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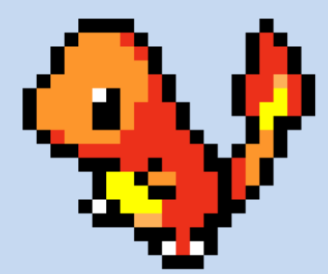# 프로그래밍 및 회로 설계 전문가

강사 이상훈
gcccompil3r@gmail.com

학생 김민호
minking12@naver.com

HC HAL Code Generator - - [TMS570LC4357ZWT]

File   Edit   View   Tools   Window   Help

TMS570LC4357ZWT   PINMUX   RTI   GIO   ESM   SCI1   SCI2

General   Driver Enable   R5-MPU-PMU   Interrupts   VIM Gene

Enable Driver Compilation

Click and mark the required modules for driver compilation from below:

- [ ] Enable RTI driver
- [ ] Enable GIO driver **
- [x] Enable SCI drivers
  - [ ] Enable SCI3 driver **
  - [ ] Enable SCI4 driver **
- [ ] Enable LIN drivers
  - [ ] Enable LIN1 driver ** /   [x] Enable SCI1 driver **
  - [ ] Enable LIN2 driver ** /   [ ] Enable SCI2 driver **
- [ ] Enable EQEP2 driver
- [x] Enable ETPWM driver
- [x] Enable ECAP driver
- [ ] Enable FEE driver

[ ] Mark/Unmark all drivers

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

Pin Muxing   Input Pin Muxing   Special Pin Muxing

N1:   N2HET1[15]   MIBSPI1NCS[4]   NONE   N2HET2[22]   NONE   ECAP1

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

Pin Muxing   Input Pin Muxing   Special Pin Muxing

B5:   GIOA[5]   NONE   NONE   EXTCLKIN   NONE   eTPWM1A

570LC4357ZWT   PINMUX   RTI   GIO   ESM   SCI1   SCI2   SCI3   SCI4   LIN

ut Pin Muxing   Special Pin Muxing

_5, for disabling selected HET1 PWM outputs
_2, for disabling selected HET2 PWM outputs
ate ADC ' Trigger Option-A'
ate ADC ' Trigger Option-B'

[ ] Enable EMIF_CLK output
[ ] EMIF Output enable

Ethernet MII/RMII select   [MII ▼]

PERR12 ▼
PERR12 ▼
PERR12 ▼

[ ] Use HET1_LOOP_SYNC for time-base sync

[x] Enable TBCLK sync**
nTZ1   [ASYNC ▼]
nTZ2   [ASYNC ▼]
nTZ3   [ASYNC ▼]
EPWM1SYNCI   [ASYNC ▼]

**Done in etpwmInit

eTPWM for AD
[x] Selecting
[x] Selecting
[x] Selecting
[x] Selecting
[x] Selecting
[x] Selecting

TMS570LC4357ZWT   PINMUX   RTI   GIO   ESM   SCI1   SCI2   SCI3   SCI4   LIN1   LIN2   MIBSF
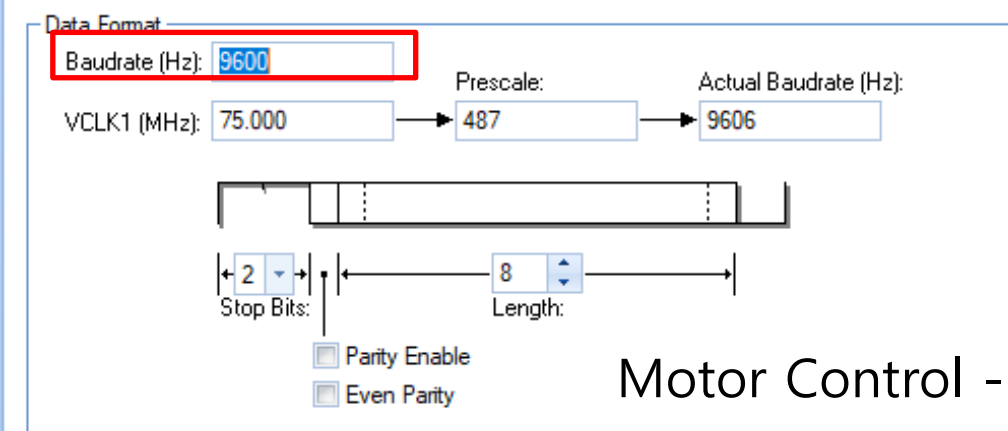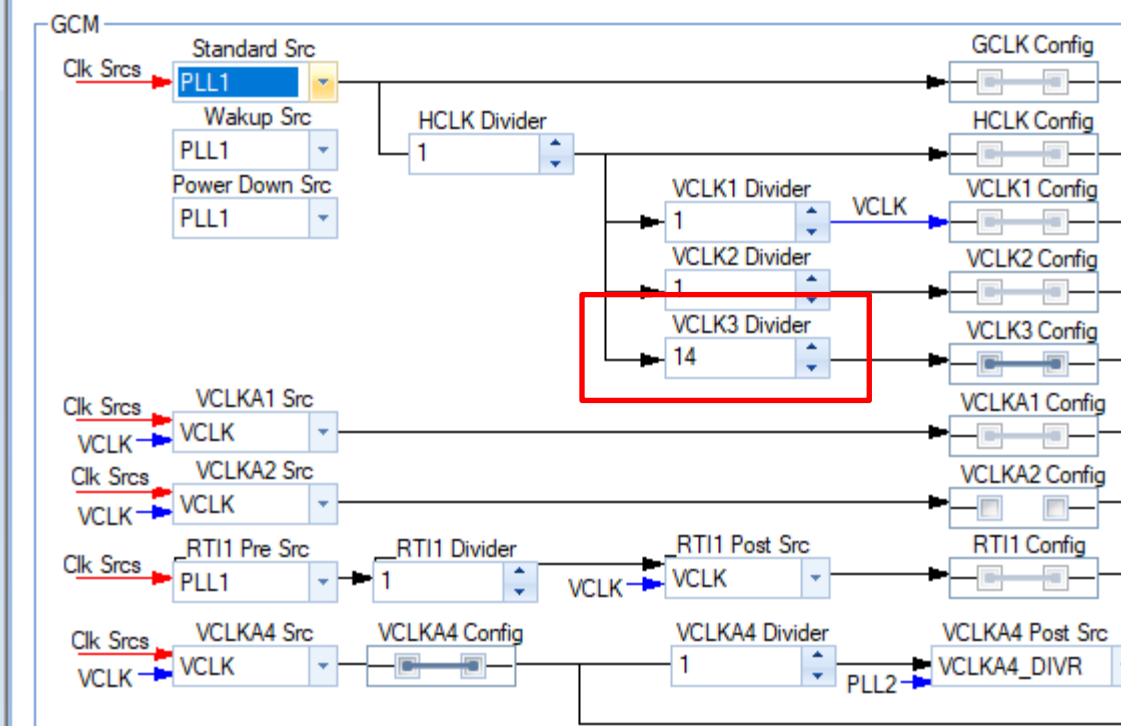
VIM RAM   VIM Channel 0-31   VIM Channel 32-63   VIM Channel 64-95   VIM Channel 96-127   RAM

103 : eTPWM7 Trip Zone   103   → IRQ   FIQ
104 : eCAP1 interrupt   104   → IRQ   FIQ
105 : eCAP2 interrupt   105   → IRQ
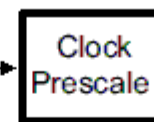
Motor Control - eCAP

Motor Control - eCAP

Motor Control - eCAP

```c
#include <HL_ecap.h>
#include <HL_etpwm.h>
#include <HL_hal_stdtypes.h>
#include <HL_reg_ecap.h>
#include <HL_reg_sci.h>
#include <HL_sci.h>
#include <HL_sys_core.h>
#include <HL_system.h>
#include <string.h>
#include "HL_system.h"
#include "stdio.h"
#include "math.h"

void send_data(sciBASE_t* sci, uint8* msg, int length);
int main(void)
{
    _enable_interrupt_();
    sciInit();
    etpwmInit();
    ecapInit();

    etpwmStartTBCLK();
    while (1)
    {
    }
    return 0;
}


void send_data(sciBASE_t* sci, uint8* msg, int length)
{
    int i;
    for (i = 0; i < length; i++)
        sciSendByte(sci, msg[i]);
}
```
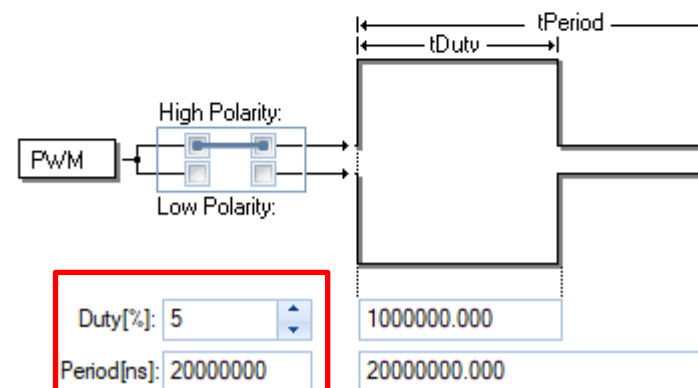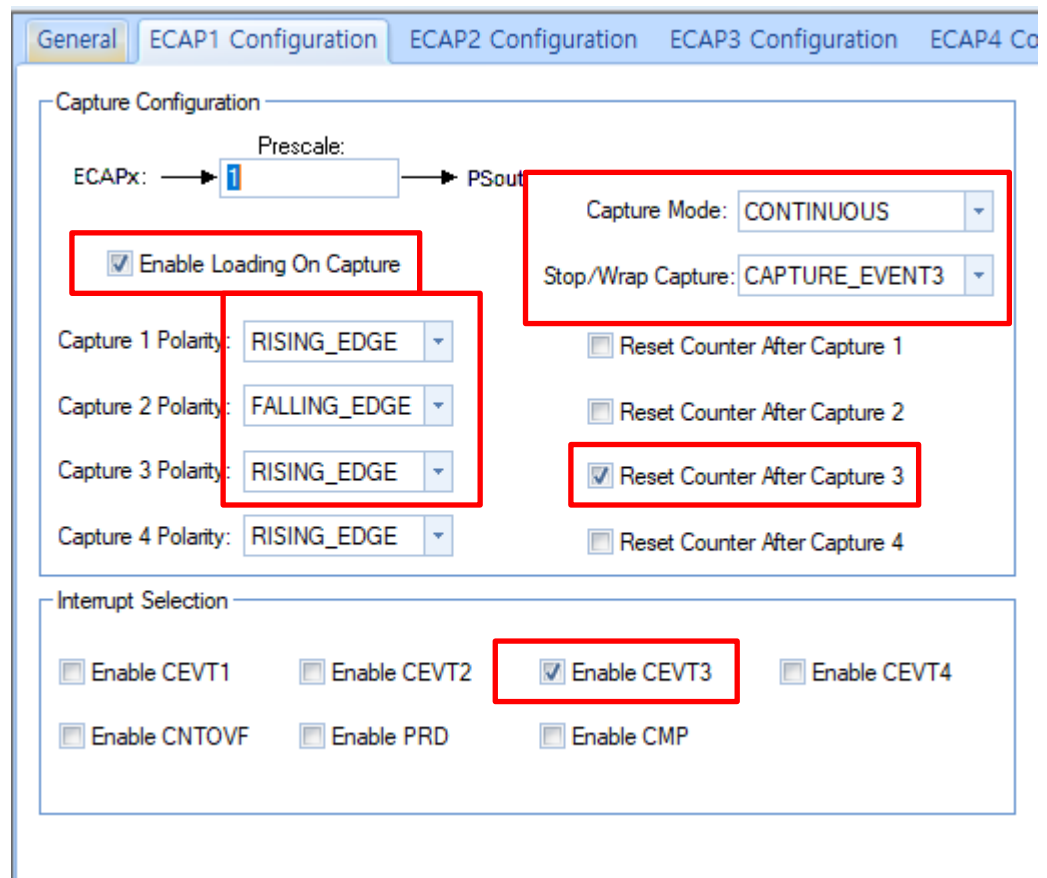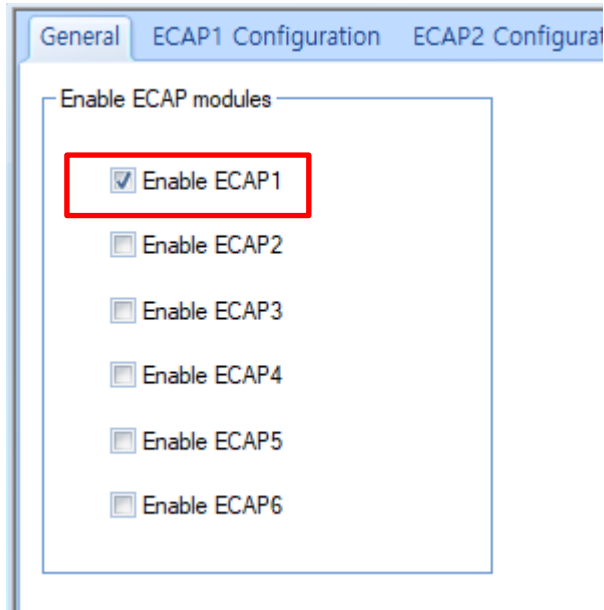
```c
void ecapNotification(ecapBASE_t *ecap, uint16 flags)
{
    uint32 ecap1, ecap2, ecap3;
    float64 duty, period, w;
    float64 pi = 3.14;
    uint8 msg[128] = { 0, };
    uint8 T[64] = { 0, };

    ecap1 = ecapGetCAP1(ecapREG1);
    ecap2 = ecapGetCAP2(ecapREG1);
    ecap3 = ecapGetCAP3(ecapREG1);

    duty = (ecap2 - ecap1) * 1000 / VCLK3_FREQ;
    period = (ecap3 - ecap1) * 1000 / VCLK3_FREQ;
    w = ((2*pi)/(432*period))*pow(10,9);

    sprintf(msg, "duty = %.3lf, period = %.3lf, W = %.3lf\r\n", duty, period, w);
    send_data(sciREG1, msg, 128);
}
```

Motor Control – eCAP code

```
duty = 135300.000, period = 268300.000, W = 54.182
duty = 136200.000, period = 269800.000, W = 53.881
duty = 132200.000, period = 262100.000, W = 55.464
duty = 138000.000, period = 273200.000, W = 53.210
duty = 134900.000, period = 267100.000, W = 54.425
duty = 135100.000, period = 268500.000, W = 54.142
duty = 134200.000, period = 266300.000, W = 54.589
duty = 137100.000, period = 271100.000, W = 53.622
duty = 134500.000, period = 266000.000, W = 54.651
duty = 132700.000, period = 264100.000, W = 55.044
duty = 133600.000, period = 265500.000, W = 54.753
duty = 135500.000, period = 268200.000, W = 54.202
duty = 135800.000, period = 269000.000, W = 54.041
duty = 134000.000, period = 265500.000, W = 54.753
duty = 137600.000, period = 273200.000, W = 53.210
duty = 135600.000, period = 267700.000, W = 54.303
duty = 135100.000, period = 268400.000, W = 54.162
duty = 132500.000, period = 262200.000, W = 55.443
duty = 137300.000, period = 272500.000, W = 53.347
duty = 134900.000, period = 267300.000, W = 54.385
duty = 135700.000, period = 268700.000, W = 54.101
duty = 134000.000, period = 265600.000, W = 54.733
duty = 137600.000, period = 271400.000, W = 53.563
duty = 134100.0
```