CPE301 – SPRING 2019

Design Assignment 2A

Student Name: Ron Joshua Recrio
Student #: 5003825419
Student Email: recrio@unlv.nevada.edu
Primary Github address: https://github.com/recrio/submissions
Directory: /DesignAssignments/DA2A

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

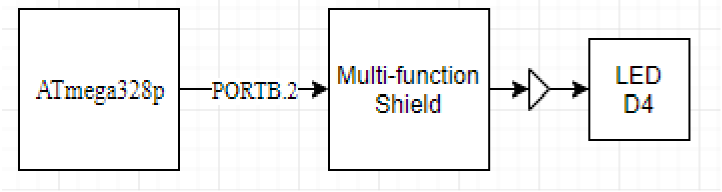
List of Components used:

ATmega328p Xplained Mini

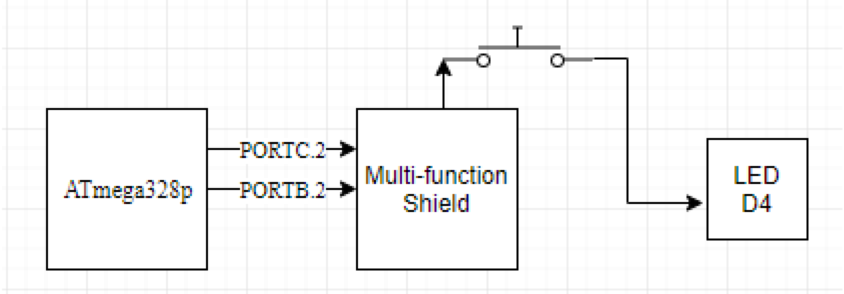
Multifunction Shield

Block diagram with pins used in the Atmega328P

Task 1:



Task 2:



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

```
.ORG 0x0000
init:
       LDI R16, 4                  ; 0b000100 or PB2
       OUT DDRB, R16               ; PB2 set as an output
       LDI R17, 0                  ; Used for toggling PB2
       LDI R20, 5                  ; 0b101 for 1024 prescaler
       STS TCCR1B, R20             ; prescaler set
       LDI R20, 0                  ; to be used for TCNT1 reset

main:
       RCALL TCNT1_RESET           ; reset TCNT1

       RCALL delay_ON              ; call for .435 second delay
       EOR R17, R16                ; XOR toggles the LED
       OUT PORTB, R17              ; should turn off the LED

       RCALL TCNT1_RESET           ; reset TCNT1

       RCALL delay_OFF             ; call for .290 second delay
       EOR R17, R16                ; toggle LED again
       OUT PORTB, R17              ; should turn on the LED

       RJMP main                   ; repeat

TCNT1_RESET:
       STS TCNT1H, R20             ; timer/counter
       STS TCNT1L, R20             ; reset
       RET

check_high_ON:
       CPI R27, 0x1A               ; check if high byte of timer matches 0x1A
       BRLT delay_ON               ; if not, recheck timer
       RET                         ; return to main

delay_OFF:                         ; TCNT should be 4531 or 0x11B3
       LDS R27, TCNT1H             ; R27 = high byte of timer/counter
       LDS R26, TCNT1L             ; R26 = low byte of timer/counter

       CPI R26, 0xB3               ; check if low byte of timer matches 0x8C
       BRSH check_high_OFF         ; if low bytes match, check high byte next

       RJMP delay_OFF              ; repeat until matching

check_high_OFF:
       CPI R27, 0x11               ; check if high byte of timer matches 0x1A
       BRLT delay_OFF              ; if not, recheck timer
       RET                         ; return to main

delay_ON:                          ; TCNT should be 6796 or 0x1A8C
       LDS R27, TCNT1H             ; R27 = high byte of timer/counter
       LDS R26, TCNT1L             ; R26 = low byte of timer/counter

       CPI R26, 0x8C               ; check if low byte of timer matches 0x8C
       BRSH check_high_ON          ; if low bytes match, check high byte next

       RJMP delay_ON               ; repeat until matching
```

3. VERIFICATION OF TASK 1 USING C

```
#include <stdio.h>
#include <avr/io.h>

int main(void)
{
    DDRB = (1<<2); //Make PB2 Output
       TCCR1B = 5;   // set prescaler to 1024

    while (1)
    {
              TCNT1 = 0; // set timer/counter to 0
              while (TCNT1 != 6796) {
              //do nothing
              }
              PORTB ^= (1<<2); // toggle PB2 using xor
              TCNT1 = 0; // reset again
              while (TCNT1 != 4531) {
              // do nothing
              }
              PORTB ^= (1<<2); // toggle PB2 using xor
    }
}
```

4. DEVELOPED CODE OF TASK 2/A

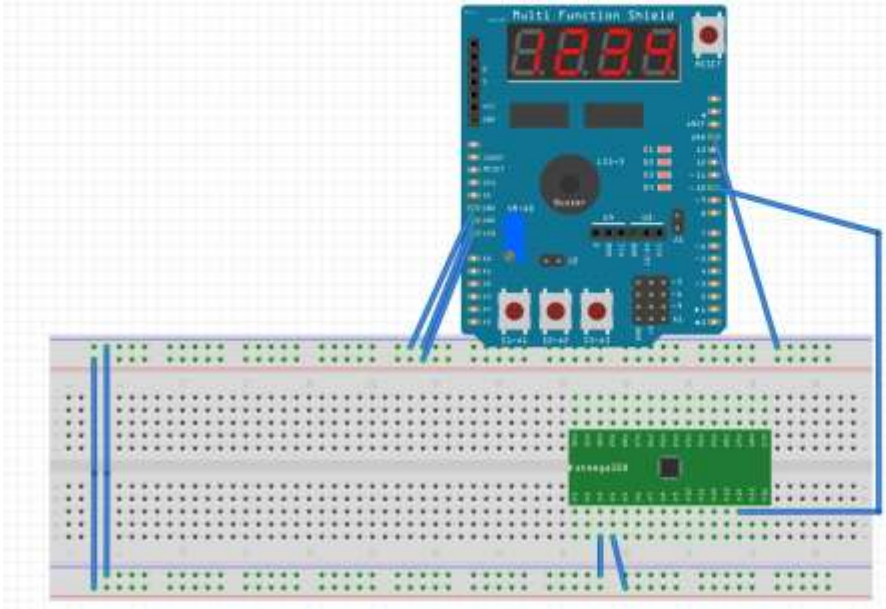
```
.ORG 0
init:
 CBI DDRC, 2         ; PC2 as input, Clears DDRC2
 LDI R16, 4          ; PB2
 OUT DDRB, R16       ; SET PB2 as output
 LDI R17,0           ; toggle for PC2
 LDI R20,5           ; 0b101 for prescaler 1024
 STS TCCR1B, R20     ; Set prescaler to 1024
 LDI R20, 0          ; For counter/timer reset
 EOR R17, R16        ; Toggle with XOR so that it starts as OFF

main:
 OUT PORTB, R17      ; Turn off LED
 SBIC PINC, 2        ; Skip if PC2 is pressed
 RJMP keep_checking  ; if not then jump

 EOR R17, R16        ; Toggle again
 OUT PORTB, R17      ; Turn on LED
 EOR R17, R16        ; Toggle back
 STS TCNT1H, R20     ; Reset
 STS TCNT1L, R20     ; Timer/Counter

 RCALL delay         ; call delay of 1.25s

 RJMP main           ; Loop again

keep_checking:
 SBI PORTB, 2        ; PB2 = 1, set it high so it will not light up
 RJMP main           ; Loop back to check for button press

delay:               ; TCNT1 should be 19,531 or 0x4C4B
 LDS R27, TCNT1H     ; high byte of timer/counter
 LDS R26, TCNT1L     ; low byte of timer/counter
 CPI R26,0x4B        ; check if low byte matches

 BRSH check_high     ; if matching, then check high byte

 RJMP delay          ; else keep checking

check_high:
 CPI R27,0x4C        ; check high byte
 BRLT delay          ; if not matching , keep checking
 RET                 ; if timer/counter is finally matching, return
```

5. VERIFICATION USING C OF TASK 2

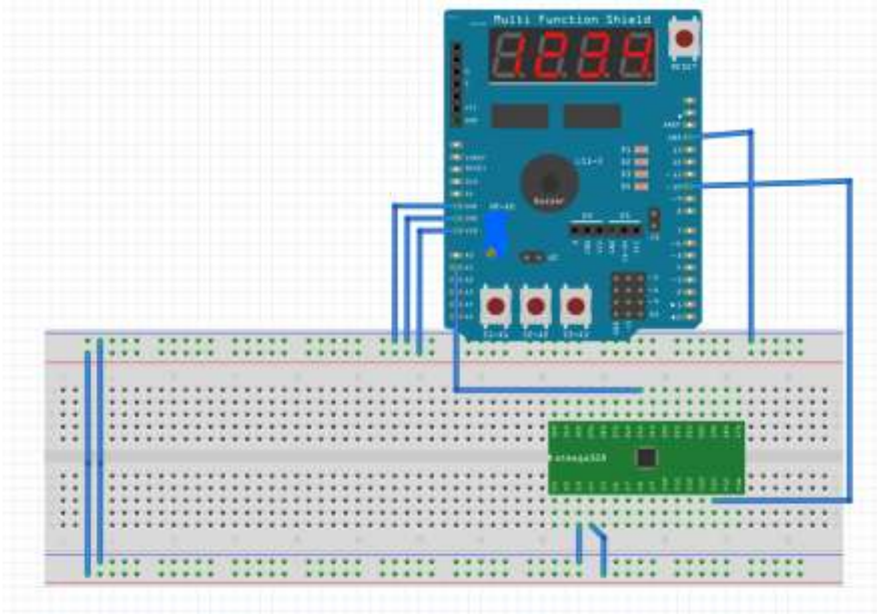
```
#define F_CPU 16000000UL //Change frequency to 16MHz
#include <avr/io.h>
#include <util/delay.h>

int main(void)
{
       DDRB |= (1<<2);      //PB2 is an output, XXXX X1XX
       PORTB |= (1<<2);     //PB2 set as high to turn off LED, XXXX X1XX
       DDRC &= (0<<2);      //PC2 is an input, XXXX X0XX
       PORTC |= (0<<2);     //PC2 set as low or unpressed, XXXX X0XX
       while (1) {
              if (!(PINC & (1 << PINC2))){ // If button is pressed
              PORTB &= ~(1<<2);    // turn on LED
              _delay_ms(1250);     // delay for 1250ms
              }
           else {                  // if button is not pressed
                  PORTB |= (1<<2); // set PB2 to high or LED off
              }
    }
}
```

6. SCHEMATICS

Task 1:

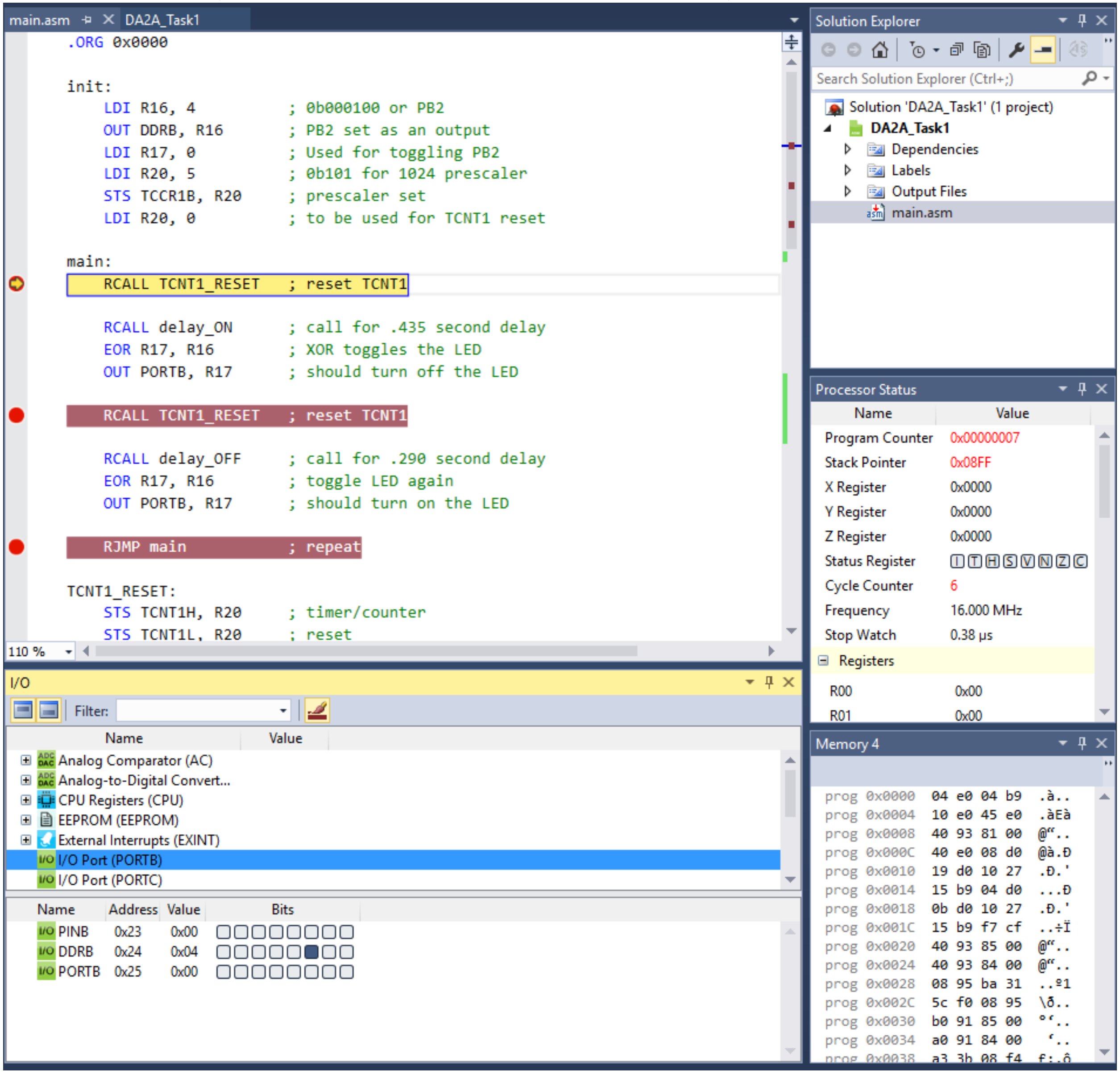


Task 2:



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

TASK 1: Initial State (Note that PORTB2 as 0 means LED is turned ON)



```
    .ORG 0x0000

    init:
        LDI R16, 4          ; 0b000100 or PB2
        OUT DDRB, R16       ; PB2 set as an output
        LDI R17, 0          ; Used for toggling PB2
        LDI R20, 5          ; 0b101 for 1024 prescaler
        STS TCCR1B, R20     ; prescaler set
        LDI R20, 0          ; to be used for TCNT1 reset

    main:
        RCALL TCNT1_RESET   ; reset TCNT1

        RCALL delay_ON      ; call for .435 second delay
        EOR R17, R16        ; XOR toggles the LED
        OUT PORTB, R17      ; should turn off the LED

        RCALL TCNT1_RESET   ; reset TCNT1

        RCALL delay_OFF     ; call for .290 second delay
        EOR R17, R16        ; toggle LED again
        OUT PORTB, R17      ; should turn on the LED

        RJMP main           ; repeat

    TCNT1_RESET:
        STS TCNT1H, R20     ; timer/counter
        STS TCNT1L, R20     ; reset
```

Processor Status

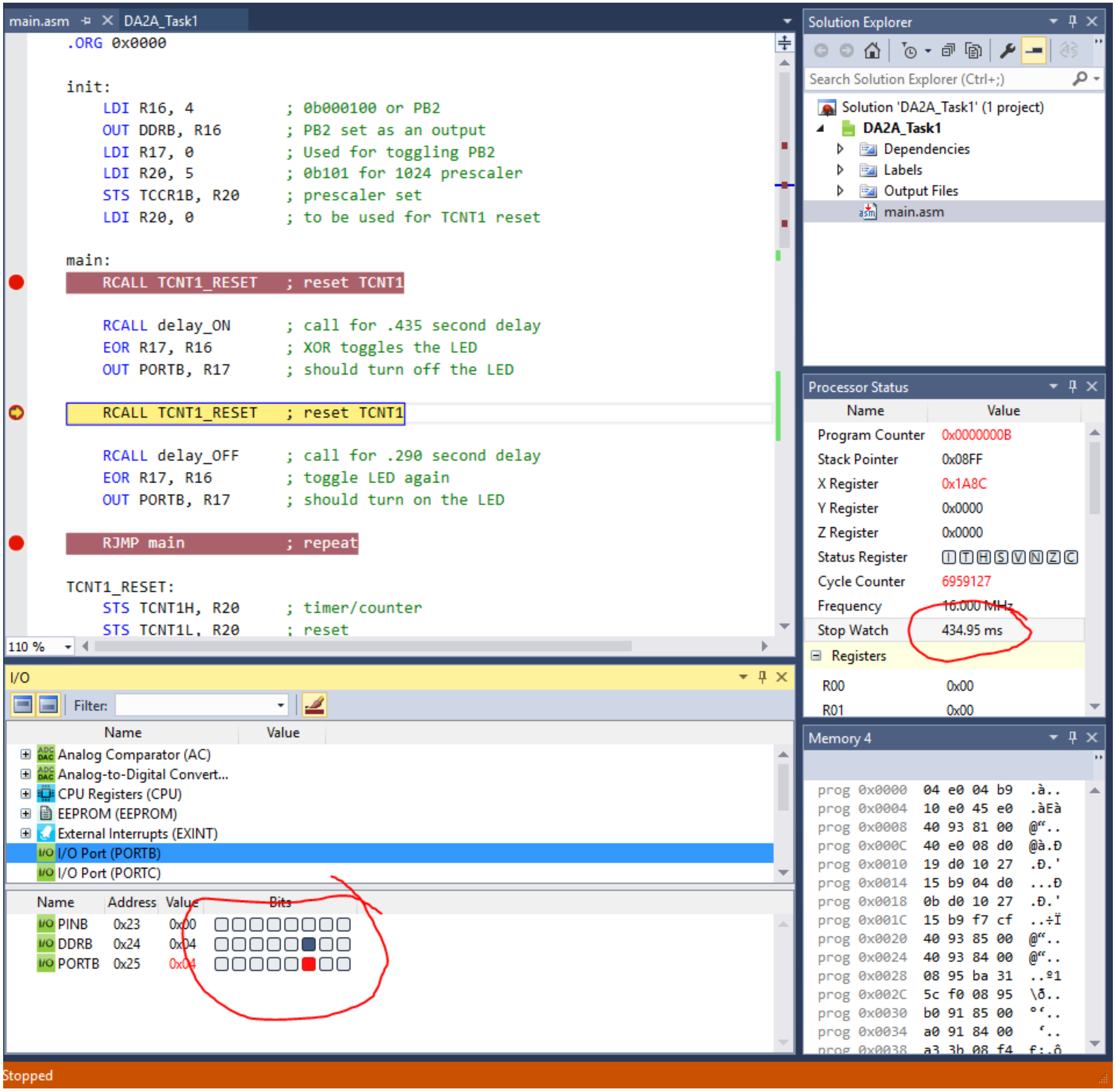
Name	Value
Program Counter	0x00000007
Stack Pointer	0x08FF
X Register	0x0000
Y Register	0x0000
Z Register	0x0000
Status Register	I T H S V N Z C
Cycle Counter	6
Frequency	16.000 MHz
Stop Watch	0.38 µs
⊟ Registers	
R00	0x00
R01	0x00

Name	Value
⊞ Analog Comparator (AC)	
⊞ Analog-to-Digital Convert...	
⊞ CPU Registers (CPU)	
⊞ EEPROM (EEPROM)	
⊞ External Interrupts (EXINT)	
I/O Port (PORTB)	
I/O Port (PORTC)	

Name	Address	Value	Bits
PINB	0x23	0x00	□□□□□□□□
DDRB	0x24	0x04	□□□□□■□□
PORTB	0x25	0x00	□□□□□□□□

```
prog 0x0000  04 e0 04 b9  .à..
prog 0x0004  10 e0 45 e0  .àEà
prog 0x0008  40 93 81 00  @“..
prog 0x000C  40 e0 08 d0  @à.Ð
prog 0x0010  19 d0 10 27  .Ð.'
prog 0x0014  15 b9 04 d0  ...Ð
prog 0x0018  0b d0 10 27  .Ð.'
prog 0x001C  15 b9 f7 cf  ..÷Ï
prog 0x0020  40 93 85 00  @“..
prog 0x0024  40 93 84 00  @“..
prog 0x0028  08 95 ba 31  ..º1
prog 0x002C  5c f0 08 95  \ð..
prog 0x0030  b0 91 85 00  °‘..
prog 0x0034  a0 91 84 00   ‘..
prog 0x0038  a3 3b 08 f4  £;.ô
```

TASK 1: After .435 Sec (PORTB2 is 1 which means LED is OFF)



```
.ORG 0x0000

init:
    LDI R16, 4          ; 0b000100 or PB2
    OUT DDRB, R16       ; PB2 set as an output
    LDI R17, 0          ; Used for toggling PB2
    LDI R20, 5          ; 0b101 for 1024 prescaler
    STS TCCR1B, R20     ; prescaler set
    LDI R20, 0          ; to be used for TCNT1 reset

main:
    RCALL TCNT1_RESET   ; reset TCNT1

    RCALL delay_ON      ; call for .435 second delay
    EOR R17, R16        ; XOR toggles the LED
    OUT PORTB, R17      ; should turn off the LED

    RCALL TCNT1_RESET   ; reset TCNT1

    RCALL delay_OFF     ; call for .290 second delay
    EOR R17, R16        ; toggle LED again
    OUT PORTB, R17      ; should turn on the LED

    RJMP main           ; repeat

TCNT1_RESET:
    STS TCNT1H, R20     ; timer/counter
    STS TCNT1L, R20     ; reset
```

Processor Status

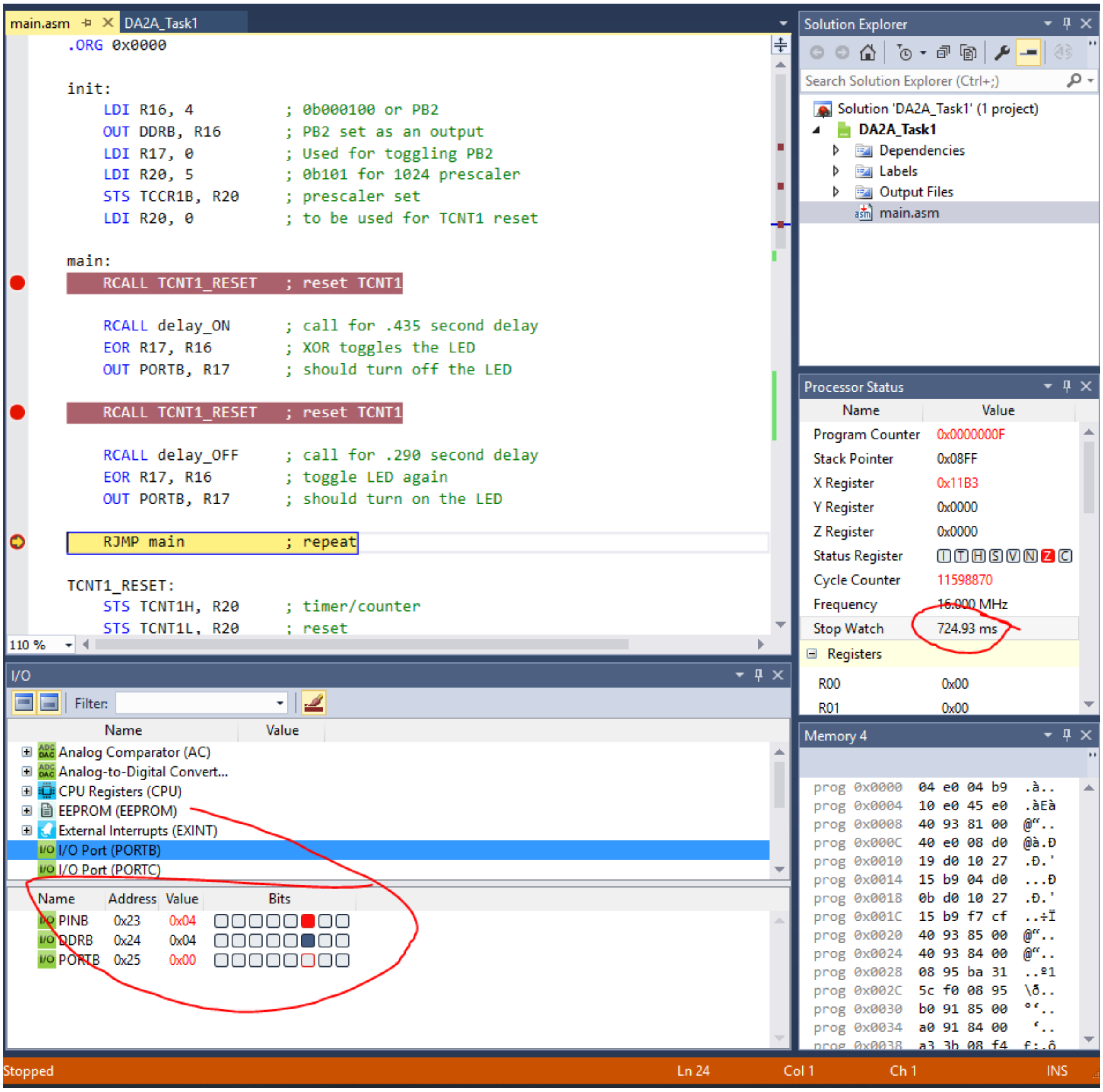
Name	Value
Program Counter	0x0000000B
Stack Pointer	0x08FF
X Register	0x1A8C
Y Register	0x0000
Z Register	0x0000
Status Register	I T H S V N Z C
Cycle Counter	6959127
Frequency	16.000 MHz
Stop Watch	434.95 ms
Registers	
R00	0x00
R01	0x00

Name	Address	Value	Bits
PINB	0x23	0x00	□□□□□□□□
DDRB	0x24	0x04	□□□□□■□□
PORTB	0x25	0x04	□□□□□■□□

Memory 4

```
prog 0x0000  04 e0 04 b9  .à..
prog 0x0004  10 e0 45 e0  .àEà
prog 0x0008  40 93 81 00  @“..
prog 0x000C  40 e0 08 d0  @à.Ð
prog 0x0010  19 d0 10 27  .Ð.'
prog 0x0014  15 b9 04 d0  ...Ð
prog 0x0018  0b d0 10 27  .Ð.'
prog 0x001C  15 b9 f7 cf  ..÷Ï
prog 0x0020  40 93 85 00  @“..
prog 0x0024  40 93 84 00  @“..
prog 0x0028  08 95 ba 31  ..º1
prog 0x002C  5c f0 08 95  \ð..
prog 0x0030  b0 91 85 00  °‘..
prog 0x0034  a0 91 84 00   ‘..
prog 0x0038  a3 3b 08 f4  £;.ô
```

TASK 1: After .290 Sec



```
.ORG 0x0000

init:
    LDI R16, 4          ; 0b000100 or PB2
    OUT DDRB, R16       ; PB2 set as an output
    LDI R17, 0          ; Used for toggling PB2
    LDI R20, 5          ; 0b101 for 1024 prescaler
    STS TCCR1B, R20     ; prescaler set
    LDI R20, 0          ; to be used for TCNT1 reset

main:
    RCALL TCNT1_RESET   ; reset TCNT1

    RCALL delay_ON      ; call for .435 second delay
    EOR R17, R16        ; XOR toggles the LED
    OUT PORTB, R17      ; should turn off the LED

    RCALL TCNT1_RESET   ; reset TCNT1

    RCALL delay_OFF     ; call for .290 second delay
    EOR R17, R16        ; toggle LED again
    OUT PORTB, R17      ; should turn on the LED

    RJMP main           ; repeat

TCNT1_RESET:
    STS TCNT1H, R20     ; timer/counter
    STS TCNT1L, R20     ; reset
```

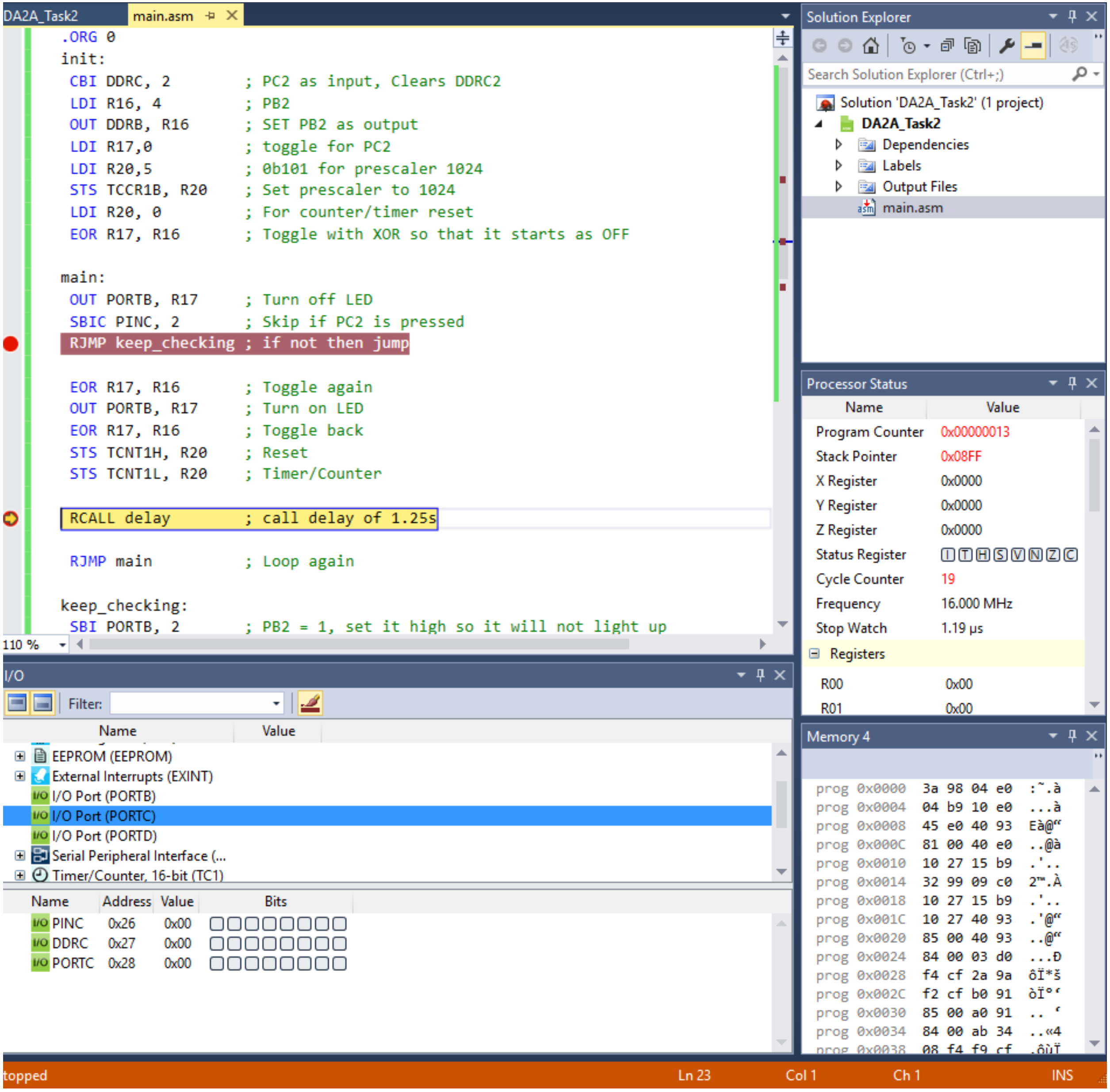
Processor Status

Name	Value
Program Counter	0x0000000F
Stack Pointer	0x08FF
X Register	0x11B3
Y Register	0x0000
Z Register	0x0000
Status Register	I T H S V N Z C
Cycle Counter	11598870
Frequency	16.000 MHz
Stop Watch	724.93 ms
Registers	
R00	0x00
R01	0x00

Name	Address	Value	Bits
PINB	0x23	0x04	□□□□□■□□
DDRB	0x24	0x04	□□□□□■□□
PORTB	0x25	0x00	□□□□□□□□

```
prog 0x0000  04 e0 04 b9  .à..
prog 0x0004  10 e0 45 e0  .àEà
prog 0x0008  40 93 81 00  @“..
prog 0x000C  40 e0 08 d0  @à.Ð
prog 0x0010  19 d0 10 27  .Ð.'
prog 0x0014  15 b9 04 d0  ...Ð
prog 0x0018  0b d0 10 27  .Ð.'
prog 0x001C  15 b9 f7 cf  ..÷Ï
prog 0x0020  40 93 85 00  @“..
prog 0x0024  40 93 84 00  @“..
prog 0x0028  08 95 ba 31  ..º1
prog 0x002C  5c f0 08 95  \ð..
prog 0x0030  b0 91 85 00  °‘..
prog 0x0034  a0 91 84 00   ‘..
prog 0x0038  a3 3b 08 f4  £;.ô
```

TASK 2: INITIAL BUTTON = 0 (or PRESSED)



```
 .ORG 0
 init:
  CBI DDRC, 2         ; PC2 as input, Clears DDRC2
  LDI R16, 4          ; PB2
  OUT DDRB, R16       ; SET PB2 as output
  LDI R17,0           ; toggle for PC2
  LDI R20,5           ; 0b101 for prescaler 1024
  STS TCCR1B, R20     ; Set prescaler to 1024
  LDI R20, 0          ; For counter/timer reset
  EOR R17, R16        ; Toggle with XOR so that it starts as OFF

 main:
  OUT PORTB, R17      ; Turn off LED
  SBIC PINC, 2        ; Skip if PC2 is pressed
  RJMP keep_checking ; if not then jump

  EOR R17, R16        ; Toggle again
  OUT PORTB, R17      ; Turn on LED
  EOR R17, R16        ; Toggle back
  STS TCNT1H, R20     ; Reset
  STS TCNT1L, R20     ; Timer/Counter

  RCALL delay         ; call delay of 1.25s

  RJMP main           ; Loop again

 keep_checking:
  SBI PORTB, 2        ; PB2 = 1, set it high so it will not light up
```

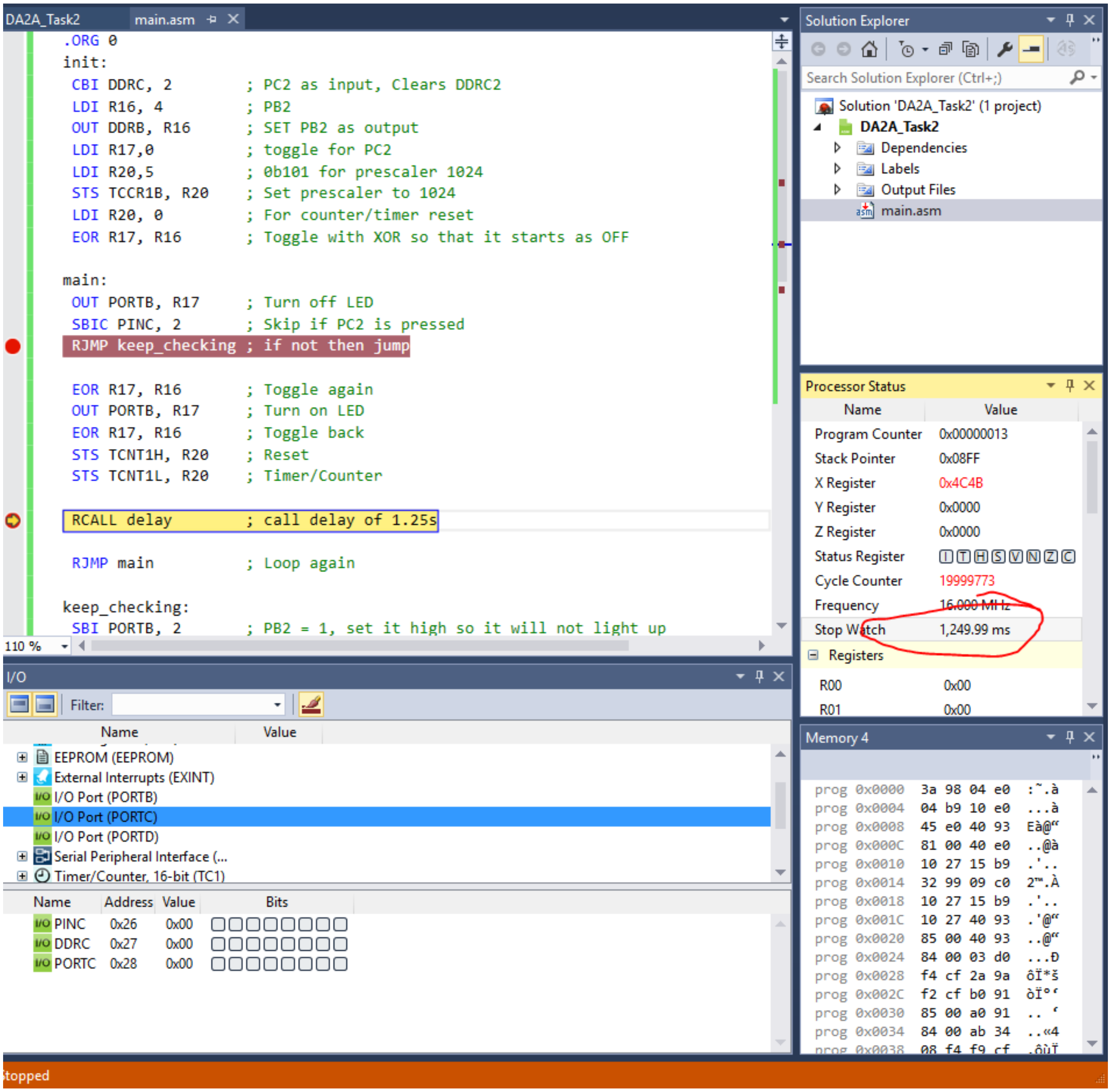
Processor Status

Name	Value
Program Counter	0x00000013
Stack Pointer	0x08FF
X Register	0x0000
Y Register	0x0000
Z Register	0x0000
Status Register	I T H S V N Z C
Cycle Counter	19
Frequency	16.000 MHz
Stop Watch	1.19 µs
⊟ Registers	
R00	0x00
R01	0x00

Name	Address	Value	Bits
PINC	0x26	0x00	□□□□□□□□
DDRC	0x27	0x00	□□□□□□□□
PORTC	0x28	0x00	□□□□□□□□

```
prog 0x0000  3a 98 04 e0  :˜.à
prog 0x0004  04 b9 10 e0  ...à
prog 0x0008  45 e0 40 93  Eà@“
prog 0x000C  81 00 40 e0  ..@à
prog 0x0010  10 27 15 b9  .'..
prog 0x0014  32 99 09 c0  2™.À
prog 0x0018  10 27 15 b9  .'..
prog 0x001C  10 27 40 93  .'@“
prog 0x0020  85 00 40 93  ..@“
prog 0x0024  84 00 03 d0  ...Ð
prog 0x0028  f4 cf 2a 9a  ôÏ*š
prog 0x002C  f2 cf b0 91  òÏ°‘
prog 0x0030  85 00 a0 91  .. ‘
prog 0x0034  84 00 ab 34  ..«4
prog 0x0038  08 f4 f9 cf  .ôùÏ
```

TASK 2: AFTER DELAY



```
.ORG 0
init:
 CBI DDRC, 2         ; PC2 as input, Clears DDRC2
 LDI R16, 4          ; PB2
 OUT DDRB, R16       ; SET PB2 as output
 LDI R17,0           ; toggle for PC2
 LDI R20,5           ; 0b101 for prescaler 1024
 STS TCCR1B, R20     ; Set prescaler to 1024
 LDI R20, 0          ; For counter/timer reset
 EOR R17, R16        ; Toggle with XOR so that it starts as OFF

main:
 OUT PORTB, R17      ; Turn off LED
 SBIC PINC, 2        ; Skip if PC2 is pressed
 RJMP keep_checking ; if not then jump

 EOR R17, R16        ; Toggle again
 OUT PORTB, R17      ; Turn on LED
 EOR R17, R16        ; Toggle back
 STS TCNT1H, R20     ; Reset
 STS TCNT1L, R20     ; Timer/Counter

 RCALL delay         ; call delay of 1.25s

 RJMP main           ; Loop again

keep_checking:
 SBI PORTB, 2        ; PB2 = 1, set it high so it will not light up
```

Processor Status

Name	Value
Program Counter	0x00000013
Stack Pointer	0x08FF
X Register	0x4C4B
Y Register	0x0000
Z Register	0x0000
Status Register	I T H S V N Z C
Cycle Counter	19999773
Frequency	16.000 MHz
Stop Watch	1,249.99 ms
⊟ Registers	
R00	0x00
R01	0x00

Name	Value
⊞ EEPROM (EEPROM)	
⊞ External Interrupts (EXINT)	
I/O Port (PORTB)	
I/O Port (PORTC)	
I/O Port (PORTD)	
⊞ Serial Peripheral Interface (...	
⊞ Timer/Counter, 16-bit (TC1)	

Name	Address	Value	Bits
PINC	0x26	0x00	□□□□□□□□
DDRC	0x27	0x00	□□□□□□□□
PORTC	0x28	0x00	□□□□□□□□

```
prog 0x0000  3a 98 04 e0  :˜.à
prog 0x0004  04 b9 10 e0  ...à
prog 0x0008  45 e0 40 93  Eà@“
prog 0x000C  81 00 40 e0  ..@à
prog 0x0010  10 27 15 b9  .'..
prog 0x0014  32 99 09 c0  2™.À
prog 0x0018  10 27 15 b9  .'..
prog 0x001C  10 27 40 93  .'@“
prog 0x0020  85 00 40 93  ..@“
prog 0x0024  84 00 03 d0  ...Ð
prog 0x0028  f4 cf 2a 9a  ôÏ*š
prog 0x002C  f2 cf b0 91  òÏ°‘
prog 0x0030  85 00 a0 91  .. ‘
prog 0x0034  84 00 ab 34  ..«4
prog 0x0038  08 f4 f9 cf  .ôùÏ
```

8. SCREENSHOT OF EACH DEMO (BOARD SETUP)

TASK 1:

TASK 2:

9. VIDEO LINKS OF EACH DEMO

DA2A Task 1- https://youtu.be/6_uJ5-JgL44
DA2A Task 2 - https://youtu.be/mwjTFpltsVw

10. GITHUB LINK OF THIS DA

https://github.com/recrio/submissions/tree/master/DesignAssignments/DA2A

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

"*This assignment submission is my own, original work*".
Ron Joshua Recrio