Trace Stewart

CPE301 – SPRING 2018

Design Assignment 2

DO NOT REMOVE THIS PAGE DURING SUBMISSION:

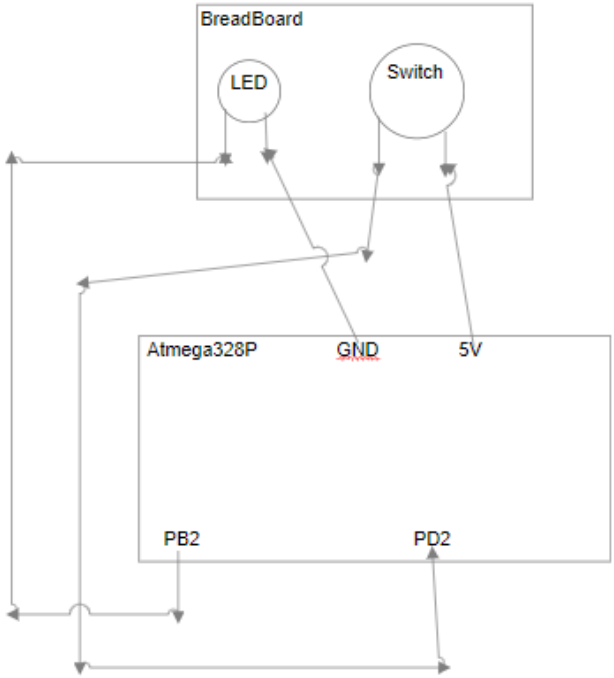
The student understands that all required components should be submitted in complete for grading of this assignment.

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

The components used for this project were the atmega xplained mini, a solderless breadboard, and LED, a push button, and some wires.

Block diagram with pins used in the Atmega328P



2. INITIAL/DEVELOPED CODE OF TASK 1

Assembly Code

```
; DA2_1.asm
;
; Created: 3/6/2018 10:12:34 AM
; Author : trace

start:
        ; constant declaration (value & register)
        .EQU    LOOPCNT = 244 ; loop count for timer
        CLR             R0                              ; R0 = 0

        ; port initialization
        LDI             R16, (1<<2)
        OUT             DDRB, R16               ; Pb.2 output
        OUT             PORTB, R0               ; PORTB = 0
```

```
        ; variable initialization
        LDI     R16, (1<<2)         ; R16 = 0x04: bit 2 = 1
        LDI     R17, LOOPCNT  ; initialize loopCnt (loop count)
        CLR     R1                  ; counter = 0

init:
        OUT     TCNT0, R0                       ; initialize Timer0 to 0
        OUT     TCCR0A, R0                      ; Timer0: normal, internal clk
        LDI     R18, (1<<CS00) | (1<<CS01)      ; Timer0: enabled, 64 prescalar
        OUT     TCCR0B, R18

timerLp:
        IN      R2, TIFR0           ; read TOV0 (overflow)
        SBRS    R2, 0               ; if (TOV0 is set), skip to next instr
        RJMP    timerLp             ; jump back to timerLp
        CLR     R18                 ; stop Timer0
        OUT     TCCR0B, R18
        LDI     R18, (1<<TOV0) ; clear TOV0 flag
        OUT     TIFR0, R18
        DEC     R17                 ; loopCnt--
        BRNE    init

toggle:
        IN      R6, PORTB           ; R6 = PORTB
        EOR     R6, R16             ; toggle bit 0 of R0
        OUT     PORTB, R6           ; toggle Pb.0
        LDI     R17, LOOPCNT  ; reinitialize loopCnt
        JMP     init
```

C Code

```
/* DA2_C.c
 *
 * Created: 3/1/2018 11:19:22 AM
 * Author : trace
*/
#define F_CPU 8000000UL
#include "avr/io.h"
#include <util/delay.h>

int main ()
{
        DDRB = 1<<2;

        while(1)
        {
                _delay_ms(250);
```

```
                PORTB ^= 1<<2;
        }
}
```

3. TASK 2

Assembly Code

```
; DA2_2.asm
;
; Created: 3/6/2018 10:18:53 AM
; Author : trace
;

.org 0x0000
        ldi r16,0b00001111 ; Make the lower 4 bits output
        out ddrb,r16 ; for port b.
        LDI R20,5 ;to set prescaler
        STS TCCR1B,R20 ;Prescaler: 1024
top:
        CBI DDRD, 2
        SBIS PIND, 0 //skip next inst if pind=0
        RJMP top
        sbi portb,2 ; Set bit 0 immediate of port b
        rcall delay ; Calling a subroutine.
        cbi portb,2 ; Clear bit 0 immediate of port b
        rjmp top ; Relative jump to label top

delay:
        LDS R29, TCNT1H ;loading upper bit of counter to R29
        LDS R28, TCNT1L ;loading lower bit of counter to R28
        CPI R28,0xFF ;comparing if lower is 0x84 10,000
        BRSH body
        RJMP delay
body:
        CPI R29,0x3D ;3906
        BRSH done
        RJMP delay
done:
        LDI R20,0x00
        STS TCNT1H,R20 ;resetting the counter to 0 for next round
        LDI R20,0x00
        STS TCNT1L,R20 ;resetting the counter to 0 for next round
        RET
```

C Code

```
/*
* DA2_C.c
 *
 * Created: 3/1/2018 11:19:22 AM
 * Author : trace
 */

#define F_CPU 16000000UL
#include <avr/io.h>
#include <util/delay.h>

int main(void)
{
	DDRB |= (1<<2);
	DDRD &= ~(1<<2);//Makes firs pin of PORTD as Input
	PORTD |= (1<<2); //Set pull up resistor

	while(1) //infinite loop
	{
		if(PIND & (1<<PD0) == 1) //If switch is pressed
		{
			_delay_ms(200);
			PORTB ^= 1<<2; //Toggle Led
			_delay_ms(1000);
			PORTB ^= 1<<2; //Toggle Led
		}
	}
}
```

4. Task 3

Assembly Code

```
; DA2_3.asm
;
; Created: 2/26/2018 1:22:46 PM
; Author : trace
;

;0.5 Second Period with 50% DC
RESET:
	SBI DDRB, 2 ;PB2 is output
	LDI R16, 0	;R16 = 0
	OUT PORTB, R16		;output Port B
	LDI R17, 0x04	;3rd bit = 1
	LDI R18, 15		;loop initialization for .5 sec
```

```
START:
        LDI R19, 0x0            ;R19 = 0
        OUT TCNT0, R19          ;Timer0 = 0
        OUT TCCR0A, R16         ;Timer0 = normal
        LDI R20, (1 << CS00) | (1 << CS02)
        OUT TCCR0B, R20         ;Enable Timer0 and set prescalar = 1024
LOOP:
        IN      R21, TIFR0      ;Check Timer0 flag register
        SBRS R21, 0     ;if overflow, dont jmp
        RJMP LOOP
        LDI R21, 0x0            ;R21 = 0
        OUT TCCR0B, R21         ;Stop Timer0
        LDI R21, (1 << TOV0)
        OUT TIFR0, R21 ;Clear overflow flag
        DEC R18         ; R21 = R21 - 1
        BRNE    START
TOGGLE:
        EOR R16, R17            ;R16 xor R17
        OUT PORTB, R16          ;TOGGLE PB2
        LDI R18, 15     ;Reinitialize loop
        RJMP START
```

C Code

```
/* DA2_C.c
 *
 * Created: 3/1/2018 11:19:22 AM
 * Author : trace
 */

#include <avr/io.h>

int main(void)
{
        unsigned int i;                         // variable for the loop below

        DDRB = (1<<2);  // set PB2
        PORTB = 0;                                              // initialize PORTB to 0


        // initialize timer0 with no prescalar and normal mode
        TCCR0A = 0;
        TCCR0B = 1;
        TCNT0 = 0;              // initialize Timer0 = 0

        while (1){
                for(i=0; i<15624; i++){
                        while((TIFR0 & (1<<0)) == 0) ;     // wait until overflow flag is set
```

```
                TIFR0 |= 1;                                     // clear overflow flag
            }
            PORTB ^= (1<<2);                    // toggle PB2
        }
}
```

5. Task 4

Assembly Code

```
; DA2_4.asm
;
; Created: 3/6/2018 10:45:18 AM
; Author : trace
;

.org 0
        jmp  START
.org 0x20
        jmp  TIM0_OVF               ; Timer0 overflow interrupt vector

START:
        ; Toggle PORTB.2 every .5 second
        SBI     DDRB,2              ;PB.2 as an output
        LDI     R18,0               ;PB.2 = 0
        OUT     PORTB,R18
        LDI     R16,0x04            ;R16 = 0x04: bit 2 = 1
        LDI  R21, 15            ;initialize loop count to 30
Begin:
        LDI     R19, 0x0            ;load Timer0 = 0
        OUT     TCNT0,R19
        OUT     TCCR0A,R18          ;Timer0: normal mode, internal clock
        LDI     R17,(1<<CS00) | (1<<CS02) ;Timer0: enabled, prescalar = 1024
        OUT     TCCR0B, R17

        ;enable interrupts
        LDI     R20, 0x01
        STS  TIMSK0, R20        ;enable overflow interrupt
        SEI                         ;enable global interrupts
Loop:
        RJMP LOOP                   ;infinite loop

;Timer0 overflow ISR
TIM0_OVF:
        LDI     R20,0x0             ;stop/disable Timer0
        OUT     TCCR0B,R20
```

```
        LDI     R20,(1<<TOV0) ;R20 = 0x01
        OUT     TIFR0,R20               ;clear TOV0 flag
        DEC  R21                        ;R21--
        BRNE DONE                       ;repeat if Timer0 hasn't overflowed 15 times

        LDI  R21, 15    ;reinitialize loop count to 15
        EOR     R18,R16                 ;toggle bit 2 of R18
        OUT     PORTB,R18               ;toggle PB.2
DONE:
        LDI     R19, 0          ;load Timer0 = 0
        OUT     TCNT0,R19
        LDI     R17,(1<<CS00) | (1<<CS02) ;Timer0: enabled, prescalar = 1024
        OUT     TCCR0B, R17
        RETI                            ;return from interrupt, interrupts enabled
```

C Code

```
/*
* DA2_C.c
 *
 * Created: 3/1/2018 11:19:22 AM
 * Author : trace
 */

#include "avr/io.h"
#include <avr/interrupt.h>

volatile int ovrflw;        // global variable for keeping track of # of times Timer0 overflows

int main(void) {
        ovrflw= 0;                              // initialize ovrflw to keep track of # of times
        // Timer0 overflows

        // port initialization
        DDRB = (1<<2);  // set PB2 as output
        PORTB = 0;              // initialize PORTB to 0


        // initialize timer0 with starting value of 0, normal mode with no pre scaler
        TCNT0 = 0;
        TCCR0A = 0;
        TCCR0B |= 1;

        // enable interrupts
        TIMSK0 |= (1 << TOIE0);         // enable overflow interrupt
        sei();                                          // enable global interrupts

        while(1) ;                              // loop forever
```

```
}


// this interrupt service routine (ISR) runs whenever an overflow on Timer0 occurs
ISR (TIMER0_OVF_vect) {
        if (ovrflw== 15624) {
                PORTB ^= (1<<2);                // toggle PB2
                ovrflw= 0;                              // reinitialize ovrflw
}
else
ovrflw++;       // increment ovrflw
}
```

6. Task 5

Assembly Code

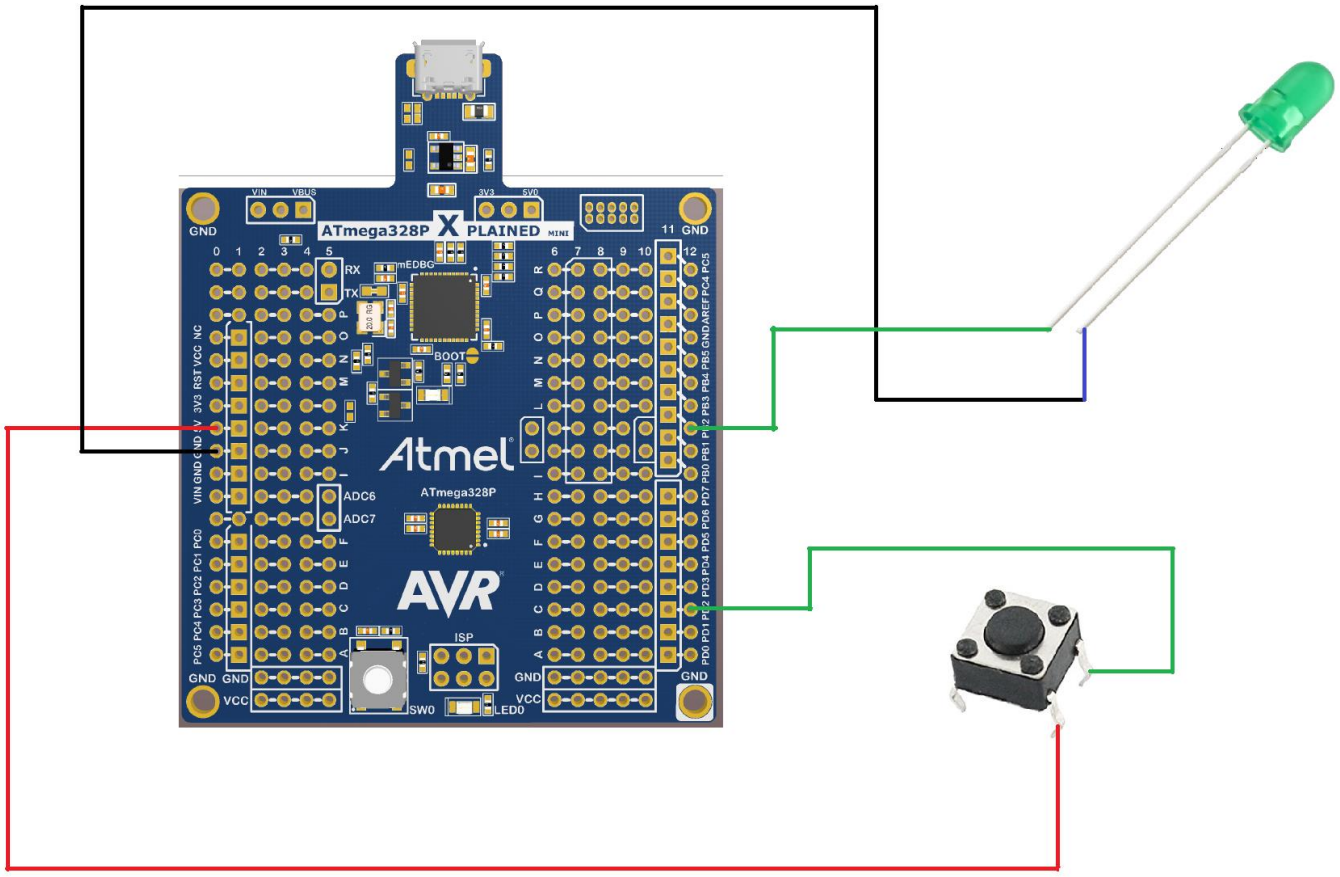
```
;
; DA2_5.asm
;
; Created: 3/10/2018 4:29:23 PM
; Author : trace
;

.ORG 0 ;location for reset
        JMP MAIN
.ORG 0x02 ;location for EXT_INT0
        JMP EX0_ISR

MAIN:
        LDI R20,HIGH(RAMEND)
        OUT SPH,R20
        LDI R20,LOW(RAMEND)
        OUT SPL,R20

        SBI DDRB,2 ;PB2 = output
        SBI PORTD,2 ;pull-up activated
        LDI R20,1<<INT0 ;Enable INT0
        OUT EIMSK,R20
        LDI R20, (1<<ISC00 | 1<<ISC01) ;Fall Edge
        STS EICRA,R20
        SEI ;Set I (Enable Interrupts)

HERE:
        JMP HERE

EX0_ISR:
        LDI R20, 1<<INTF0
```

```
        STS EIFR, R20 ; clear flag
        IN R21,PORTB
        LDI R22,0x04
        EOR R21,R22
        OUT PORTB,R21
RETI
```

C Code

```
/*
* DA2_C.c
 *
 * Created: 3/1/2018 11:19:22 AM
 * Author : trace
 */

#include <avr/io.h>
#include <avr/interrupt.h>

#define F_CPU 16000000UL
#include <util/delay.h>

int main(void)
{
        DDRD = (1<<2); //PD2 Input
        //PORTD |= (1<<2); //pull-up activated
        DDRB = (1<<2); //Makes PB2 output

        EIMSK = 1<<INT0;                                        // Enable INT0
        EICRA = 1<<ISC01 | 1<<ISC00;    // Trigger INT0 on rising edge

        sei();                                  //Enable Global Interrupt

        while(1);

}

//Interrupt Service Routine for INT0
ISR(INT0_vect)
{

                PORTB ^= (1<<PB2); //Toggle PB2
                _delay_ms(1000);
                PORTB ^= (1<<PB2); //Toggle PB2
                EIFR |= (1<<INTF0); // clear the INT0 flag

}
```

7. SCHEMATICS



8. SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)

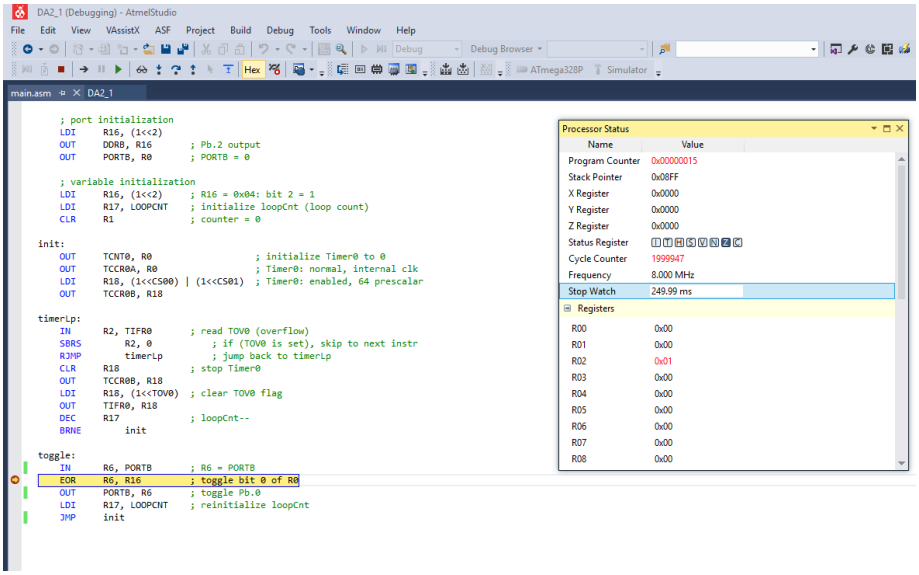


Figure 1 Half a cycle of Task 1 Assembly Code. 0.25 Seconds

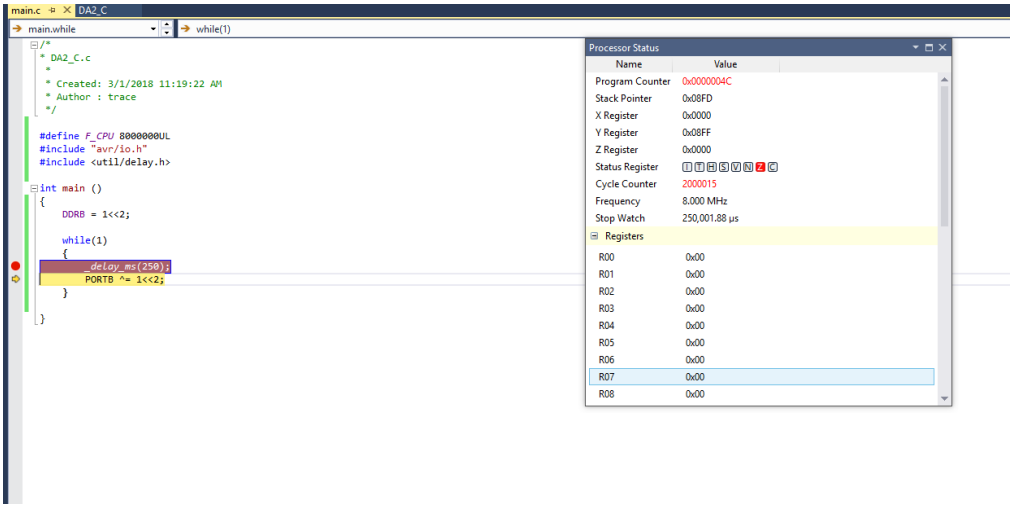


Figure 2: Half a cycle of Task 1 C Code. 0.25 Seconds

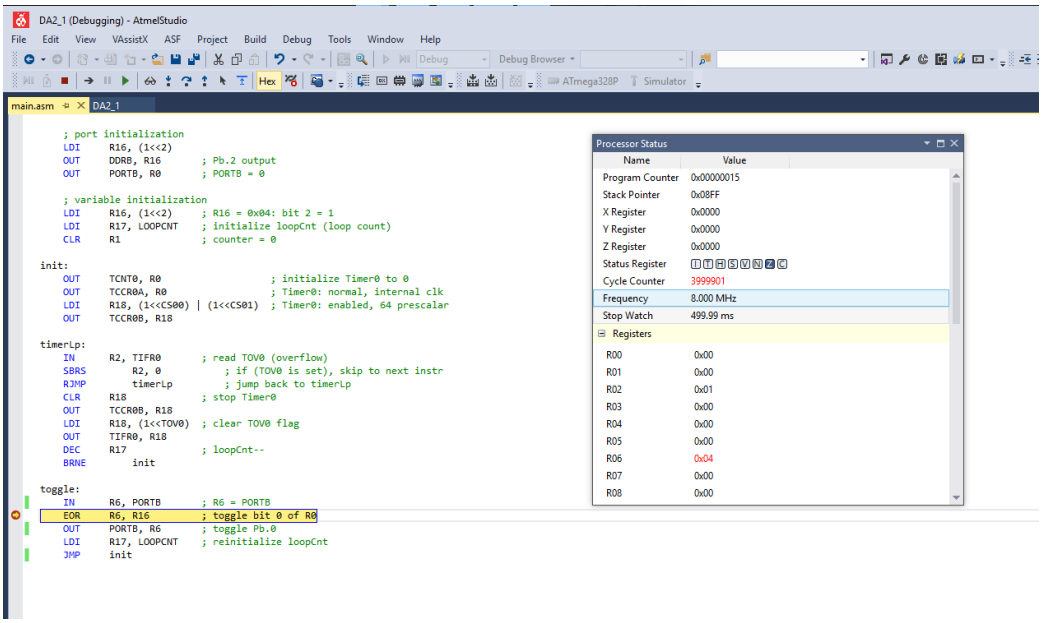


Figure 3: Full cycle of task 1, Assembly Code. 0.5 Seconds

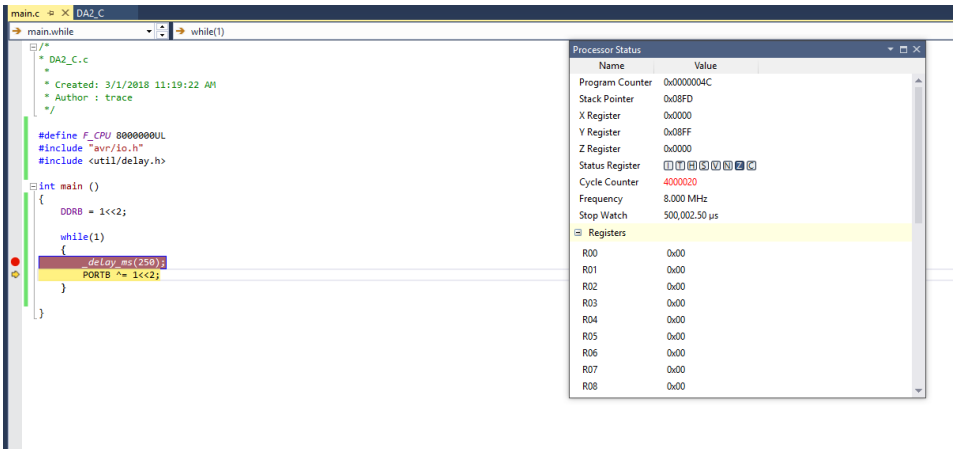


Figure 4: Full cycle of task 1, C Code. 0.5 Seconds

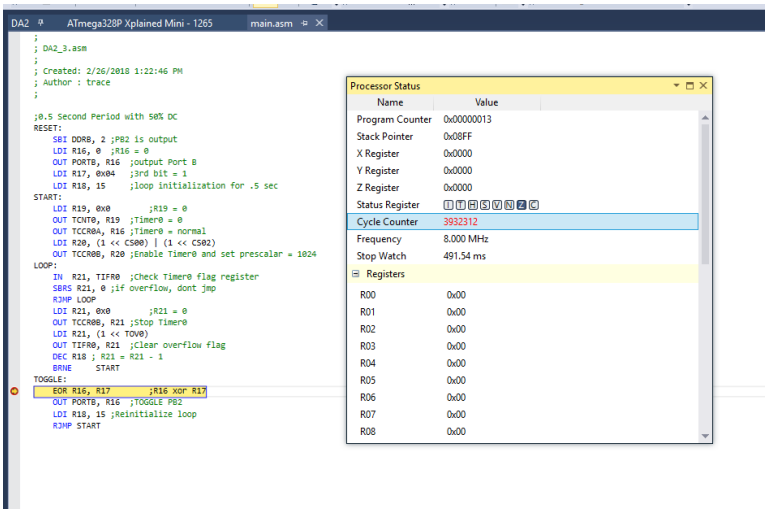


Figure 5: Full Cycle of Task 3 Assembly Code. 0.5 Seconds.

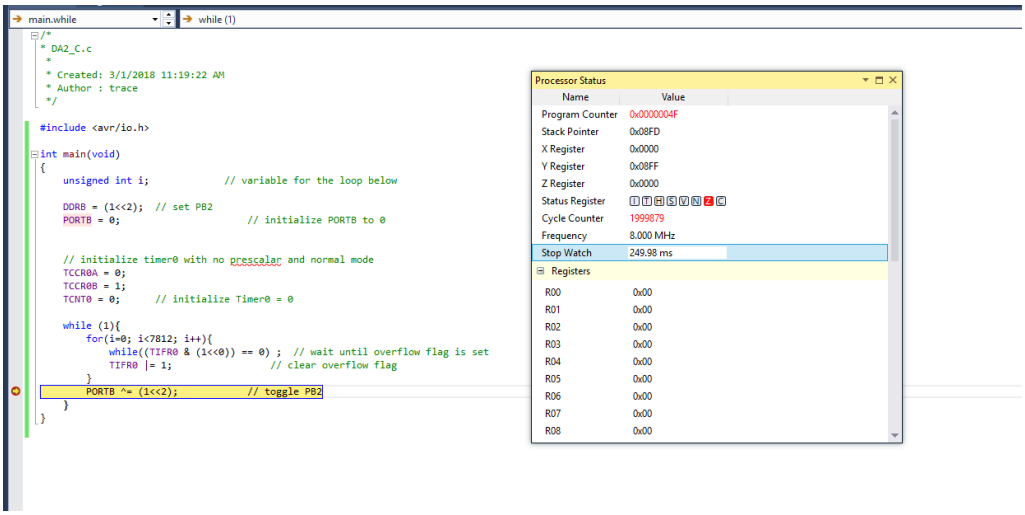


Figure 6: Half Cycle of Task 3 C code. 0.25 Seconds

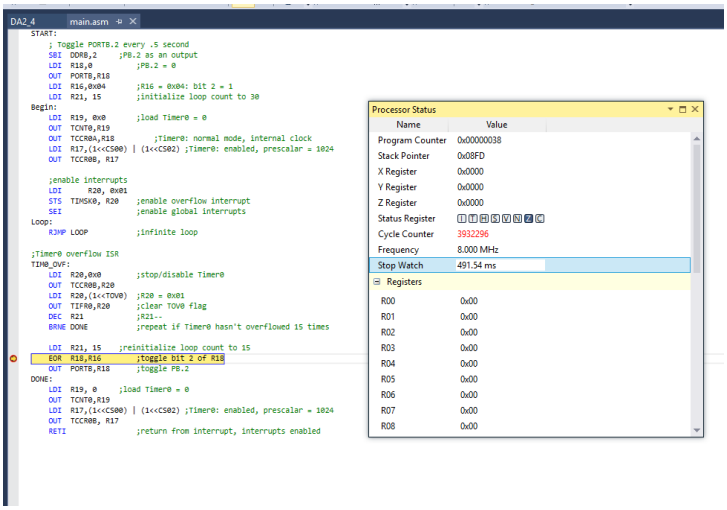


Figure 7: Full Cycle of Task 4 Assembly Code. 0.5 Seconds

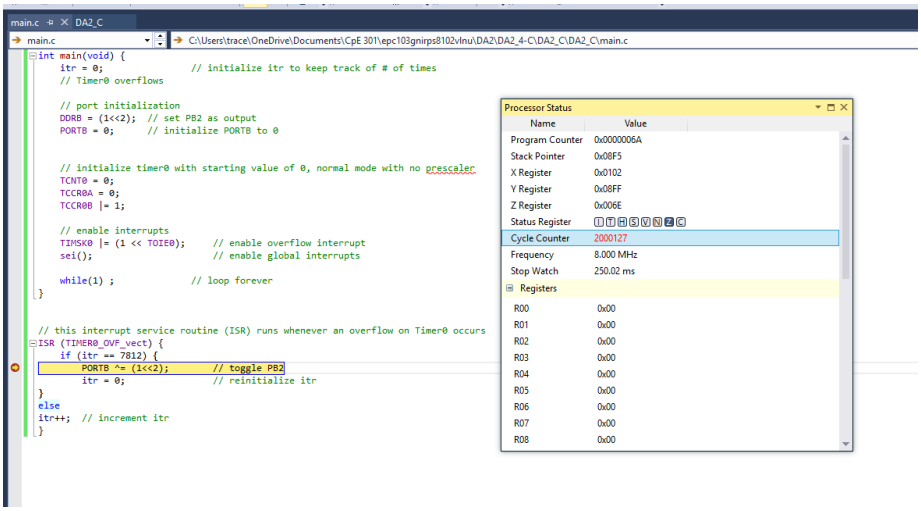


Figure 8: Half cycle of Task 4 C Code. 0.25 Seconds

9. SCREENSHOT OF EACH DEMO (BOARD SETUP)

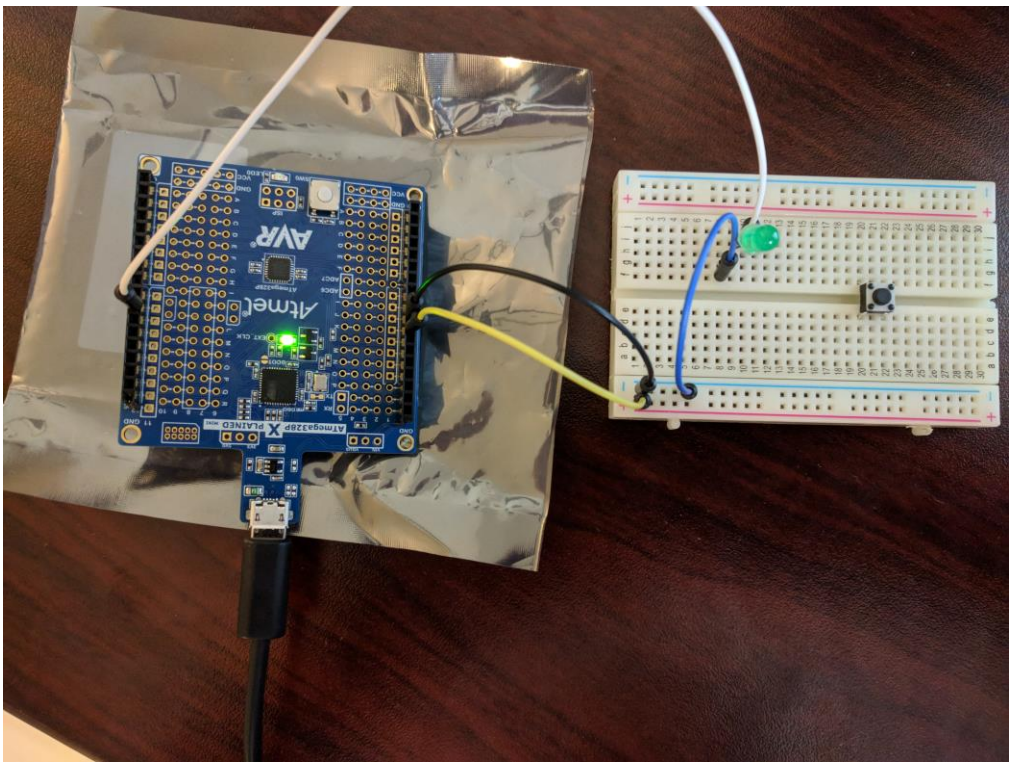


Figure 9: Board Setup for Task 1, Task 3, and Task 4

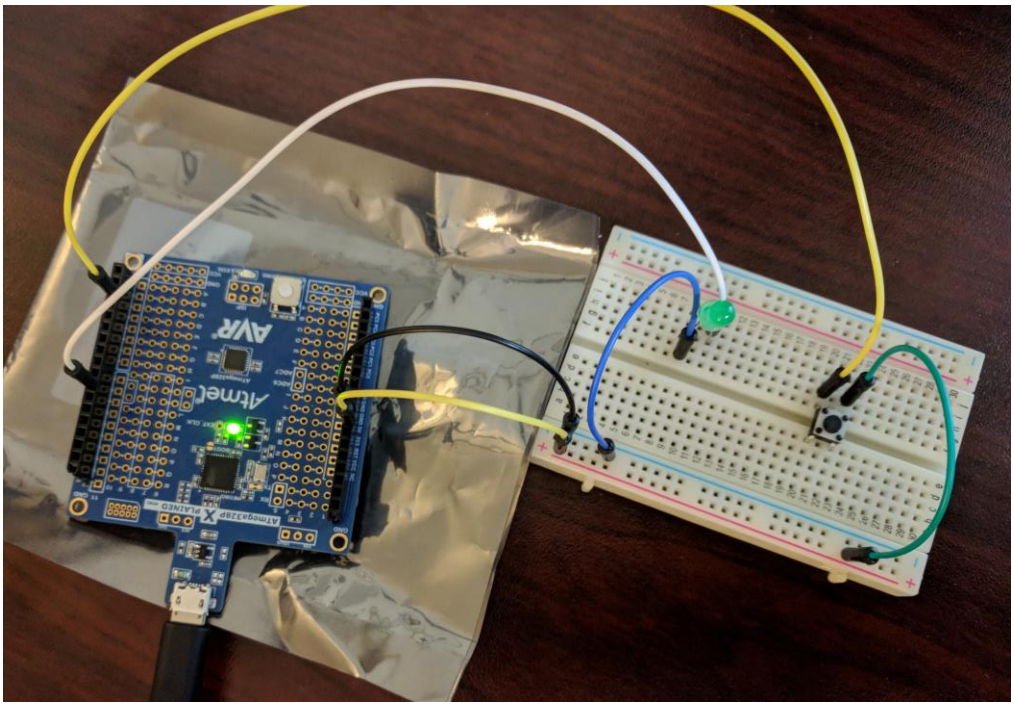


Figure 10: Board setup for Task 2 and Task 5

10. VIDEO LINKS OF EACH DEMO

Task 1 Assembly: https://www.youtube.com/watch?v=DoauVUG7VW8
Task 1 C: https://www.youtube.com/watch?v=ve6Gr7Gz-vQ
Task 2 Assembly: https://www.youtube.com/watch?v=KDIGaYRWjEc
Task 2 C: https://www.youtube.com/watch?v=A-hojAG77uE
Task 3 Assembly: https://www.youtube.com/watch?v=-HLhe0F6zU8
Task 3 C: https://www.youtube.com/watch?v=boD4b2jlvZ8
Task 4 Assembly: https://www.youtube.com/watch?v=FhFOzb6HhTY
Task 4 C: https://www.youtube.com/watch?v=CXvC9ACjI4Q
Task 5 Assembly: https://www.youtube.com/watch?v=-aQgYOmq2vc
Task 5 C: https://www.youtube.com/watch?v=5_HqqGo9MB4

11. GITHUB LINK OF THIS DA

https://github.com/TraceStewart/epc103gnirps8102vlnu/tree/master/DA2

Student Academic Misconduct Policy
http://studentconduct.unlv.edu/misconduct/policy.html

"*This assignment submission is my own, original work*".
Trace Stewart