CPE301 – SPRING 2019

Design Assignment 2CT1

Student Name: Billy Maddex

Student #: 2000928390

Student Email: [maddex@unlv.nevada.edu](mailto:maddex@unlv.nevada.edu)

Primary Github address: <https://github.com/billymaddex/fluffy-chainsaw>

Directory: DA2C/DA2CT1

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**

Atmega328PB-XMINI

Multi-Function Arduino Module

1. **INITIAL/MODIFIED/DEVELOPED CODE OF TASK 2/B**

/\*

\* DA2BT2.c

\*

\* Created: 10/5/2019 2:28:51 PM

\* Author : Billy

\*/

#define F\_CPU 1600000UL

#include <avr/io.h>

#include <avr/interrupt.h>

#include <util/delay.h>

// long flash on secondary LED

int light2 (void)

{

// turn the light on

PORTB &= ~(1 << 2);

// call delay

\_delay\_ms(13330);

// turn the light off

PORTB |= (1 << 2);

return 0;

}

// external interrupt handler

ISR (INT0\_vect)

{

if (~PIND & (1 << 2))

{

light2();

}

}

// pin change interrupt handler

ISR (PCINT1\_vect)

{

if (~PINC & (1 << 3))

{

light2();

}

}

int main(void)

{

// Set DDR B,C,D to known initial value

DDRB = 0x00;

DDRC = 0x00;

DDRD = 0x00;

// set PORT B,C,D to know initial value

PORTB = 0xFF;

PORTC = 0xFF;

PORTD = 0xFF;

// set PORTB 2 and 3 to output

DDRB |= (1 << 3);

DDRB |= (1 << 2);

// set PINC 3 to input

DDRC &= ~(1 << 3);

// configure interrupt handling

// enable INT0, falling edge trigger

EIMSK = (1 << INT0);

EICRA = 0x02;

// enable PCINT11 (PINC.3)

PCMSK1 = 0x08;

PCICR = (1 << PCIE1);

// enable global interrupt

sei();

// LED pulse

while (1)

{

// turn the light on

PORTB &= ~(1 << 3);

// call delay

\_delay\_ms(2500);

// turn the light off

PORTB |= (1 << 3);

// call delay

\_delay\_ms(3750);

// if switch is pressed, flash the other light

/\*if (~PINC & (1 << 3))

{

light2();

}\*/

}

}

1. **DEVELOPED MODIFIED CODE OF TASK 1/C from TASK 2/B**

/\*

\* DA2CT1.c

\*

\* Created: 10/12/2019 9:46:07 AM

\* Author : Billy

\*/

#define F\_CPU 16000000UL

#include <avr/io.h>

#include <avr/interrupt.h>

// timer based delay function, clock 0

int clock0\_ms (int ms)

{

// set clock 1 to normal operation

TCCR0A = 0x00;

// 250 ticks

TCNT0 = 0x06;

// start clock 1 with prescalar = 64

TCCR0B |= (1 << CS01) | (1 << CS00);

// run the clock for 0.001s ms times

for (int i = ms; i > 0; i--)

{

// poll the clock

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

// 250 ticks

TCNT0 = 0x06;

// reset overflow flag

TIFR0 = 0x01;

}

// stop clock 1

TCCR0B &= ~(1 << CS01) & ~(1 << CS00);

return 0;

}

// timer based delay function, clock 0

int clock2\_ms (int ms)

{

// set clock 2 to normal operation

TCCR2A = 0x00;

// 250 ticks

TCNT2 = 0x06;

// start clock 2 with prescalar = 64

TCCR2B |= (1 << CS22);

// run the clock for 0.001s ms times

for (int i = ms; i > 0; i--)

{

// poll the clock

while ((TIFR2 & 0x01) == 0);

// 250 ticks

TCNT2 = 0x06;

// reset overflow flag

TIFR2 = 0x01;

}

// stop clock 2

TCCR2B &= ~(1 << CS22);

return 0;

}

// long flash on secondary LED

int light2 (void)

{

// turn the light on

PORTB &= ~(1 << 2);

// call delay

clock2\_ms(1333);

// turn the light off

PORTB |= (1 << 2);

return 0;

}

// external interrupt handler

ISR (INT0\_vect)

{

if (~PIND & (1 << 2)) light2();

}

// pin change interrupt handler

ISR (PCINT1\_vect)

{

if (~PINC & (1 << 3)) light2();

}

int main(void)

{

// Set DDR B,C,D to known initial value

DDRB = 0x00;

DDRC = 0x00;

DDRD = 0x00;

// set PORT B,C,D to know initial value

PORTB = 0xFF;

PORTC = 0xFF;

PORTD = 0xFF;

// set PORTB 2 and 3 to output

DDRB |= (1 << 3);

DDRB |= (1 << 2);

// set PINC 3 to input

DDRC &= ~(1 << 3);

// configure interrupt handling

// enable INT0, falling edge trigger

EIMSK = (1 << INT0);

EICRA = 0x02;

// enable PCINT11 (PINC.3)

PCMSK1 = 0x08;

PCICR = (1 << PCIE1);

// enable global interrupt

sei();

// LED pulse

while (1)

{

// turn the light on

PORTB &= ~(1 << 3);

// call timer based delay

clock0\_ms(250);

// turn the light off

PORTB |= (1 << 3);

// call delay

clock0\_ms(375);

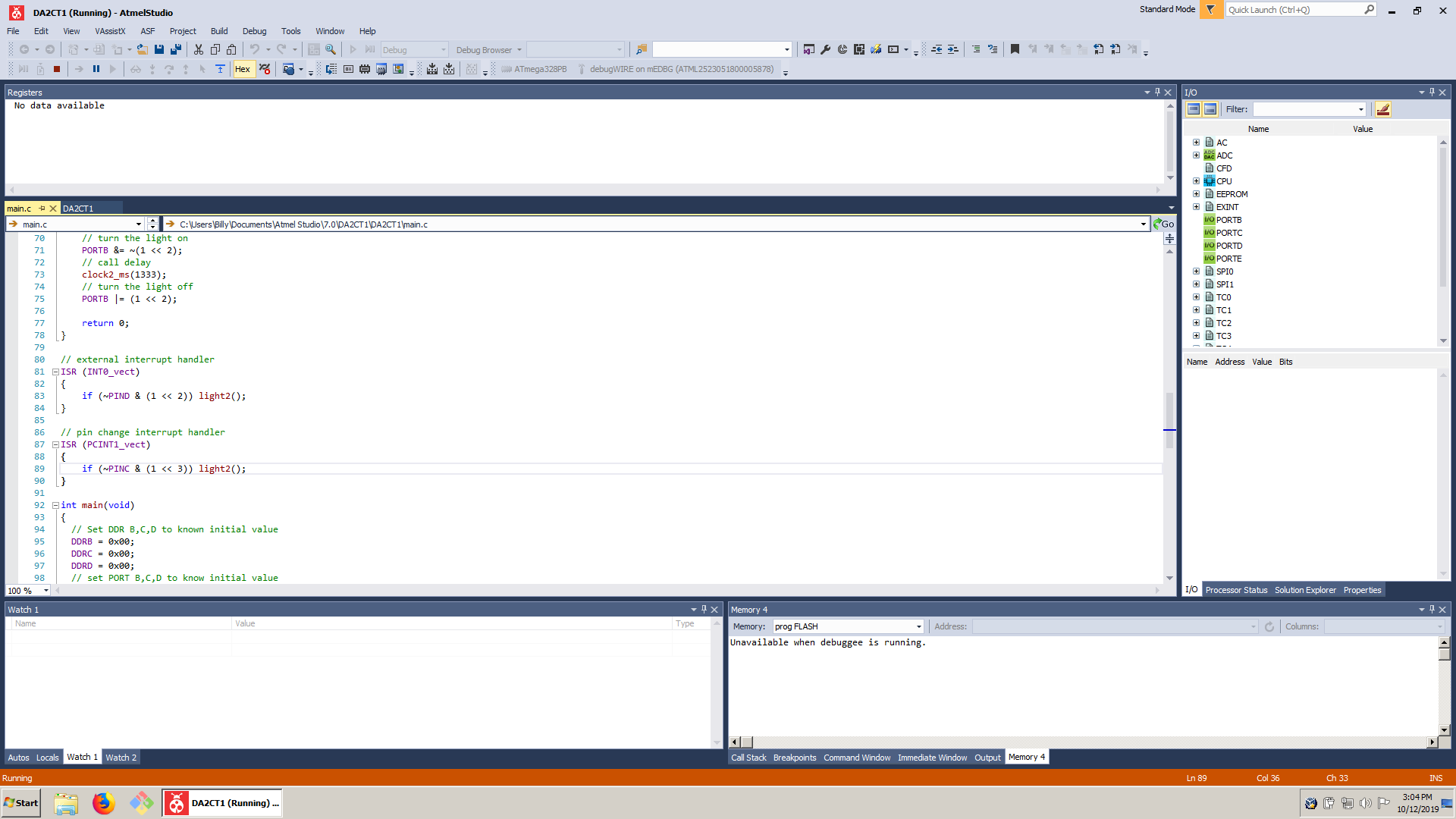
}

}

1. **SCHEMATICS**

Use fritzing.org

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



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



1. **VIDEO LINKS OF EACH DEMO**

https://youtu.be/OZI-edV0L1o

1. **GITHUB LINK OF THIS DA**

<https://github.com/billymaddex/fluffy-chainsaw/tree/master/DesignAssignments/DA2C/DA2CT1>

**Student Academic Misconduct Policy**

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

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

Billy Maddex