#### **CPE301 - SPRING 2018**

# Design Assignment 2

#### **DO NOT REMOVE THIS PAGE DURING SUBMISSION:**

The student understands that all required components should be submitted in complete for grading of this assignment.

NO	SUBMISSION ITEM	COMPLETED (Y/N)	MARKS (/MAX)
1	COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS		
2.	INITIAL CODE OF TASK 1/A		
3.	INCREMENTAL / DIFFERENTIAL CODE OF TASK 2/B		
3.	INCREMENTAL / DIFFERENTIAL CODE OF TASK 3/C		
3.	INCREMENTAL / DIFFERENTIAL CODE OF TASK 4/D		
3.	INCREMENTAL / DIFFERENTIAL CODE OF TASK 5/E		
4.	SCHEMATICS		
5.	SCREENSHOTS OF EACH TASK OUTPUT		
5.	SCREENSHOT OF EACH DEMO		
6.	VIDEO LINKS OF EACH DEMO		
7.	GOOGLECODE LINK OF THE DA		

#### 1. COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS

One  $220\Omega$  resistor Two  $10K\Omega$  resistors One pushbutton One red LED (See schematics for diagrams)

### 2. INITIAL/DEVELOPED CODE OF TASK 1/A

No initial code for any tasks.

#### 3. MODIFIED CODE OF TASK 1/A

```
Task 1 assembly code:
```

```
.org 0x00
    LDI R16, HIGH(RAMEND)
    OUT SPH, R16
    LDI R16, LOW(RAMEND)
    OUT SPL, R16
    SBI DDRB, 2
                        ;set PORT2 as output
    LDI R17, 0
                         ;used to intialize PB2 OFF
                         turn PB2 off;
    OUT PORTB, R17
    LDI R16, 4
                         ;used to toggle LED
 LOOP:
                        ;call delay subroutine
    rcall myDelay
                         ;toggle bits of R17
    EOR R17, R16
    OUT PORTB, R17
                         ;toggle LED
    RJMP LOOP
                         ;continue looping
 myDelay:
    LDI R18, 250
                        ;counter for delay
 L1:
    LDI R19, 250
                        ;second nested counter for delay
 L2:
                        ;take 1 clock cycle 250*250 times
    NOP
    DEC R19
                         ;take 1 clock cycle 250*250 times and decrement 2nd count
                        ;keeping delaying if not zero
    BRNE L2
                         ;decrement primary counter
    DEC R18
                         ;go into nested loop if not zero
    BRNE L1
    RET
                          ;finished .25 second delay
     Task 1 C code:
 #include <avr/io.h>
 #define F_CPU 1000000UL
 #include <util/delay.h>
□int main()
 {
      DDRB |= (1<<2);
                                     //set PB2 as output
      PORTB &= ~(1<<2);
                                      //set PB2 OFF
      while (1)
                                    //Turn on LED
          PORTB |= (1<<2);
           _delay_ms(250);
                                     //wait 250ms
                                 //Turn off LED
          PORTB &= ~(1<<2);
         _delay_ms(250); //wait 250ms
      }
 }
```

#### 4. MODIFIED CODE OF TASK 2/B

Task 2/B assembly code:

```
.org 0x00
      LDI R16, HIGH(RAMEND)
      OUT SPH, R16
      LDI R16, LOW(RAMEND)
OUT SPL, R16
      SBI DDRB, 2
                              ;set PORT2 as output
                             jused to intialize PB2 OFF
      LDI R17, 0
                             ;set PB2 OFF
      OUT PORTB, R17
                             ;set PD2 as input
;set PD2 as input
      CBI DDRD, 2
      LDI R17 ,0x04
                             ;set PD2 as input
      OUT PORTD, R17
      LDI R25, 0
  LOOP:
      IN R16, PIND
                             ;R16 gets PIND values
      CPI R16, 0x00
                             ;check if button was pressed
                             ;if not 0 keep polling
      BRNE LOOP
      OUT PORTB, R17
                              ;Turn on LED
      rcall myDelay250ms
                             ;call 250ms delay 4 times for
      rcall myDelay250ms
                             ;overall 1 second delay
      rcall myDelay250ms
      rcall myDelay250ms
      OUT PORTB, R25
                              ;turn off LED
      jmp LOOP
  myDelay250ms:
      LDI R18, 250
                              ;counter for delay
      LDI R19, 250
                              ;second nested counter for delay
  L2:
      NOP
                             ;take 1 clock cycle 250*250 times
                              ;take 1 clock cycle 250*250 times and decrement 2nd count
      DEC R19
      BRNE L2
                              ;keeping delaying if not zero
      DEC R18
                              ;decrement primary counter
      BRNE L1
                              ;go into nested loop if not zero
                              ;finished .25 second delay
      RET
      Task 2/B C code:
 #include <avr/io.h>
 #define F_CPU 1000000UL
 #include <util/delay.h>

☐ int main(void)

 {
      DDRB |= 0xFF;
                                //set PORTB as output
                                //initialize LED OFF
      PORTB = 0x00;
      PORTD = 0x04;
                                //turn on pull-up
                               //while button is not pressed
      while (1)
      {
          if(PIND & 0x04)
                                    //if the button isn't pressed
          {
              //do nothing
          }
          else
          {
              PORTB |= (1<<2);
                                    //Turn on LED
                                   //wait 1 second
               _delay_ms(1000);
              PORTB &= ~(1<<2);
                                   //Turn off LED
          3
      }
```

#### MODIFIED CODE OF TASK 3/C

Task 3/C assembly code:

```
.org 0
      LDI R16, 4
                           ;used to toggle LED
                           ;used to initialize TCCR0A
      LDI R18, 0
      SBI DDRB, 2
                           ;PB2 as output
                           ;needed to toogle led
      LDI R17,0
      OUT PORTB, R17
                           turn LED off;
  begin:
      LDI R20, 12 ;250ms delay with 1024 prescaler
OUT TCNT0, R20 ;load value into timer
LDI R20, 5 ;to set prescaler
OUT TCCR0B, R20 ;Prescaler: 1024
OUT TCCR0A, R18 ;Timer0, normal mode, initalize clock
  loop:
      IN R20, TIFR0
                           ;read in TIFR0
      SBRS R20, 0 ;if TOV0 is set skip next instruction
    RJMP LOOP ;keep polling
      LDI R20, 0
      OUT TCCR0B, R20 ;stop the timer
      LDI R20, 1
      OUT TIFRØ, R20
                           ;reset TOV0 flag
                           ;XOR to toogle led
      EOR R17, R16
      OUT PORTB, R17
                           ;toggle LED
      RJMP begin
                            :reset
      Task 3/C C code:
 #include <avr/io.h>
□int main(void)
 {
                               //set PB2 as output
//turn PB2 LED off
     DDRB |= (1<<2);
PORTB &= ~(1<<2);
TCCRØA = 0;
                                  //Timer0, normal mode, initialize clock
                                 //prescaler of 1024
     TCCRØB = 5;
     TCNT0 = 12;
                                  //250ms delay value
     while (1)
         //reset o...
//reset counter
                                         //toggle LED
              PORTB ^= (1<<2);
             TIFR0 |= (1<<TOV0);
                                           //reset overflow bit
             TCNT0 = 12;
         }
     }
 }
```

#### 6. MODIFIED CODE OF TASK 4/D

Task 4/D assembly code:

```
.org 0x0
    jmp MAIN
                                 ;addr for Timer1 overflow
.org 0x1A
   jmp T1_OV_ISR
.org 0x100
MAIN:
    ldi R17, HIGH(RAMEND)
                                 ;initialize the stack
    out SPH, R17
    ldi R17, LOW(RAMEND)
out SPL, R17
    sbi DDRB, 2
                        ;set PB2 as output
    ldi R17, 0
    out PORTB, R17
                             ;turn off LED initially
                            ;upper bits of 61630
;set high bits of counter
    ldi R17, 0xF0
    sts TCNT1H, R17
                             ;lower bits of 61630
    ldi R17, 0xBE
                             ;set low bits
    sts TCNT1L, R17
    ldi R17, 0
    sts TCCR1A, R17
                             ;normal mode
    ldi R17, 3
    sts TCCR1B, R1/
ldi R17, (1<<TOIE1)
R17 ;set flag bit
                             ;set prescaler to 64
                    ;enable the interrupt
    sei
again:
    jmp again
                                  ;loop until interrupt occurs
T1_OV_ISR:
    LDI R20, 1<<TOV1
                            clear the flag bit;
    sts TIFR1, R20
                             ;flag bit cleared
    IN R16, PORTB
                             ;read in PB2
    LDI R17, 0x04
EOR R16, R17
                          ;toggle PB2
;toggle LED
;reload upper bits
    OUT PORTB, R16
    ldi R17, 0xF0
    sts TCNT1H, R17
                             ;reset the couter
                            reload lower bits;
    ldi R17, 0xBE
    sts TCNT1L, R17 ; res
RETI ; return
                              ;reset the counter
  RETI
```

#### Task 4/D C code:

```
#include <avr/io.h>
 #include <avr/interrupt.h>
□int main()
 {
                            //set PB2 as output
     DDRB |= 0x04;
     PORTB = 0;
                              //initialize LED off
                              //normal mode
//set pre-scaler to 64
     TCCR1A = 0;
      TCCR1B = 3;
     TCNT1 = 61630;
     TCNT1 = 61630; //set timer value
TIMSK1 = (1<<TOIE1); //enable overflow interrupt
     sei ();
                                //enable interrupts
     while(1)
          //wait for interrupt
}
□ISR (TIMER1 OVF vect)
                            //reset flag bit
//toggle LED
      TIFR1 |= (1<<TOV1);
     PORTB ^= 0x04;
     TCNT1 = 61630;
                               //reset timer
 }
```

#### **MODIFIED CODE OF TASK 5/E** 7.

Task 5/E assembly code:

```
.ORG Ø
                                  ;location for reset
     JMP MAIN
 .ORG 0x02
                                  ;location for EXT_INT0
    JMP EX0 ISR
 .ORG 0x1A
                                  ;location for TIM1_OVF
     JMP T1_OV_ISR
 MAIN:
     LDI R20, HIGH(RAMEND)
                                 ;initialize the stack
     OUT SPH,R20
     LDI R20, LOW(RAMEND)
     OUT SPL,R20
     SBI DDRB,2
                             ;PC.3 = output
                             ;pull-up activated
     SBI PORTD,2
     LDI R20,1<<INT0
                             ;Enable INT0
     OUT EIMSK,R20
                             ;Enable INT0
     LDI R20, (1<<ISC01)
                             ;Configure to falling edge triggered
     sts EICRA,R20
     LDI R20, (1<<TOIE1)
                             ;Enable timer1 overflow interrupt
                             ;Enable TOIE1
;Set I (Enable Interrupts)
     STS TIMSK1, R20
     SET
 HERE:
     JMP HERE
                             ;wait for interrupt
 EX0_ISR:
     LDI R20, 1<<INTF0
                             ;load in flag position
                             ;clear flag
     OUT EIFR, R20
                             ;send 1 to PB2
     LDI R22,0x04
     OUT PORTB, R22
                             turn on LED;
     LDI R17, 0xF0
                             ;get upper bits of 61630
                             ;set high bits of counter
     STS TCNT1H, R17
                             ;get lower bits of 61630
     LDI R17, 0xBE
     STS TCNT1L, R17
                             ;set lower bits
     LDI R17, 0
     STS TCCR1A, R17
                             ;normal mode
     LDI R17, 4
     STS TCCR1B, R17
                             ;set prescaler to 256
     RETI
                             ;finish interrupt
T1_OV_ISR:
    LDI R20, 1<<TOV1
                            ;clear flag bit
    STS TIFR1, R20
    LDI R17, 0
                             ;turn off LED
    OUT PORTB, R17
    STS TCCR1B, R17
                             ;turn off timer
    RETT
                            ;finish interrupt
Task 5/E C code:
```

```
#include <avr/io.h>
#include <avr/interrupt.h>
□int main()
          DDRB = 1<<2;

PORTD = 1<<2;

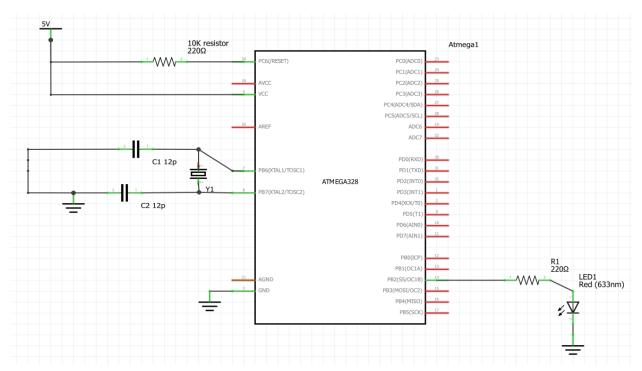
EIMSK = (1<<INT0);

EICRA = (1<<ISC01);

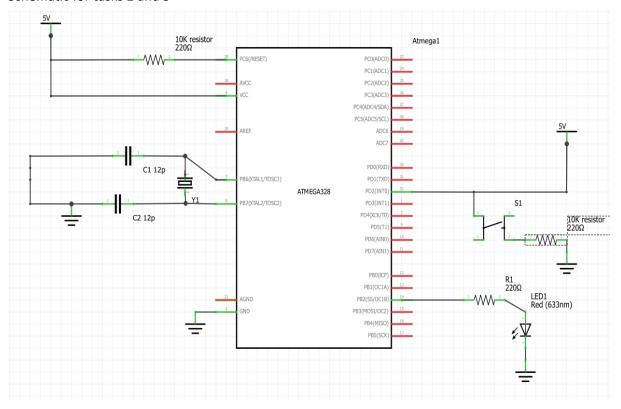
TIMSK1 = (1<<TOIE1);
                                                         //set PB2 as output
                                                        //set PB2 as output
//set up pull up resistor
//enable external interrupt 0
//falling edge trigger
//overflow interrupt timer1 enabled
           sei();
           while (1)
                  //wait for interrupts
|
|}
|□ISR (INTØ_vect)
           EIFR |= (1<<INTF0);
PORTB |= (1<<2);
TCNT1 = 61630;
TCCR1A = 0;
TCCR1B = (1<<CS12);
                                                         //reset flag
                                                         //turn on LED
                                                         //set counter value
//normal mode
                                                         //set prescaler to 256
  L 3-
□ISR (TIMER1_OVF_vect)
          TIFR1 |= (1<<TOV1);
PORTB &= ~(1<<2);
TCCR1B = 0;
                                                         //reset flag
                                                         //turn LED off
//turn off timer
```

#### 8. SCHEMATICS

Schematic for tasks 1,3, and 4

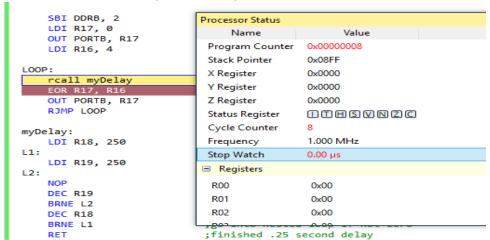


#### Schematic for tasks 2 and 5

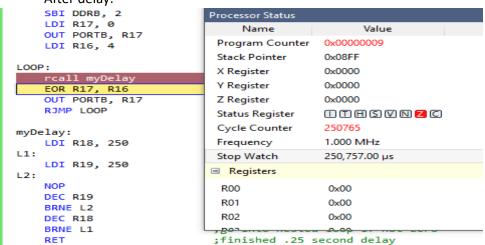


#### 9. SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)

Task 1/A assembly before delay:



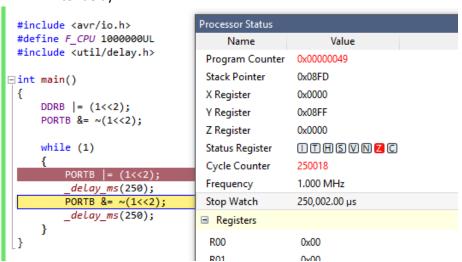
#### After delay:



Task 1/A C code before delay:

```
Processor Status
 #include <avr/io.h>
 #define F_CPU 1000000UL
                                                               Value
                                           Name
 #include <util/delay.h>
                                       Program Counter
                                                        0x00000042
                                       Stack Pointer
                                                        0x08FD
⊡int main()
 {
                                       X Register
                                                        0x0000
      DDRB |= (1<<2);
                                       Y Register
                                                        0x08FF
      PORTB &= ~(1<<2);
                                       Z Register
                                                        0x0000
      while (1)
                                       Status Register
                                                        ITHSVNZC
                                       Cycle Counter
                                                        16
           PORTB |= (1<<2);
                                       Frequency
                                                        1.000 MHz
           <u>_delay_ms(250);</u>
PORTB &= ~(1<<2);
                                       Stop Watch
                                                        0.00~\mu s
           _delay_ms(250);
                                      ■ Registers
 }
                                                         0x00
                                       R00
                                                         0x00
                                       R01
```

#### After delay:



Task 2/B assembly before button pushed:

LUI KI/, Ø	ion pusheu.	
OUT PORTB, R17	Processor Status	
CBI DDRD, 2	Name	Value
LDI R17 ,0x04	Program Counter	0x000000F
OUT PORTD, R17 LDI R25, 0	Stack Pointer	0x08FF
	X Register	0x0000
LOOP:	Y Register	0x0000
IN R16, PIND CPI R16, 0x00	Z Register	0x0000
BRNE LOOP	Status Register	ITHSVN <b>Z</b> C
OUT PORTB, R17	Cycle Counter	16
rcall myDelay250ms rcall myDelay250ms	Frequency	1.000 MHz
rcall myDelay250ms	Stop Watch	0.00 μs
rcall myDelay250ms	■ Registers	
OUT PORTB, R25	R00	0x00
Jiip Loor	R01	0x00
myDelay250ms:	R02	0x00
LDI R18, 250	BUS	0~00

#### After button pushed:

Auter pactor pastical			
LDI KI/, A	SUSPILIO INITAL	LVE PDZ VEE	
OUT PORTB, R17	Processor Status		
CBI DDRD, 2	Name	Value	
LDI R17 ,0x04	Program Counter	0x00000013	
OUT PORTD, R17 LDI R25, 0	Stack Pointer	0x08FF	
	X Register	0x0000	
LOOP:	Y Register	0x0000	
IN R16, PIND CPI R16, 0x00	Z Register	0x0000	
BRNE LOOP	Status Register	UTHSVNZC	
OUT PORTB, R17	Cycle Counter	1003044	
rcall myDelay250ms	Frequency	1.000 MHz	
rcall myDelay250ms rcall myDelay250ms	Stop Watch	1,003,028.00 µs	
rcall myDelay250ms	■ Registers		
OUT PORTB, R25	R00	0x00	
jmp LOOP	R01	0x00	
myDelay250ms:	R02	0x00	
LDI R18, 250	R03	0×00	
1.11			

Task 2/B C code before button pushed:

```
Processor Status
  #include <avr/io.h>
  #define F_CPU 1000000UL
                                                         Value
                                      Name
 #include <util/delay.h>
                                  Program Counter
                                                   0x00000048
                                  Stack Pointer
                                                   0x08FD
⊡int main(void)
                                  X Register
                                                   0x0000
 {
      DDRB |= 0xFF;
                                  Y Register
                                                   0x08FF
      PORTB |= 0x00;
                                                   0x0000
                                  Z Register
      PORTD = 0x04;
                                  Status Register
                                                   UTHSVNZC
      while (1)
                                  Cycle Counter
                                  Frequency
                                                   1.000 MHz
          if(PIND & 0x04)
                                                   0.00 µs
                                  Stop Watch
               //do nothing
                                 ■ Registers
                                  R00
                                                    0x00
          else
                                  R01
                                                    0x00
               PORTB |= (1<<2)
                                  R02
                                                    0x00
                delay_ms(1000)
```

#### After button pushed:

```
Processor Status
 #include <avr/io.h>
 #define F CPU 100000UL
                                       Name
                                                            Value
 #include <util/delay.h>
                                   Program Counter
                                                     0x00000052
                                   Stack Pointer
                                                     0x08FD

    int main(void)

                                   X Register
                                                     0x0000
 {
      DDRB |= 0xFF;
                                   Y Register
                                                     0x08FF
      PORTB |= 0x00;
                                   Z Register
                                                     0x0000
      PORTD = 0x04;
                                   Status Register
                                                     IITHS VNZC
      while (1)
                                   Cycle Counter
                                                     1000023
                                   Frequency
                                                     1.000 MHz
           if(PIND & 0x04)
                                   Stop Watch
                                                     1,000,002.00 µs
               //do nothing
                                  ■ Registers
                                    R00
                                                      0x00
           else
                                    R01
                                                      0x00
               PORTB |= (1<<2)
                                    R02
                                                      0x00
               _delay_ms(1000)
PORTB &= ~(1<<2
                                    RN3
//lurn of
                                                     EDIO
      }
```

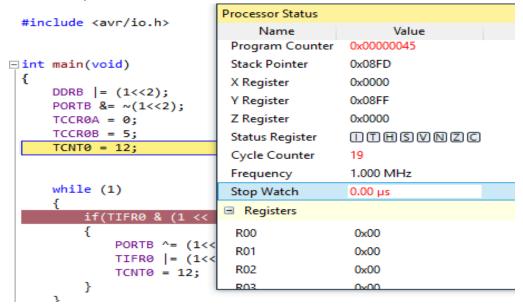
Task 3/C assembly before timer overflow:

; Replace with your app	lica	Processor Status	
.org 0	-110	Name	Value
LDI R16, 4	;us	Program Counter	0x0000000A
LDI R18, 0	;us	Stack Pointer	0x08FF
SBI DDRB, 2 LDI R17,0	;PB ;ne	X Register	0x0000
OUT PORTB, R17	;tu	Y Register	0x0000
begin:		Z Register	0x0000
LDI R20, 12	;25	Status Register	UTHSVNZC
OUT TCNT0, R20 LDI R20, 5	;lo ;to	Cycle Counter	10
OUT TCCR0B, R20	;Pr	Frequency	1.000 MHz
OUT TCCRØA, R18	;⊤i	Stop Watch	0.00 µs
loop: IN R20, TIFR0	;re	■ Registers	
SBRS R20, 0	;if	R00	0x00
10111 2001	,,,,	R01	0x00
LDI R20, 0		R02	0x00
OUT TCCR0B, R20	;st	RU3	0~00

#### After timer overflow:

; Replace with your appl	Processor Status	
.org 0	Name	Value
LDI R16, 4	Program Counter	0x000000D
LDI R18, 0 SBI DDRB, 2	Stack Pointer	0x08FF
LDI R17,0	X Register	0x0000
OUT PORTB, R17	Y Register	0x0000
begin: LDI R20, 12	Z Register	0x0000
OUT TCNT0, R20	Status Register	UTHSVNZC
LDI R20, 5	Cycle Counter	249869
OUT TCCR0B, R20 OUT TCCR0A, R18	Frequency	1.000 MHz
loop:	Stop Watch	249,859.00 μs
IN R20, TIFR0	; □ Registers	
SBRS R20, 0 RJMP LOOP	R00	0x00
	R01	0x00
LDI R20, 0	R02	0x00
OUT TCCR0B, R20	RUS	0~00

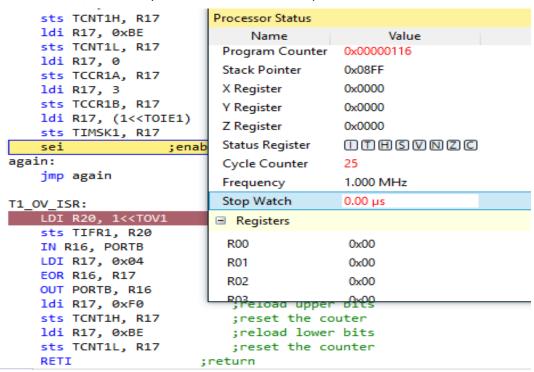
Task 3/C C code before timer overflow:



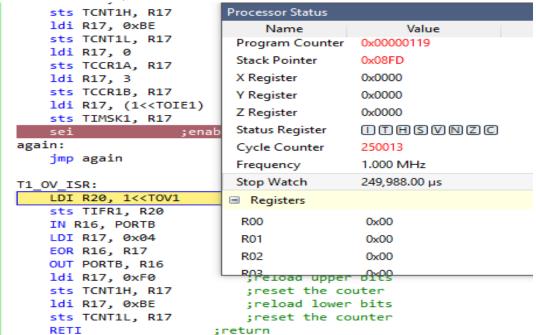
#### After timer overflow:

```
Processor Status
 #include <avr/io.h>
                                    Name
                                                       Value
                                Program Counter
                                                 0x0000004B
                                Stack Pointer
                                                 0x08FD
□int main(void)
 {
                                X Register
                                                 0x00000
      DDRB |= (1<<2);
                                                 0x08FF
                                Y Register
      PORTB &= ~(1<<2);
      TCCRØA = 0;
                                Z Register
                                                 0x0000
      TCCRØB = 5;
                                Status Register
                                                 ITHSVNZC
      TCNT0 = 12;
                                Cycle Counter
                                                 249877
                                Frequency
                                                 1.000 MHz
      while (1)
                                Stop Watch
                                                 249,858.00 µs
                               Registers
           if(TIFR0 & (1 <<
                                R00
                                                  0x00
               PORTB ^= (1<<
                                R01
                                                  0x00
               TIFR0 |= (1<<
                                                  0x00
                                R02
               TCNT0 = 12;
           }
                                 RN3
                                                  0
```

Task 4/D assembly code before timer interrupt overflow:



#### After interrupt occurs:



#### Task 4/D C code before timer overflow interrupt:

```
DDRB |= 0x04;
                               Processor Status
      PORTB = 0;
                                     Name
                                                         Value
      TCCR1A = 0;
                                 Program Counter
                                                  0x00000050
      TCCR1B = 3;
      TCNT1 = 61630;
TIMSK1 = (1<<TOIE1);
                                 Stack Pointer
                                                  0x08FD
                                 X Register
                                                  0x0000
      sei ();
                                 Y Register
                                                  0x08FF
                                                  0x0000
                                 Z Register
      while(1)
                                 Status Register
                                                  IITHS VNZC
           //wait for interr
                                 Cycle Counter
                                 Frequency
                                                   1.000 MHz
 }
                                 Stop Watch
                                                  0.00 µs
☐ISR (TIMER1_OVF_vect)
                                Registers
      TIFR1 |= (1<<TOV1);
                                 R00
                                                   0x00
      PORTB ^= 0x04;
                                 R01
                                                   0x00
      TCNT1 = 61630;
                                                   0x00
                                 R02
                                 בחם
                                                   ۸.۸۸
```

#### After interrupt occurs:

```
DDRB = 0x04;
                              Processor Status
     PORTB = 0;
                                                      Value
                                   Name
     TCCR1A = 0;
                               Program Counter
                                               0x00000059
     TCCR1B = 3;
                               Stack Pointer
     TCNT1 = 61630;
                                                0x08F6
     TIMSK1 = (1 << TOIE1);
                               X Register
                                                0x00000
     sei ();
                               Y Register
                                                0x08FF
                               Z Register
                                                0x0000
     while(1)
                                                Status Register
          //wait for interr
                               Cycle Counter
                                                250024
                               Frequency
                                                1.000 MHz
}
                               Stop Watch
                                                249,995.00 µs
∃ISR (TIMER1_OVF_vect)
                              Registers
      TIFR1 = (1 << TOV1);
                               R00
                                                0x00
     PORTB ^= 0x04;
                                                0x00
                               R01
     TCNT1 = 61630;
                               R02
                                                0x00
 }
                               RU3
                                                0
```

### Task 5/E assembly before timer interrupt:

,	•	
LDI R17, 0xF0	Processor Status	
STS TCNT1H, R17	Name	Value
LDI R17, 0xBE STS TCNT1L, R17	Program Counter	0x0000003B
LDI R17, 0	Stack Pointer	0x08FF
STS TCCR1A, R17	X Register	0x0000
LDI R17, 4	Y Register	0x0000
STS TCCR1B, R17	Z Register	0x0000
	Status Register	<b>I</b> THSVNZC
T1_OV_ISR:	Cycle Counter	99
LDI R20, 1< <tov1 STS TIFR1, R20</tov1 	Frequency	1.000 MHz
LDI R17, 0	Stop Watch	0.00 µs
OUT PORTB, R17 STS TCCR1B, R17	☐ Registers	
RETI	R00	0x00
	R01	0x00
	R02	0x00
	I	

After interrupt:		
LDI R17, 0xF0	Processor Status	
STS TCNT1H, R17	Name	Value
LDI R17, 0xBE STS TCNT1L, R17	Program Counter	0x0000003E
LDI R17, 0	Stack Pointer	0x08FD
STS TCCR1A, R17	X Register	0x0000
LDI R17, 4	Y Register	0x0000
STS TCCR1B, R17	Z Register	0x0000
	Status Register	OTHSVNZC
T1_OV_ISR:		1000046
LDI R20, 1< <tov1< td=""><td>Cycle Counter</td><td></td></tov1<>	Cycle Counter	
STS TIFR1, R20	Frequency	1.000 MHz
STS TIFR1, R20 LDI R17, 0	-	
STS TIFR1, R20 LDI R17, 0 OUT PORTB, R17	Frequency	1.000 MHz
STS TIFR1, R20 LDI R17, 0	Frequency Stop Watch	1.000 MHz
STS TIFR1, R20 LDI R17, 0 OUT PORTB, R17 STS TCCR1B, R17	Frequency Stop Watch Registers	1.000 MHz 999,947.00 μs
STS TIFR1, R20 LDI R17, 0 OUT PORTB, R17 STS TCCR1B, R17	Frequency Stop Watch Registers	1.000 MHz 999,947.00 μs 0x00

#### Task 5/E before timer interrupt:

```
Processor Status
     while (1)
                                                            Name
                                                                               Value
                                                        Program Counter 0x0000005D
          //wait for interrupts
                                                        Stack Pointer
                                                                         0x08FD
 }
                                                                         0x0000
                                                        X Register
□ISR (INT0_vect)
                                                                         0x08FF
                                                        Y Register
      EIFR |= (1<<INTF0);
                                //reset flag
                                                                         0x0000
                                                        Z Register
     PORTB |= (1<<2);
                               //turn on LED
                                                        Status Register
                                                                         THSVNZC
     TCNT1 = 61630;
                               //set counter value
                                                        Cycle Counter
                                //normal mode
     TCCR1A = 0;
                                                                         1.000 MHz
      TCCR1B = (1 << CS12);
                                //set prescaler to 25
                                                        Frequency
                                                        Stop Watch
                                                                         0.00 µs
                                                       ■ Registers
∃ISR (TIMER1 OVF vect)
                                                         R00
                                                                         0x00
     TIFR1 |= (1<<TOV1);
                                //reset flag
                                                                         0x00
                                                         R01
      PORTB &= ~(1<<2);
                                //turn LED off
                                //turn off timer
      TCCR1B = 0;
                                                         R02
                                                                          0x00
```

#### After interrupt:

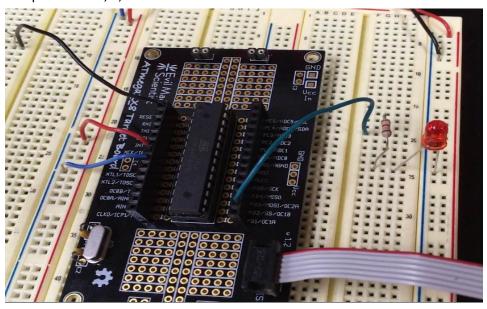
```
Processor Status
      while (1)
                                                             Name
                                                                                Value
                                                         Program Counter
                                                                          0x0000006C
          //wait for interrupts
                                                         Stack Pointer
                                                                          0x08F8
                                                         X Register
                                                                          0x0000
□ISR (INT0_vect)
                                                         Y Register
                                                                          0x08FF
      EIFR |= (1<<INTF0);
                                //reset flag
                                                         Z Register
                                                                          0x0000
      PORTB |= (1<<2);
                                //turn on LED
                                                                          OTHSVNZC
                                                         Status Register
                                //set counter value
      TCNT1 = 61630;
                                                         Cycle Counter
                                                                          999989
      TCCR1A = 0;
                                //normal mode
      TCCR1B = (1<<CS12);
                                //set prescaler to 25
                                                                          1.000 MHz
                                                         Frequency
                                                         Stop Watch
                                                                          999,954.00 µs
                                                        ■ Registers

☐ISR (TIMER1_OVF_vect)

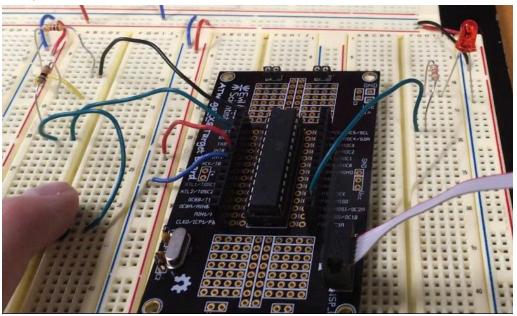
                                                          R00
                                                                          0x00
      TIFR1 |= (1<<TOV1);
                                 //reset flag
                                                          R01
                                                                          0x00
                                //turn LED off
      PORTB &= ~(1<<2);
      TCCR1B = 0;
                                //turn off timer
                                                          R02
                                                                           0x00
```

## 10. SCREENSHOT OF EACH DEMO (BOARD SETUP)

Setup for tasks 1, 3, and 4



Set up for tasks 2 and 5



#### 11. VIDEO LINKS OF EACH DEMO

Tasks 1/3/4 - https://www.youtube.com/watch?v=Q2s7c8BWkkk

Tasks 2/5 - https://www.youtube.com/watch?v=0\_fHTzq70SA

#### 12. GITHUB LINK OF THIS DA

https://github.com/nhanuscin/submit/tree/master/DA2

**Student Academic Misconduct Policy** 

http://studentconduct.unlv.edu/misconduct/policy.html

"This assignment submission is my own, original work".

Nathan Hanuscin