Task-7.1 Introduction

(1) Partial block diagram for the internal resources of TC2 of ATmega328 Microcontroller

(2) The 8-bit Timer/Counter2 register has the symbolic name: TCNT2. The TC2 has no association with any physical pin like TC0 and TC1. This indicates that the TCNT1 cannot be driven by external pulses. The TC2 has various modes of operation:

(a) Normal Mode: In this mode, the TC2 works as an up-counter register. It begins counting from a pres-set value and reaches at the maximum count of 0FFh. The arrival of the next clocking pulse makes TC2 to roll-over from all 1s to all 0s. As a result, the TOV2 flag assumes LH state. If interrupt switches K13 and K14 are enabled, the MCU is interrupted and goes to location 0012h to execute the ISRTOV2.

(b) Pulse Width Modulator: A pulse width modulated (PWM) signal can be obtained at Pin-5 (Arduino DPin-3) of the MCU. Both T (period) and PW (pulse width) of the output signal are changeable.

(c) Many more operating modes (can be found in the data sheets).

(3) The driving clock of TCNT2 comes from 16 MHz oscillator after being divided by prescaler.

(4) The OCR2B (Output Compare Register B of TC2), OCF2B Output Compare Match Flag B of TC2), and Wave Generator come into picture when the TC2 is operated in PWM Mode.

(5) Working principles of PWM Mode (non-inverting) at Pin-5 (OC2B):

(a) PWM refers to the continuous change in the ON period (Pulse Width, PW) of a fixed frequency signal. Fast PWM (FPWM) refers to a particular type of PWM which can operate relatively at a high frequency (Period, T).

(b) Single Slope PWM (SSFPWM) is a kind of PWM in which the frequency remains confined with the ‘up-counting time’ of the TC2.

(c) Non-inverting PWM means that the OC2B pin is initially kept at LH state.
The Timing Diagram for Single Slope FPWM at OC2B pin is depicted in Fig-7.2.

After the PWM mode set, the TC2 starts counting from 00h; its output faces continuous comparison with the content of OC2R. The OC2B-pin is initially kept at LH (non-inverting); but, it can also be kept at LL (inverting).

When the content of TCNT2 and OC2R becomes equal, a CME (compare match event) occurs; the OC2B-pin goes to LL-state and remains in this state until the TC2 rolls-over by CRE (count reset event). The CRE event forces the OC2B-pin to assume LH-state again. Thus, we see that a complete signal cycle (PWM) has appeared within the ‘up-counting time’ of TC2.

The ON period (PW) of the OC2B signal can be varied dynamically by changing the contents of OC2R register.

The CME and CRE events set the OC2FB and TOV2 flags of TIFR2 (TC2 Interrupt Flag Register). These flag bits can be effectively used to interrupt the MCU for taking actions if there any like updating the content of OCR2 register.

In PWM Mode, the OCR2B register is ‘double buffered’ which means that the user program can write new value into OCR2B at any time. The value will first enter into a temporary register and then into the actual OCR2B at the next CRE event.

The ATmega328 is capable of generating the following six PWM signals:

Figure-7.3: PINS of ATmega328/ArduinoUno showing PWM signals signatures
**Task-7.2**  
Generation of Non-inverting SSFPWM Signal (62.50 KHz) at OC2B-pin of ATmega328 using ArduinoUNO. Change PW at 3-sec interval by polling TOV1 flag.

A Fast Single Slope PWM (FSSPWM) signal will be generated at OC2B-pin (Pin-5/DPin-3/PD3) of the ArduinoUNO board containing Atmega328 microcontroller. The frequency of the PWM signal would be 62.50 KHz (Period 16 µS). The program will begin with an initial PW (Pulse Width) of 2 µS. At the end-of-4 seconds delay, the PW will be augmented by another 2 µS. The 4-sec time delay will be generated by polling the TOV1 flag of TC1.

(1) The procedures:

**START:**

```assembly
nop
L1:  ; Initialize Stack
L1A: Initialize TC1 as 4-sec delay counter
     Initialize others as needed
L2:  Configure TC2 for SSFPWM Mode at OC2B-pin and start TC2 to produce FPWM of 62.50 KHz
L1D: Start TC2 by selecting clkI0
L3:  if (5-sec has not elapsed)  
      Goto L3
      Reset TOV1
      Reload TCNT1 by preset value of 4-dsec time delay
L4:  On the built-in LED (L) of ArduinoUNO
L5:  Increase PW by 2 µS
L6:  Goto L3 to modulate PW
```

(2) Assembly Codes for the instructions of Step-1

```assembly
.include "m328def.inc"
.cseg
.org 0x0000
RESET:                    nop
     rjmp 0x0040
.org 0x0040
START:                    nop
     L1: ;-- Stack Top at 0x0900----
         ldi r16, 0x90
         out sph, r16
         ldi r16, 0x00
         out spl, r16
     L1A: ;-- configure TC1 as 4-sec delay counter--- clkI0 = 16 MHz/1024 = 15625 Hz----------
         ldi r16, 0x00
         out TCCR1A, r16 ; Normal TC1 operation
         out TCCR1B, r16 ; TC1 is OFF
         ;-------------------
         ldi r16, 0x0B ; 4-sec Time Delay parameter: 10000h – 4x15625 = 0BDCh
         out TCNT1H, r16
         ldi r16, 0xDC
         out TCNT1L
         ;-------------------
         ldi r16, 0x05 ; TC1 is ON
         out TCC1B, r16
```
--; store 2 µS delay parameter in r20 for TC2; clkIO = 16 MHz
ldi r20, 20h ; 16x10^6 x 2x10^-6 = 32 = 20h
--; PORT as output------
ldi r16, 0xFF
out DDRD, r16
--; load CCR2 with initial PW----
out OCR2, r20
L2: ;-- configure TC2 as SSFPWM -- clkIO = 16 MHz; N, prescaler = 1; -- f_OC2B (62.50 KHz) = clkIO/(N*256)
ldi r16, 0x23 ; COM2A1 COM2A0 COM2B1 COM2B0 x x WGM21 WGM20 = 00100011
out TCCR2A
ldi r16, 0x01 ; FOC2A FOC2B x x WGM22 CS22 CS21 CS20 = 00000001
out TCCR2B
L3: ;-- check if 4-sec has gone by polling TOV1
in r16, TIFR
ror r16
brcc L3 ; 4-sec has not elapsed
--; clear TOV1—
ldi r16, 0x04
out TIFR, r16
--; reload preset value in TCNT1—
ldi r16, 0x0B ; 4-sec Time Delay parameter: 10000h – 4x15625 = 0BDCh
out TCNT1H, r16
ldi r16, 0x0C
out TCNT1L
L4: ;---ON/OFF Built-in LED (L) of Arduino——
;----------------------
L5: ;-- increase PW by 2 µS---
add r20, 0x20
out OCR2, r20
L6: ;-- repeat the process-----
rjmp L3
.exit ; end of program

(3) Arduino C codes for the ASM Program of Step-2 (P72OC2BPWM.ino)

unsigned char x=0x20; // value for initial PW = 2 µS
void setup()
{
  pinMode(13, OUTPUT);
  TCCR1A = 0x00; //TC1 normal counter operation
  TCCR1B = 0x00; //TC1 is OFF
  TCNT1 = 0x0BDC; //Time Delay parameter for 4-sec Time Delay
  TCCR1B = 0x05; //TC1 is ON with clkio = 16 MHz/1024 = 15625 Hz
  OCR2B = x; // initial PW = 2 µS
  DDRD = 0xFF; // PORTD as output
  TCCR2A = 0x23; // TC2 as SSFPWM foc2B = clkIO/(N*256) = 62.50 KHz
  TCCR2B = 0x01; // TC2 as SSFPWM; clkIO = 16 MHz
}

void loop()
{
while ((bitRead(TIFR1, 0)) != 1) // checking if 4-sec has elapsed
{
    bitWrite(TIFR1, 0, 1); // TOV1 is cleared
    TCNT1 = 0x0BDC; // reload preset value

digitalWrite(13, digitalRead(13)^1); // Latch LED (L)
x = x+0x20; // PW changes by 2 µS
OCR2B = x; // modulate PW by 2 µS
}

(4) Let us compile, upload, and execute the program of Step-3.
(5) Connect a Ch-0 of an oscilloscope at DPin-3 of the ArduinoUNO Board.
(6) Observe that the oscilloscope shows a PWM signal of 16 µS period (62.50 KHz). The PW changes by 2 µS at every 4-sec interval. Also, observe that L of the Arduino Board changes its state from ON to OFF and vice versa at 4-sec interval.