


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


강사 – Innova Lee(이상훈)

gcccompil3r@gmail.com

학생 – 정한별

hanbulkr@gmail.com

etPWM (led 제어, 모터 제어) 프로그래밍

2018. 7. 1. 18:17

https://blog.naver.com/hanbulkr/221310434686

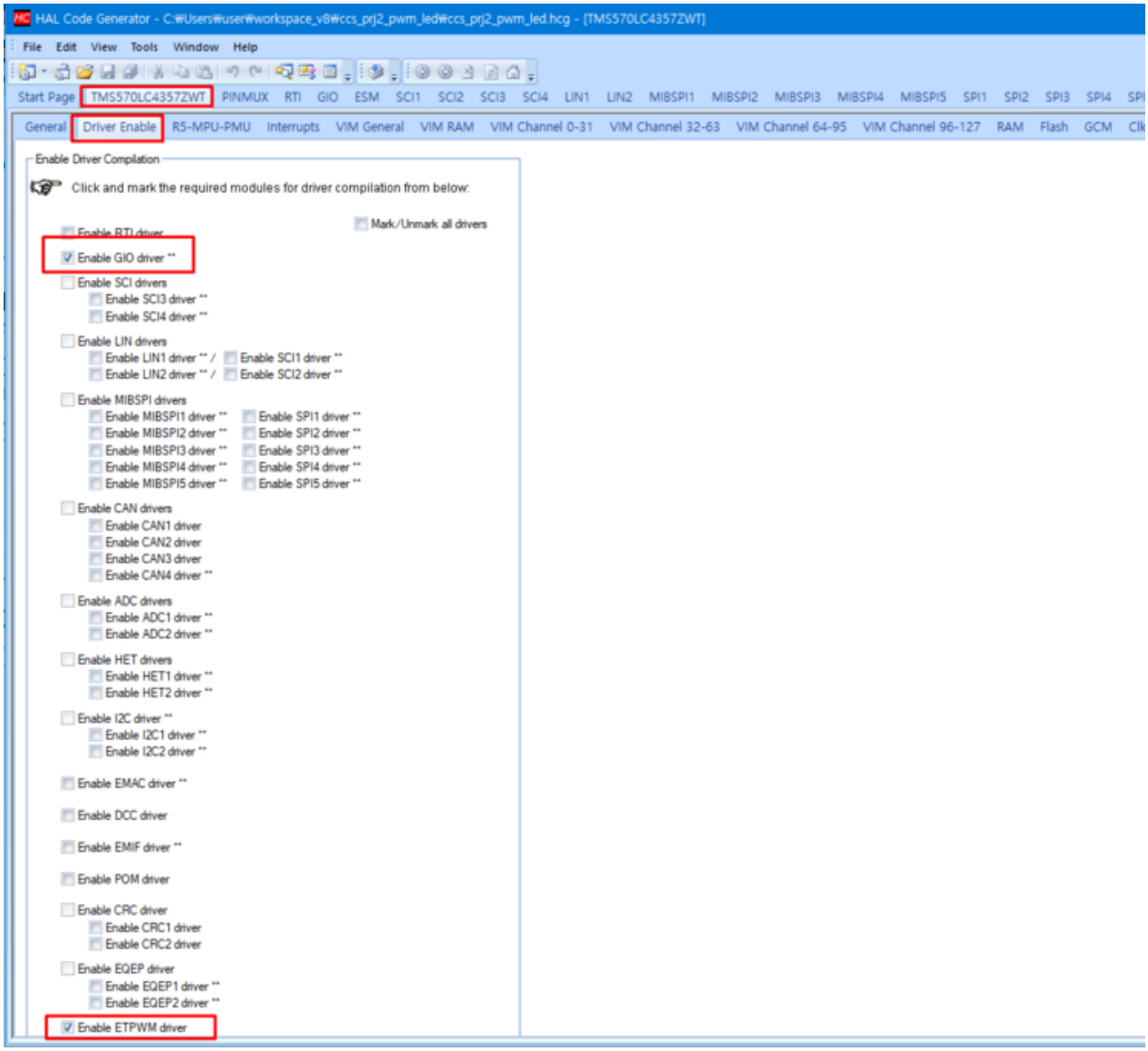
## 1. duty 비를 조정하여 led pwm 제어를 실시 한다.

### 1) HALCoGen 설정을 한다.

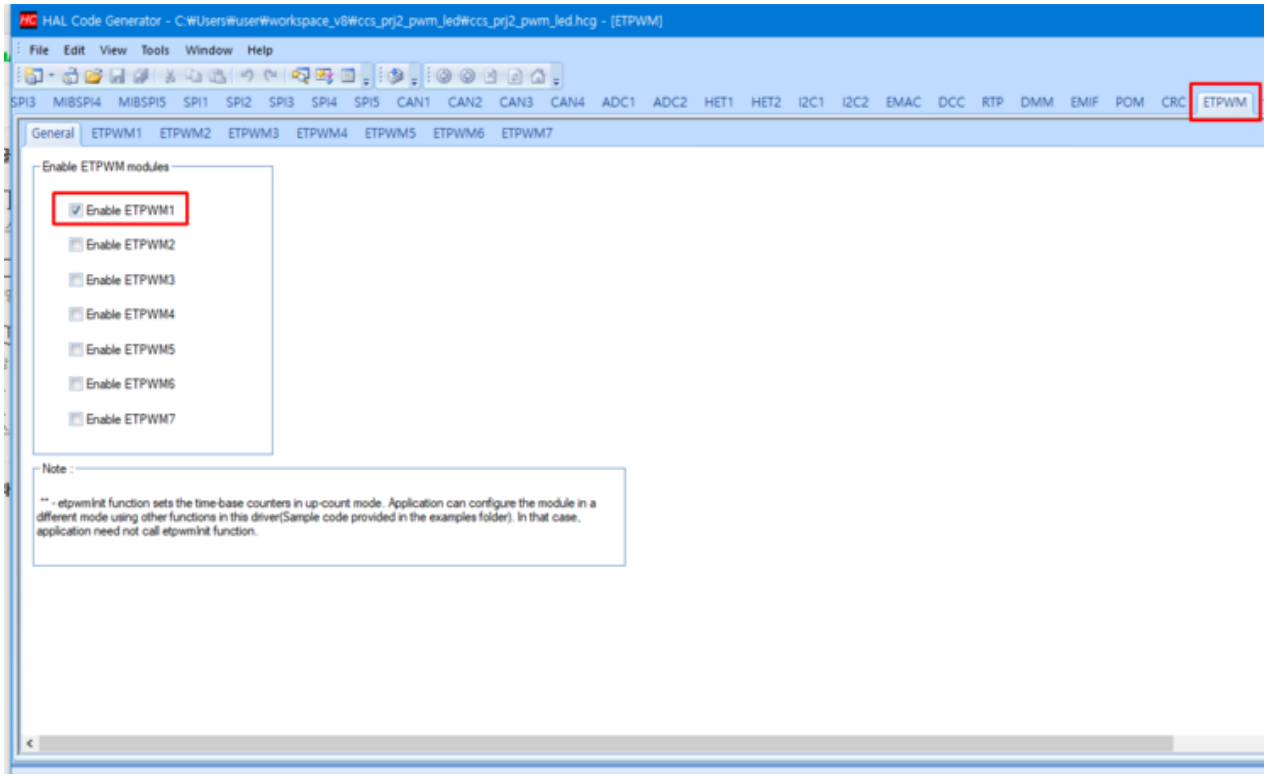
- 앞에서 포스팅 한 기본 설정과 같이 프로젝트를 생성한다.
https://blog.naver.com/hanbulkr/221309388402

'별''자리' : 네이버 블로그
그냥 개인 여행 기록 블로그에요. 보고 참고하셔…
blog.naver.com

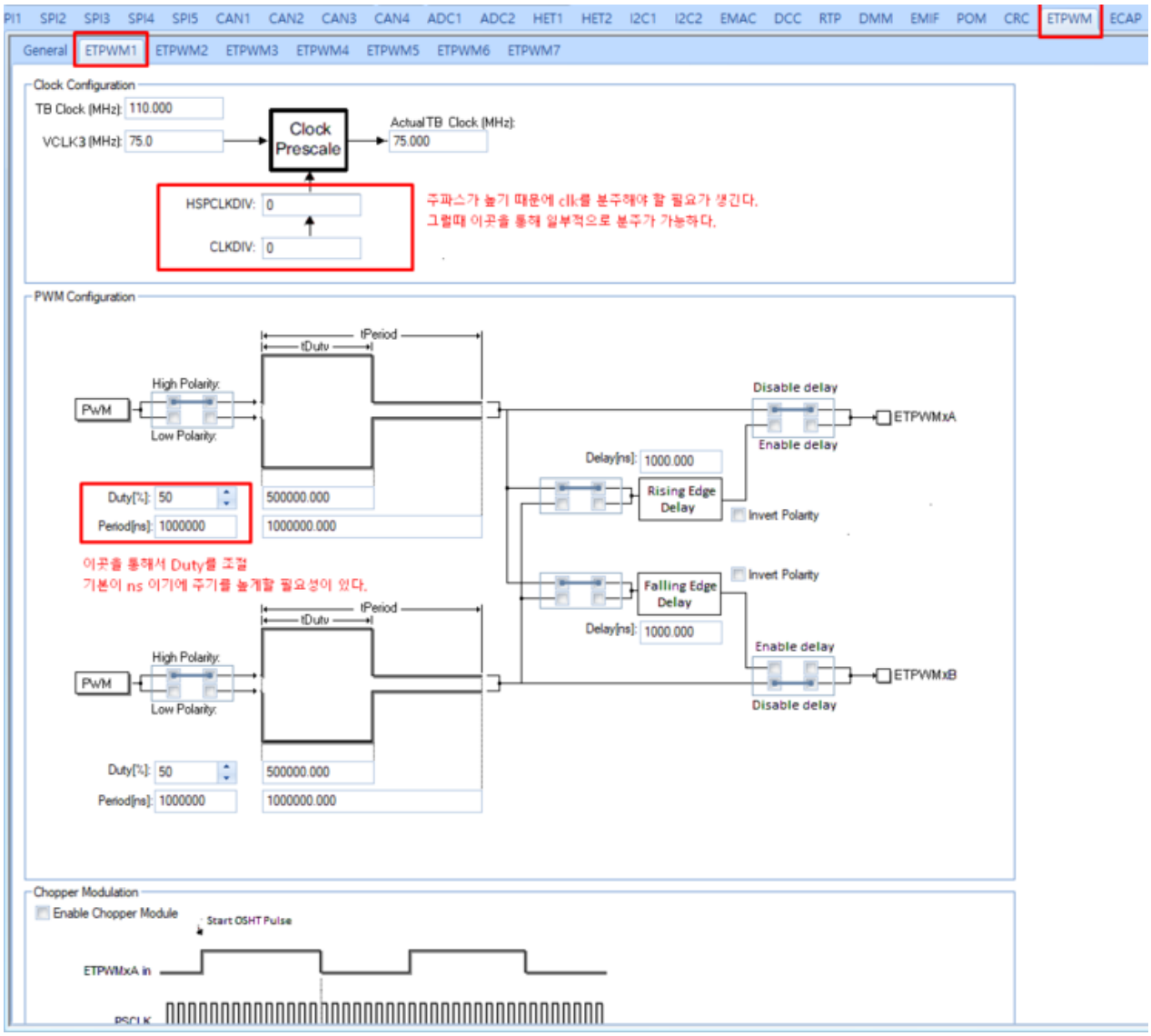
- 밑에와 같이 led를 키기 위해 GIO 와 ETPWM 기능을 사용 할 것이기 때문에 드라이버를 추가해 준다.




- ETPWM 탭에서 밑에 그림과 같이 설정해 준다. pwm1번을 사용하기로 한다.

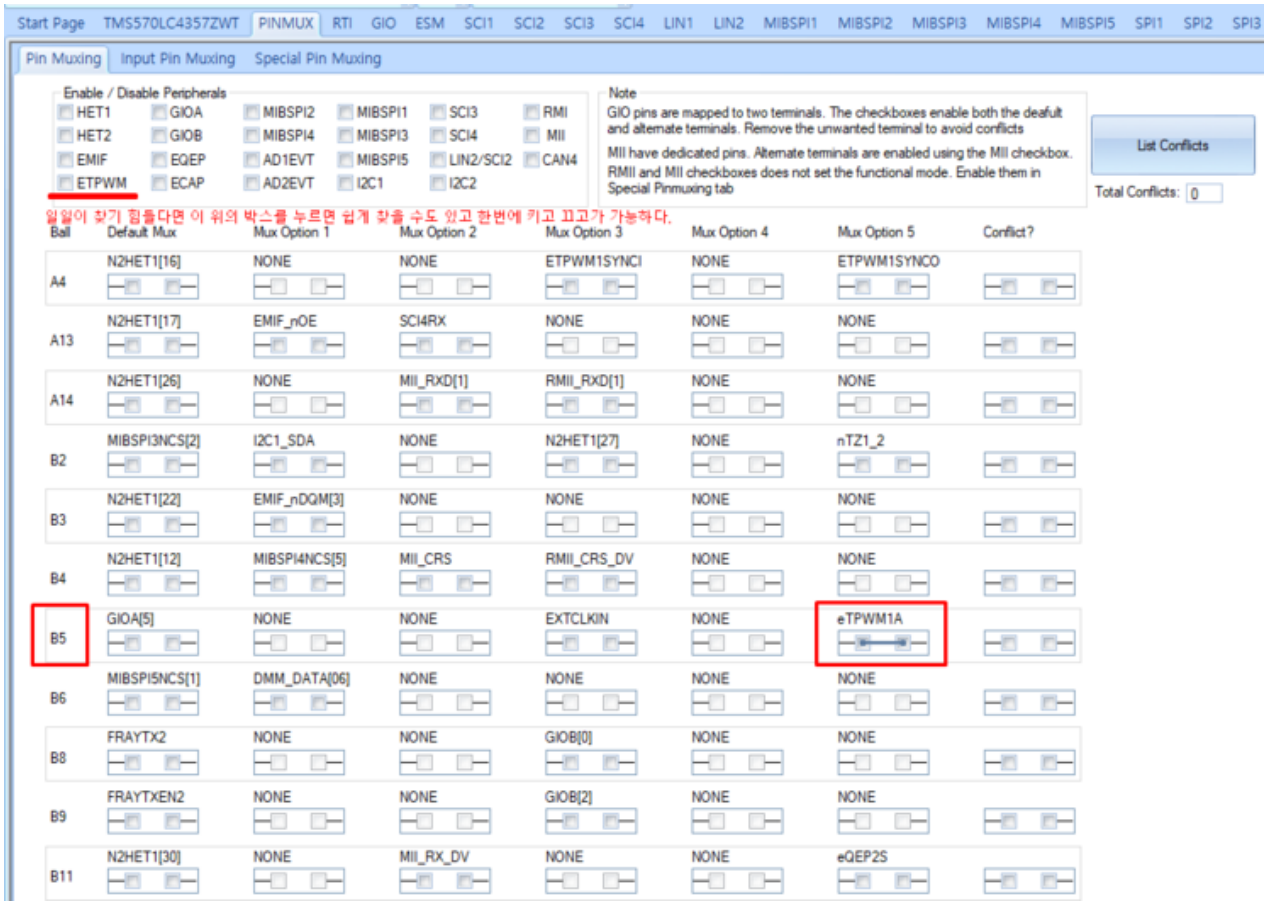


- ETPWM1의 설정을 하여 준다. (duty , period, 분주 등을 설정 해준다.)




- PINMUX 설정을 한다.

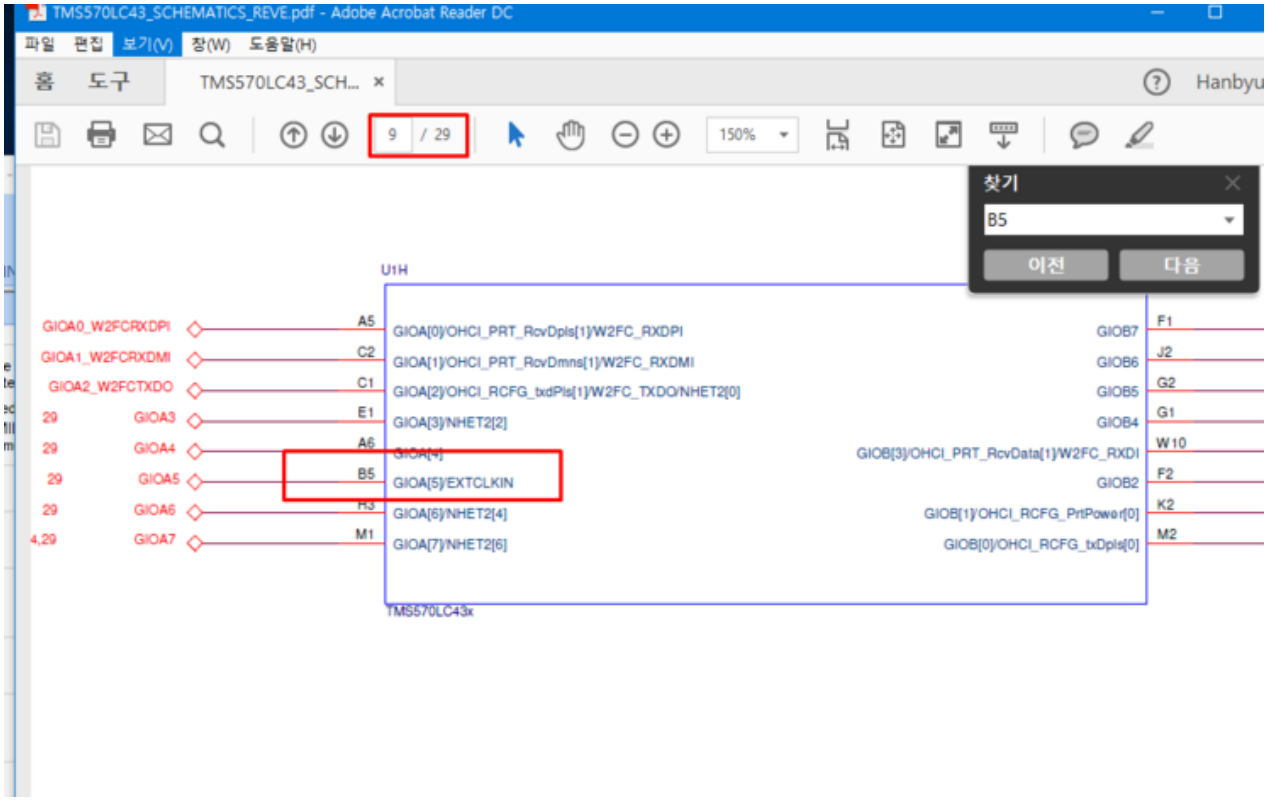
( PINMUX에 물려있는 정보를 보아하니 etPWM은 PINMUX에서 따로 핀을 설정해 그 PIN에서 제어하는 것이어서 GIO사용이 사실 필요가 없어졌다. ( driver enable 설정에서 꺼주어도 된다. ) )




- 위의 사진을 보면 B5의 eTPWM1A를 볼 수 있다. 아까 위에서 설정한 PWM의 1번핀을 B5에 물려 사용할 것이라는 뜻이다. B5위치가 어딘지 찾아 보려면 (기본설치)에 있는 Schematic에서 찾아본다.
https://blog.naver.com/hanbulkr

'별''자리' : 네이버 블로그
그냥 개인 여행 기록 블로그에요. 보고 참고하셔…
blog.naver.com

- 사진을 보면 B5에서 GIOA[5]로 쓰고 있는 핀이라고 알려 주고 있다.
한마디로 eTPWM1A = GIOA[5]핀 이라는 뜻이다.




## 2) CCS에서 프로그램 하기.

```
/** @file HL_sys_main.c
*   @brief Application main file
*   @date 07-July-2017
*   @version 04.07.00
*
*   This file contains an empty main function,
*   which can be used for the application.
*/

/*
* Copyright (C) 2009-2016 Texas Instruments Incorporated - www.ti.com
*
*
*  Redistribution and use in source and binary forms, with or without
*  modification, are permitted provided that the following conditions
*  are met:
*
*    Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
*
*    Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the
*    distribution.
*
*    Neither the name of Texas Instruments Incorporated nor the names of
*    its contributors may be used to endorse or promote products derived
*    from this software without specific prior written permission.
*
*  THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
```

```
*  "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
*  LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
*  A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
*  OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
*  SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
*  LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
*  DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
*  THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
*  (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
*  OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*
*/


/* USER CODE BEGIN (0) */
/* USER CODE END */

/* Include Files */

#include "HL_sys_common.h"
#include "HL_system.h"
#include "HL_etpwm.h"

uint32 value =0;
uint32 idx = 0;
uint32 duty_arr[25] = {1000,1200,1400,1600,1800,
                       2000,2200,2500,2800,3300,
                       3500,3800,4400,6000,8000,
                       10000,20000,25000,30000,40000,
                       45000,50000,60000,65000,70000};

void pwmSet(void);
void delay(uint32);

int main(void)
{
    //etpwmREG1->TBPRD = 74999U;
    //현재 period 의 정보. 74999 로 설정되어 있다.
    //아까 설정한  1000000nsec 는 VCLK3(75M) 과 곱해져 TBPRD 값이 나온다.
    etpwmInit();
    etpwmStartTBCLK();
    delay(10000);

    for(;;)

        pwmSet();
        delay(1000000);
    }

    return 0;
}
void pwmSet(void)
{
    value =duty_arr[idx % 25];
    etpwmSetCmpA(etpwmREG1, value);
```

```
    idx++;
}

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


/* USER CODE BEGIN (4) */
/* USER CODE END */
```

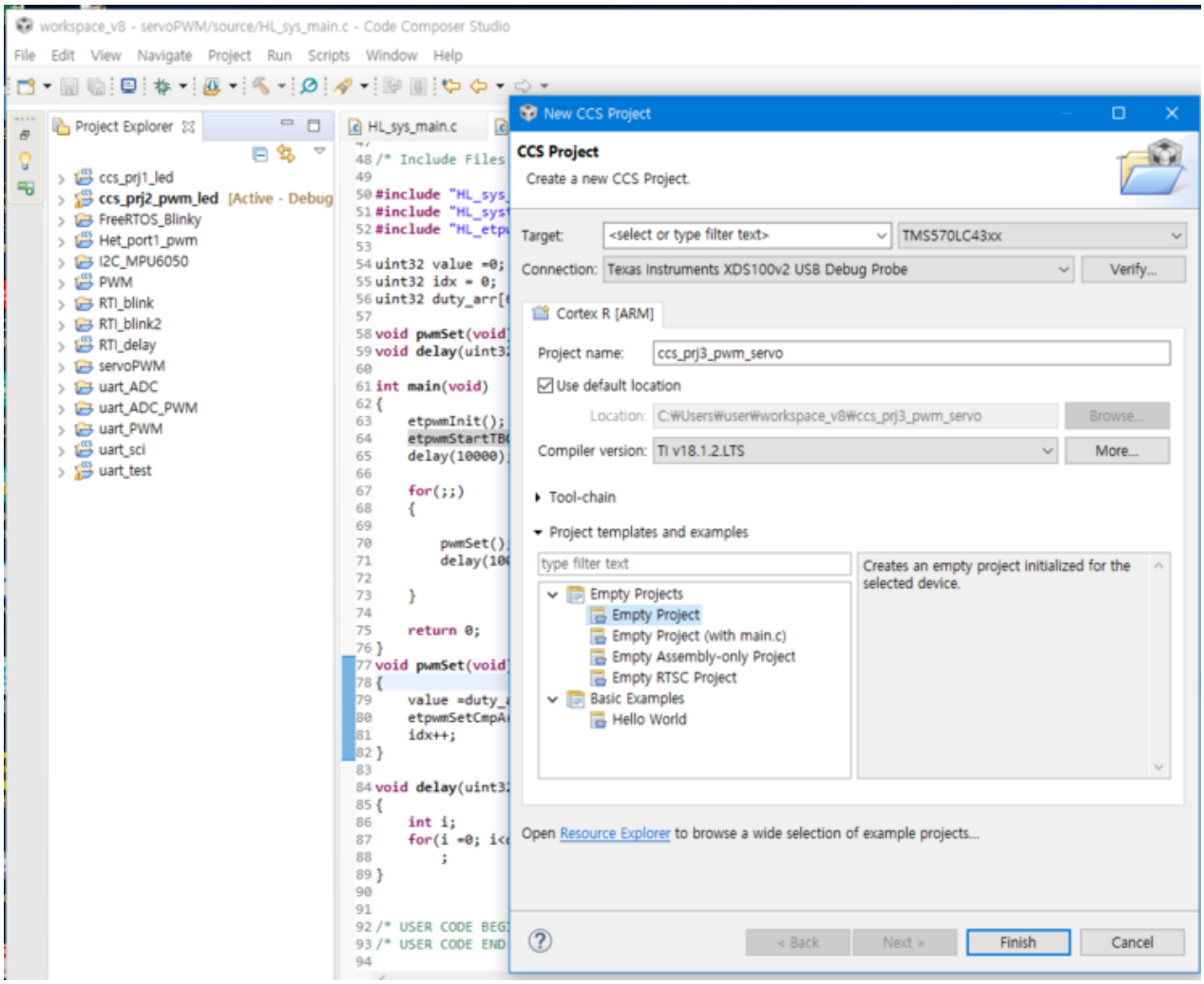
위와 같이 하여 실행 하면 된다.
( 사실 배열의 값을 다 정해줄 필요는 없다. 일정한 값을 지속적으로 더하고 최대 주기 74999를 넘기 전에 0으로 만들어 주면 된다.) 그럼 더욱 부드럽게 표현 할 수 있다. )

딜레이를 쓰는 이유는 아직 rti를 다루는 내용을 안 했기 때문이다.

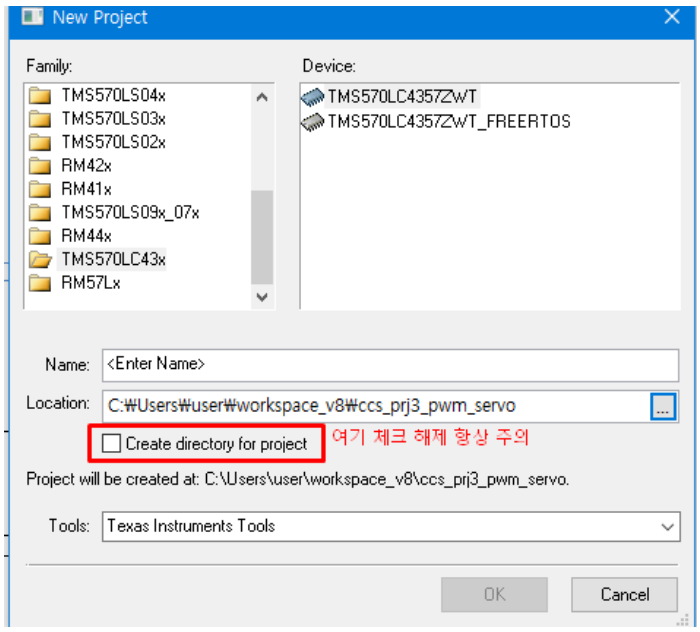
## 2. duty 비를 조정하여 servo pwm 제어를 실시 한다.

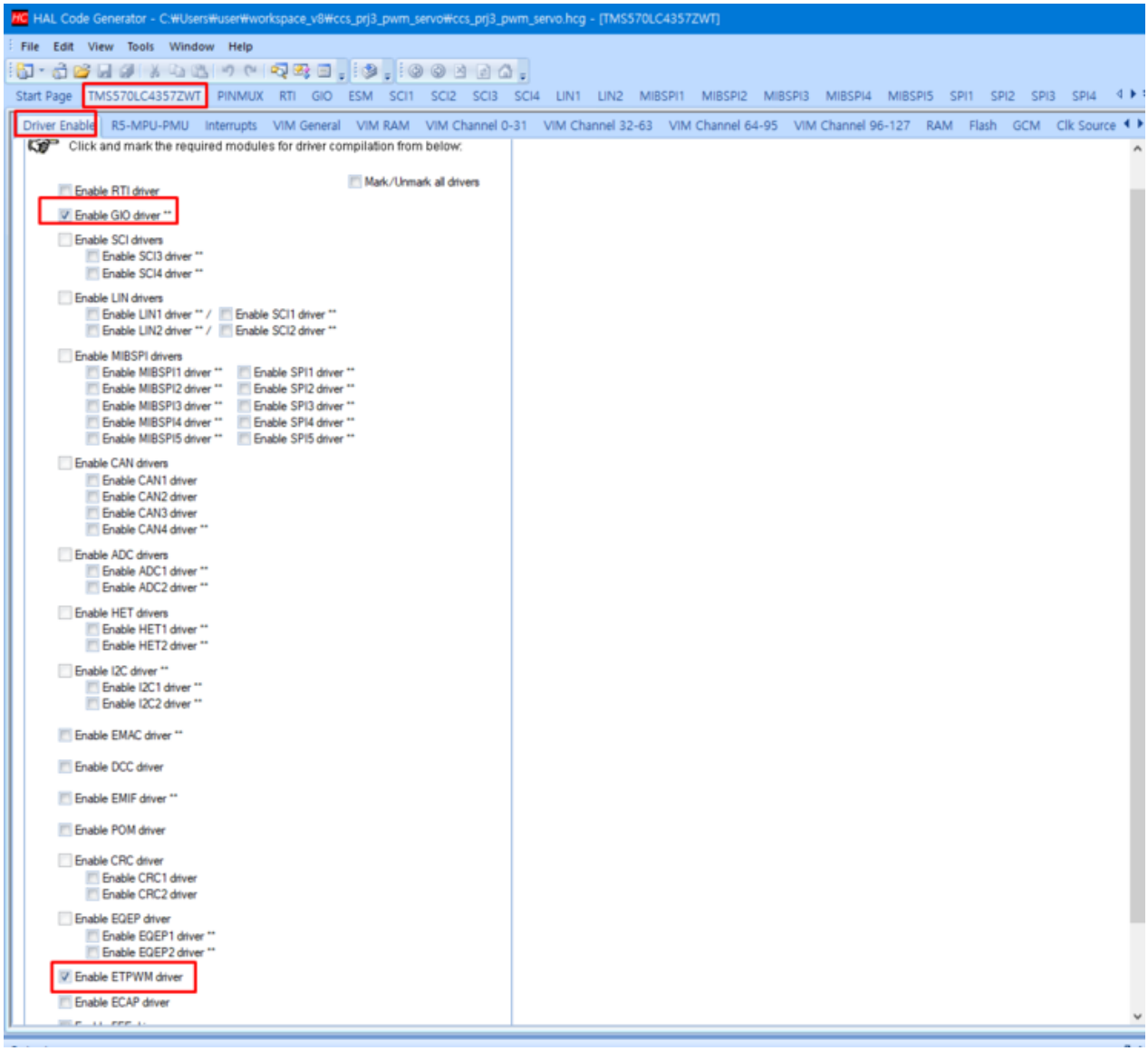
- 위와 같은 절차로 빠르게 만들어 보자

( 주의 ! = servo는 20ms의 파형이 들어가야 동작을 한다. 따라서 pwm의 주기를 20ms로 맞춘다.)

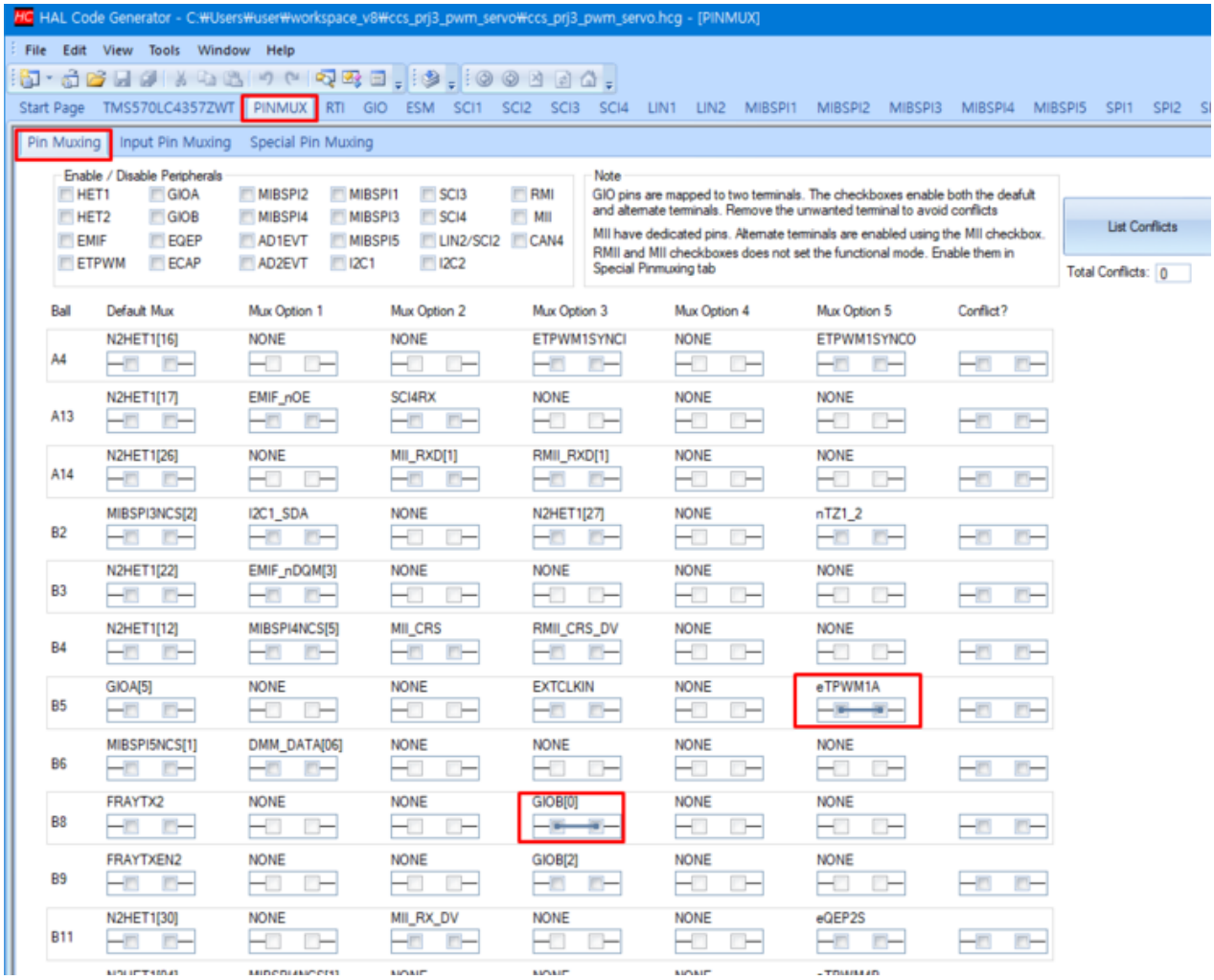


1. 아까는 사진을 안올렸지만 저번에 설명 했듯이 미리 ccs 프로젝트를 만든다.
2. HALCoGen 에서 프로젝트 경로를 똑같게 해준다.

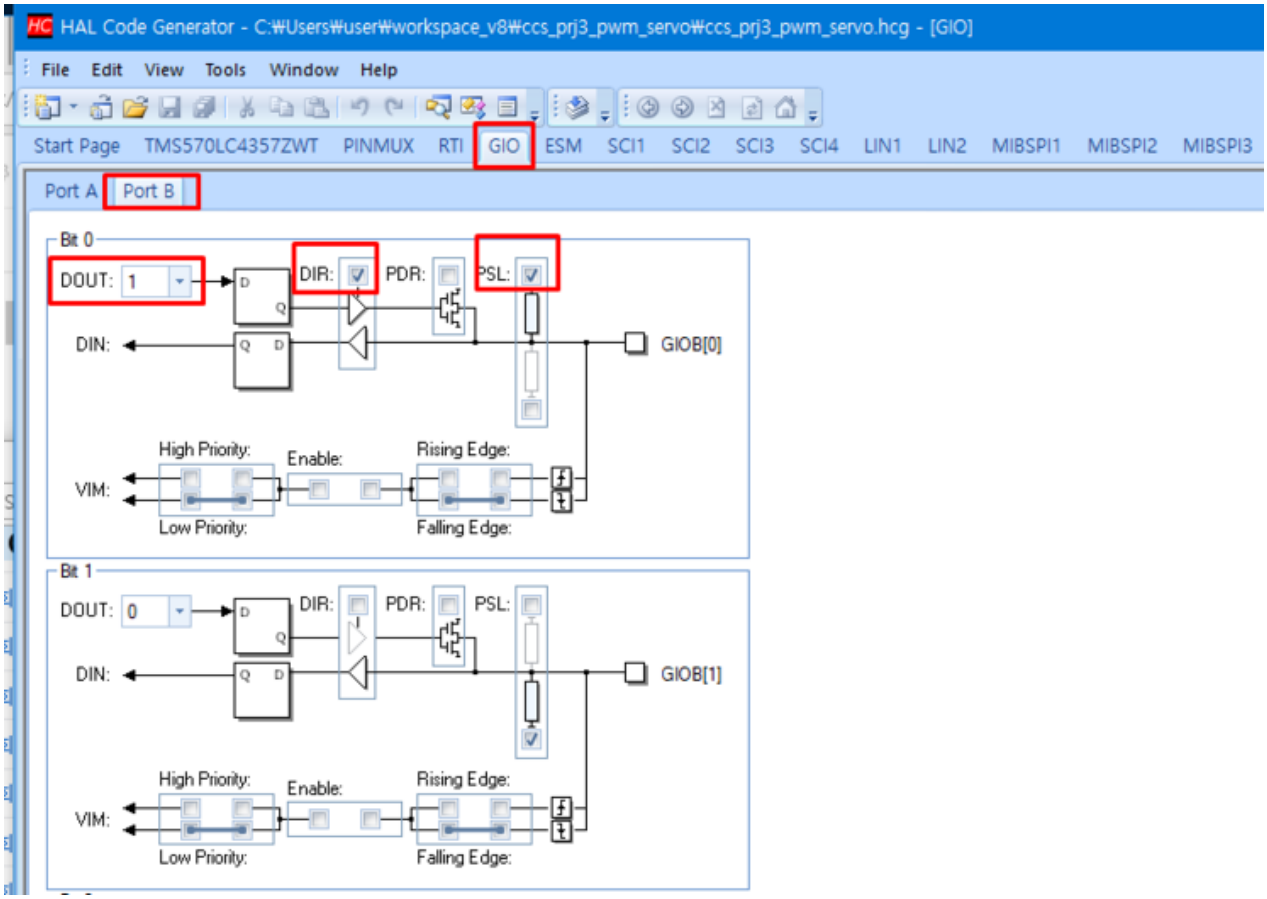


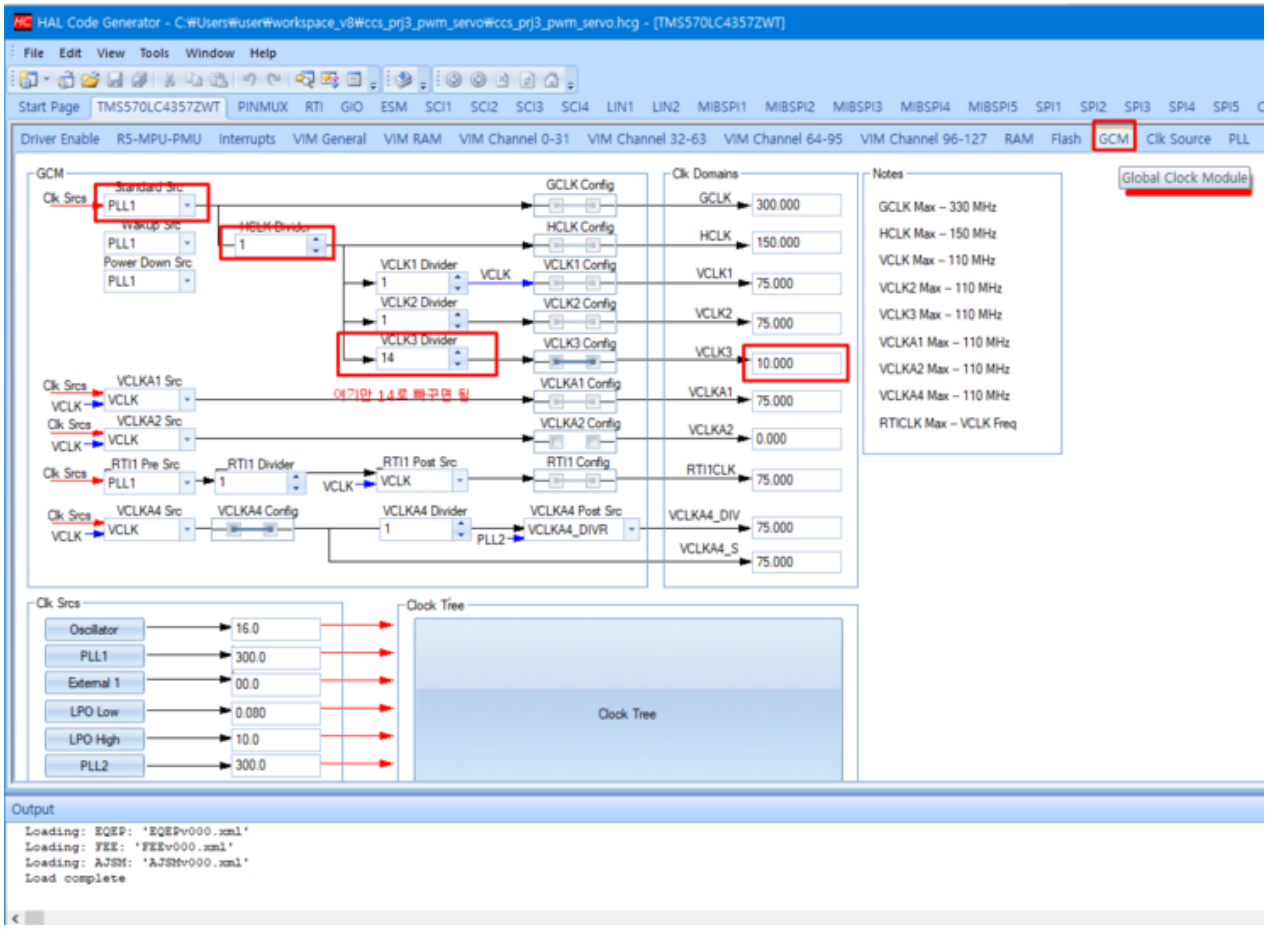

- servo 의 전원을 포트에서 주기로 한다.




- 위어서 mux 설정한 핀 B[0]의 상태를 설정해 준다.



- 여기서 위와 다르게 GCM 을 조절한다. 전체적으로 모듈의 clock을 조절할 수 있다.
  VCLK3 (PWM을 관리하는 클럭)을 14로 바꾸어 준다.

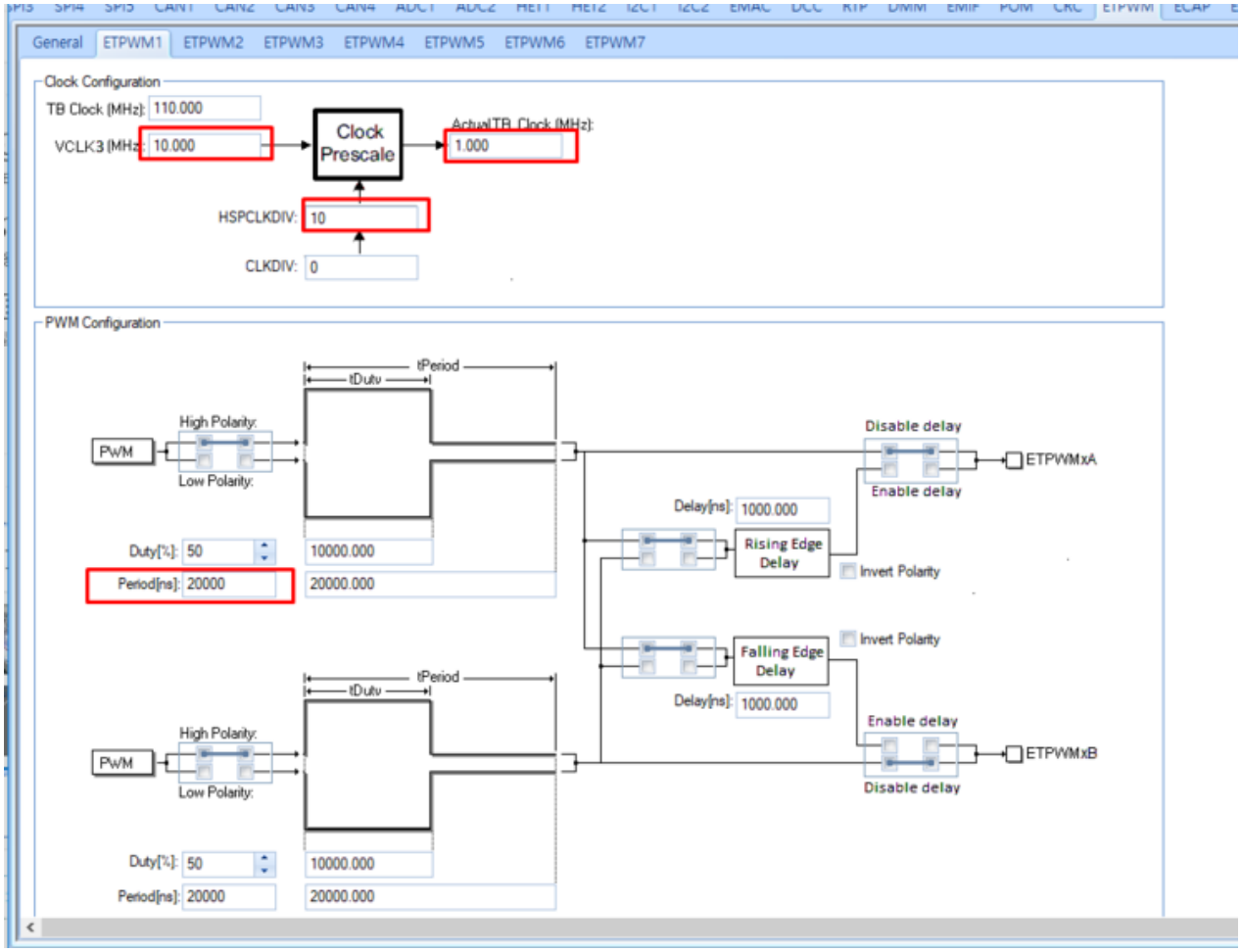



- GCM에 의해서  밑에 그림의 VCLK3이 바뀌는 것을 볼 수 있다.
- 또한 20m/s를 맞추기 위해 10으로 한번더 나누어 준다.
- *T = 1/f 이다* , 그럼

만들려는 20m/s (주기) =  내장된 clk(1M) 동안 몇번 카운드 해야 하는가.→ x/ 1M (주파수)

20m/s (T) = x/ 1M (1/f) ,  20m * 1M =  x  , x= 20000

여기서 x는 1M를 자를 시간카운트의 단위 이기 때문에 주기라고 볼 수 있다.




- 위에서 한 프로그램 소스와 같다. 모터가 각도에 따라 움직이는 것을 볼 수 있다.
- etpwmInit() 을 F3으로 드라이빙 하면 TBPRB 를 찾을 수 있다.

```
#include "HL_sys_common.h"
#include "HL_system.h"
#include "HL_etpwm.h"

uint32 value =0;
uint32 idx = 0;
uint32 duty_arr[6] = {1000,1200,1400,1600,1800,2000};

void pwmSet(void);
void delay(uint32);

int main(void)
{
    etpwmInit();
    etpwmStartTBCLK();
    delay(10000);

    for(;;)
    {
        pwmSet();
        delay(10000000);
    }

    return 0;
}
```

```
void pwmSet(void)
{
    value =duty_arr[idx % 6];
    etpwmSetCmpA(etpwmREG1, value);
    idx++;
}

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

- 위와 같이 코딩만 하면 끝이 아니다.
- TBPRD의 값은 다른 값으로 되어 있을 확률이 있으니 확인을 해주자.

```
70 /* Requirements : HL_CONQ_EPWM_SR2 */
71 void etpwmInit(void)
72 {
73 /* USER CODE BEGIN (1) */
74 /* USER CODE END */
75
76     /** @b initialize @b ETPWM1 */
77
78     /** - Sets high speed time-base clock prescale bits */
79     etpwmREG1->TBCTL = (uint16)5U << 7U;
80
81     /** - Sets time-base clock prescale bits */
82     etpwmREG1->TBCTL |= (uint16)((uint16)0U << 10U);
83
84     /** - Sets time period or frequency for ETPWM block both PWMA and PWMB*/
85     etpwmREG1->TBPRD = 19999U;
86
87     /** - Setup the duty cycle for PWMA */
88     etpwmREG1->CMPA = 10U;
89
90     /** - Setup the duty cycle for PWMB */
91     etpwmREG1->CMPB = 10U;
92
93     /** - Force EPWMxA output high when counter reaches zero and low when counter reaches Compare A value */
94     etpwmREG1->AQCTLA = ((uint16)((uint16)ActionQual_Set   << 0U)
95                        | (uint16)((uint16)ActionQual_Clear << 4U));
96
97     /** - Force EPWMxB output high when counter reaches zero and low when counter reaches Compare B value */
98     etpwmREG1->AQCTLB = ((uint16)((uint16)ActionQual_Set   << 0U)
99                        | (uint16)((uint16)ActionQual_Clear << 8U));
100
101    /** - Mode setting for Dead Band Module
102    *     -Select the input mode for Dead Band Module
103    *     -Select the output mode for Dead Band Module
104    *     -Select Polarity of the output PWMs
105    */
106    etpwmREG1->DBCTL = ((uint16)((uint16)0U << 5U)    /* Source for Falling edge delay(0-PWMA, 1-PWMB) */
```

- 프로그램을 구어보자.