Time-Base Period Register (TBPRD) [offset = 08h]**

| 15 | 0 |
|---|---|
| TBPRD | |
| R/W-0 | |

LEGEND: R/W = Read/Write; *-n* = value after reset

**Table 35-26. Time-Base Period Register (TBPRD) Field Descriptions**

| Bits | Name | Description |
|---|---|---|
| 15-0 | TBPRD | These bits determine the period of the time-base counter. This sets the PWM frequency.<br>Shadowing of this register is enabled and disabled by the TBCTL[PRDLD] bit. By default this register is shadowed.<br>• If TBCTL[PRDLD] = 0, then the shadow is enabled and any write or read will automatically go to the shadow register. In this case, the active register will be loaded from the shadow register when the time-base counter equals 0.<br>• If TBCTL[PRDLD] = 1, then the shadow is disabled and any write or read will go directly to the active register, that is the register actively controlling the hardware.<br>• The active and shadow registers share the same memory map address. |




CAN 통신을 위하여 Message1 은 TX 로, Message2 는 RX 로 각각 설정해준다

# ccs 코드

```
#include "HL_sys_common.h"
#include "HL_system.h"
#include "HL_can.h"
#include "HL_etpwm.h"
#include "HL_esm.h"
#include "HL_sys_core.h"
#include "stdio.h"
#define D_COUNT 8
#define D_SIZE 8
uint32 cnt = 0;
uint32 error = 0;
uint32 tx_done = 0;
uint8 tx_data[D_COUNT] = { 1, 2, 3, 4, 4, 3, 2, 1 }; //전송 데이터
uint8 rx_data [D_COUNT] = { 0 };
uint32_t checkPackets(uint8_t *src_packet, uint8_t *dst_packet, uint32_t psize);

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

int main(void)
{

    etpwmInit();
    canInit();
    //dcan_enable_int();
/* Pin MUX 레지스터에 접근(gpio핀에 etPWM을 연결) */
    etpwmStartTBCLK();

    printf("start\n");
    canEnableErrorNotification(canREG1);
    //canIoSetDirection(canREG1, canMESSAGE_BOX1, canMESSAGE_BOX2);

    while (1)
    {

        printf("transmit CAN message\n");
        delay(10000000);
         /*HCG에서 canMESSAGE_BOX1을 TX로 설정*/
        canTransmit(canREG1, canMESSAGE_BOX1, (const uint8 *)&tx_data[0]);

         /*HCG에서 canMESSAGE_BOX2을 RX로 설정*/
        if (canIsRxMessageArrived(canREG1, canMESSAGE_BOX2))
        {
            canGetData(canREG1, canMESSAGE_BOX2, (uint8 *) &rx_data[0]);

           /*받은 데이터의 첫번 째 배열에 있는 값 확인*/
            if(*rx_data == 0){
                printf("직진\n");
            }else if(*rx_data == 1){
                printf("우회전\n");
            }else if(*rx_data == 2){
```

```
            printf("좌회전\n");
        }else
            printf("error\n");

        printf("rx_data : %x\n", rx_data[1]);

        if(rx_data[0] == 1) //우회전
            etpwmREG1->CMPA = 1875 + (20 * rx_data[1]);
        else if(rx_data[0] == 2) //좌회전
            etpwmREG1->CMPA = 1875 - (20 * rx_data[1]);
        else if(rx_data[0] == 0) //직진
            etpwmREG1->CMPA = 1875;
        else
            printf("error");


    }
  }
}

/** - Sets time period or frequency for ETPWM block both PWMA and PWMB*/
    etpwmREG1->TBPRD = 24999U;

/** - Setup the duty cycle for PWMA */
    etpwmREG1->CMPA = 1250U;
```

TBPRD값의 5%인 1250이 CMPA의 값이 된다.

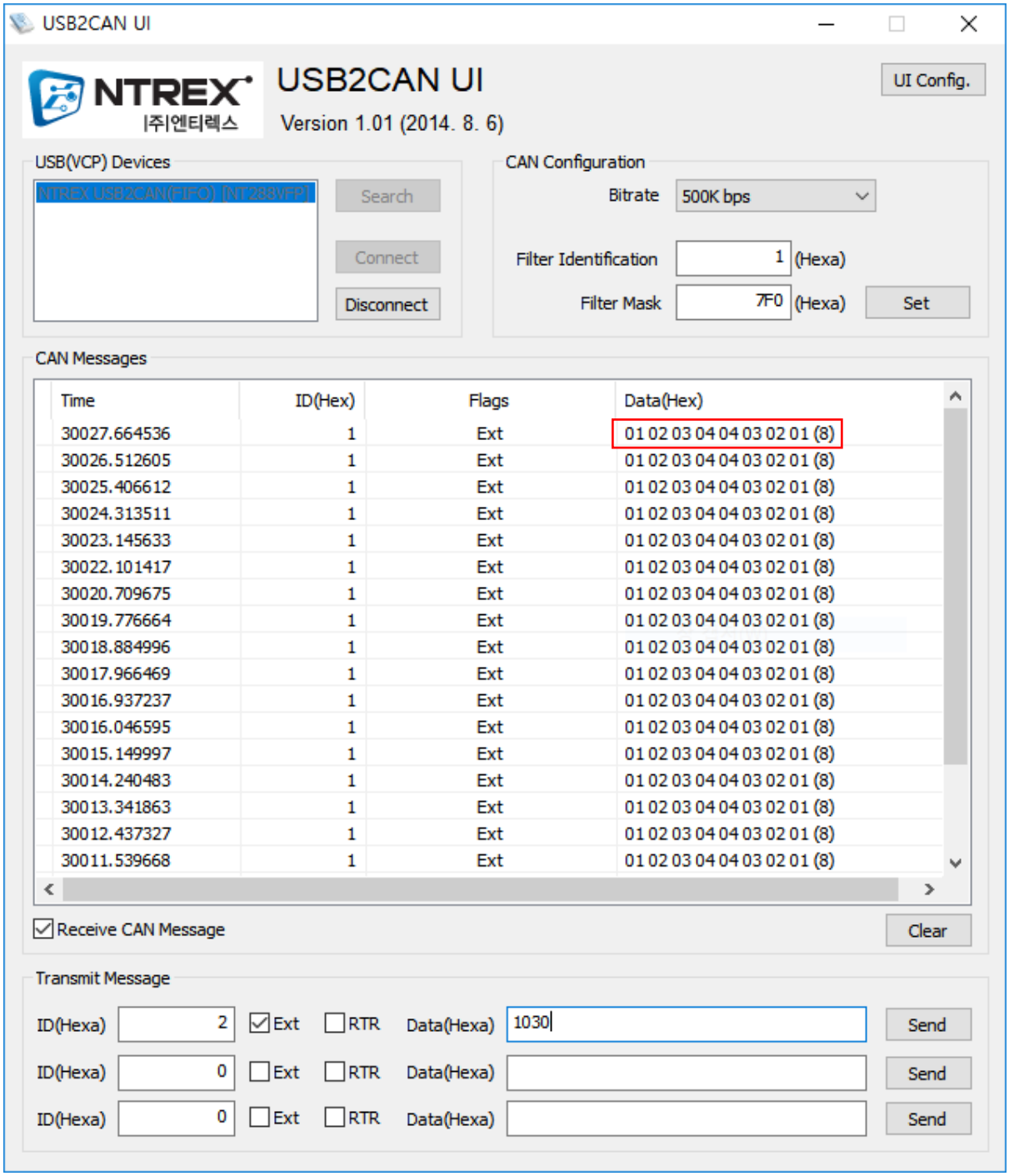
SERVO MOTOR SG90은 5% ~ 10%까지 컨트롤이 가능

따라서 25000의 5% ~ 10%인 1250 ~ 2500까지를 etpwmREG1->CMPA에 넣어서

모터의 각도를 제어했다.

| **CMPA** | 1250 | 1875 | 2500 |
|---|---|---|---|
| **%** | 5% | 7.5% | 10% |
| **각도** | -30 | 0 | 30 |

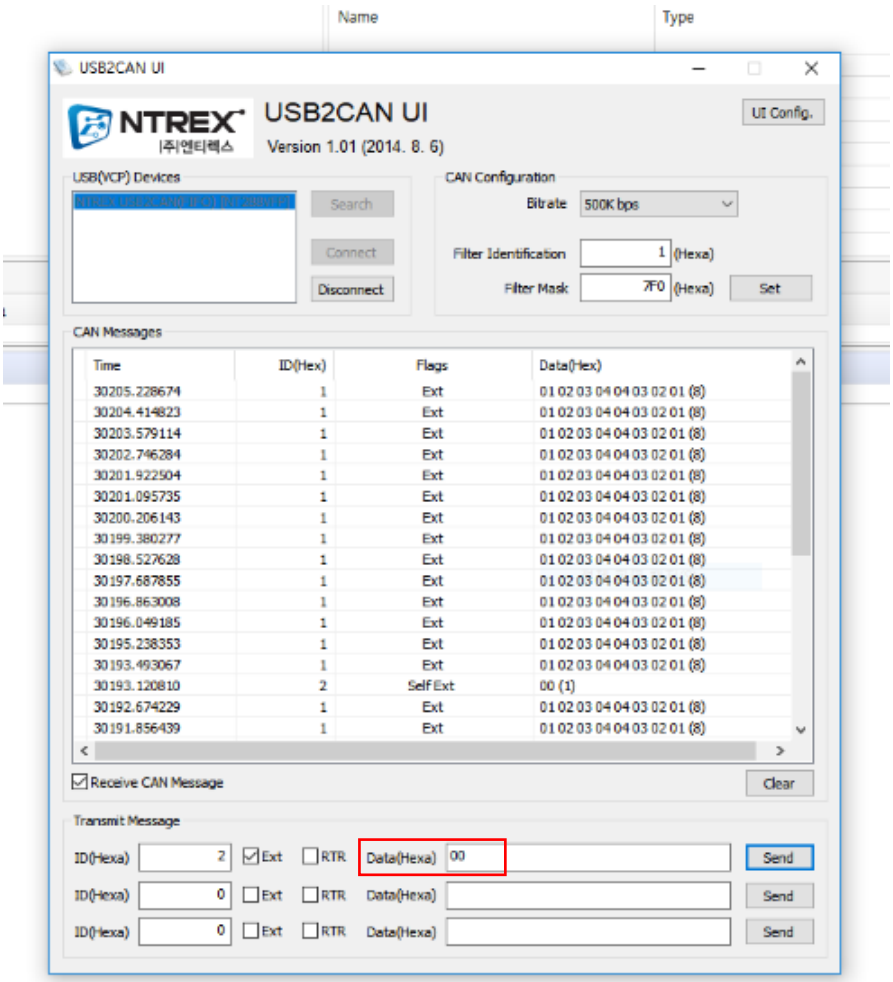
# 결과


USB2CAN UI
NTREX
㈜엔티렉스
USB2CAN UI
Version 1.01 (2014. 8. 6)
UI Config.
USB(VCP) Devices
NTREX USB2CAN(FIFO) [NT288VFP]
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)
30027.664536 1 Ext 01 02 03 04 04 03 02 01 (8)
30026.512605 1 Ext 01 02 03 04 04 03 02 01 (8)
30025.406612 1 Ext 01 02 03 04 04 03 02 01 (8)
30024.313511 1 Ext 01 02 03 04 04 03 02 01 (8)
30023.145633 1 Ext 01 02 03 04 04 03 02 01 (8)
30022.101417 1 Ext 01 02 03 04 04 03 02 01 (8)
30020.709675 1 Ext 01 02 03 04 04 03 02 01 (8)
30019.776664 1 Ext 01 02 03 04 04 03 02 01 (8)
30018.884996 1 Ext 01 02 03 04 04 03 02 01 (8)
30017.966469 1 Ext 01 02 03 04 04 03 02 01 (8)
30016.937237 1 Ext 01 02 03 04 04 03 02 01 (8)
30016.046595 1 Ext 01 02 03 04 04 03 02 01 (8)
30015.149997 1 Ext 01 02 03 04 04 03 02 01 (8)
30014.240483 1 Ext 01 02 03 04 04 03 02 01 (8)
30013.341863 1 Ext 01 02 03 04 04 03 02 01 (8)
30012.437327 1 Ext 01 02 03 04 04 03 02 01 (8)
30011.539668 1 Ext 01 02 03 04 04 03 02 01 (8)
Receive CAN Message
Clear
Transmit Message
ID(Hexa) 2 Ext RTR Data(Hexa) 1030 Send
ID(Hexa) 0 Ext RTR Data(Hexa) Send
ID(Hexa) 0 Ext RTR Data(Hexa) Send

Console

can:CIO

```
transmit CAN message
transmit CAN message
transmit CAN message
transmit CAN message
transmit CAN message
transmit CAN message
```



데이터 00전송

```
transmit CAN message
transmit CAN message
transmit CAN message
transmit CAN message
transmit CAN message
직진
rx_data : 45
transmit CAN message
transmit CAN message
transmit CAN message
transmit CAN message
transmit CAN message
transmit CAN message
transmit CAN message
transmit CAN message
transmit CAN message
transmit CAN message
transmit CAN message
transmit CAN message
transmit CAN message
```

데이터 0130 전송

```
transmit CAN message
transmit CAN message
transmit CAN message
transmit CAN message
transmit CAN message
우회전
rx_data : 30
transmit CAN message
transmit CAN message
transmit CAN message
transmit CAN message
transmit CAN message
transmit CAN message
transmit CAN message
transmit CAN message
transmit CAN message
transmit CAN message
transmit CAN message
transmit CAN message
transmit CAN message
transmit CAN message
transmit CAN message
transmit CAN message
```

# *문제점*

소수점 단위 처리불가