CPE301 – SPRING 2019

Design Assignment 2A

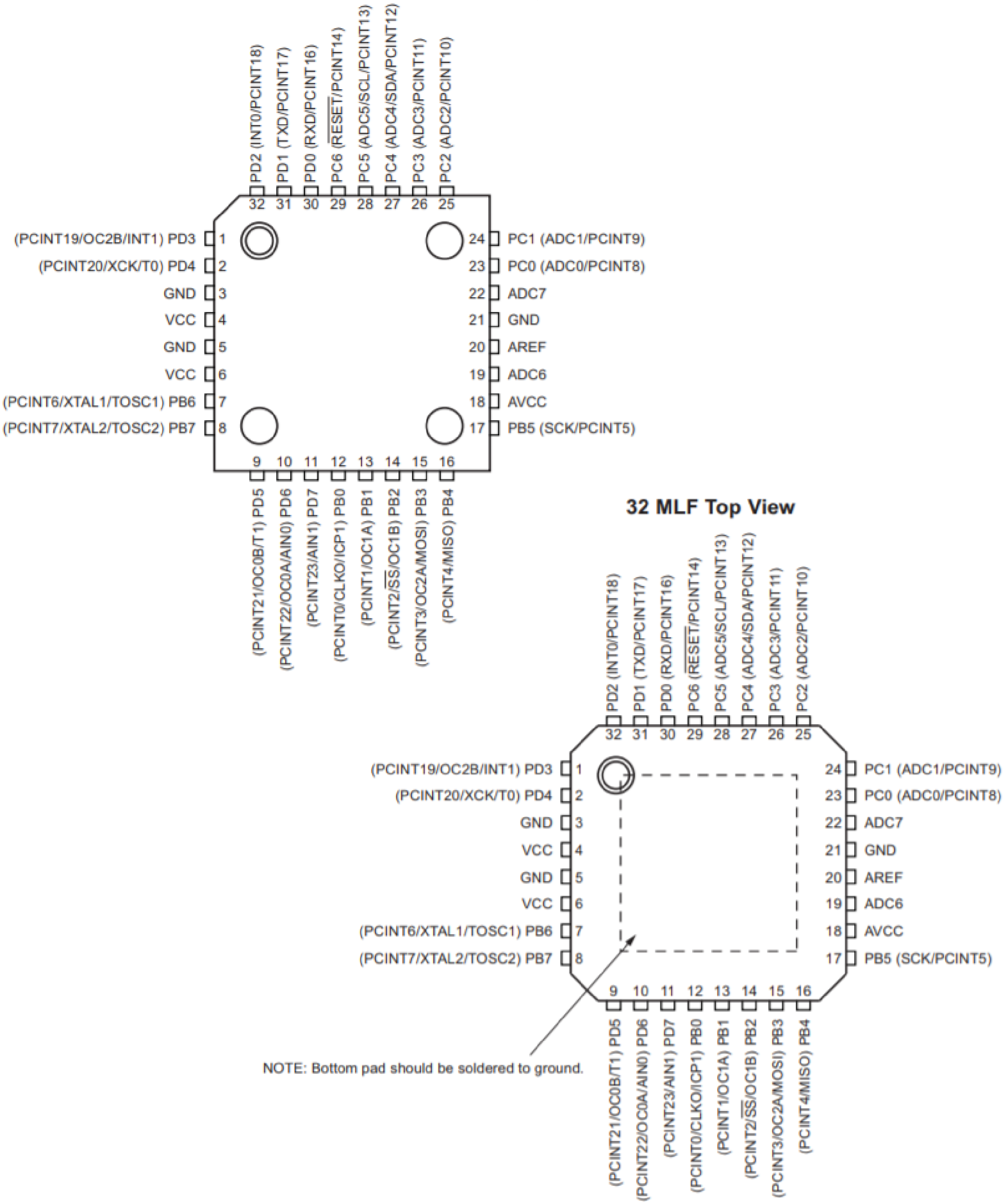
Student Name: Tyler Gardenhire
Student #: 8000450294
Student Email: gardenhi@unlv.nevada.edu
Primary Github address: gardenhi@unlv.nevada.edu
Directory: https://github.com/tylergardenhire/submission_projects.git

Submit the following for all Labs:

1. In the document, for each task submit the modified or included code (only) with highlights and justifications of the modifications. Also, include the comments.
2. Use the previously create a Github repository with a random name (no CPE/301, Lastname, Firstname). Place all labs under the root folder ESD301/DA, sub-folder named LABXX, with one document and one video link file for each lab, place modified asm/c files named as LabXX-TYY.asm/c.
3. If multiple asm/c files or other libraries are used, create a folder LabXX-TYY and place these files inside the folder.
4. The folder should have a) Word document (see template), b) source code file(s) and other include files, c) text file with youtube video links (see template).

1. COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS

Atmel Studio 7 w/ AVR assembly and simulator and Atmega328p board used.



PortB2 is connected to LED3 and PortC2 is connected to SW1.

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

Task 1 Assembly Code:

```
.org 0
sbi ddrb, 2           ;set portb2 as output
ldi r20, 5
sts tccr1b, r20       ;set clock prescaler to 1024

begin: sbi portb, 2  ;set portb2 to high
       ldi r21, 0x00
       sts tcnt1h, r21 ;set high bits of timer to 0
       sts tcnt1l, r21 ;set low bits of timer to 0
       rcall delayon ;call first delay function
       cbi portb, 2  ;clear portb2
       ldi r21, 0x00
       sts tcnt1h, r21      ;set high bits of timer to 0
       sts tcnt1l, r21 ;set low bits of timer to 0
       rcall delayoff       ;call second delay function
       rjmp begin           ;infinite loop


delayon:
       lds r29, tcnt1h      ;load high bits of timer1
       lds r28, tcnt1l ;load low bits of timer1
       cpi r28, 0xa9 ;compare low bits of timer1 to 0x01a9 (representing 0.435 seconds)
       brsh bodyon
       rjmp delayon
bodyon:
       cpi r29, 0x01 ;compare high bits of timer1 to 0x01a9
       brlt delayon
       ret

delayoff:
       lds r29, tcnt1h      ;load high bits of timer1
       lds r28, tcnt1l      ;load low bits of timer1
       cpi r28, 0x1b ;compare low bits of timer1 to 0x011b (representing 0.290 seconds)
       brsh bodyoff
       rjmp delayoff
bodyoff:
       cpi r29, 0x01 ;compare high bits of timer1 to 0x01a9
       brlt delayoff
       ret
```

Task 1 C Code:

```
#include <avr/io.h>                     //standard avr library
#define F_CPU 1000000UL                 //set clock speed in MHz
#include <util/delay.h>                 //delay library for above clock speed


int main(void)
{

       DDRB |= (1 << PB2);              //define portb2 as an output to blink the LED

       while(1)                                //infinite loop to see the correct output
on the board
```

```
       {
              PORTB |= (1 << PB2); //turn led ON
              _delay_ms(435);                            //delay for 435ms
              PORTB &= ~(1 << PB2);       //turn led OFF
              _delay_ms(290);                            //delay for 290ms
       }
```

3. DEVELOPED MODIFIED CODE OF TASK 2/A from TASK 1/A

Task 2 Assembly Code:

```
.org 0
cbi ddrc, 2                        ;set portc2 as input (switch)
sbi ddrb, 2                        ;set portb2 as output (led)
ldi r20, 5
sts tccr1b, r20                    ;set clock prescaler to 1024

again: sbic pinc, 2         ;check if portc2 is receiving input from switch
       rjmp over                   ;if so, jump to over
       cbi portb, 2         ;if not, clear portb2
       rjmp again                  ;infinite loop
over: sbi portb, 2          ;set portb2 to high
       ldi r21, 0x00
       sts tcnt1h, r21             ;set high bits of timer1 to 0
       sts tcnt1l, r21             ;set low bits of timer1 to 0
       rjmp delayon         ;jump to delay (1.25 second delay)
       rjmp again                  ;infinte loop




delayon:
       lds r29, tcnt1h             ;load high bits of timer1
       lds r28, tcnt1l             ;load low bits of timer1
       cpi r28, 0xc5        ;compare low bits of timer1 to 0x04c5 (representing 1.25
seconds)
       brsh bodyon
       rjmp delayon
bodyon:
       cpi r29, 0x04        ;compare high bits of timer1 to 0x04c5
       brlt delayon
       ret
```

Task 2 C Code:

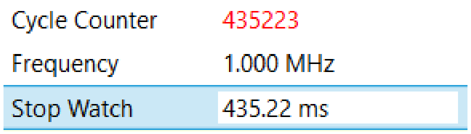
```
#include <avr/io.h>                       //standard avr library
#define F_CPU 1000000UL                   //set clock speed in MHz
#include <util/delay.h>                   //delay library for above clock speed


int main(void)
{

       DDRB |= (1 << PB2);                //define portb2 as an output to blink the LED
       DDRC &= (0 << PB2);                //define portc2 as an input (switch)
       PORTC |= (1 << PB2);        //enable pull-up

       while(1)                                  //infinite loop
       {
        if (PINC)                           //if there is input at portc2
        {
            PORTB |= (1 << PB2);//enable portb2
            _delay_ms(1250);    //delay for 1.25 seconds
        }
        else
        PORTB &= ~(1 << PB2);       //else clear portb2
    }

    return 1;
}
```

4. SCHEMATICS

N/A

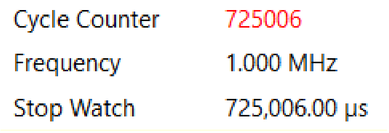
5. SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)

Task 1:
Assembly:
725 ms * 60% duty cycle = 435 ms, the following screenshot was taken at end of first delay call.



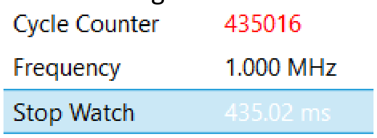
Cycle Counter	435223
Frequency	1.000 MHz
Stop Watch	435.22 ms

Full period is 725 ms, the following screenshot was taken at the end of one loop.



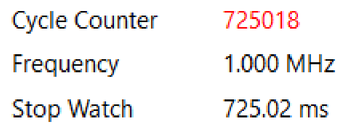
Cycle Counter	725006
Frequency	1.000 MHz
Stop Watch	725,006.00 µs

C:
The following screenshot was taken at end of first delay call.



Cycle Counter	435016
Frequency	1.000 MHz
Stop Watch	435.02 ms

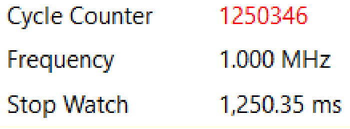
Full period is 725 ms, the following screenshot was taken at the end of one loop.



Cycle Counter	725018
Frequency	1.000 MHz
Stop Watch	725.02 ms

Task 2:
Assembly:

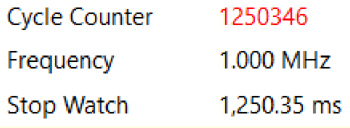
Full period is 1.25 s, the following screenshot was taken at the end of one loop.



Cycle Counter	1250346
Frequency	1.000 MHz
Stop Watch	1,250.35 ms

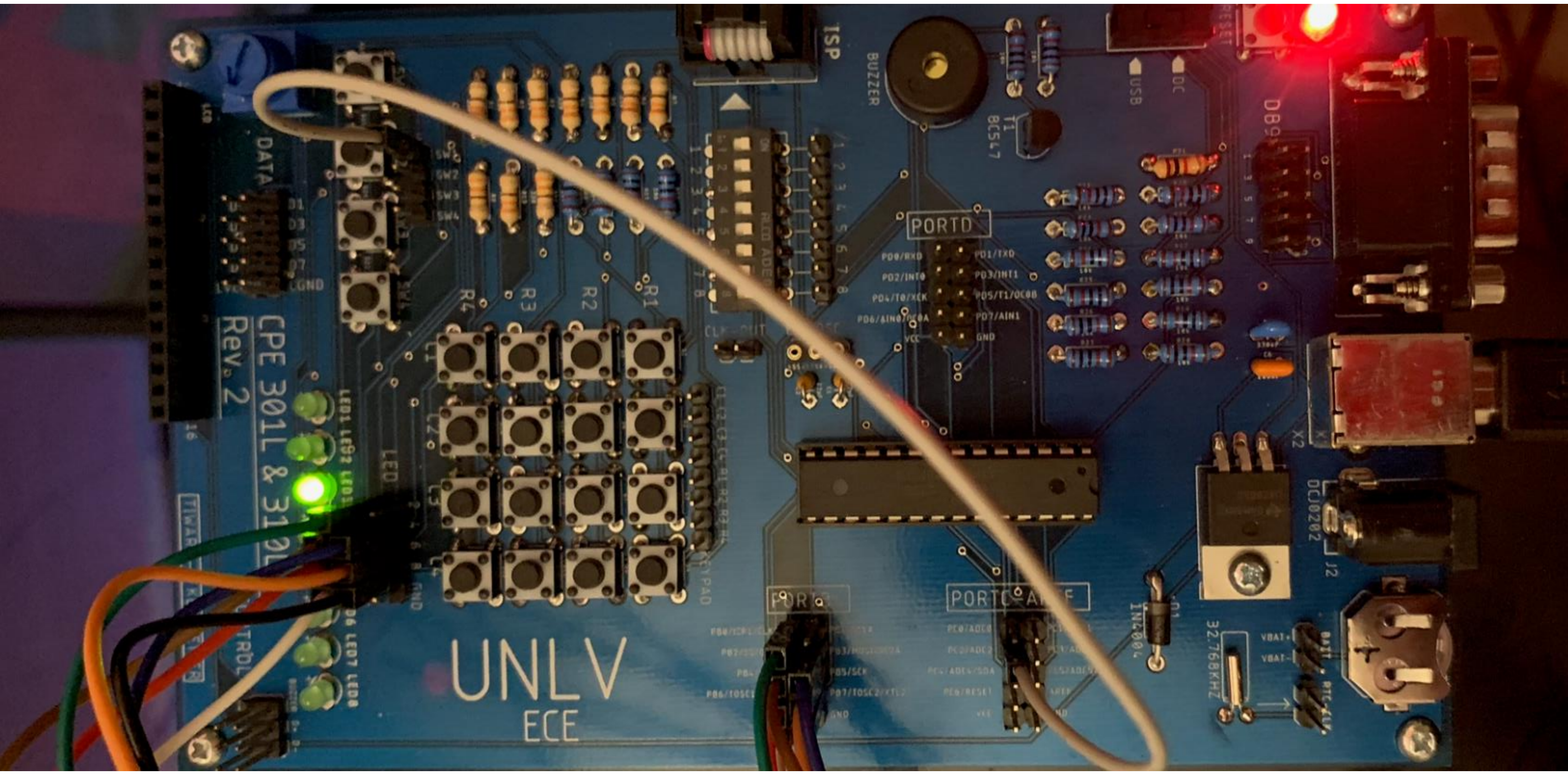
C:

Full period is 1.25 s, the following screenshot was taken at the end of one loop.



Cycle Counter	1250346
Frequency	1.000 MHz
Stop Watch	1,250.35 ms

6. SCREENSHOT OF EACH DEMO (BOARD SETUP)



7. VIDEO LINKS OF EACH DEMO

https://youtu.be/PYId6ZwDTrc

8. GITHUB LINK OF THIS DA

https://github.com/tylergardenhire/submission_projects.git

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

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