CPE301 – SPRING 2019

Design Assignment 3B

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: DA3B

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 3A**

/\*

\* DA3A.c

\*

\* Created: 10/19/2019 9:26:49 PM

\* Author : Billy

\*/

// 16MHz CPU clock

#define F\_CPU 16000000UL

// 9600 baud rate setting

#define UBRR\_9600 103

#include <avr/io.h>

#include <avr/interrupt.h>

#include <util/atomic.h>

#include <stdlib.h>

#include <stdio.h>

// global constants and variables

// terminal output variables

int ival;

float fval;

char sendme[80];

// terminal output function declarations

void uart\_init(unsigned int ubrr);

void uart\_send\_string(char \*data);

// TIMER1 CTC interrupt handler

ISR (TIMER1\_COMPA\_vect )

{

// get a random integer

ival = rand();

fval = (float) ival / rand();

// print to the terminal: a string, a random int, a floating point

ATOMIC\_BLOCK(ATOMIC\_FORCEON)

{

snprintf(sendme, sizeof(sendme), "A string every second!\nA random integer: %d\nA floating point: %f\r\n\r\n", ival, fval);

uart\_send\_string(sendme);

}

}

int main(void)

{

// configure terminal and test output

uart\_init(UBRR\_9600);

// configure interrupt handling

// enable TIMER1 COMPA interrupt

TIMSK1 |= (1 << OCIE1A);

// set TIMER1 to CTC operations

TCCR1B |= (1 << WGM12);

// TIMER1 COMPA = X

OCR1A = 15620;

// start TIMER1 with prescalar = 1024

TCCR1B |= (1 << CS12) | (1 << CS10);

// enable global interrupt

sei();

// loop forever

while (1);

}

// configure terminal output

void uart\_init(unsigned int ubrr)

{

// set buad rate to 9600

UBRR0H = (unsigned char)(ubrr>>8);

UBRR0L = (unsigned char)ubrr;

// set terminal to output

UCSR0B = (1 << TXEN0);

// set ouput mode

UCSR0C = (1 << UCSZ01) | (1 << UCSZ00);

}

void uart\_send\_string(char \*data)

{

while((\*data != '\0'))

{

while (!(UCSR0A & (1 <<UDRE0)));

UDR0 = \*data;

data++;

}

}

1. **DEVELOPED MODIFIED CODE OF TASK B from TASK A**

/\*

\* DA3A.c

\*

\* Created: 10/19/2019 9:26:49 PM

\* Author : Billy

\*/

// 16MHz CPU clock

#define F\_CPU 16000000UL

// 9600 baud rate setting

#define UBRR\_9600 103

#include <avr/io.h>

#include <avr/interrupt.h>

#include <util/atomic.h>

#include <stdlib.h>

#include <stdio.h>

// terminal output function declarations

void uart\_init(unsigned int ubrr);

void uart\_send\_string(char \*data);

// ADC function declarations

void adc\_init (void);

// TIMER1 CTC interrupt handler

ISR(TIMER1\_COMPA\_vect)

{

// start ADC conversion

ADCSRA |= (1 << ADSC);

}

ISR(ADC\_vect)

{

// collect and compute ADC value

// value stored in ADC

//uint16\_t voltage = ADC;

//float temp = (((5000 / 1024) \* voltage) / 1000.0);

float cel = 5 \* ADC / 1000.0;

float fahr = 1.8 \* cel + 32;

// output string

char report[80];

// generate output string

snprintf(report, sizeof(report), "Current temperature: %.1f° C\t%.1f° F\r\n\r\n", cel, fahr);

// output to terminal

ATOMIC\_BLOCK(ATOMIC\_FORCEON)

{

uart\_send\_string(report);

}

}

int main(void)

{

// configure terminal

uart\_init(UBRR\_9600);

// configure and start ADC

adc\_init();

// configure interrupt handling

// enable TIMER1 COMPA interrupt

TIMSK1 |= (1 << OCIE1A);

// set TIMER1 to CTC operations

TCCR1B |= (1 << WGM12);

// TIMER1 COMPA = X

OCR1A = 15620;

// start TIMER1 with prescalar = 1024

TCCR1B |= (1 << CS12) | (1 << CS10);

// enable global interrupt

sei();

// loop forever

while (1);

}

// configure terminal output

void uart\_init(unsigned int ubrr)

{

// set buad rate to 9600

UBRR0H = (unsigned char)(ubrr>>8);

UBRR0L = (unsigned char)ubrr;

// set terminal to output

UCSR0B = (1 << TXEN0);

// set ouput mode

UCSR0C = (1 << UCSZ01) | (1 << UCSZ00);

}

void uart\_send\_string(char \*data)

{

while((\*data != '\0'))

{

while (!(UCSR0A & (1 <<UDRE0)));

UDR0 = \*data;

data++;

}

}

// configure ADC input

void adc\_init (void)

{

// set PORTC to input for ADC

DDRC = 0x00;

ADMUX =

// Reference Selection Bits

// AVcc with external capacitor at AREF

(0 << REFS1) |

(1 << REFS0) |

// ADC Left Adjust Result

(1 << ADLAR) |

// Analog Channel Selection Bits

// ADC4 (PC4 PIN27)

(1 << MUX2) |

(0 << MUX1) |

(1 << MUX0);

ADCSRA =

// ADC ENable

(1 << ADEN) |

// ADC Start Conversion

(0 << ADSC) |

// ADC Auto Trigger Enable

(0 << ADATE) |

// ADC Interrupt Flag

(0 << ADIF) |

// ADC Interrupt Enable

(1 << ADIE) |

// ADC Prescaler Select Bits

// prescalar = 128

(1 << ADPS2) |

(1 << ADPS1) |

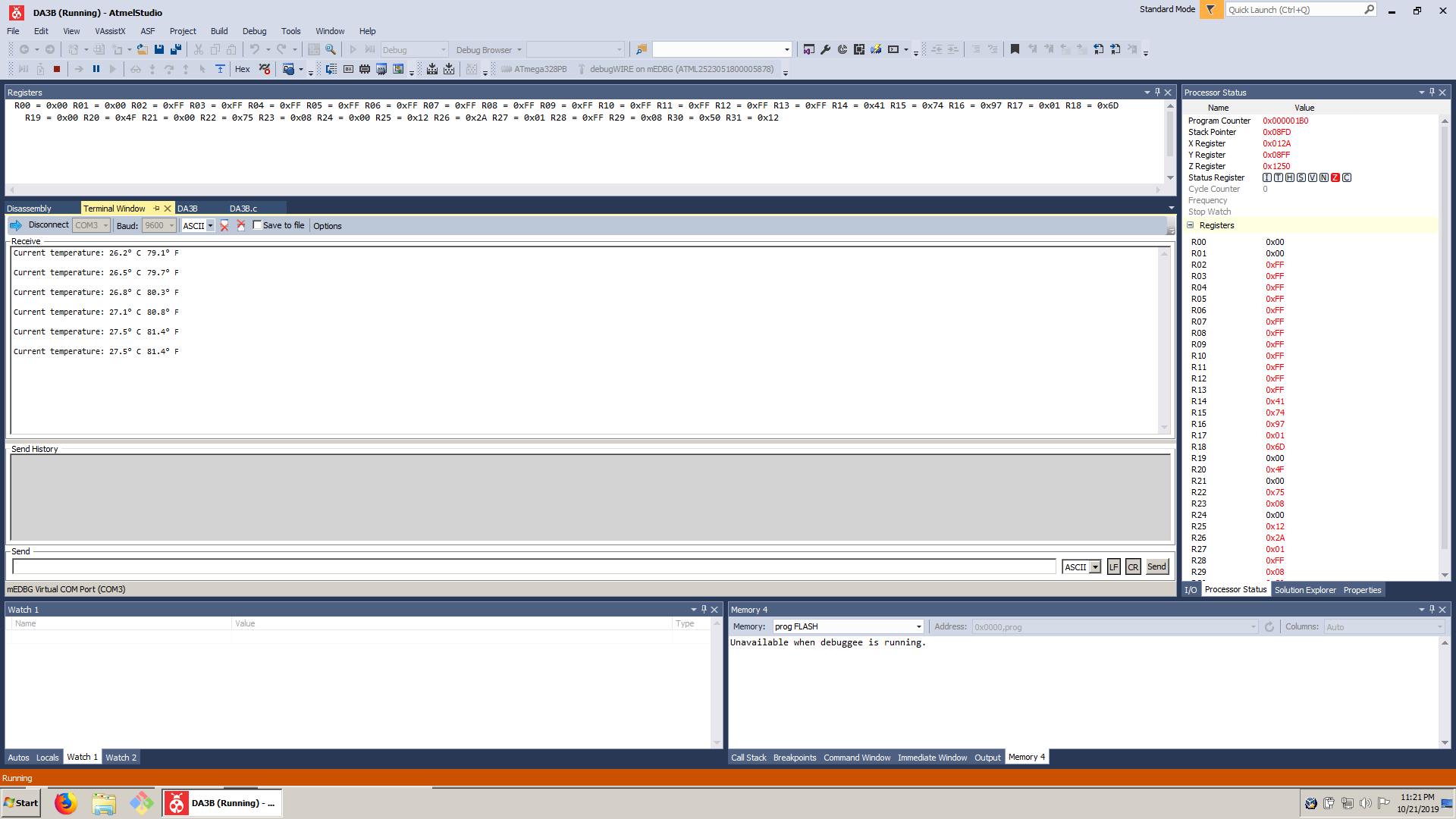
(1 << ADPS0);

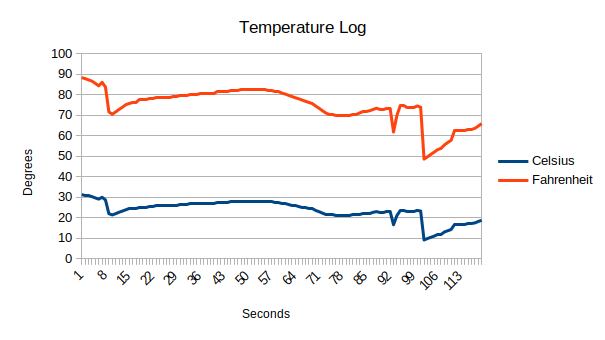
}

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://youtu.be/IwnsnzIeUsI

1. **GITHUB LINK OF THIS DA**

<https://github.com/billymaddex/fluffy-chainsaw/tree/master/DesignAssignments/DA3B>

**Student Academic Misconduct Policy**

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

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

Billy Maddex