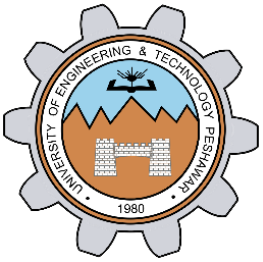
ASSIGNMENT 04



FALL 2022

EMBEDDED SYSTEMS

Submitted by : **SHAHZADA FAHIM JAN**

Registration no: **19PWCSE1765**

Semester**: 7th**

Section : **A**

"On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work."

Student signature _____

Submitted to:

Dr. BILAL HABIB

DECEMBER 11, 2022

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

Q No 01:

This task consists of two parts,

A. **PWM Generation:** 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

   Bonus Points: Run it on Actual board and show results on Oscilloscope.

CODE:

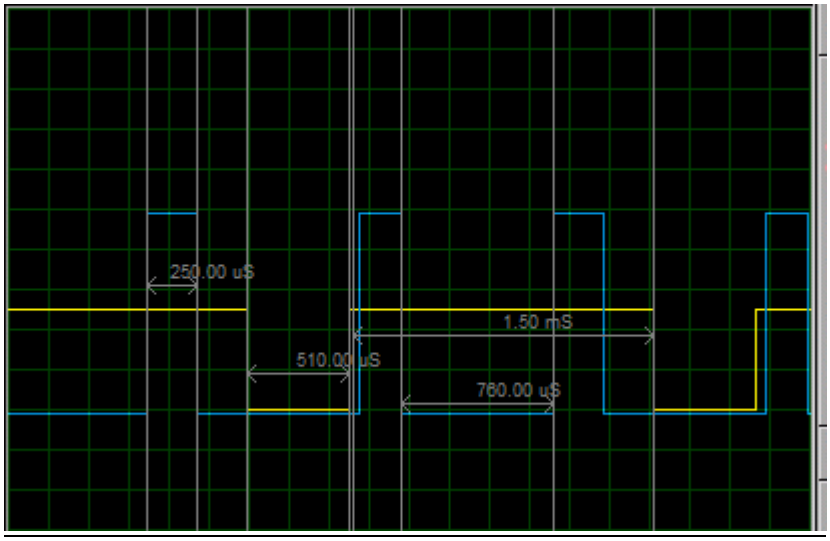
```
#include <MSP430.h>
#include<stdint.h>
#include<stdio.h>
  //For Changing the Duty Cycle in Interrupt
  int ON_Time_X=0, ON_Time_Y=0;
  //Total Cylces Signal X Signal Y
  int FREQUENCY_X=2000;
  int FREQUENCY_Y=1000;
  // UP TIME AND DOWN TIME
  int UP_TIME_X=1500;
  int DOWN_TIME_X=500;
  int UP_TIME_Y=250;
  int DOWN_TIME_Y=750;
  //variables for new frequency
  int X1=0;
  int X2=0;
  int Y1=0;
  int Y2=0;
  int NEW_FREQUENCY_1;
  int NEW_FREQUENCY_2;
int main (void)
{

   WDTCTL = WDTPW | WDTHOLD;//stop watch  dog timer
   BCSCTL1=CALBC1_1MHZ;
   DCOCTL=CALDCO_1MHZ;

   P1DIR= BIT2 | BIT3 ; // // SETS  the ouput  P1.2  and P1.3
   // Trigerring Configuaration
   P2IES=0; //H -> L
   // enbales the port interrupt at P2.1
   P2IE=BIT1;
```

```
   // Clearing the Flags
   P2IFG=~BIT1;
  P1OUT =BIT2 | BIT3;
  P2DIR=BIT2;
  P2OUT=~BIT2;

   //Signal X

   TA0CCR1=UP_TIME_X; //       75% Duty Cycle 0.75(2000)=1500

   //Signal Y
   TA0CCR2= UP_TIME_Y; //  25% Duty Cycle 0.25(1000)=250


   TA0CCTL1=CCIE ; // interrupt enabler for TACCR1
   TA0CCTL2=CCIE ;// interrupt enabler for TACCR2
   TA0CCTL0=CCIE ; // interrupt enabler for TACCR0
   //Timer configuration
   TA0CTL= MC_2 | TASSEL_2 | ID_0| TACLR |TAIE ; //continuous mode Timer :
SMCLK divider:1

 while(1)
 {
     __bis_SR_register(LPM4_bits | GIE);
     //Sleeping Mode
 }

}
//Timer0 A
 #pragma vector = TIMER0_A1_VECTOR //TIMERA1_VECTOR  for TA0CCR1
__interrupt void TA1_ISR()
 {
switch(TA0IV)
{
 case 2: //For TA0CCR1 Flag
 {
   P1OUT ^=BIT2;
 if(ON_Time_X==0)
{
          TA0CCR1+=DOWN_TIME_X;// DOWN TIME
          ON_Time_X=1;
}
 else
{
          TA0CCR1+=UP_TIME_X; //UP TIME
          ON_Time_X=0;
```

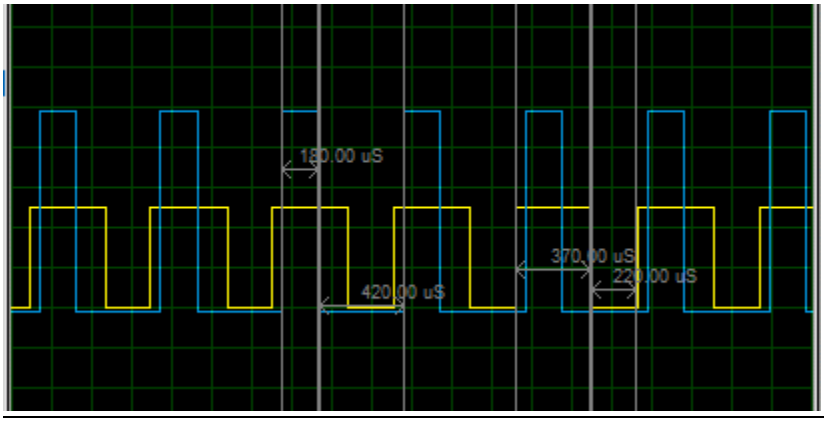
```
}

 }
 break;
 case 4: // For TA0CCR2 Flag

 {
   P1OUT^=BIT3; // Toggles the outputs at P1.3 when CCIFG2 sets

 if(ON_Time_Y==0)
 {
         TA0CCR2+=DOWN_TIME_Y;// DOWN TIME
         ON_Time_Y=1;
 }
else
 {
    TA0CCR2+=UP_TIME_Y; // UP TIME
    ON_Time_Y=0;
 }

 }
 break;
 }
 }

 //P2.1 INTERRUPT

 #pragma vector=PORT2_VECTOR
 __interrupt void port_2(void)
 {
 //ON Time OFF Time
  FREQUENCY_X=FREQUENCY_X-100; // 2000 1900 1800
  if(FREQUENCY_X==100)
   FREQUENCY_X=2000;
  FREQUENCY_Y=FREQUENCY_Y+100; //1000  1100  1200
  if(FREQUENCY_Y==2000)
    FREQUENCY_Y=1000;

  NEW_FREQUENCY_1=1000000/FREQUENCY_X; // cycles=1MHz/F
  NEW_FREQUENCY_2=1000000/FREQUENCY_Y;
  X1=75*(NEW_FREQUENCY_1/100);
  X2=25*(NEW_FREQUENCY_2/100);
  UP_TIME_X=X1 ;  DOWN_TIME_X=X2;
  Y1=25*(NEW_FREQUENCY_2/100);
  Y2=75*(NEW_FREQUENCY_2/100);
  UP_TIME_Y=Y1; DOWN_TIME_Y=Y2;
```

```
  P2IFG=~BIT1 ;// clears the flag of P2.1

}

//.Timer A.

#pragma vector = TIMERA0_VECTOR //TIMERA0_VECTOR ----> for TA0CCR0
__interrupt void TA0_ISR()

 {
if( FREQUENCY_Y >= FREQUENCY_X)
{
  P2OUT =BIT2; // Turn ON Led at P2.2;
 }
 }
```

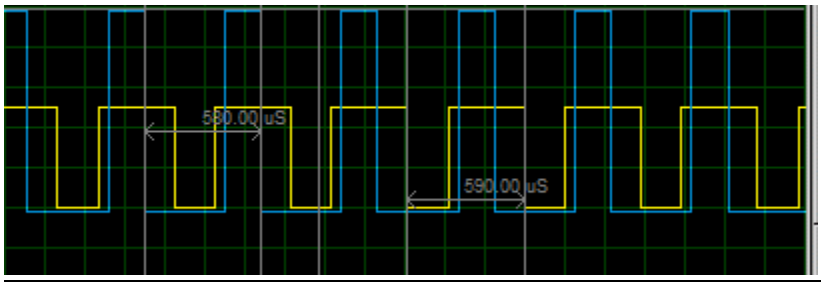
OUTPUT:



WHEN BUTTON PRESSED:



AGAIN PRESSING BUTTON:



CIRCUIT: