CPE301 – SPRING 2019

Design Assignment 2C

Student Name: David Flores

Student #: 2000881655

Student Email: flored6@unlv.nevada.edu

Primary Github address: https://github.com/David-Floress/submission\_da.git

Directory :DA2C:

**\*\*\*Big note if C folders do not appear rename first folder (DA2C) to “CPE310L-Lab8” That way the directory it right\*\*\***

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

Arduino Shield

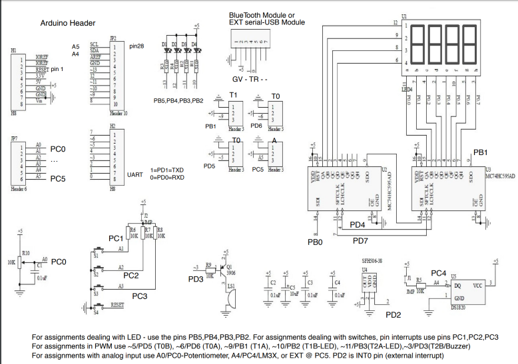
Wires

Stepper Motor

Servo Motor

ULN2003A Driver

Potentiometer



I used PB0 PB1 PB2 and PB3 as output. I had PC0 for the potentiometer

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

Task 1 Part 1:

/\*

\* DA2C.c

\*

\* Created: 5/10/2019 7:42:49 AM

\* Author : user11

\*/

//Task 1 Part1

#define *F\_CPU* 16000000L

#include <avr/io.h>

int main()

{

*uint8\_t* OVFCount = 0; //overflow flag counter

DDRB |= (1<<2); //set pb2 as an output

PORTB |= (1<<2); //led output

TCNT0 = 0x1D; //starts the counter

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

while(1)

{

while((TIFR0 & 0x01) == 0); //while flag is 0

TCNT0 = 0x1D; //starting counter for tcnt0

TIFR0 = 0x01; //resets the overflow flag

//OVFCount++;

if(OVFCount == 19 ) //overflow counter counts up to this

{

PORTB ^= (1<<2); //turns led off

}

else if(OVFCount == 49) //overflow counter counts up to this

{

PORTB ^= (1<<2); //keeps the led on

OVFCount = 0; //resets the overflow count

}

OVFCount++; //increment overflow count

}

return 0;

}

Task 1 Part2

/\*

\* Task 1 part2.c

\*

\* Created: 5/10/2019 8:45:32 AM

\* Author : user11

\*/

// Task 1 part2

#define *F\_CPU* 16000000UL

#include <avr/io.h>

#include <util/delay.h>

int main(void)

{

DDRB |= (1<<2); //sets pb2 as an input

PORTB |= (1<<2); //sets pb2 high for LED

DDRC &= (0<<2); //sets PC2 as an input

PORTC |= (0<<2); //sets PC2 low

TCCR0A = 0;

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

int OVFCount = 0;

while (1)

{

if (!(PINC & (1<<PINC2))) //checks if the pushbutton is pressed

{

OVFCount = 0;

TCNT0 = 0;

}

while ((TIFR0 & 0x01) == 0);

TCNT0 = 0x05; //starts count at this value

TIFR0 = 0x01; //rests the overflow flag

OVFCount++; //overflow flag counter increment

if (OVFCount <= 78) //when the overflow counter is less than or equal

to 78

{

PORTB = (0<<2); //portb2 led is on

}

else

{

PORTB = (1<<2); //portb2 led is off

}

}

return 0;

}

Task 2 Part 1

/\*

\* Task 2 part1.c

\*

\* Created: 5/11/2019 7:51:40 PM

\* Author : user11

\*/

//Task 2 part1

#define *F\_CPU* 16000000UL

#include <avr/io.h>

#include <util/delay.h>

#include <avr/interrupt.h>

int main(void)

{

DDRB |= (1<<2); //sets pb2 as an output

PORTB |= (1<<2); //sets pb2 led

//TCNT0 = 0x1D;

TCNT0 = 0;

TCCR0A = 0;

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

TIMSK0 |= (1<< TOIE0); //enables an interrupt

sei(); //enables global interrupt

while(1)

{

}

}

ISR(TIMER0\_OVF\_vect)

{

*uint8\_t* OVFCount = 0;

while(1)

{

while((TIFR0 & 0x01) == 0); //while flag is 0

TCNT0 = 0x1D; //starting counter for tcnt0

TIFR0 = 0x01; //resets the overflow flag

//OVFCount++;

if(OVFCount == 19 ) //overflow counter counts up to this

{

PORTB ^= (1<<2); //turns led off

}

else if(OVFCount == 49) //overflow counter counts up to this

{

PORTB ^= (1<<2); //this keeps led on

OVFCount = 0; //resets the overflow count

}

OVFCount++; //increment overflow count

}

}

Task 2 Part 2

/\*

\* Task2 part2.c

\*

\* Created: 5/11/2019 9:05:32 PM

\* Author : user11

\*/

// Task 2 part2

#define *F\_CPU* 16000000UL

#include <avr/interrupt.h>

#include <avr/io.h>

int main(void)

{

DDRB |= (1<<2); //sets pb2 as an output

PORTB |= (1<<2); //sets pb2 led

TCNT0 = 0;

TCCR0A = 0;

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

TIMSK0 |= (1<< TOIE0); //enables the interrupt

sei(); //enables global interrupt

while(1)

{

}

}

ISR(TIMER0\_OVF\_vect)

{

int OVFCount = 0;

while (1)

{

if (!(PINC & (1<<PINC2))) //checks if pushbutton is pressed

{

OVFCount = 0;

TCNT0 = 0;

}

while ((TIFR0 & 0x01) == 0);

TCNT0 = 0x05; //starts at this value to count from

TIFR0 = 0x01; //rests the overflow flag

OVFCount++; //overflow flag counter increment

if (OVFCount <= 78) //when overflow counter is less than or equal

to 78

{

PORTB = (0<<2); //then portb2 led is on

}

else

{

PORTB = (1<<2); //then portb2 led is off

}

}

}

Task 3 Part 1

/\*

\* Task3 part1.c

\*

\* Created: 5/11/2019 10:12:17 PM

\* Author : user11

\*/

//Task 3 Part1

#define *F\_CPU* 16000000UL

#include <avr/io.h>

#include <util/delay.h>

#include <avr/interrupt.h>

int main(void)

{

DDRB |= (1<<2); //sets pb2 as an output

PORTB |= (1<<2); //sets for pb2 led

TCNT0 = 0;

TCCR0A = (1<<WGM01); //CTC mode

TCCR0B = (1<<CS00) | (1 << CS02); //sets prescaler to 2014

TIMSK0 |= (1<<OCIE0A); //sets interrupt compare match

sei(); //enables global interrupt

while(1)

{

}

}

ISR(TIMER0\_COMPA\_vect)

{

*uint8\_t* OVFCount = 0;

while(1)

{

while((TIFR0 & 0x01) == 0); //while flag is 0

TCNT0 = 0x1D; //starts the count for TCNT0

TIFR0 = 0x01; //resets the overflow flag

//OVFCount++;

if(OVFCount == 19 ) //overflow counter counts up to this

{

PORTB ^= (1<<2); //turns led off

}

else if(OVFCount == 49) //overflow counter counts up to this

{

PORTB ^= (1<<2); //keeps the led on

OVFCount = 0; //resets the overflow count

}

OVFCount++; //increment overflow count

}

}

Task 3 Part 2

/\*

\* Task3 part2.c

\*

\* Created: 5/11/2019 11:34:04 AM

\* Author : user11

\*/

//DA2CT3\_2.c

#define *F\_CPU* 16000000UL

#include <avr/io.h>

#include <util/delay.h>

#include <avr/interrupt.h>

int main(void)

{

DDRB |= (1<<2); //sets pb2 as an output

PORTB |= (1<<2); //sets pb2 led

TCNT0 = 0;

TCCR0A = (1<<WGM01); //CTC mode

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

TIMSK0 |= (1<<OCIE0A); //sets interrupt on compare match

sei(); //enables global interrupt

while(1)

{

}

}

ISR(TIMER0\_COMPA\_vect)

{

int OVFCount = 0;

while (1)

{

if (!(PINC & (1<<PINC2))) //checks if the pushbutton is pressed

{

OVFCount = 0;

TCNT0 = 0;

}

while ((TIFR0 & 0x01) == 0);

TCNT0 = 0x05; //starts to count from this value

TIFR0 = 0x01; //rests the overflow flag

OVFCount++; //overflow flag counter increment

if (OVFCount <= 78) //when overflow counter is less than or equal

to 78

{

PORTB = (0<<2); //then portb2 led is on

}

else

{

PORTB = (1<<2); //then portb2 led is off

}

}

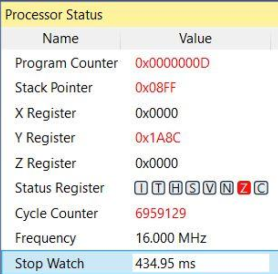
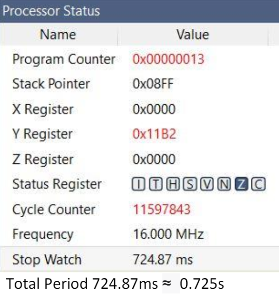
}

1. **SCHEMATICS**

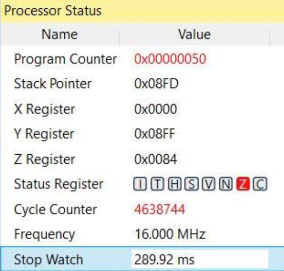
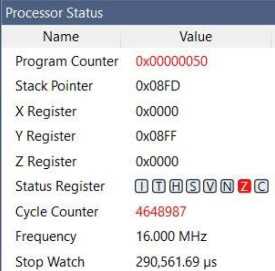
N/A

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

Duty Cycle and Period for Task 1 2 and 3 Part 1

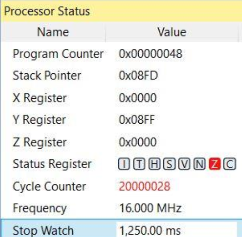
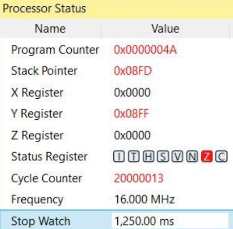
 

60% Duty Cycle Total Period ^

40% Duty Cycle Total Period^

1.25 second Delay

1. **SCREENSHOT OF EACH DEMO (BOARD SETUP)**
2. **VIDEO LINKS OF EACH DEMO**

**Task 1 Part 1 and 2**

<https://www.youtube.com/watch?v=Go8k-Aox40Q>

**Task 2 Part 1 and 2**

<https://www.youtube.com/watch?v=HftV4GAcpeg>

**Task 3 Part 1 and 2**

<https://www.youtube.com/watch?v=R3gXKWCUIII>

1. **GITHUB LINK OF THIS DA**

**Student Academic Misconduct Policy**

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

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

David Flores