CPE301 – SPRING 2019

Design Assignment DA2A

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/DA2C

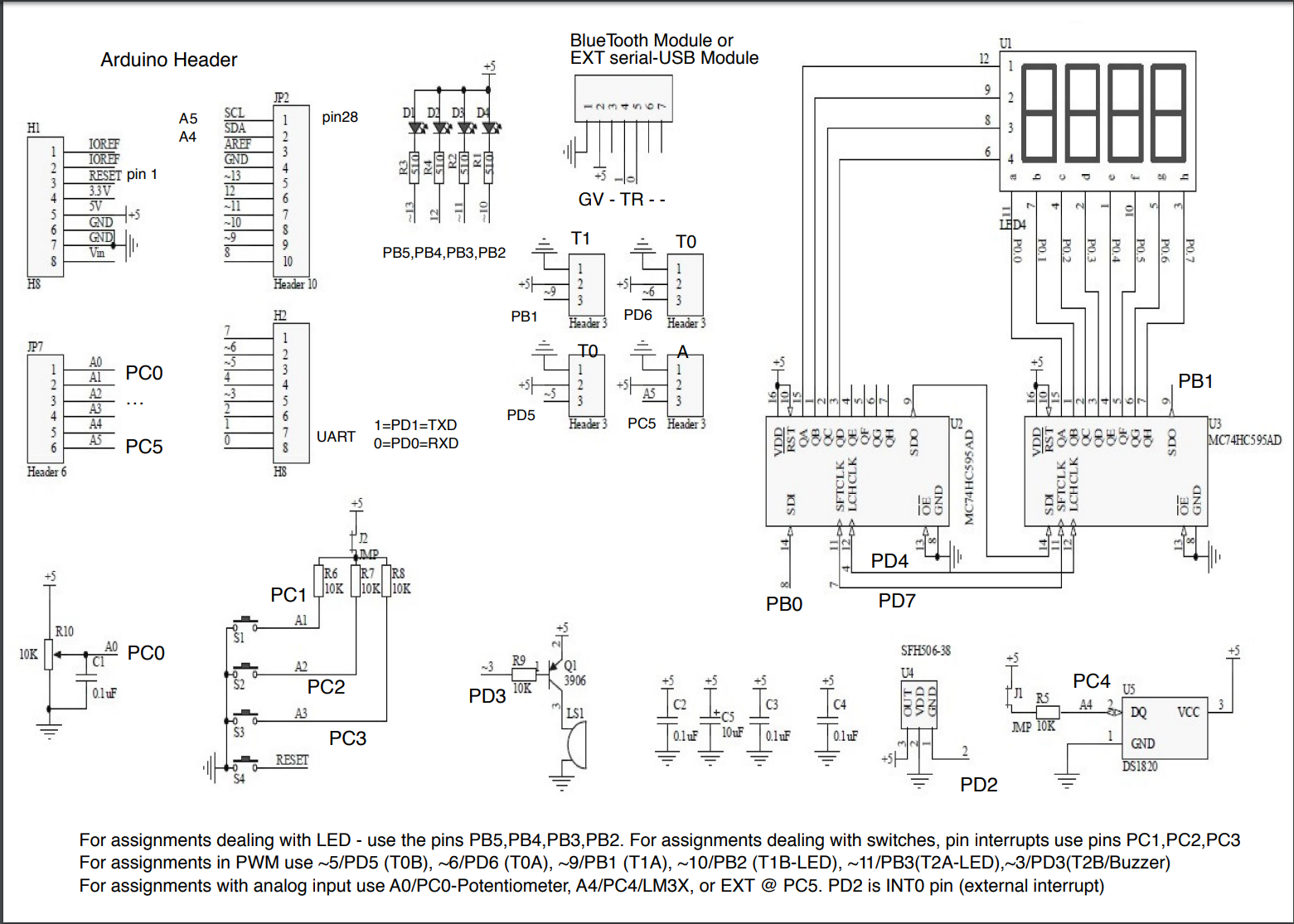
Submit the following for all Labs:

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).

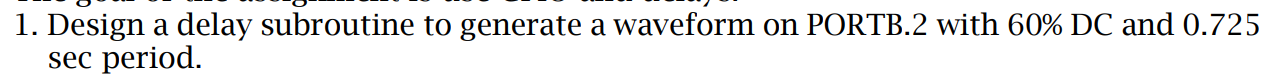
1. **COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS**

ATMega328P mini

Multi-shield



1. **INITIAL/MODIFIED/DEVELOPED CODE 2/A Task 1. C-CODE:**



C-CODE PORTION:

/\*

\* CPE301\_DA2\_C\_task1.c

\*

\* Author : Cody

\*/

#include <stdio.h>

#include <avr/io.h>

//0.725s period

//60% = 0.435s. TCNT = 6796 = 1A8C

//40% = 0.29s. TCNT = 4530 = 11B2

int main(void)

{

DDRB |= (1<<2); // Set all PORTB out

TCCR1B = 0x0D; // Set pre-scaler to 1024 and CTC Mode

OCR1A = 0x1A8C; // Set top of OCR1A to 0.435s

int flag = 0; // set flag variables

int flag2 = 1;

// Run loop for 60% DC

while(1){

flag = TIFR1 & 0x04;

if (flag == 0x04 && flag2 == 0){

PORTB &= ~(1<<2); //turn off LED

TIFR1 = 0x04; //reset clear flag

OCR1A = 0x11B2; //set delay for LED off

flag2 = 1;

}

else if(flag== 0x04 && flag2 == 1){

PORTB |= (1<<2); // turn on LED

TIFR1 = 0x04; // reset clear flag

OCR1A = 0x1A8C; // set delay for LED on

flag2 = 0;

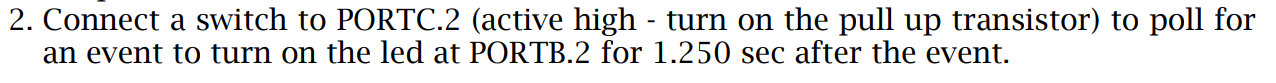
}

}

return 0;

}

1. **DEVELOPED MODIFIED CODE OF TASK 2/A TASK 2. C-CODE:**



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

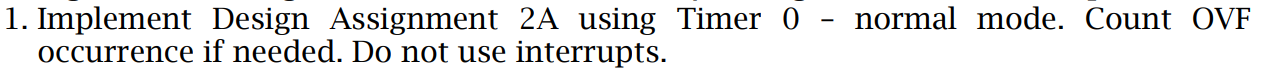
}

}

return 0;

}

1. **DA2C TASK 1 DEVELOPED MODIFIED CODE OF TASK 2/A TASK 1. C-CODE:**



1. /\*
2. \* CPE301\_DA2C\_task1.c
3. \*
4. \* Created: 3/19/2019 11:30:40 PM
5. \* Author : Cody
6. \*/
7. #include <stdio.h>
8. #include <avr/io.h>
9. //0.725s period
10. //60% = 0.435s. TCNT = 6796 = 1A8C
11. //40% = 0.29s. TCNT = 4530 = 11B2
12. int main(void)
13. {
14. DDRB |= (1<<2)|(1<<5); // led 2 and 5 to output
15. PORTB |= (1<<5);
16. TCCR0B |= (1<<CS00)|(1<<CS02); // Set pre-scaler to 1024
18. //declare vital variables
19. int ovFlowCount = 0;
20. int ovCheck = 0;
21. int flag = 1;
22. int dc60 = 27; //overflow count to reach 6796
23. int dc40 = 18; //overflow count to reach 4530
24. TCNT0 = 0; //initialize clock
25. // Run loop for 60% DC
26. while(1){
27. ovCheck = TIFR0 & 0x01; //variable to check if overflow met.
29. if(ovCheck == 1){
30. ovFlowCount ++;
31. TIFR0 |= (1<<TOV0); //reset overflow flag
32. }
34. if (flag == 0 && ovFlowCount == dc40){
35. PORTB &= ~(1<<2); //turn on LED
36. TCNT0 = 0; //reset counter
37. ovFlowCount = 0; //reset ov counter
38. flag = 1;
39. }
40. else if(flag == 1 && ovFlowCount == dc60){
41. PORTB |= (1<<2); // turn off LED
42. TCNT0 = 0; //reset counter
43. ovFlowCount = 0; //reset ov counter
44. flag = 0;
45. }
46. }
47. return 0;
48. }
49. **DA2C TASK 1 DEVELOPED MODIFIED CODE OF TASK 2/A TASK 2. C-CODE:**

/\*

\* CPE301\_DA2C\_task1\_p2.c

\*

\* Created: 3/19/2019 11:43:34 PM

\* Author : Cody

\*/

#include <stdio.h>

#include <avr/io.h>

//TCNT FOR 1.25s = 19531.25. Overflow count = 77

int main(void){

//initialize registers

DDRB |= (1<<2)|(1<<5); //set portb 2 to output

PORTB |= (1<<2)|(1<<5); //set pb2 to high

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

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

TCCR0B |= (1<<CS00)|(1<<CS02); // Set pre-scaler to 1024

//set local variables

int ovFlowCount = 0;

int ovCheck = 0;

int delayCount = 77; //amount of overflows to reach desired delay

while (1) {

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

{

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

TCNT0 = 0; //start counter at 0

while(ovFlowCount <= delayCount ){

ovCheck = TIFR0 & 0x01; //variable to check if overflow met.

if (ovCheck == 1){

ovFlowCount ++; //increase overflow count

TIFR0 |= (1<<TOV0); //reset sreg flag

}

}

ovFlowCount = 0; //reset overflow count

}

else {

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

}

}

return 0;

}

1. **DA2C TASK 2 DEVELOPED MODIFIED CODE OF TASK 2/A TASK 1. C-CODE:**



/\*

\* CPE301\_DA2C\_task2.c

\*

\* Created: 3/20/2019 2:00:02 AM

\* Author : Cody

\*/

#include <avr/io.h>

#include <stdio.h>

#include <avr/interrupt.h>

//declare vital variables

int ovFlowCount = 0;

int dc60 = 27; //overflow count to reach 6796 ov count 27

int dc40 = 18; //overflow count to reach 4530 ov count 18

int flag = 1;

//0.725s period

//60% = 0.435s. TCNT = 6796 = 1A8C

//40% = 0.29s. TCNT = 4530 = 11B2

int main(void)

{

DDRB |= (1<<2)|(1<<5); // Set all PORTB out

PORTB |= (1<<5);

TCCR0B |= (1<<CS00)|(1<<CS02); // Set pre-scaler to 1024

TIMSK0 |= (1<<TOIE0); //enable timer interrupts

sei(); //enable interrupts

TCNT0 = 0; //initialize clock

while(1);

}

ISR(TIMER0\_OVF\_vect){

ovFlowCount ++;

if (flag == 1 && ovFlowCount == dc40){

PORTB &= ~(1<<2);

ovFlowCount = 0; //reset overflow counter

flag = 0;

}

else if (flag == 0 && ovFlowCount == dc60){

PORTB |= (1<<2);

ovFlowCount = 0; //reset overflow counter

flag = 1;

}

}

1. **DA2C TASK 2 DEVELOPED MODIFIED CODE OF TASK 2/A TASK 2. C-CODE:**

/\*

\* CPE301\_DA2C\_task2\_p2.c

\*

\* Created: 3/19/2019 11:56:36 PM

\* Author : Cody

\*/

#include <avr/io.h>

#include <stdio.h>

#include <avr/interrupt.h>

int flag = 0;

//TCNT FOR 1.25s = 19531.25. Overflow count = 77

int main(void){

//initialize registers

DDRB |= (1<<2)|(1<<5); //set portb 2 to output

PORTB |= (1<<2)|(1<<5); //set pb2 to high

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

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

TCCR0B |= (1<<CS00)|(1<<CS02); // Set pre-scaler to 1024

//int flag = 0;

TIMSK0 |= (1<<TOIE0); //enable timer interrupt

sei(); //enable interrupt

while (1) {

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

{

flag = 1; //if pressed then turn on led

}

else {

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

flag = 0; //keep flag off to keep from flase positives

}

}

return 0;

}

ISR(TIMER0\_OVF\_vect){

int ovFlowCount = 0;

int ovCheck = 0;

int delayCount = 77; //amount of overflows to reach desired delay

if (flag == 1){

PORTB &= ~(1<<2);

TCNT0 = 0; //start counter at 0

while(ovFlowCount <= delayCount ){

ovCheck = TIFR0 & 0x01; //variable to check if overflow met.

if (ovCheck == 1){

ovFlowCount ++; //increase overflow count

TIFR0 |= (1<<TOV0); //reset sreg flag

}

}

ovFlowCount = 0;

PORTB |= (1<<2);

flag = 0;

}

}

1. **DA2C TASK 3 DEVELOPED MODIFIED CODE OF TASK 2/A TASK 1. C-CODE:**



/\*

\* CPE301\_DA2C\_task3.c

\*

\* Created: 3/20/2019 2:27:21 AM

\* Author : Cody

\*/

#include <avr/io.h>

#include <stdio.h>

#include <avr/interrupt.h>

//declare vital variables

int matchCount = 0;

int dc60 = 27; //overflow count to reach 6796 ov count 27

int dc40 = 18; //overflow count to reach 4530 ov count 18

int flag = 1; //flag used to toggle between duty cycles

int cycleMatch = 255; //compare to 255 and clear each time

//0.725s period

//60% = 0.435s. TCNT = 6796 = 1A8C

//40% = 0.29s. TCNT = 4530 = 11B2

int main(void)

{

DDRB |= (1<<2)|(1<<5); // Set all PORTB out

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

TCCR0A |= (1<<WGM01); //set ctc mode on OCRA

TCCR0B |= (1<<CS00)|(1<<CS02); // Set pre-scaler to 1024

TIMSK0 |= (1<<OCIE0A); //enable compare interrupts

OCR0A = cycleMatch;

sei(); //enable interrupts

while(1); //infinite loop

}

ISR(TIMER0\_COMPA\_vect){

matchCount ++;

if (flag == 1 && matchCount == dc40){

PORTB &= ~(1<<2); //toggle LED

matchCount = 0; //reset overflow counter

flag = 0;

}

else if (flag == 0 && matchCount == dc60){

PORTB |= (1<<2); //toggle LED

matchCount = 0; //reset overflow counter

flag = 1;

}

}

1. **DA2C TASK 3 DEVELOPED MODIFIED CODE OF TASK 2/A TASK 1. C-CODE:**

/\*

\* CPE301\_DA2C\_task3\_p2.c

\*

\* Created: 3/20/2019 1:17:36 PM

\* Author : Cody

\*/

#include <avr/io.h>

#include <stdio.h>

#include <avr/interrupt.h>

//set global variables

int flag = 0;

int matchCount = 0;

int cycleMatch = 255;

//TCNT FOR 1.25s = 19531.25. Overflow count = 77

int main(void){

//initialize registers

DDRB |= (1<<2)|(1<<5); //set portb 2 to output

PORTB |= (1<<2)|(1<<5); //set pb2 to high

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

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

//Set modes

TCCR0A |= (1<<WGM01); //set ctc mode on OCRA

TCCR0B |= (1<<CS00)|(1<<CS02); // Set pre-scaler to 1024

TIMSK0 |= (1<<OCIE0A); //enable compare interrupts

OCR0A = cycleMatch;

sei(); //enable interrupt

while (1) {

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

{

flag = 1; //if pressed then turn on led

}

else {

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

flag = 0;

}

}

return 0;

}

ISR(TIMER0\_COMPA\_vect){

int ovFlowCount = 0;

int matchCheck = 0;

int delayCount = 77; //amount of overflows to reach desired delay

if (flag == 1){

flag = 0; //toggle flag

PORTB &= ~(1<<2);

TCNT0 = 0; //start counter at 0

while(matchCount <= delayCount ){

matchCheck = TIFR0 & 0x02; //variable to check if compare is met

if (matchCheck == 0x02){

matchCount ++; //increase match count

TIFR0 |= (1<<OCF0A); //reset sreg flag

}

}

matchCount = 0; //reset match count

PORTB |= (1<<2); //turn off LED

}

}

1. **SCHEMATICS**

N/A

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

**DA2C TASK1 PART 1:**

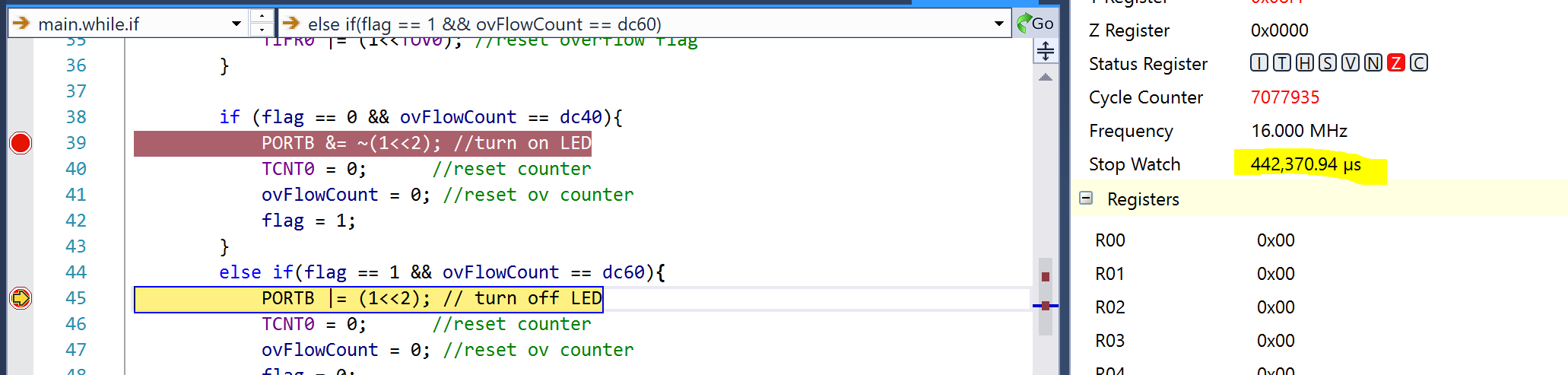


Figure : First DC

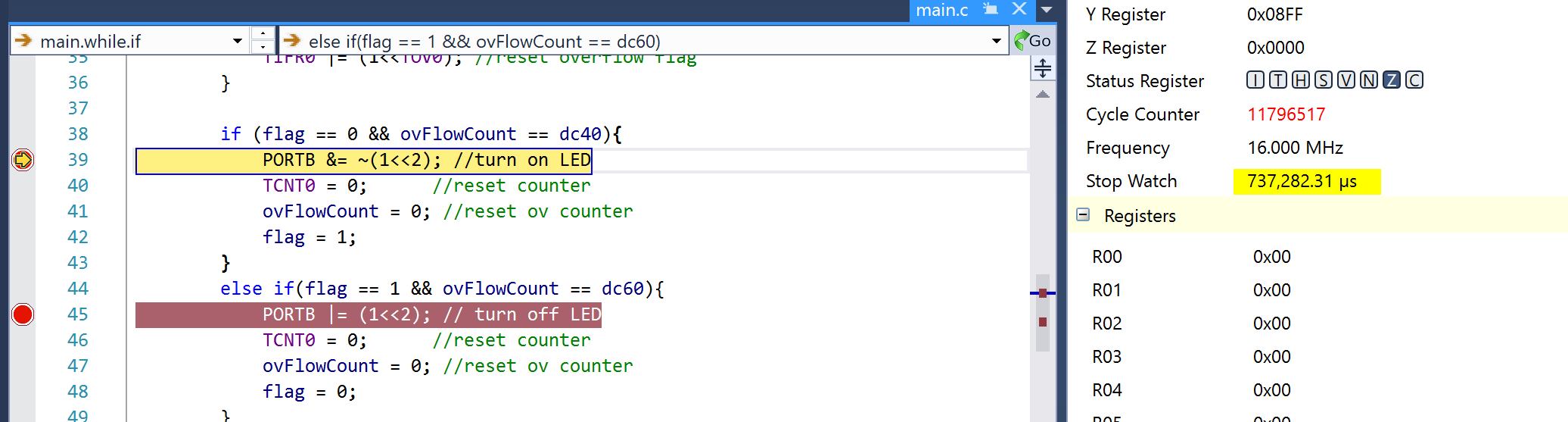


Figure : Second DC for a total of .74s

**DA2C TASK1 PART 2:**

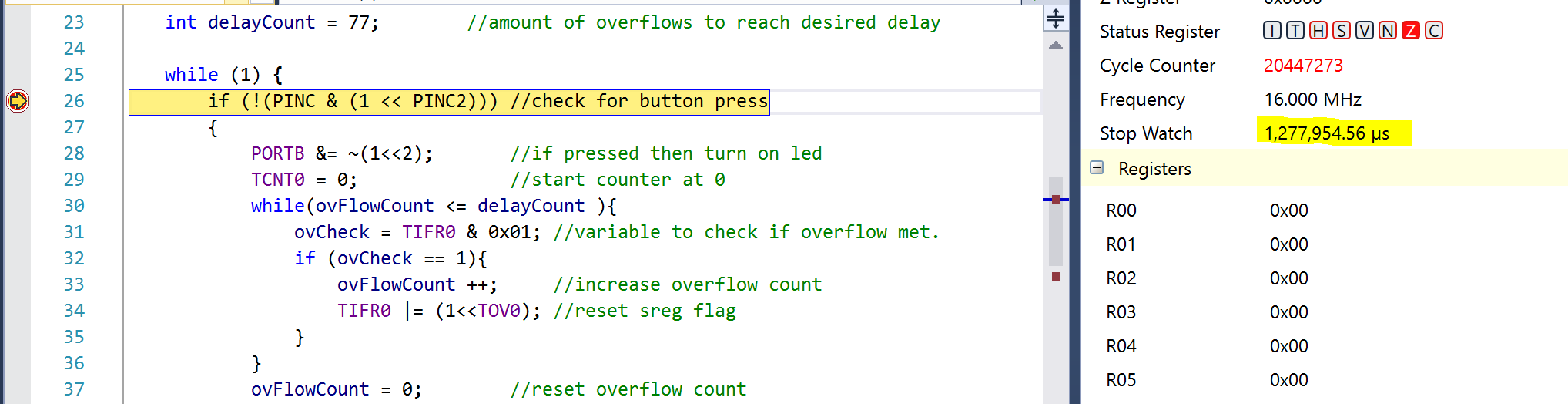


Figure : After running through one cycle of a button press

**DA2C TASK2 PART 1:**

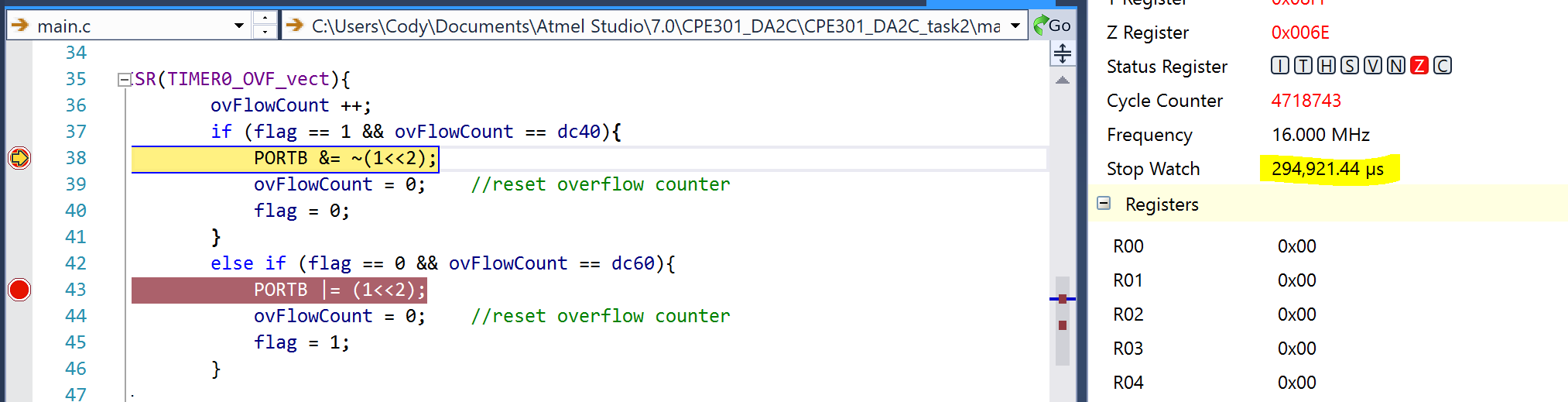


Figure : After first DC

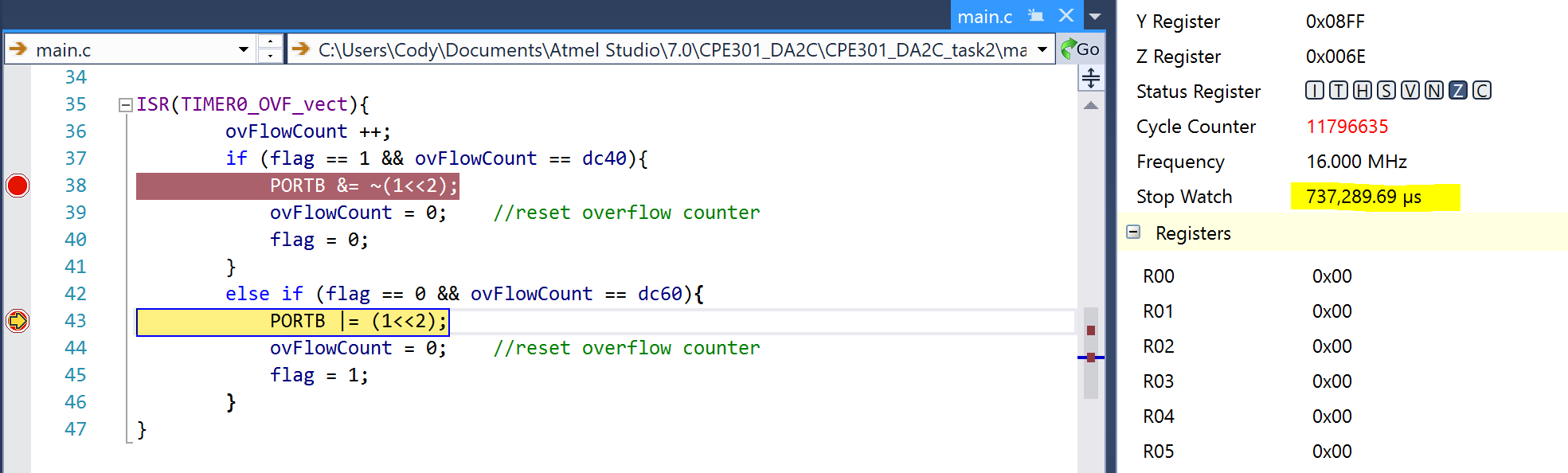


Figure : After second DC for a total of .74s

**DA2C TASK2 PART 2:**

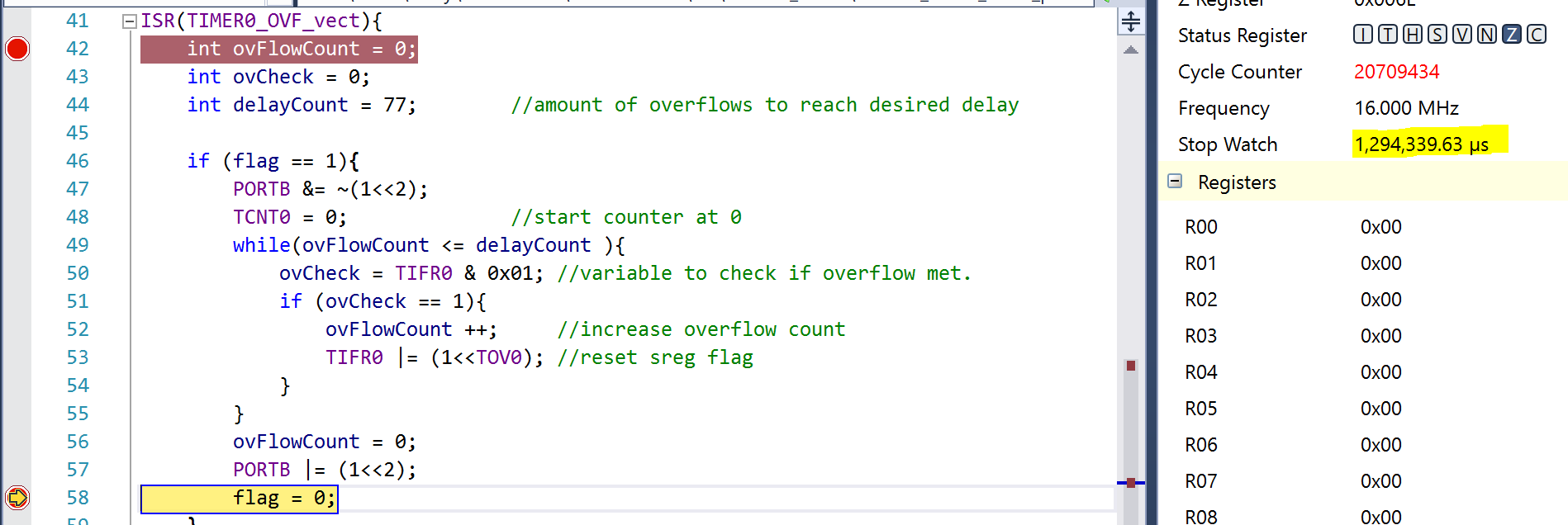


Figure : Clock after one button press

**DA2C TASK3 PART 1:**

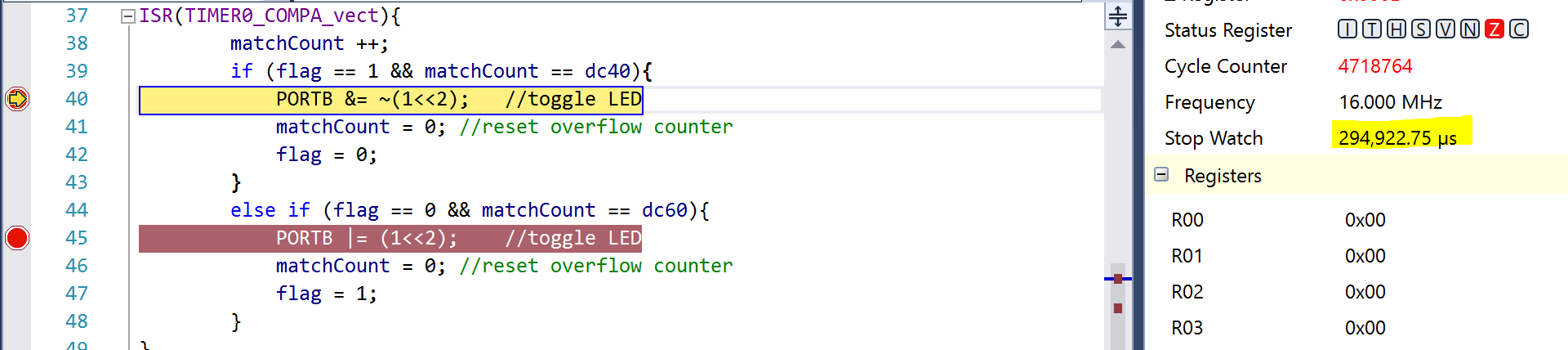


Figure : After first DC

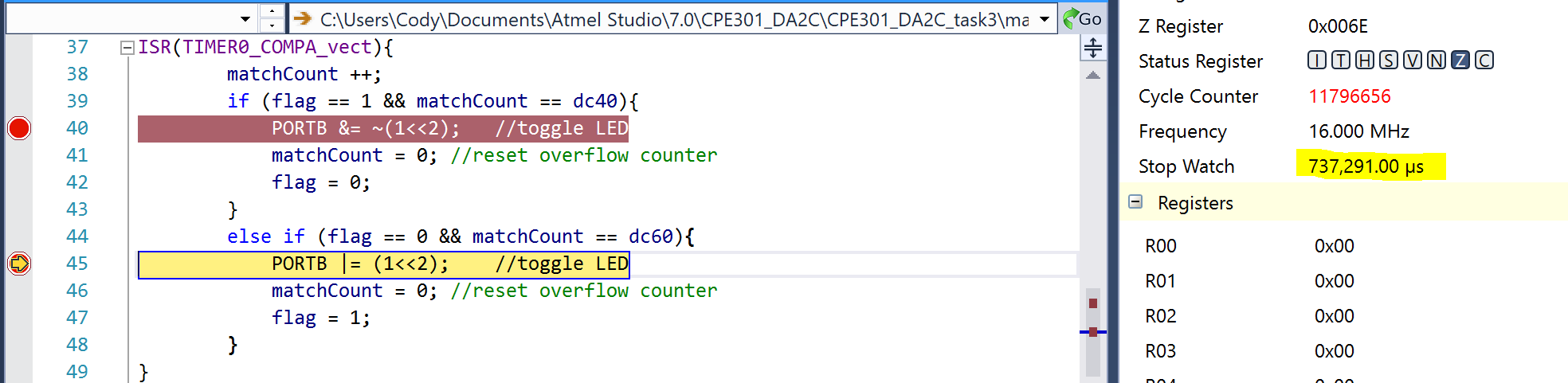


Figure : After second DC for a total of .74s

**DA2C TASK3 PART 2:**

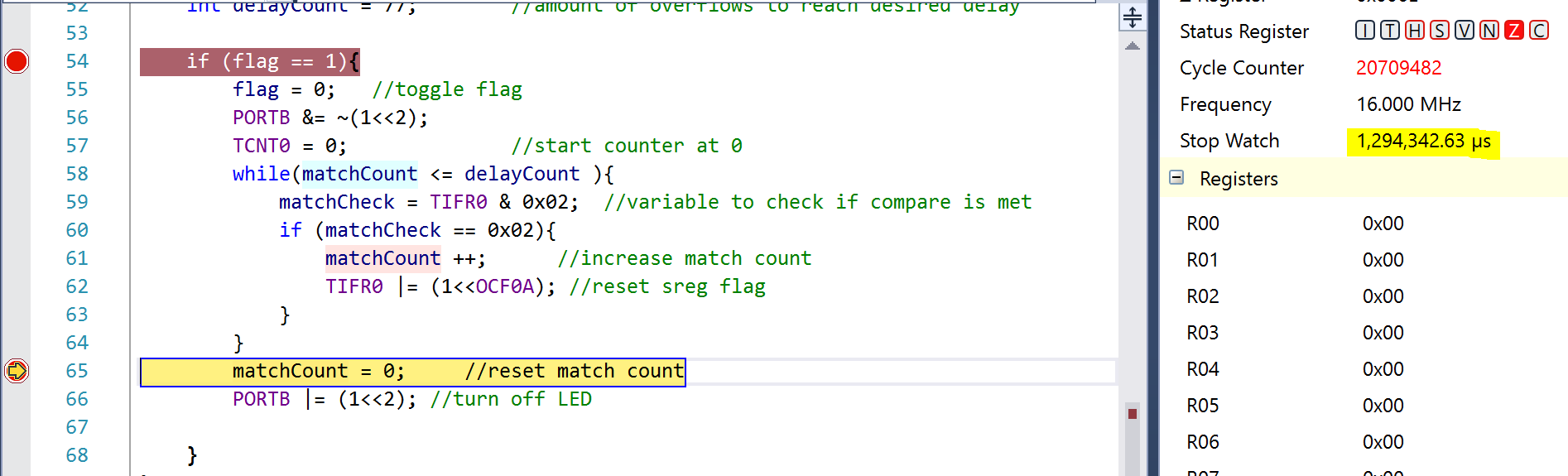
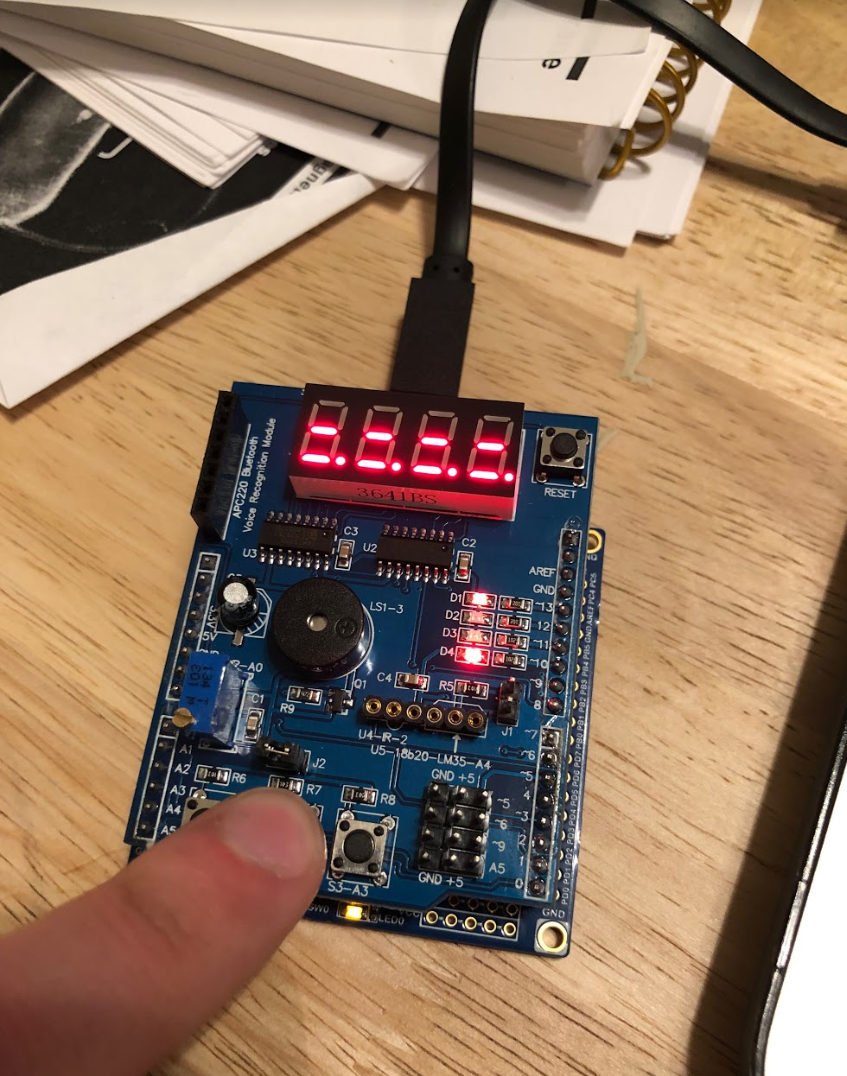
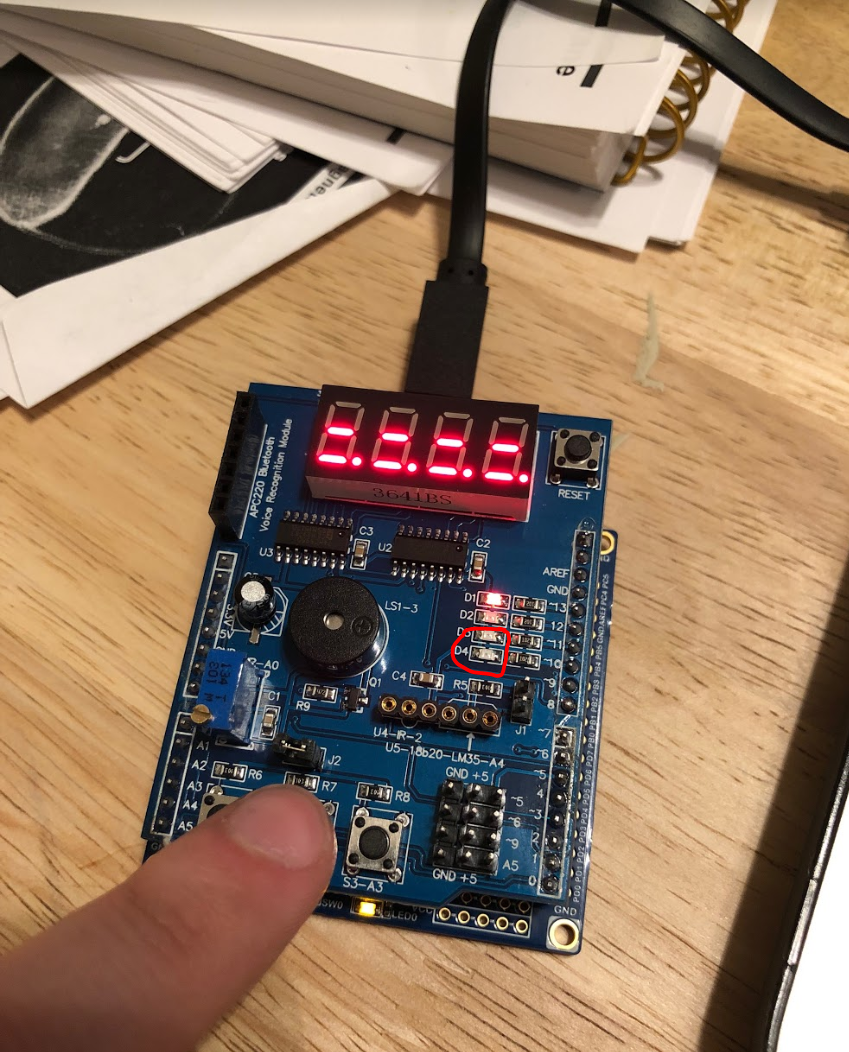


Figure : Clock after one button press

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

**The board setup of each task is the same so only one set of pictures will be listed. The first task just featured the bottom LED blinking**

**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**

**All 6 demos are featured in the below link.**

https://youtu.be/L3jjRQ-M9Og

1. **GITHUB LINK OF THIS DA**

https://github.com/elev8rProcrastinator/submission\_da/tree/master/DA2C

**Student Academic Misconduct Policy**

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

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

Cody McDonald