CPE301 – SPRING 2019

Design Assignment 2C

Student Name: Mat Tomljenovic

Student #: 1012254710

Student Email: tomljeno@unlv.nevada.edu

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

Directory: https://github.com/matcroatia/DA2B

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

List of components: Internal resistor, Pin C PORT B, Atmega328P

Pins: PB02 and PC02, LED1 and SW1 switch.

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

C CODE 1A:

/\*

\* Assignment 2C.c

\*

\* Created: 3/16/2019 11:28:38 PM

\* Author : tomljeno

\*/

#include <avr/io.h>

#define *F\_CPU* 8000000UL

int main(void)

{

*uint8\_t* OVFCount = 0; //initiates unit overflow counter

DDRB |= (1 << PB2); //sets PB2 as output

TCCR0B = 0x05; //initiates timer0

TCNT0 = 0x00; //initiates counter

while(1)

{

while((TIFR0 & 0x01) == 1) //loop while overflow flag is set

{

TCNT0 = 0x00; //initiate counter to 0

TIFR0 = 0x01; //set timer overflow count to 1

OVFCount++; //increment overflow counter

if (OVFCount == 13) //set overflow count limit

{

PORTB ^= (1 << PB2); //turns on LED

OVFCount = 0; //resets over flow count to 0

while((TIFR0 & 0x01) == 1) //loop while overflow flag is set

{

TCNT0 = 0x68; //counter initiated at 0x68

TIFR0 = 0x01; //set timer overflow count to 1

OVFCount++; //increment overflow counter

if (OVFCount == 9) //set overflow count limit

{

PORTB &= ~(1<<PB2); //turn off LED

OVFCount = 0; //reset overflow counter to 0

}

}

}

}

}

}

C CODE 1B:

/\*

\* DA2C 1B.c

\*

\* Created: 3/23/2019 6:36:03 PM

\* Author : Mat Tomljenovic

\*/

#include <avr/io.h>

#define *F\_CPU* 8000000UL

int main(void)

{

*uint8\_t* OVFCount = 0; //initiates unit overflow counter

TCCR0B = 0x05; //initiates timer0

TCNT0 = 0x00; //initiates counter

DDRB = 0xFF; //sets PB2 as output

while(1) //loop for while LED on

{

//while((TIFR0 & 0x01) == 1) //loop while overflow flag is set

//{

if ((PINC & (1 << PC2)) == (1<<PC2)) //if to light up LED when button is pressed

{

PORTB = 0xFF; //Turns ON LED

TCNT0 = 0x00; //initiate counter to 0

TIFR0 = 0x01; //set timer overflow count to 1

OVFCount++; //increment overflow counter

if (OVFCount == 39) //set overflow count limit

{

PORTB = 0;

OVFCount = 0; //Turns OFF LED

}

}

//}

}

}

C CODE 2A:

/\*

\* Assignment 2C.c

\*

\* Created: 3/16/2019 11:28:38 PM

\* Author : tomljeno

\*/

#define *F\_CPU* 8000000UL

#include <avr/io.h>

#include <avr/interrupt.h>

int main(void)

{

DDRB |= (1 << PB2); //sets PB2 as output

TCCR0B = 0x05; //initiates timer0

TCNT0 = 0x00; //initiates counter

TIMSK0 |= (1 << TOIE0); //enable external interrupt 0

sei(); //enable interrupts

while(1)

{

}

}

ISR (TIMER0\_OVF\_vect) //ISR for external interrupt 0

{

*uint8\_t* OVFCount = 0; //initiates unit overflow counter

while((TIFR0 & 0x01) == 1) //loop while overflow flag is set

{

TCNT0 = 0x00; //initiate counter to 0

TIFR0 = 0x01; //set timer overflow count to 1

OVFCount++; //increment overflow counter

if (OVFCount == 13) //set overflow count limit

{

PORTB ^= (1 << PB2); //turns on LED

OVFCount = 0; //resets over flow count to 0

while((TIFR0 & 0x01) == 1) //loop while overflow flag is set

{

TCNT0 = 0x68; //counter initiated at 0x68

TIFR0 = 0x01; //set timer overflow count to 1

OVFCount++; //increment overflow counter

if (OVFCount == 9) //set overflow count limit

{

PORTB &= ~(1<<PB2); //turn off LED

OVFCount = 0; //reset overflow counter to 0

}

}

}

}

}

C CODE 2B:

/\*

\* DA2C 2B.c

\*

\* Created: 3/23/2019 9:27:39 PM

\* Author : Some guy

\*/

#define *F\_CPU* 8000000UL

#include <avr/io.h>

#include <avr/interrupt.h>

#include <util/delay.h>

int main(void)

{

DDRB |= (1 << PB2); //sets PB2 as output

TCCR0B = 0x05; //initiates timer0

TCNT0 = 0x00; //initiates counter

TIMSK0 |= (1 << TOIE0); //enable external interrupt 0

sei(); //enable interrupts

while(1)

{

}

}

ISR (TIMER0\_OVF\_vect) //ISR for external interrupt 0

{

DDRB |= (1 << PB2); //Makes first pin of PORTB as Output

DDRC &= ~(1<<PC2);//Makes first pin of PORTC as Input

DDRB = 0xFF; //sets PB2 as output

while(1) //loop for while LED on

{

if (PINC & (1<<PC2) == 1) //if to light up LED when button is pressed

{

PORTB = 0xFF; //Turns ON LED

*\_delay\_ms*(1200); //1.2 second delay

PORTB = 0; //Turns OFF LED

}

}

}

C CODE 3A:

/\*

\* DA2C 3A.c

\*

\* Created: 3/23/2019 10:38:50 PM

\* Author : Mat Tomljenovic

\*/

#define *F\_CPU* 16000000UL

#include <avr/io.h>

#include <util/delay.h>

#include <avr/interrupt.h>

int main(void)

{

DDRB |= (1 << PB2); //sets POTRB as output

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

TCNT0 = 0; //initiate timer

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

TCCR0B |= 0x05; //enable timer 0

sei(); //enable interrupt

while (1)

{

}

}

*uint8\_t* OVFCount=0; //intialize overflo counter

ISR (TIMER0\_COMPA\_vect) //timer0 overflow interrupt

{

TCNT0=0x00; //resets the timer

OVFCount++; //increment overflow count

if (OVFCount == 12) //overflow for delay off

{

PORTB= ~(1 << PORTB2); //turn LED off

}

if(OVFCount == 22) //overflow delay on

{

PORTB = (1 << PORTB2); //turns LED on

OVFCount = 0; //reset overflow counter

}

}

C CODE 3B:

/\*

\* DA2C 3B.c

\*

\* Created: 3/23/2019 10:52:37 PM

\* Author : Mat Tomljenovic

\*/

#include <avr/io.h>

#define *F\_CPU* 16000000UL

#include <util/delay.h>

#include <avr/interrupt.h>

int main(void)

{

DDRB |= (1 << PB2); //set PB2 as output port

PORTC = (1 << PINC2); //set PINC as input

TCNT0 = 0; //initiate timer

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

TCCR0B |= 0x05; //set pre-scaler to 1024

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

sei (); //enable interrupt

while (1)

{

}

}

ISR(TIMER0\_COMPA\_vect) //compare interrupt loop

{

*uint8\_t* OVFCount=0; //initiate overflow count

TCNT0=0x00; //reset the timer to zero

OVFCount++; //increment overflow count

DDRB = 0x02; //sets PB2 as output

while(1) //loop for while LED on

{

if ((PINC & (1 << PC2)) == (1<<PC2)) //if to light up LED when button is pressed

{

PORTB = 0xFF; //Turns ON LED

*\_delay\_ms*(600); //1.2 second delay

PORTB = 0; //Turns OFF LED

}

}

}

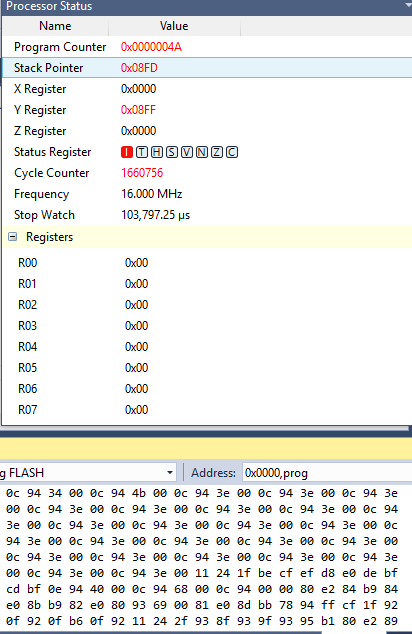
1. **DEVELOPED MODIFIED CODE OF TASK 2/A from TASK 1/A**

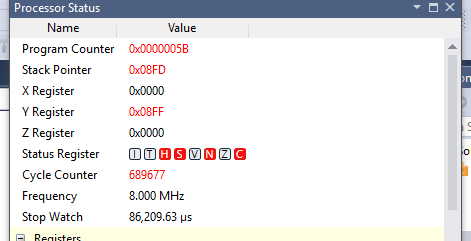
Same as above.

1. **SCHEMATICS**

Use fritzing.org

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





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

<https://www.youtube.com/watch?v=Kh4f8_boG_4>

<https://www.youtube.com/watch?v=boSG3OWyR7M>

<https://www.youtube.com/watch?v=x71AFqKIAZo>

<https://www.youtube.com/watch?v=FGN0Wt88Wus>

<https://www.youtube.com/watch?v=bmX4T3SJV30>

<https://www.youtube.com/watch?v=nNc04HObMYk>

1. **GITHUB LINK OF THIS DA**

https://github.com/matcroatia/DA2C

**Student Academic Misconduct Policy**

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

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

Mat Tomljenovic