MCU에서 모터 Calibration 설정

BLDC

보통 ESC에서 인식하는 신호

입력전압 : 3~5V

주기 : 20ms

Pulse : 1~2ms

Min(optional) : the pulse width, in microseconds, corresponding andgle on the motor(544)

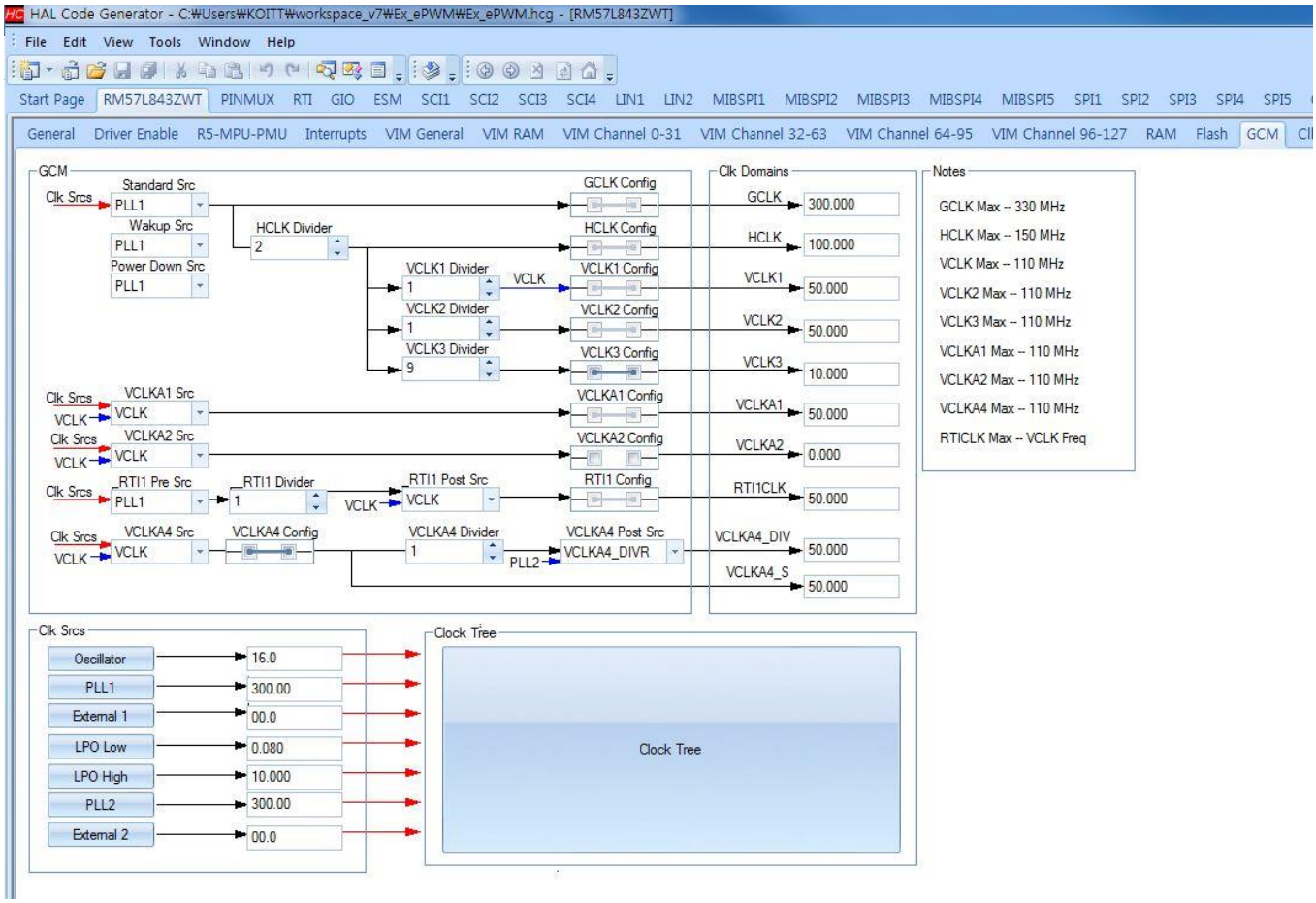
Max(optional) : the pulse width, , in microseconds, corresponding andgle on the motor(2400)

SERVO

the pulse widths involved range from 0.5ms/20ms (2.5%) and 2.5/20 (12.5%) of the total pulse period

*GCM 설정

- HCLK Divider : 1 -> 2
- VCLK3 Divider : 1 -> 9
- CLK Domains : 10.000

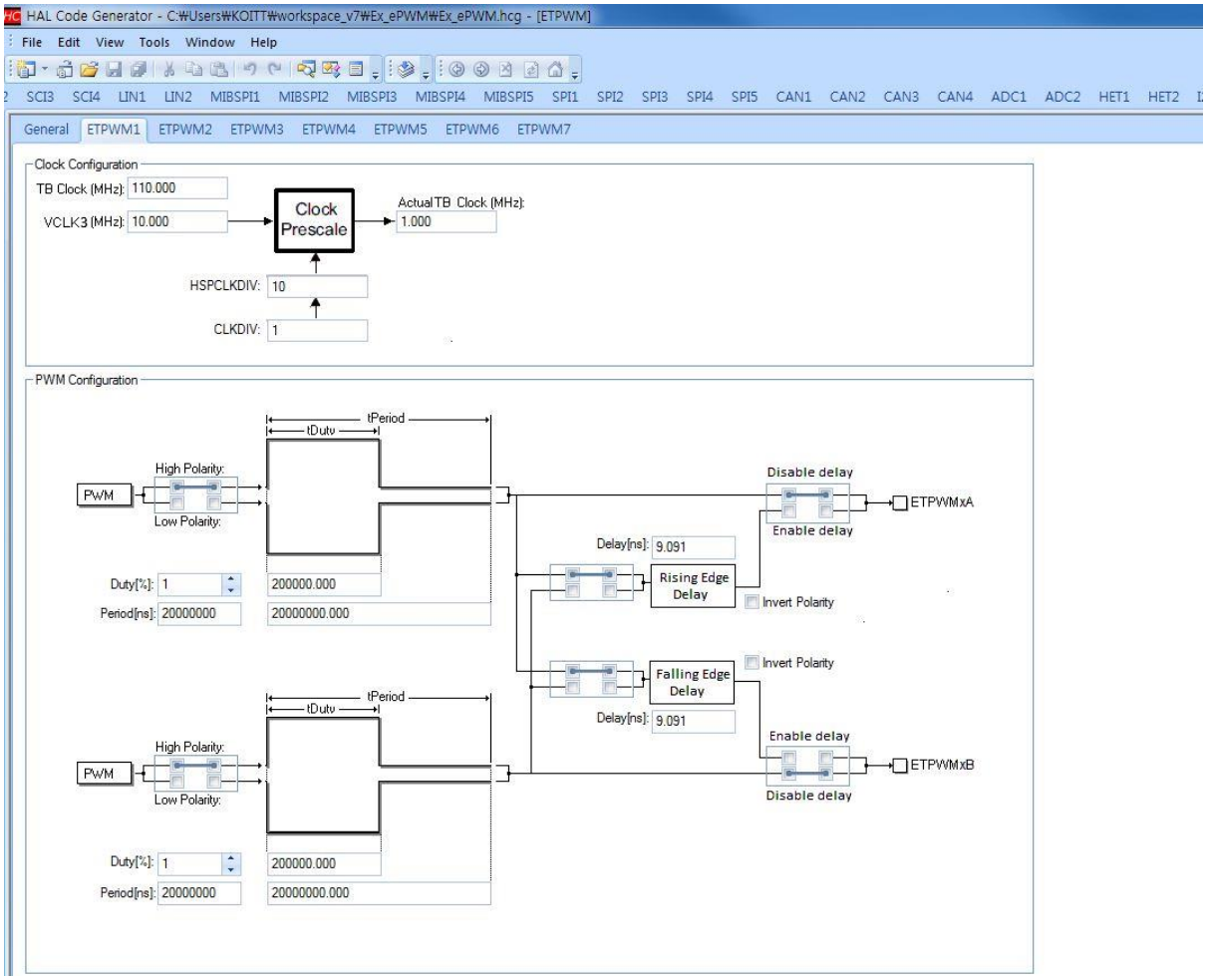


*ETPWM 설정

* HSPCLKDIV : 0 -> 10

* CLKDIV : 0 -> 1

10MHz 가 된 VCLK3를 1/10하여 ePWM 모듈의 동작 클럭을 1MHz로 분주



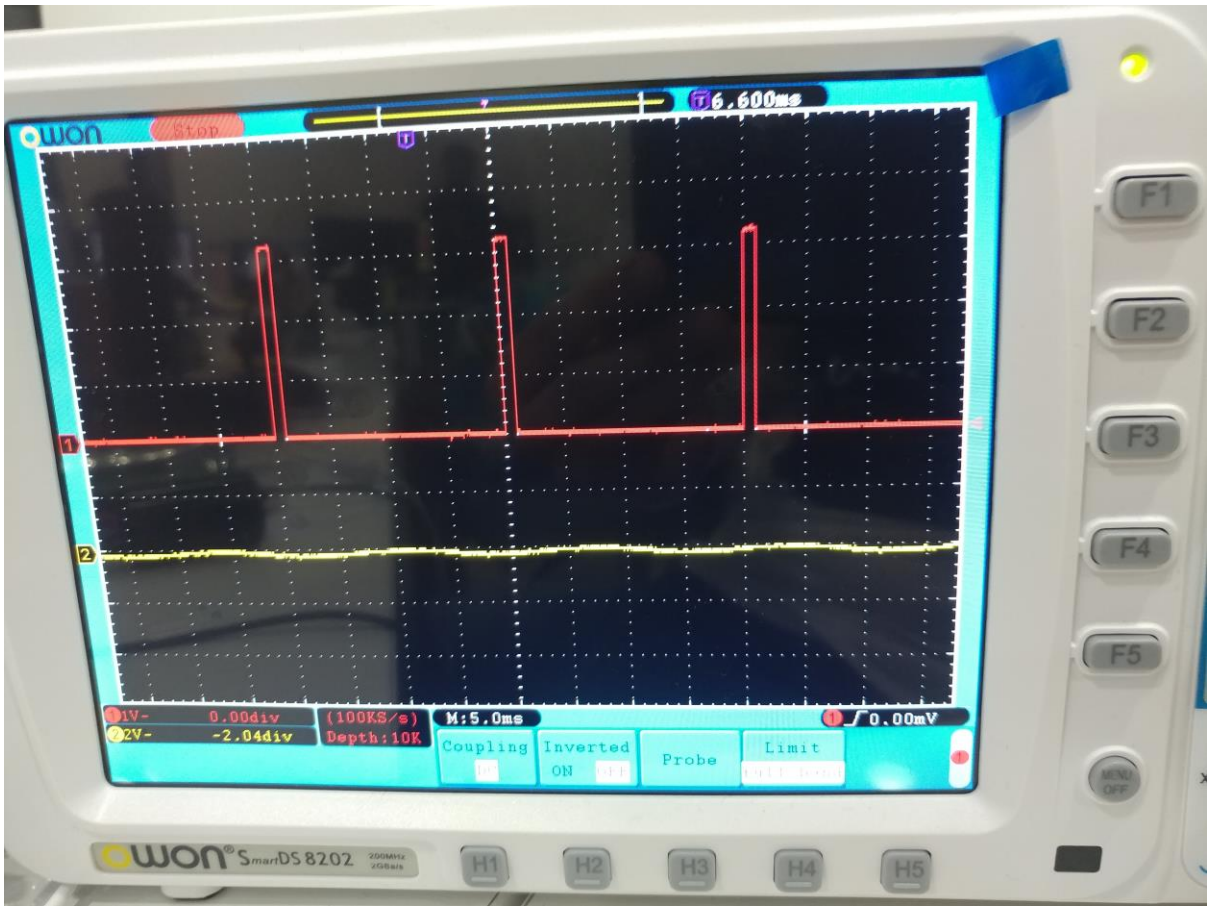
- 코드

UART로 1~9 숫자를 ASCII코드로 receiveData에 받고 이것을 실제 숫자로 변환한다

ePWM1B = receiveData*1000;

// ePWM1B에 1~9(receiveData) * 1000를 해서 입력한다.

etpwmSetCmpB(etpwmREG1, ePWM1B);



1을 입력하면 1ms (5% duty)가 나온다

2을 입력하면 2ms (10% duty) .....

```c
#include "HL_sys_common.h"
#include "HL_system.h"
#include "HL_etpwm.h"
#include "HL_sci.h"
#include <string.h>
#include <stdio.h>

#define TSIZE1  10
#define TSIZE2  5
#define TSIZE3  4

unsigned int ePWM1B = 90;

void sciDisplayText(sciBASE_t *sci, uint32 *text, uint32 length);
void wait(uint32 time);

#define UART sciREG1

uint32 receiveData = 0;

int main(void)
{
    sciInit();      /* SCI/SCI-Lin 초기화, 송수신 파라미터, Stop Bits : 2 */
    etpwmInit();

    while(1)
    {
        etpwmStartTBCLK();

        receiveData = sciReceiveByte(UART);
        //sciDisplayText(UART, &receiveData, TSIZE1);    /* Text 송신 */

        receiveData = receiveData-48;
        printf("receiveData = %d\n", receiveData);

        if(receiveData > 0){

            ePWM1B = receiveData*1000;
            if(ePWM1B > 0){
            etpwmSetCmpB(etpwmREG1, ePWM1B);
            printf("11 ePWM1B = %d\n", ePWM1B);
            }
            wait(4000);

        }else if(receiveData == 0){
            ePWM1B = receiveData;
            if(ePWM1B >= 0){
            etpwmSetCmpB(etpwmREG1, ePWM1B);
            printf("22 ePWM1B = %d\n", ePWM1B);
            }
            wait(4000);
        }else{
            printf("-------\n");
        }


        wait(4000);
    }

}
```