CPE301 – SPRING 2019

Design Assignment 2C

Student Name: Ron Joshua Recrio

Student #: 5003825419

Student Email: recrio@unlv.nevada.edu

Primary Github address: https://github.com/recrio

Directory: /DesignAssignments/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**

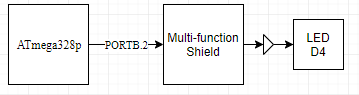
List of Components used

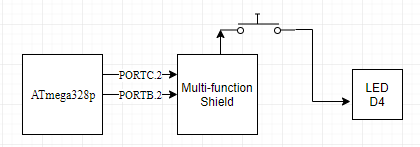
ATmega328p Xplained Mini

Multifunction Shield

Block diagram with pins used in the Atmega328P

**Task 1-3 for Task 1:**

  
**Task 1-3 for Task 2:**



1. **INITIAL CODE OF DA2A**

**DA2A Task 1:**

#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

}

}

**DA2A 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

}

}

}

1. **DEVELOPED CODE OF TASK 1/C FROM DA2A**

**DA2C Task 1 for Task 1:**

#define *F\_CPU* 16000000UL

#include <avr/io.h>

#include <stdio.h>

int main(void)

{

DDRB = (1<<DDB2); // Make PB2 Output

PORTB = (0<<DDB2); // Turn on LED

TCCR0A = 0; // Normal Mode

TCCR0B = 5; // Set prescaler to 1024

int ovrflow = 0; // overflow counter

while (1)

{

TCNT0 = 0; // Reset counter

ovrflow = 0; // Reset overflow counter

// Delay for 6796

while (ovrflow < 26) { // Gets to 6656

while ((TIFR0 & 0x01) == 0) {}

ovrflow++; // increment ovrflow

TCNT0 = 0; // reset counter

TIFR0 = 1; // reset ovf flag

}

while (TCNT0 < 140) {} // 6656+140 = 6796

PORTB ^= (1<<DDB2); // Turn off LED

TCNT0 = 0; // reset counter

ovrflow = 0; // reset ovrflow counter

// Delay for 4531

while (ovrflow < 17) { // 4352

while ((TIFR0 & 0x01) == 0) {}

ovrflow++; // increment ovrflow

TCNT0 = 0; // reset counter

TIFR0 = 1; // reset ovf flag

}

while (TCNT0 < 179) {} // 4352+179 = 4531

PORTB ^= (1<<DDB2); // Turn on LED

}

}

**DA2C Task 1 for Task 2:**

#define *F\_CPU* 16000000UL

#include <avr/io.h>

#include <stdio.h>

int main(void)

{

DDRB = (1<<DDB2); // Make PB2 Output

DDRC = (0<<DDC2); // Make PC2 Input

PORTB = (1<<DDB2); // Turn off LED

PORTC = (1<<DDC2); // Turn on pull-up transistor

TCCR0A = 0; // Normal Mode

TCCR0B = 5; // Set prescaler to 1024

int ovrflow = 0; // overflow counter

while (1)

{

if (!(PINC & (1 << PINC2))) { // if button pressed

PORTB ^= (1<<DDB2); // Turn on LED

TCNT0 = 0; // Reset counter

ovrflow = 0; // Reset overflow counter

// Delay for 1.25 sec (19531 TCNT)

while (ovrflow < 76) { // Gets to 19456

while ((TIFR0 & 0x01) == 0) {}

ovrflow++; // increment ovrflow

TCNT0 = 0; // reset counter

TIFR0 = 1; // reset ovf flag

}

while (TCNT0 < 75) {} // 19456+75 = 19531

PORTB ^= (1<<DDB2); // Turn off LED

}

}

}

1. **DEVELOPED CODE OF TASK 2/C FROM DA2A**

**DA2C Task 2 for Task 1:**

#define *F\_CPU* 16000000UL

#include <avr/io.h>

#include <stdio.h>

#include <avr/interrupt.h>

int ovrflow = 0; // global ovrflow counter

int main(void)

{

DDRB = (1<<DDB2); // Make PB2 Output

PORTB = (0<<DDB2); // Turn on LED

TIMSK0 |= (1<<TOIE0); // Set up interrupt

TCCR0A = 0; // Normal Mode

sei(); // interrupt enable

TCCR0B = 5; // Set prescaler to 1024

while (1)

{

}

}

ISR (TIMER0\_OVF\_vect) {

ovrflow++; //increment ovrflow

if (ovrflow == 26) { // delay for .435s

TCNT0 = 0;

while (TCNT0 < 140) {}

PORTB ^= (1<<DDB2); // Turn OFF

TCNT0 = 0; // reset counter

}

else if (ovrflow == 43) { // delay for .29s

TCNT0 = 0;

while (TCNT0 < 179) {}

PORTB ^= (1<<DDB2); // Turn ON

ovrflow = 0; // reset ovrflow

TCNT0 = 0; // reset counter

}

TCNT0 = 0;

}

**DA2C Task 2 for Task 2:**

#define *F\_CPU* 16000000UL

#include <avr/io.h>

#include <stdio.h>

#include <avr/interrupt.h>

int ovrflow = 0; // global ovrflow counter

int main(void)

{

DDRB = (1<<DDB2); // Make PB2 Output

DDRC = (0<<DDC2); // Make PC2 Input

PORTB = (1<<DDB2); // Turn off LED

PORTC = (1<<DDC2); // Turn on pull-up transistor

TIMSK0 |= (1<<TOIE0); // Set up interrupt

TCCR0A = 0; // Normal Mode

sei();

TCCR0B = 5; // Set prescaler to 1024

while (1)

{

if (!(PINC & (1 << PINC2))) { // if button pressed

PORTB ^= (1 << DDB2); // Turn on LED

TCNT0 = 0;

ovrflow = 0;

while (!(PORTB & (1<<PORTB2))) {} // while on

}

ovrflow = 0;

}

}

ISR (TIMER0\_OVF\_vect) {

ovrflow++; //increment ovrflow

if (ovrflow == 76) {

TCNT0 = 0;

while (TCNT0 < 75) {}

PORTB ^= (1<<DDB2); // Turn off LED

}

TCNT0 = 0; //reset counter

}

1. **DEVELOPED CODE OF TASK 3/C FROM DA2A**

**DA2C Task 3 for Task 1:**

#define *F\_CPU* 16000000UL

#include <avr/io.h>

#include <stdio.h>

#include <avr/interrupt.h>

int ovrflow = 0; // global ovrflow counter

int main(void)

{

DDRB |= (1<<DDB2); // Make PB2 Output

PORTB &= (0<<DDB2); // Turn on LED

TIMSK0 |= (1<<OCIE0A); // Set up interrupt

TCCR0A |= (1<<WGM01); // Normal Mode

OCR0A = 0xFF;

TCNT0 = 0;

sei(); // interrupt enable

TCCR0B |= (1<<CS02) | (1<<CS00); // Set prescaler to 1024

while (1)

{

}

}

ISR (TIMER0\_COMPA\_vect) {

ovrflow++; //increment ovrflow

if (ovrflow == 26) { // delay for .435s

TCNT0 = 0;

while (TCNT0 < 140) {}

PORTB ^= (1<<DDB2); // Turn OFF

TCNT0 = 0; // reset counter

}

else if (ovrflow == 43) { // delay for .29s

TCNT0 = 0;

while (TCNT0 < 179) {}

PORTB ^= (1<<DDB2); // Turn ON

ovrflow = 0; // reset ovrflow

TCNT0 = 0; // reset counter

}

}

**DA2C Task 3 for Task 2:**

#define *F\_CPU* 16000000UL

#include <avr/io.h>

#include <stdio.h>

#include <avr/interrupt.h>

int ovrflow = 0; // global ovrflow counter

int main(void)

{

DDRB |= (1<<DDB2); // Make PB2 Output

DDRC &= (0<<DDC2); // Make PC2 Input

PORTB |= (1<<DDB2); // Turn off LED

PORTC |= (1<<DDC2); // Turn on pull-up transistor

TIMSK0 |= (1<<OCIE0A); // Set up interrupt

TCCR0A |= (1<<WGM01); // Normal Mode

OCR0A = 0xFF;

TCNT0 = 0;

sei();

TCCR0B |= (1<<CS02) | (1<<CS00); // Set prescaler to 1024

while (1)

{

if (!(PINC & (1 << PINC2))) { // if button pressed

PORTB ^= (1 << DDB2); // Turn on LED

TCNT0 = 0;

ovrflow = 0;

while (!(PORTB & (1<<PORTB2))) {} // while on

}

ovrflow = 0;

}

}

ISR (TIMER0\_COMPA\_vect) {

ovrflow++; //increment ovrflow

if (ovrflow == 76) {

TCNT0 = 0;

while (TCNT0 < 75) {}

PORTB ^= (1<<DDB2); // Turn off LED

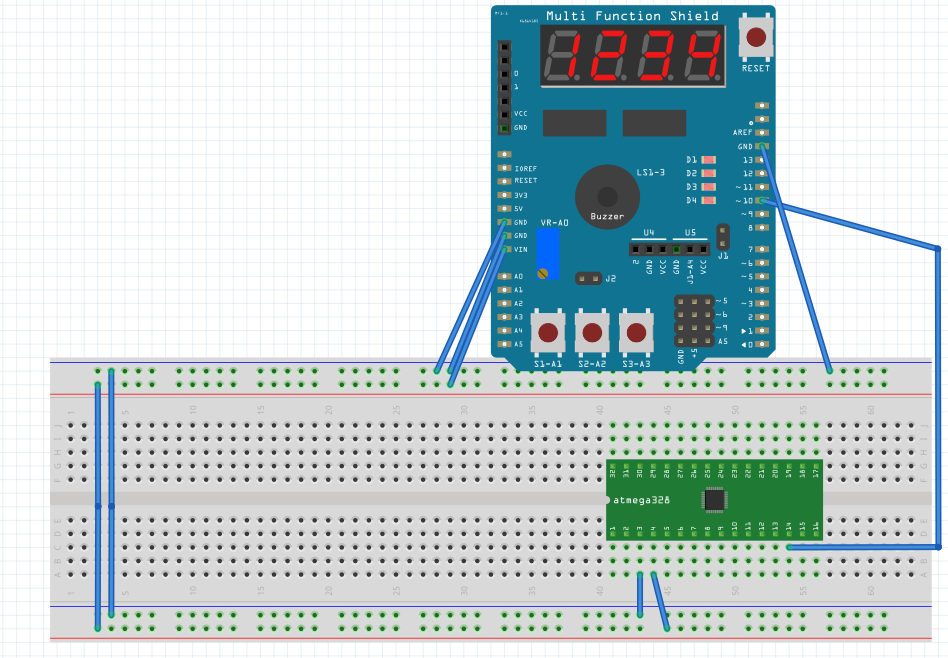
}

TCNT0 = 0; //reset counter

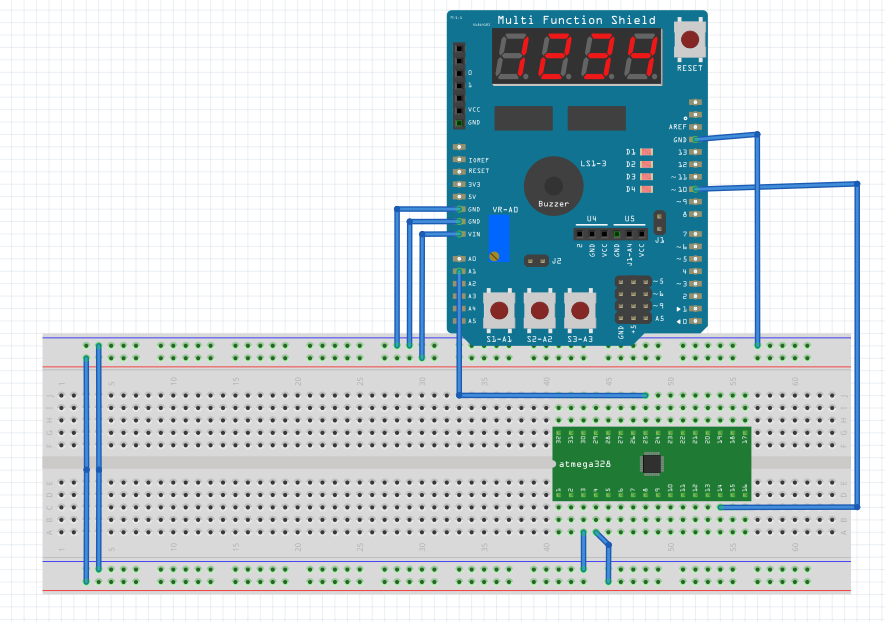
}

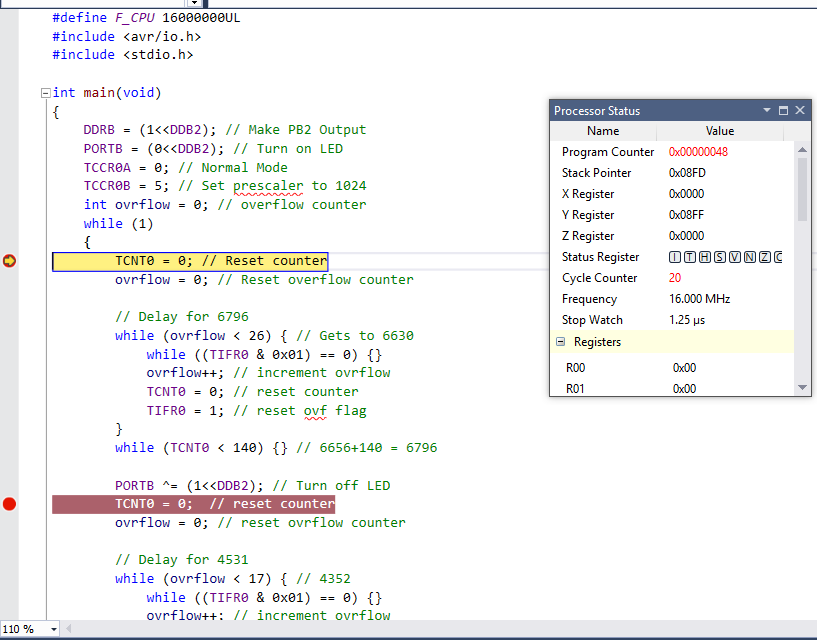
1. **SCHEMATICS**

**Task 1-3 for Task 1:**



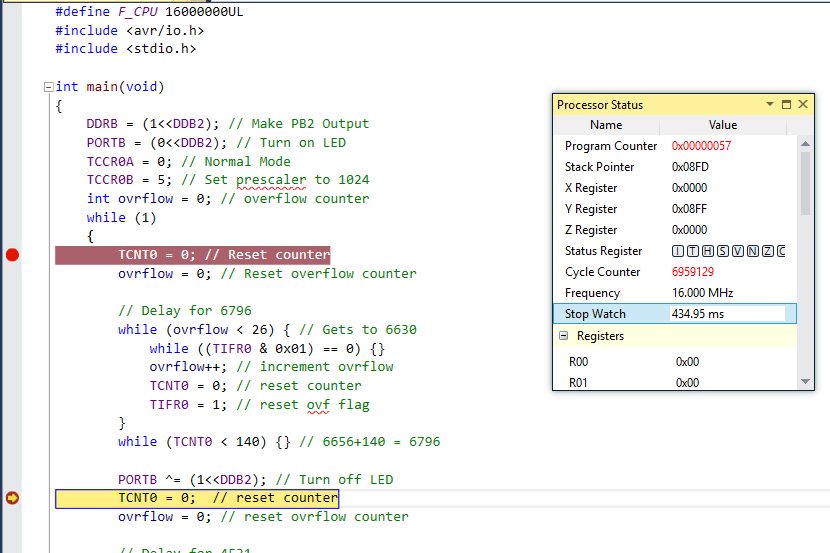
**Task 1-3 for Task 2:**

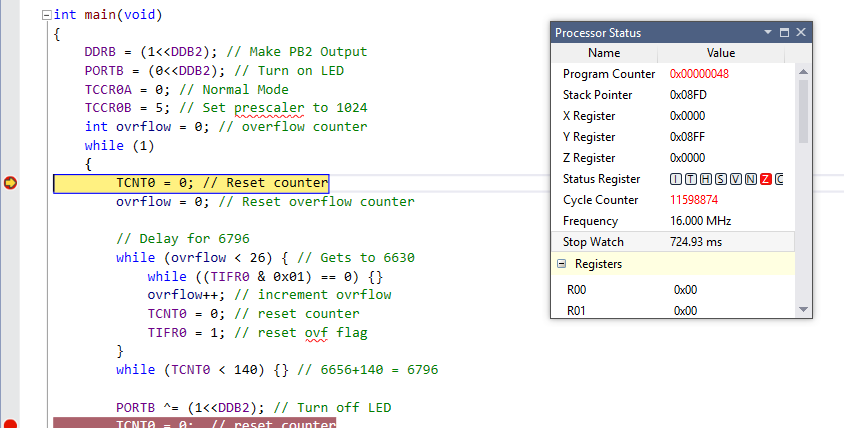
****

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

**DA2C Task 1\_1 (BEFORE):**

**DA2C Task 1\_1 (AFTER):**

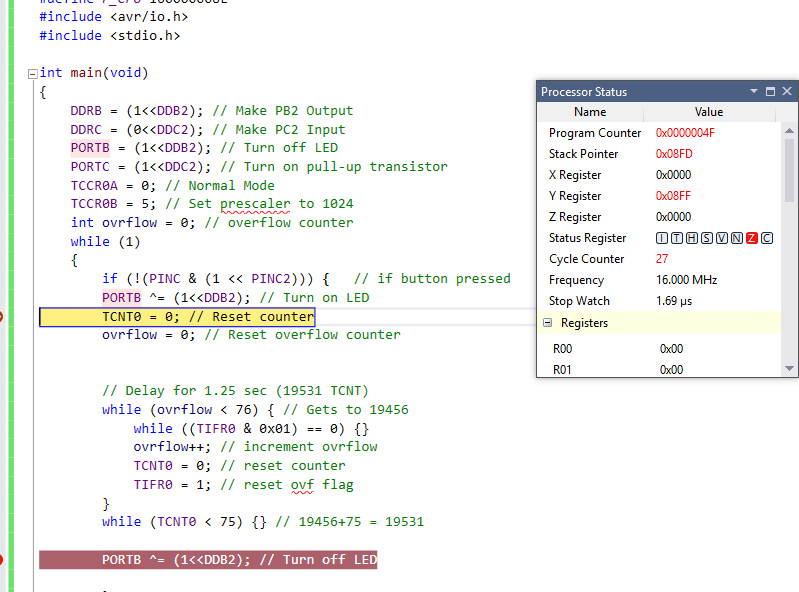




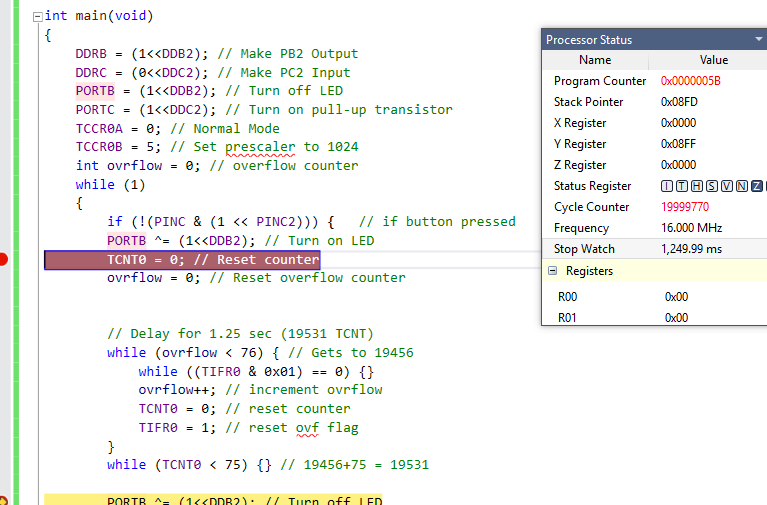
**DA2C Task 1\_1 Waveform:**



**DA2C Task 1\_2 (BEFORE):**

****

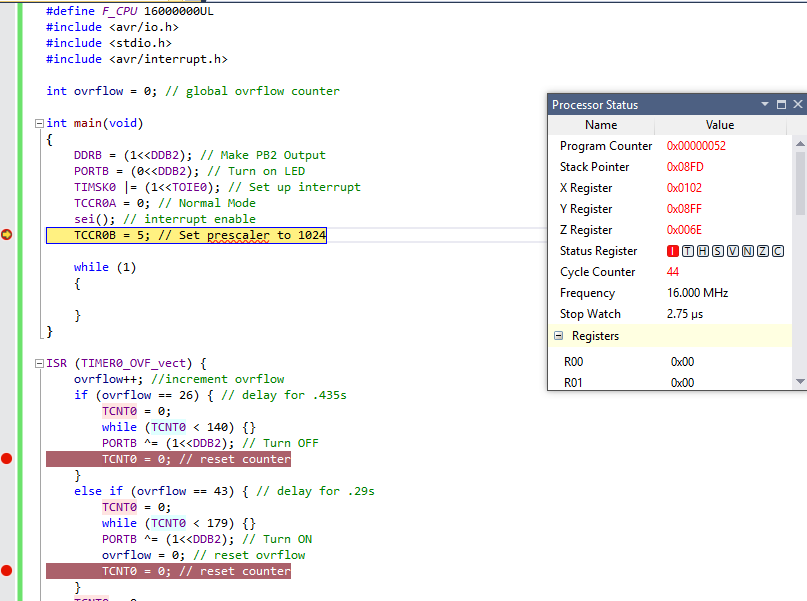
**DA2C Task 1\_2 (AFTER):**

****

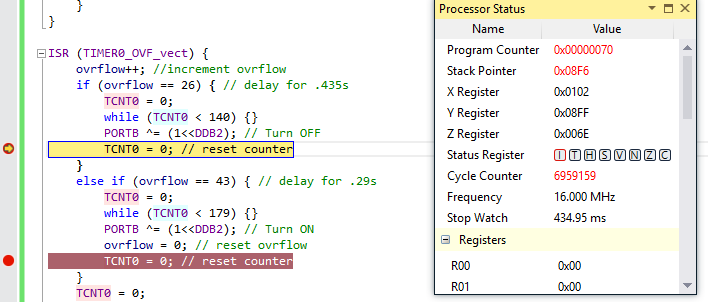
**DA2C Task 1\_2 Waveform:**

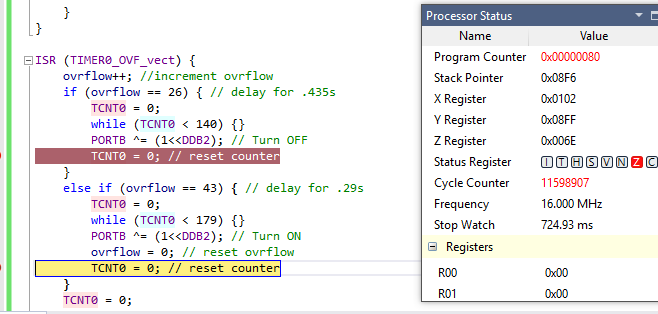


**DA2C Task 2\_1 (BEFORE):**

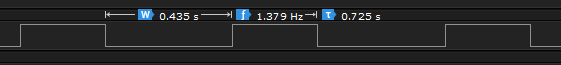
****

**DA2C Task 2\_1 (AFTER):**

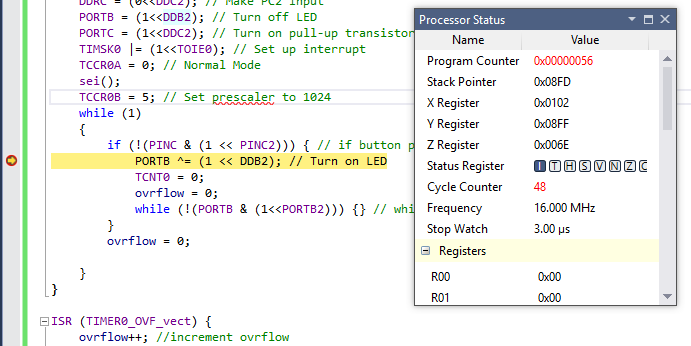
****

****

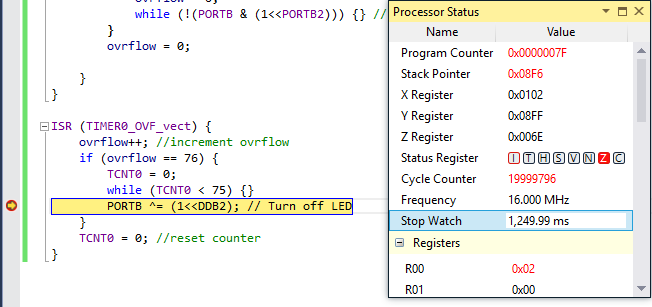
**DA2C Task 2\_1 Waveform:**

****

**DA2C Task 2\_2 (BEFORE):**

****

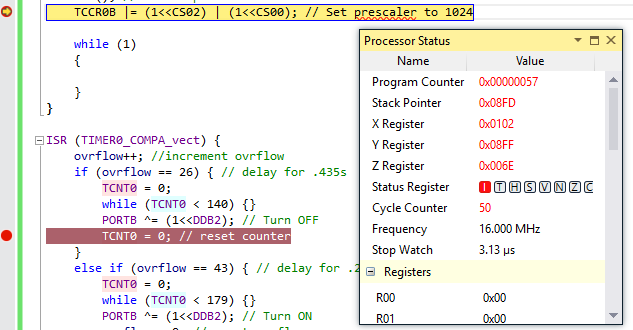
**DA2C Task 2\_2 (AFTER):**

****

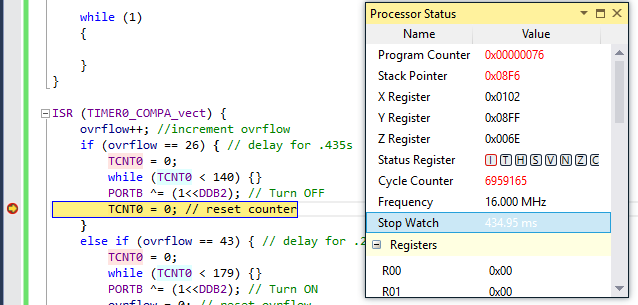
**DA2C Task 2\_2 Waveform:**

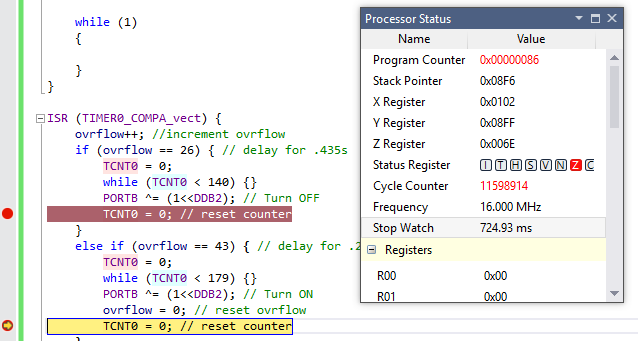
****

**DA2C Task 3\_1 (BEFORE):**

****

**DA2C Task 3\_1 (AFTER):**

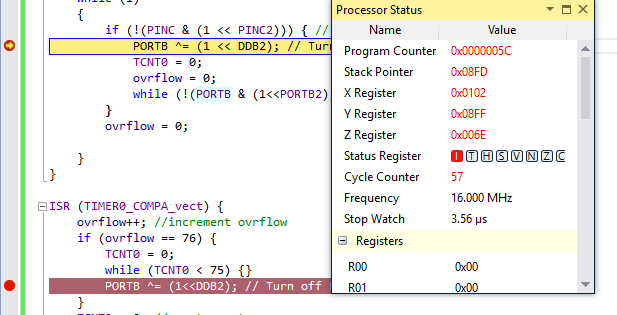
****

****

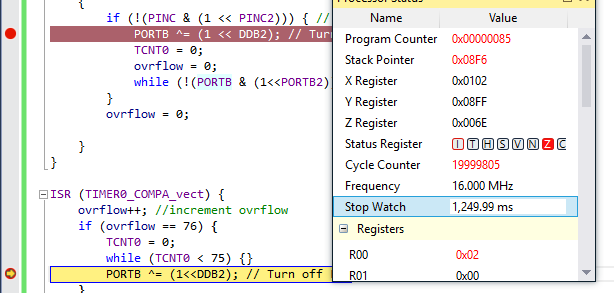
**DA2C Task 3\_1 Waveform:**

****

**DA2C Task 3\_2 (BEFORE):**

****

**DA2C Task 3\_2 (AFTER):**

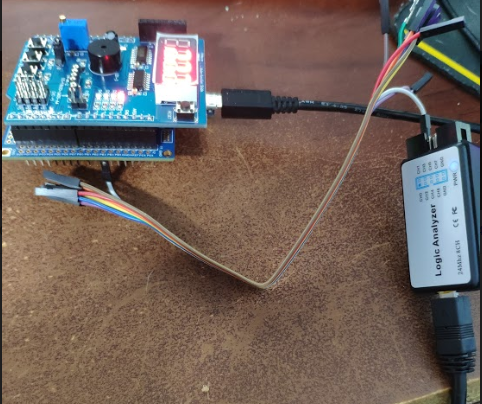
****

**DA2C Task 3\_2 Waveform:**

****

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

**Task 1-3:**



1. **VIDEO LINKS OF EACH DEMO**

<https://youtu.be/hjMDncpwhuI>

1. **GITHUB LINK OF THIS DA**

<https://github.com/recrio/submissions/tree/master/DesignAssignments/DA2C>

**Student Academic Misconduct Policy**

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

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

Ron Joshua Recrio