CPE301 – SPRING 2019

Design Assignment DA2B

Student Name: Cody McDonald

Student #: 5000382538

Student Email: mcdonc4@unlv.nevada.edu

Primary Github address: https://github.com/elev8rProcrastinator/submission\_da.git

Directory: https://github.com/elev8rProcrastinator/submission\_da/tree/master/DA2B

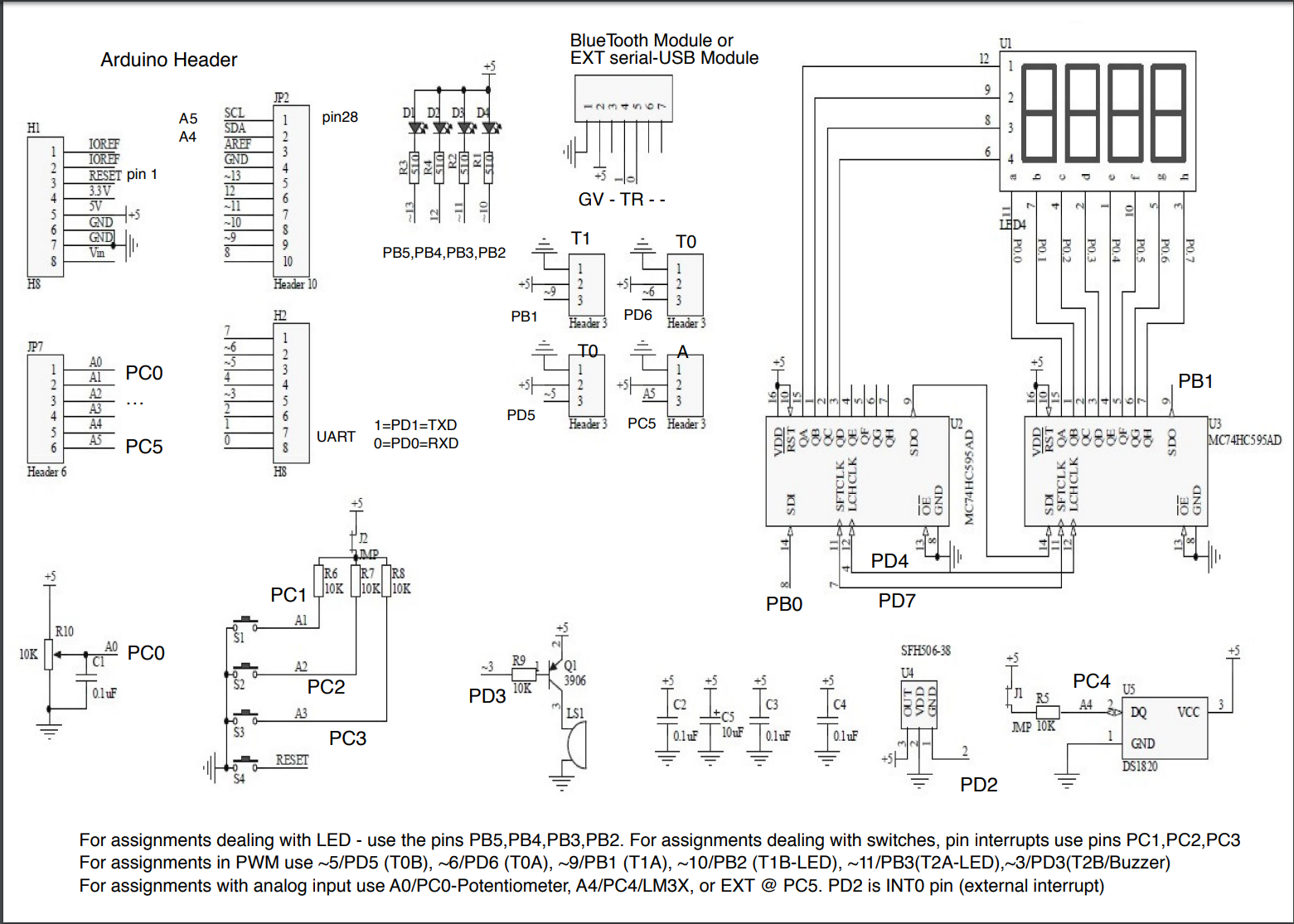
Submit the following for all Labs:

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).

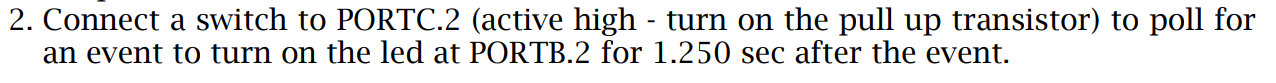
1. **COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS**

ATMega328P mini

Multi-shield



1. **INITIAL CODE OF TASK DA.2. ASSEMBLY AND C-CODE:**



ASSEMBLY:

; CPE301\_DA2\_assembly\_task2.asm

;

; Created: 3/2/2019 3:31:09 PM

; Author : Cody

; TASK2

;TCNT for 1.25s = 0x4C4A

.ORG 0x00 ;start memory at 0x00

LDI R20,5 ;set clock pre-scaler to 1024 and normal mode

STS TCCR1B,R20 ;attach value to correct register

CBI DDRC, 2 ;set pin 2 to an input

LDI R16,0b00111100 ;set led pins to outputs

OUT DDRB, R16 ;attach to ddr

LDI R16, 0xFF ;make sure all leds are off

OUT PORTB, R16

AGAIN:

LDI R16, 0b00111100 ;turn led's off

OUT PORTB, R16 ;output to port b

SBIC PINC,2 ;skip next line if switch 2 is pressed

RJMP OVER ;jump to over subroutine

LDI R16, 0b00111000 ;turn on led

OUT PORTB, R16 ;output to port

LDI R20,0x00 ;reset the counter to 0

STS TCNT1H,R20

STS TCNT1L,R20

RCALL DELAY ;go to delay subroutine

RJMP AGAIN

OVER: ;subroutine to loop if button isn't pressed

SBI PORTB,2

RJMP AGAIN

DELAY:

LDS R29, TCNT1H ;loading upper bit of counter to R29

LDS R28, TCNT1L ;loading lower bit of counter to R28

CPI R28,0x4A ;comparing if lower 8 bits of timer is 0x4A

BRSH BODY ;if lower bits of timer have reached desired amount, check the upperbits

RJMP DELAY ;otherwise, keep checking lower bits

BODY:

CPI R29,0x4C ;check to see if upper timer bits have reached the desired value

BRLT DELAY ;if not, recheck the lower bits

RET ;once the timer reached the desired value, toggle the LED

C-CODE:

/\*

\* CPE301\_DA2\_C\_task2.c

\*

\* Created: 3/2/2019 12:04:23 AM

\* Author : Cody

\*/

#define *F\_CPU* 16000000UL //set clock speed to 16MHz

//include important header files

#include <avr/io.h>

#include <stdio.h>

#include <util/delay.h>

int main(void){

//initialize registers

DDRB |= (1<<2); //set portb 2 to output

PORTB |= (1<<2); //set pb2 to high

DDRC &= (0<<2); //set ddrc to input

PORTC |= (0<<2); //set portc2 to low

while (1) {

if (!(PINC & (1 << PINC2))) //check for button press

{

PORTB &= ~(1<<2); //if pressed then turn on led

*\_delay\_ms*(1250); //keep led on for 1.25s

}

else {

PORTB |= (1<<2); //if not pressed then keep led off

}

}

return 0;

}

1. **DEVELOPED MODIFIED CODE OF TASK DA.2. ASSEMBLY AND C-CODE:**

**Assembly:**

1. ;
2. ; CPE301\_DA2B\_assembly\_task2.asm
3. ;
4. ; Author : Cody
5. ; TASK2
6. .include <m328pdef.inc>
7. .ORG 0x00 ;start memory at 0x00
8. JMP MAIN
9. .ORG 0X10 ;jump memory back to 0x10
10. JMP PCINT1\_ISR
11. MAIN:
12. ;Set stack pointers
13. LDI R20, HIGH(RAMEND)
14. OUT SPH, R20
15. LDI R20, LOW(RAMEND)
16. OUT SPL, R20
17. ;Initialize ports
18. LDI R17, 0b0011\_1000 ;reset register
19. LDI R19, 4 ;used to toggle 4th LED
20. LDI R21, 0xFF ;used to set DDRB high
22. OUT DDRB, R21 ;set led ports to high
23. OUT PORTB, R21 ;set leds to low
25. OUT DDRC, R19 ;set ddrc to inputs
26. ;Set modes for interrupt
27. LDI R20,0b0000\_0100
28. STS PCMSK1, R20
29. OUT PORTC, R20
30. LDI R20, (1<<PCIE1)
31. STS PCICR, R20
32. SEI ;start interrupt
33. ;Run an infinite loop
34. HERE:
35. OUT PORTB,R21 ;keep LED off unless
36. JMP HERE
37. ;interrupt subroutine
38. PCINT1\_ISR:
39. EOR R17,R19
40. OUT PORTB,R17
41. ;delay subroutine
42. ldi r22, 102
43. ldi r23, 118
44. ldi r24, 194
45. L1: dec r24
46. brne L1
47. dec r23
48. brne L1
49. dec r22
50. brne L1
51. RETI ;return back to loop

**C-Code:**

/\*

\* CPE301\_DA2B\_C.c

\*

\* Created: 3/2/2019 12:04:23 AM

\* Author : Cody

\*/

#define *F\_CPU* 16000000UL //set clock speed to 16MHz

//include important header files

#include <avr/io.h>

#include <stdio.h>

#include <util/delay.h>

#include <avr/interrupt.h>

int main(void){

//initialize registers

DDRB |= (1<<2)|(1<<5);

PORTB |= (1<<5); //turn off LED

PORTB |= (1<<2); //set pb2 to high

DDRC &= (0<<1)|(0<<2)|(0<<3); //set ddrc to input

PORTC |= (1<<1)|(1<<2)|(1<<3); //set portc 1/2/3 to high

// Enable PC at pins 9/10/11

PCMSK1 |= (1<<PCINT9)|(1<<PCINT10)|(1<<PCINT11); //set corresponding pins in msk

PCICR |= (1<<PCIE1); // turn them on in register

sei(); // start interrupt

while (1) ;

}

ISR (PCINT1\_vect){ //delay subroutine

PORTB ^= (1<<2);

*\_delay\_ms*(1250);

}

1. **SCHEMATICS**

N/A

1. **SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)**

**TASK 1/DA2B Assembly:**

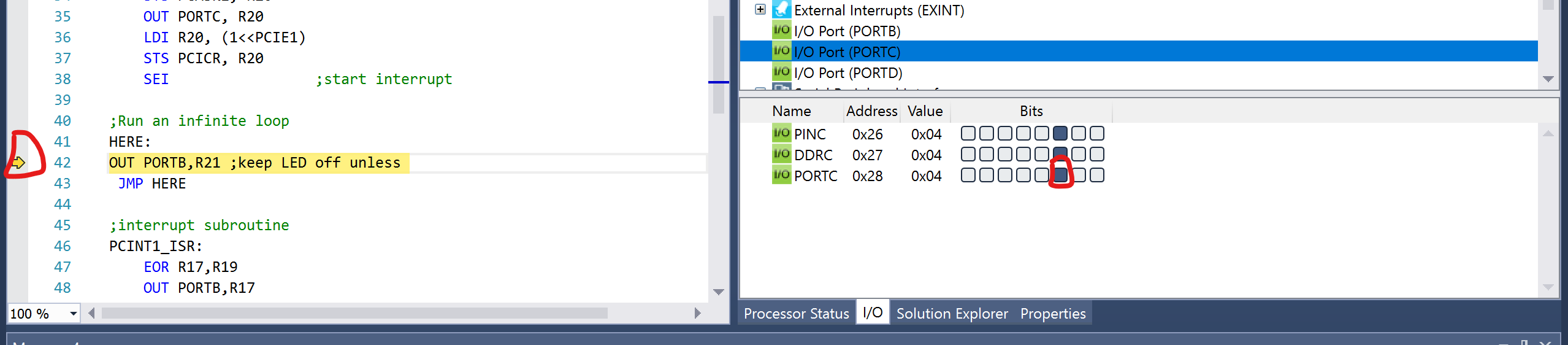
****

Figure : The button is set high and remains outside of the interrupt loop

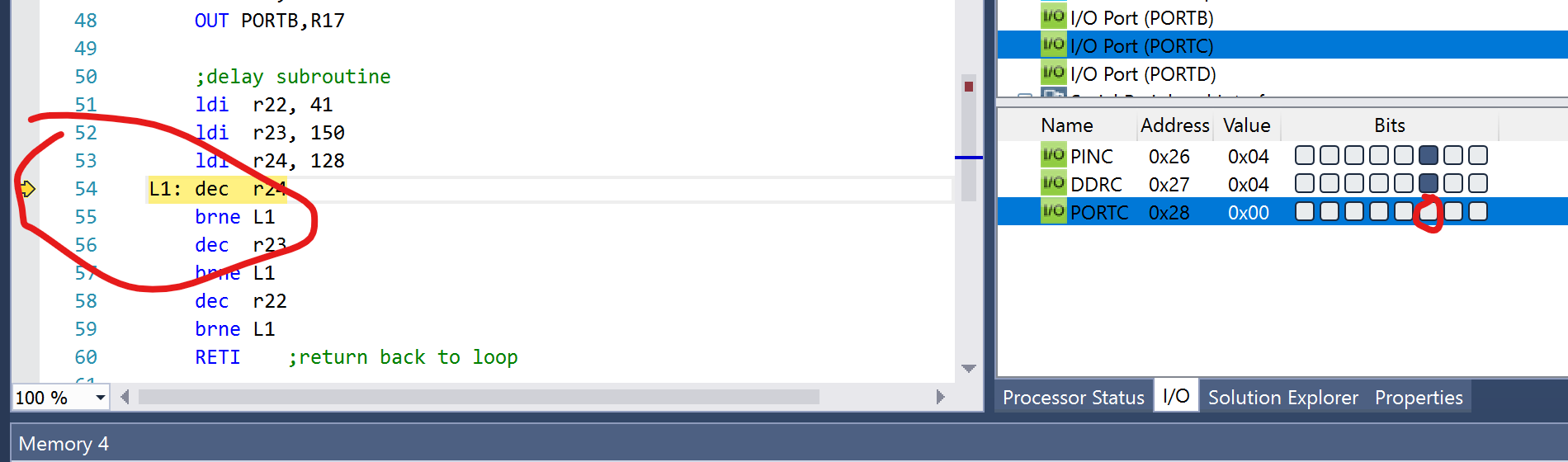
****

Figure : after button is depressed the code will enter the delay subroutine

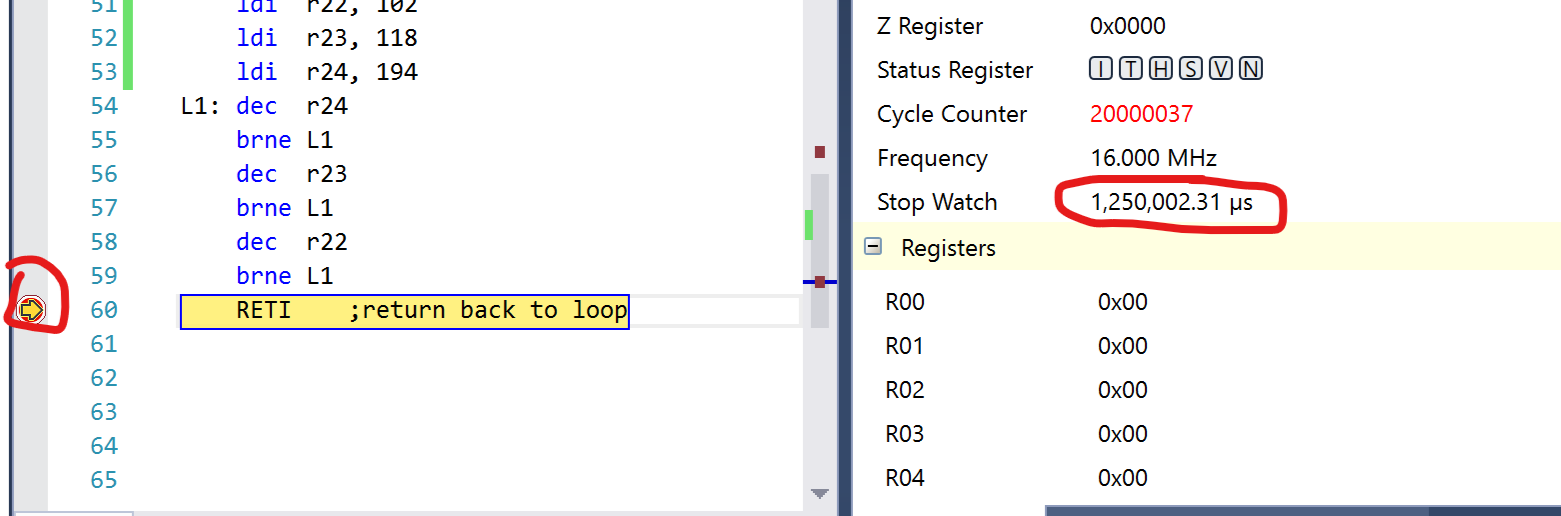
****

Figure : This shows the amount of delay in the delay subroutine which adds up to 1.25s

**TASK 1/DA2B C-Code:**

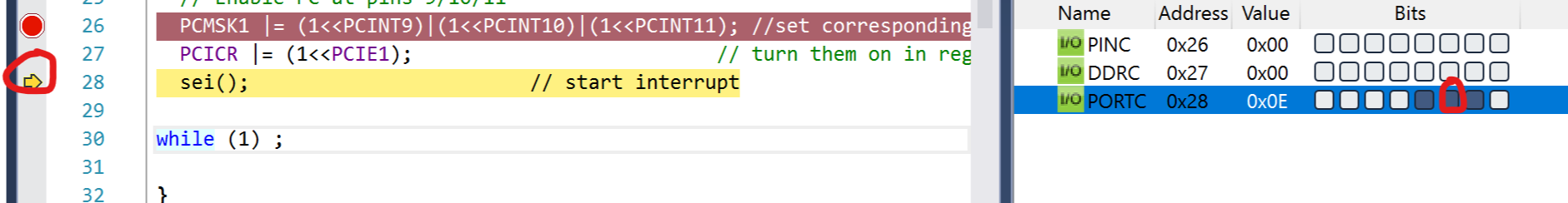
****

Figure :Before the button is depressed the code remains outside of the delay and LED subroutine

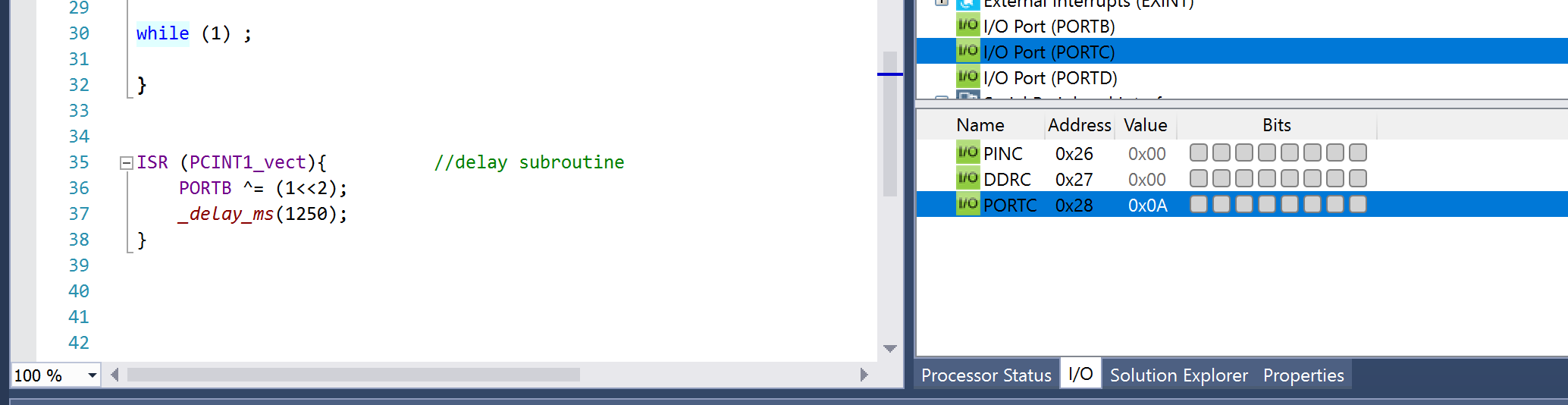
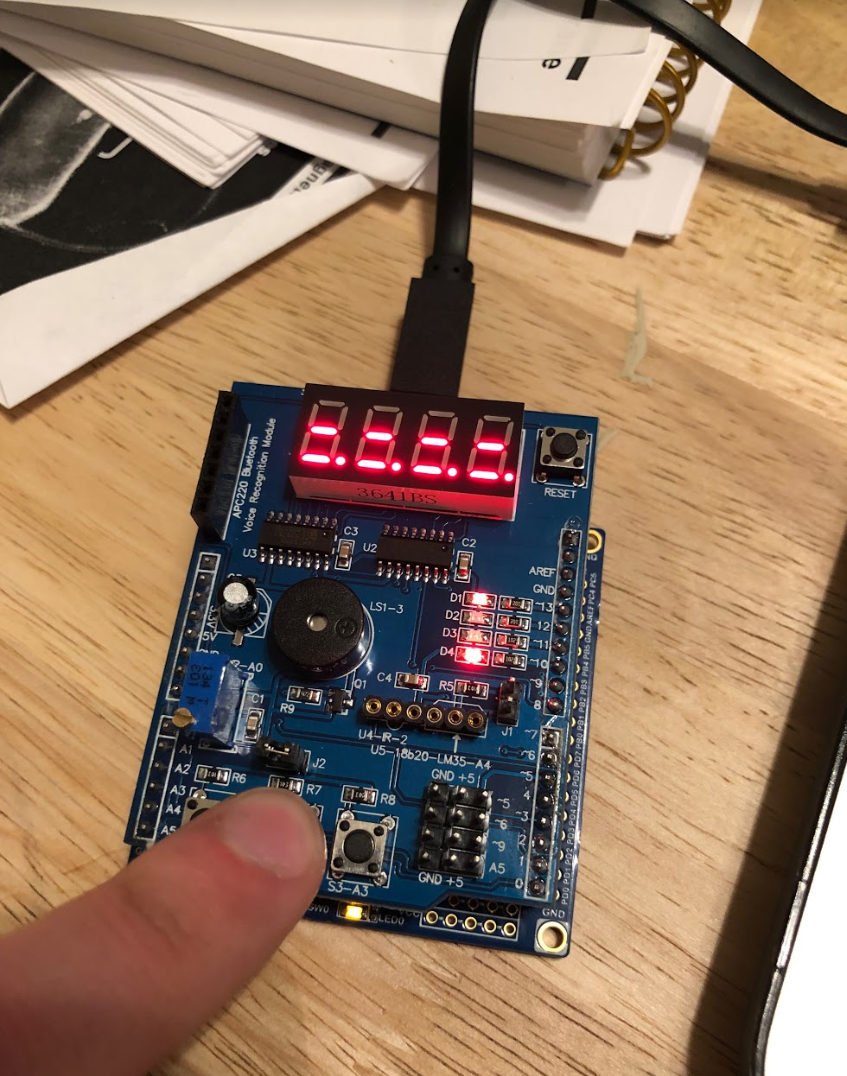
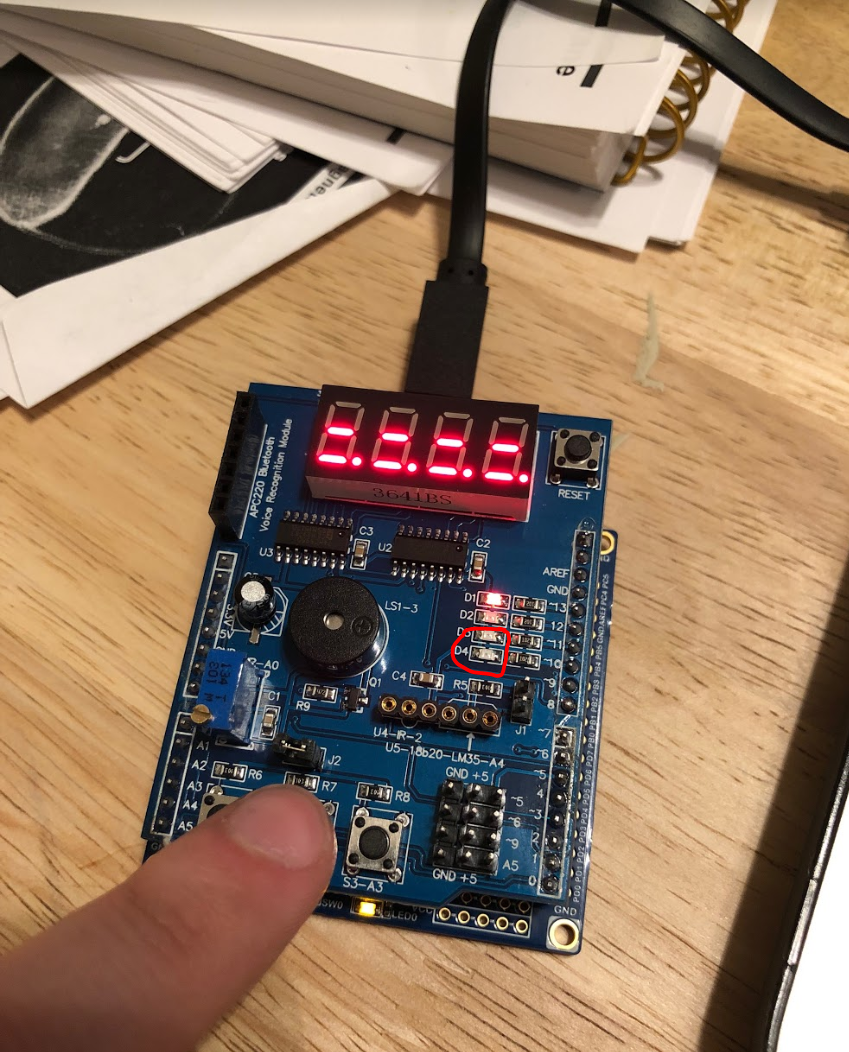
****

Figure : Afterwards the the button is depressed and enters the subroutine. The delay function creates a 1.25ms delay which I believe doesn't need to be proven to show the delay.

1. **SCREENSHOT OF EACH DEMO (BOARD SETUP)**

**The board setup is the same as DA2, so I will use the same screen shot. The same button must be pressed along with the same LED lighting up.**

**Below is the multi-shield sitting on top of the AVR. The two pictures demonstrate the effect of a button press for the second task.**



1. **VIDEO LINKS OF EACH DEMO**

**Both demos are featured in the below link.**

https://www.youtube.com/watch?v=uRLvSl80LxY

1. **GITHUB LINK OF THIS DA**

https://github.com/elev8rProcrastinator/submission\_da/tree/master/DA2B

**Student Academic Misconduct Policy**

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

“This assignment submission is my own, original work”.

Cody McDonald