```
//-----------------------------------------------------------------------
{
TCCR2A = TCCR2A | 0x30;
TCCR2B = TCCR2B & 0xF8 | 0x01;
analogWrite (11,117);
analogWrite (3,137);
}
void loop()
{
  // put your main code here, to run repeatedly:
  analogWrite (3,255-(analogRead(A0)/4));
  analogWrite (11,analogRead(A0)/4);
}
```

1. TCCR2A = TCCR2A | 0x30;

The present content (reset value 0x00, or Arduino value, or user's pre-set value)) will be bit-wise ORed with 0x30 (0011 0000). As a result, the final content of TCCR2A register is: 0x30 (0011 0000).

TC2 Control Register A TCCR2A

Bit	7	6	5	4	3	2	1	0
	COM2A1	COM2A0	COM2B1	COM2B0			WGM21	WGM20
Access	R/W	R/W	R/W	R/W			R/W	R/W
Reset	0	0	0	0			0	0

Figure-1: layout of TCCR2A register

2. TCCR2B = TCCR2B & 0xFB | 0x01;

The order of evaluation is from left-to-right. So, the present content (reset value, 0x00) will be bit-wised ANDed with 0xF8 (1111 1000); then, the result (0x00) will be bit-wise ORed with 0x01 (0000 0001). As a result, the final content of TCCR2B register is: 0x01 (0000 0001).

TC2 Control Register B TCCR2B

Bit	7	6	5	4	3	2	1	0
	FOC2A	FOC2B			WGM22	CS2[2:0]		
Access	R/W	R/W			R/W	R/W	R/W	R/W
Reset	0	0			0	0	0	0

Figure-2: Layout of TCCR2B register

The outcome of the initialization of these two register is:

Timer-2 of TC2 is configured to work as T2 with clkTC2 = clkYSY/1 = 16 MHz.

3. analogWrite (11, 117);

This is a built-in function of the Arduino. The execution of this function provides the following result (there is no need of any initialization):

(a) A 490 Hz SQW (square wave signal) appears at DPin-11 (Digital Pin-11 of Arduino UNO0. This pin is internally connected with PB3-pin of PORTB register. This is a PWM (pulse width Modulated signal) signal whose duty cycle (ON period of the wave) can be varied by changing the value of the 2nd argument (present value 117).

(b) The duty cycle of the PWM is proportion to 117. The period of the signal is: 2040 µs.

4. analogWrite (3, 137);

This is a built-in function of the Arduino. The execution of this function provides the following result (there is no need of any initialization):

(a) A 490 Hz SQW (square wave signal) appears at DPin-3 (Digital Pin-3 of Arduino UNO0. This pin is internally connected with PD3-pin of PORTD register. This is a PWM (pulse width Modulated signal) signal whose duty cycle (ON period of the wave) can be varied by changing the value of the 2nd argument (present value 137).

(b) The duty cycle of the PWM is proportion to 137. The period of the signal is: 2040 µs.

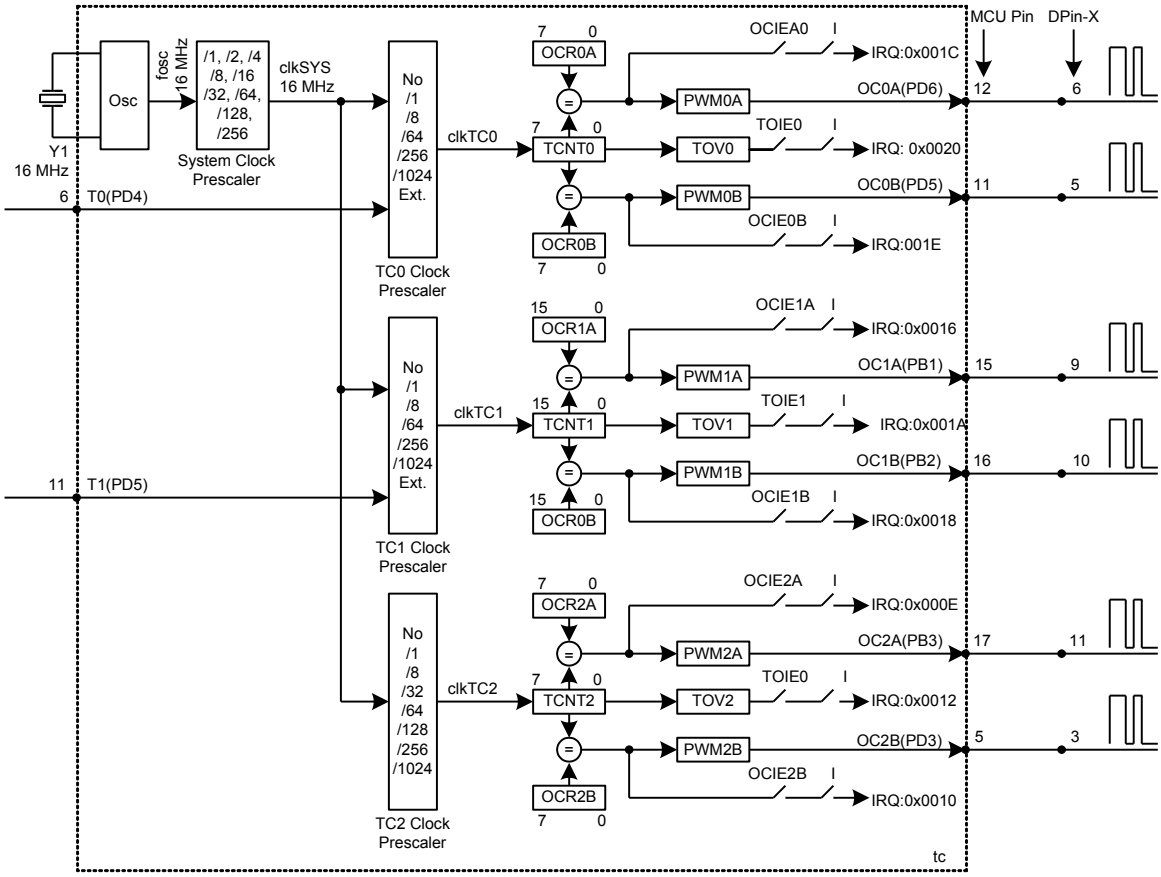

Figure-3: PWM Module of Atmega328 MCU

5. analogWrite (3,255-(analogRead(A0)/4));
   analogWrite (11,analogRead(A0)/4);

Now, the duty cycle (pulse width) of the PWM signals at DPins-3, 11 are being changed by the argument -2 which is: 255 – analogRead(A0)/4.

The execution of the instruction '**analogWrite (3,255-(analogRead(A0)/4));**' is involved with the following steps:
(a) Executes the instruction analogRead(A0)/4;. This instruction reads an analog value (range: 0 to 1023/4 = 0 – 255) from Ch-0 of the ADC Module of the ATmega328 MCU. The value is subtracted from 255, and then the result is used to determine the pulse width of the 490 Hz PWM signal. By varying the value of Pot, the duty cycle of the PWM signal can be continuously changed.

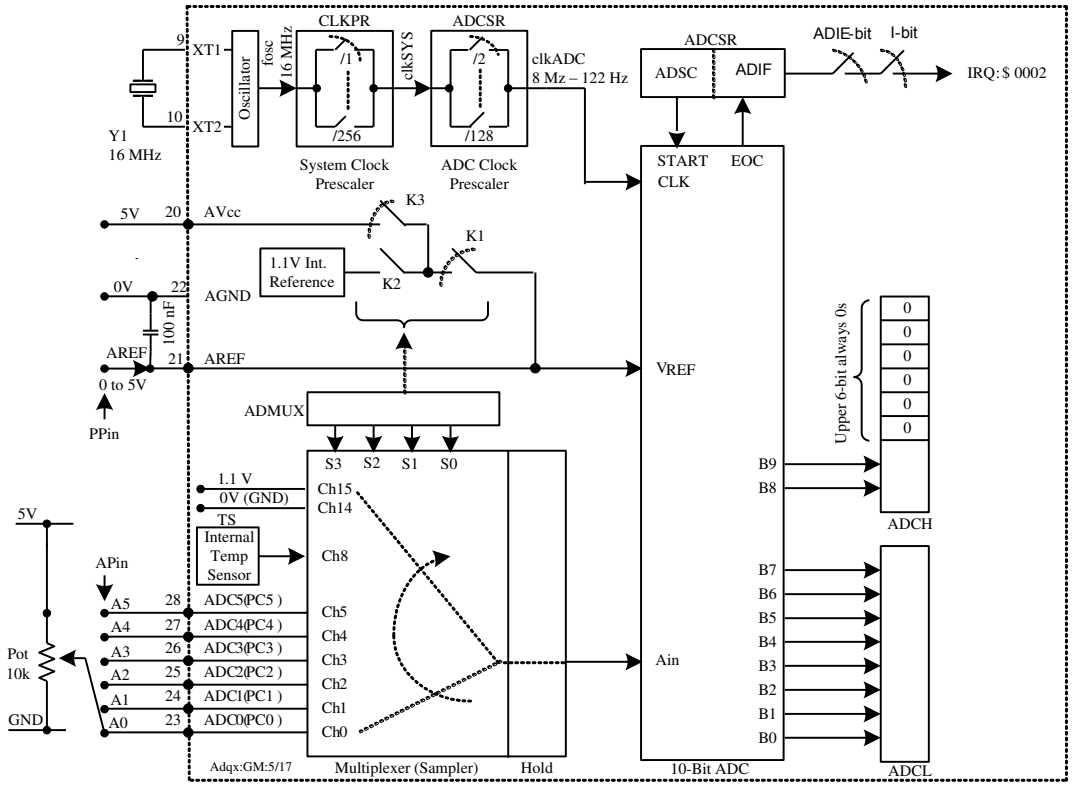

Figure-4: ADC Module of ATmega328

6. Generation of Non-inverting SSFPWM Signal (62.50 KHz) at OC2B-pin of ATmega328 using ArduinoUNO. Change PW at 2-µs quanta at 4-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 ATmega328P 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) Arduino C codes for the ASM Program (P72OC2BPWM.ino)

```
unsigned char x = 0x20;          // value for initial PW = 2 uS     ; 16x10^6 x 2 x 10^-6 = 32 = 20h
void setup()
{
  pinMode(13, OUTPUT);     //Built-in LED (L) of Arduino UNO will blink at
  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 clkTC1 = clkSYS(16 MHz/1024 = 15625 Hz

  OCR2B = x;                  // initial PW = 2 uS
  DDRD = 0xFF;                // PORTD as output

  TCCR2A = 0x23;              // TC2 as SSFPWM foc2B = clkSYS/(N*256) = 62.50 KHz
  TCCR2B = 0x01;              // TC2 as SSFPWM; clkTC2 = clkSYS/N

}

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 uS
  OCR2B = x;                           // modulate PW by 2 uS
}
```

(2) Let us compile, upload, and execute the program of Step-1.

(3) Connect a Ch-0 of an oscilloscope at DPin-3 of the ArduinoUNO Board.

(4) 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.

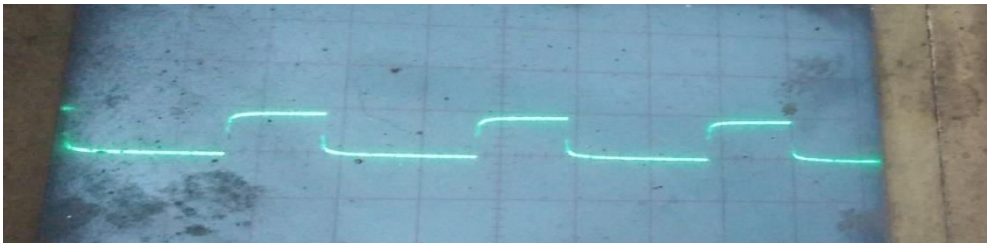

Figure-5: *Oscillograph for 62.50 KHz PW signal at OC2B-pin (DPin-3) of ATmega328*

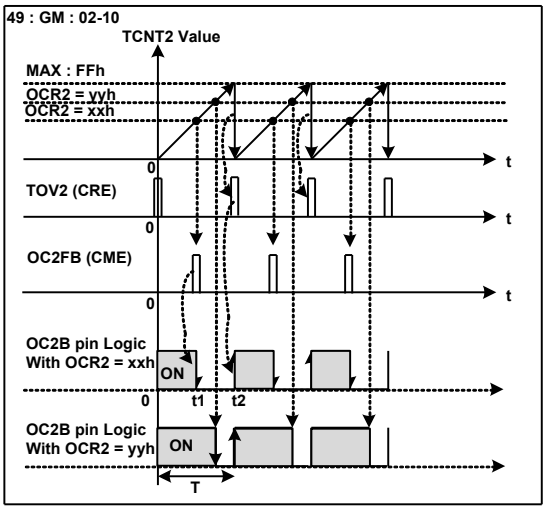

Figure-6: *Timing diagram of TC2 generated SSFPWM signal (OC2B) at DPin-3*