

```

/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////
////      Lab 3 - PWM
////      -SAMPLE CODE DOES NOT WORK-
////      - set up appropriate registers
////      - set up TC pointer
/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////

```

```

/*
 * Include header files for all drivers
 */
#include <clock.h>
#include <conf_clocks.h>

```

```

//setup a the correct TC pointer for the corresponding pins. (use
table 5-1 as a reference)

```

```

/* Set correct PA pins as TC pins for PWM operation */

```

```

void enable_port(void)
{
    //setup pins

```

```

}

```

```

/* Perform Clock configuration to source the TC
1) ENABLE THE APBC CLOCK FOR THE CORREECT MODULE
2) WRITE THE PROPER GENERIC CLOCK SELETION ID*/

```

```

void enable_tc_clocks(void)
{
    PM->APBCMASK.reg |= _____; // PM_APBCMASK_____ is
in the ____ position

```

```

    uint32_t temp=_____; // ID for _____ is
_____ (see table 14-2)
    temp |= 0<<8; // Selection Generic clock generator 0
    GCLK->CLKCTRL.reg=temp; // Setup in the CLKCTRL register
    GCLK->CLKCTRL.reg |= 0x1u << 14; // enable it.

```

```

}

```

```

/* Configure the basic timer/counter to have a period of_____ or a
frequency of _____ */

```

```

void enable_tc(void)
{
    enable_port();
    enable_tc_clocks();

```

```

        /* Set up CTRLA */
        /*
        Set the
        1) counter mode
        2) prescaler
        3) set the PRESCSYNC bits to PRESC from table 27-4 in the
datasheet
        */

        /* Write a suitable value to fix duty cycle and period.*/

        //Wait for sync operation

        /*Enable TC */
    }

int main (void)
{
    system_clock_init();

    /* Enable the timer*/
    enable_tc();

    while(1)
    {

    }

}

```