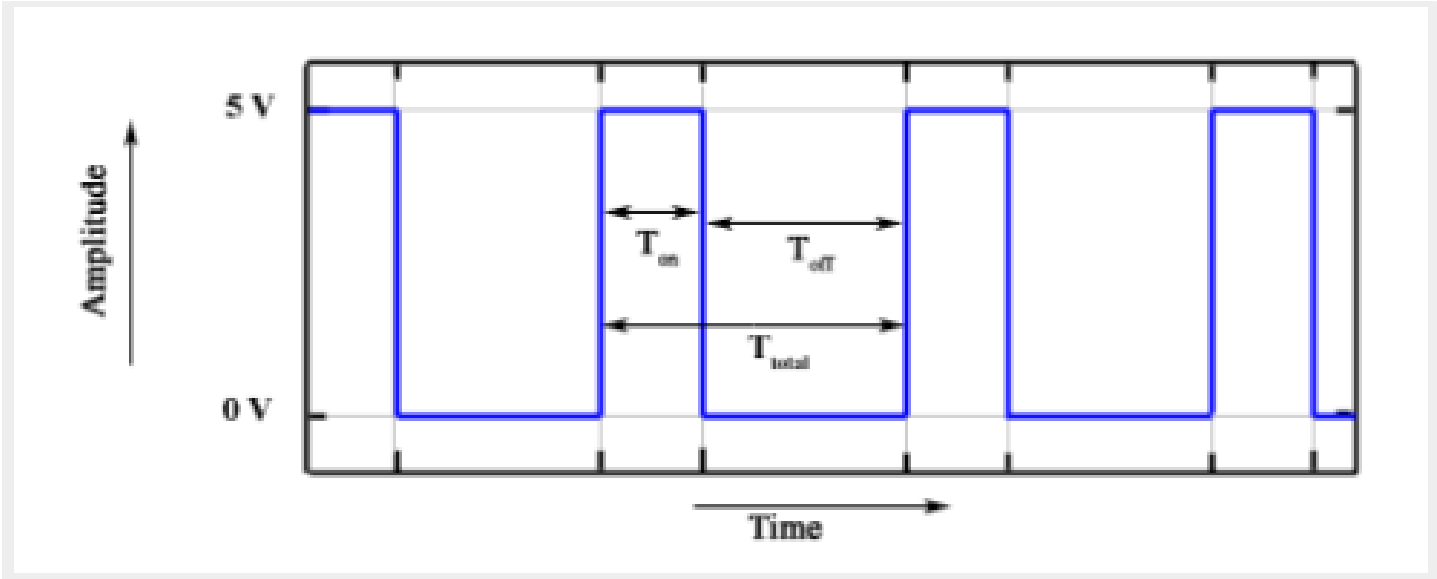
Lab:09
Generating a PWM Waveform

Engr.Shahzada Fahim Jan

PWM Waveform

- PWM is an abbreviation of Pulse Width Modulation
- Pulse width modulation is basically a square wave with a varying high and low time.



- There are various terms associated with PWM:

➢On-Time: Duration of time signal is high

➢Off-Time: Duration of time signal is low

➢Period: It is represented as the sum of on-time and off-time of PWM signal

➢Duty cycle: It is represented as percentage of time signal remains on during the period of the PWM signal

- It is used in those engineering projects where you want an analog output.
- For example, you want to control the speed of your DC motor then you need PWM pulse.
- Using PWM signal you can move your motor at any speed from 0 to its max speed.
- Similarly suppose you want to dim your LED light, again you will use PWM pulse.

- Period
- Suppose T_{on} denotes the on-time and T_{off} denotes the off time of signal.
- Duty Cycle
- Duty cycle is calculated as on-time to the period of time. Using the period calculated above,

Generate a signal of 500 Hz with duty cycle 40%

- **<u>CALCULATIONS:</u>**
- As we have given frequency =**500Hz**
- We know that Time period **(T)=1/f**
- **T=1/500=0.002** or **2ms**
- Now we know that duty cycle=uptime/total time
- Duty cycle=**40% or 0.4**
- **0.4=uptime/0.002**
- Uptime=**0.0008 or 800micros**
- **As total time =2ms so the we have uptime of 800micrs and then 1200mics**

Table 7.5: Output unit modes *(Courtesy of Texas Instruments, Inc.)*

MODx	Mode	Description
000	Output	The output signal OUTx is defined by the OUTx bit. The OUTx signal updates immediately when OUTx is updated
001	Set	Output set when timer reaches the TACCRx value. It remains set until a timer reset, or until another mode affecting the output is selected
010	Toggle/Reset	The output is toggled when the timer counts to the TACCRx value. It is reset when the timer counts to the TACCR0 value
011	Set/Reset	The output is set when the timer counts to the TACCRx value. It is reset when the timer counts to the TACCR0 value
100	Toggle	The output is toggled when the timer counts to the TACCRx value. The output period is double the timer period
101	Reset	Output reset when timer reaches the TACCRx value. It remains reset until another output mode is selected and affects the output
110	Toggle/Set	The output is toggled when the timer counts to the TACCRx value. It is set when the timer counts to the TACCR0 value
111	Reset/Set	The output is reset when the timer counts to the TACCRx value. It is set when the timer counts to the TACCR0 value

```
Description: This program generates two PWM outputs on P4.0,P8.3 using
Timer1_A configured for up mode. The value in CCR0, 1000-1, defines the PWM
period and the values in CCR1 and CCR2 the PWM duty cycles. Using ~1MHz
SMCLK as TACLK, the timer period is ~1ms with a 75% duty cycle on P1.2
and 25% on P1.3.
ACLK = n/a, SMCLK = MCLK = TACLK = 1MHz


          MSP430FR4133
        ---------------
    /|\|                 |
     | |                 |
     --|RST              |
       |                 |
       |     P4.0/TA1.1  |--> CCR1 - 75% PWM
       |     P8.3/TA1.2  |--> CCR2 - 25% PWM
```

```c
#include <msp430.h>

int main(void)
{
    WDTCTL = WDTPW | WDTHOLD;                 // Stop WDT

    // Configure GPIO
    P4DIR |= BIT0;                            // P4.0 and P8.3 output
    P8DIR |= BIT3;
    P4SEL0 |= BIT0;                           // P4.0 and P8.3 options select
    P8SEL0 |= BIT3;

    // Disable the GPIO power-on default high-impedance mode to activate
    // previously configured port settings
    PM5CTL0 &= ~LOCKLPM5;
```

```
    TA1CCR0 = 1000-1;                         // PWM Period
    TA1CCTL1 = OUTMOD_7;                      // CCR1 reset/set
    TA1CCR1 = 750;                            // CCR1 PWM duty cycle
    TA1CCTL2 = OUTMOD_7;                      // CCR2 reset/set
    TA1CCR2 = 250;                            // CCR2 PWM duty cycle
    TA1CTL = TASSEL__SMCLK | MC__UP | TACLR;  // SMCLK, up mode, clear TAR

    __bis_SR_register(LPM0_bits);             // Enter LPM0
    __no_operation();                         // For debugger
}
```

TASKS:

- 1) Generate a signal of 500Hz with 40% duty cycle
- 2) Generate a signal of 600Hz with 60% duty cycle on P1.3

Hint: use timer

3) Generate a signal of 100Hz with 40% duty cycle on P1.2 When a user presses a button at P2.3 the signal change to 300Hz with 60% duty cycle.

Button pressed means Press and release.

This task consists of two parts,

A. Generate a signal x of 2KHz with 75% duty cycle on P1.2. Similarly, generate another signal y of 1KHz with 25% duty cycle on P1.3. As soon a user presses a button on P2.1, x frequency drops by 100Hz and y increases by 100Hz. If x crosses y, an LED at P2.2 is turned ON.

Use low power mode when nothing is happening. Additionally, use interrupts and not polling in your program.

a. Use Timer interrupt for delay creation