CPE301 – SPRING 2019

Design Assignment 4B

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: DA4B

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

Breadboard Arduino Module

1K Ohm Potentiometer

Stepper Motor and Driver

1. **INITIAL/MODIFIED/DEVELOPED CODE OF TASK 4BT1**

/\*

\* DA4B.c

\*

\* Created: 12/3/2019 11:42:58 PM

\* Author : Billy

\*/

#define F\_CPU 16000000UL

#include <avr/io.h>

#include <avr/interrupt.h>

// global variable to control motor step

int step = 0;

// ADC conversion complete vector

// update CTC COMPA on each conversion

ISR(ADC\_vect)

{

unsigned int voltage = ADCH;

// TIMER1 COMPA = X

OCR1A = (( (float) voltage / 255.0) \* 1480) + 80;

}

// TIMER1 CTC interrupt handler

ISR(TIMER1\_COMPA\_vect)

{

// check step variable to move through full step sequence

if (step == 0)

{

PORTD = 0x09;

}

else if (step == 1)

{

PORTD = 0x03;

}

else if (step == 2)

{

PORTD = 0x06;

}

else if (step == 3)

{

PORTD = 0x0C;

}

// increment step

step++;

// reset step when greater than 3

if (step > 3)

{

step = 0;

}

}

// configure TIMER1 for CTC mode with interrupt enabled, prescalar 1024

void CTC\_init()

{

// configure interrupt handling

// enable TIMER1 COMPA interrupt

TIMSK1 |= (1 << OCIE1A);

// set safe initial value for COMPA

OCR1A = 1560;

// set TIMER1 to CTC operations

TCCR1B |= (1 << WGM12);

// start TIMER1 with prescalar = 1024

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

}

// enable ADC on PINC5, free running, interrupt enable

void ADC\_init()

{

ADMUX =

// Reference Selection Bits

// AVcc with external capacitor at AREF

(0 << REFS1) |

(1 << REFS0) |

// ADC Left Adjust Result

(1 << ADLAR) |

// Analog Channel Selection Bits

// ADC5 (PC5 PIN28)

(0 << MUX3) |

(1 << MUX2) |

(0 << MUX1) |

(1 << MUX0);

ADCSRA =

// ADC ENable

(1 << ADEN) |

// ADC Start Conversion

(1 << ADSC) |

// ADC Auto Trigger Enable

(1 << ADATE) |

// ADC Interrupt Flag

(0 << ADIF) |

// ADC Interrupt Enable

(1 << ADIE) |

// ADC Prescaler Select Bits

// prescalar = 128

(1 << ADPS2) |

(1 << ADPS1) |

(1 << ADPS0);

ADCSRB =

// disable analog comparator

(0 << ACME) |

// enable free running mode

(0 << ADTS2) |

(0 << ADTS1) |

(0 << ADTS0);

}

int main(void)

{

// initialize ADC

ADC\_init();

// initialize TIMER1

CTC\_init();

// enable those interrupts

sei();

// set PORTD 0,1,2,3 to output

DDRD = 0x0F;

// the usual infinite loop

while (1);

}

1. **DEVELOPED MODIFIED CODE OF TASK 4BT2**

/\*

\* DA4BT2.c

\*

\* Created: 12/4/2019 10:59:47 AM

\* Author : Billy

\*/

#define F\_CPU 16000000UL

#include <avr/io.h>

#include <avr/interrupt.h>

// ADC conversion complete vector

// update CTC COMPA on each conversion

ISR(ADC\_vect)

{

unsigned int voltage = ADCH;

// TIMER2 COMPB = X

// uses top 4 bits of ADC to select value from 16 to 32

OCR2B = (voltage >> 4) + 16;

}

// configure TIMER2 for PWM mode with with 50 Hz frequency. prescalar 1024, COMP2A 156

void PWM\_init()

{

// set PORTD3 to output for PWM

DDRD = (1 << DDRD3);

OCR2A = 156;

TCCR2B |= (1 << WGM22) | (1 << CS22) | (1 << CS21) | (1 << CS20);

TCCR2A |= (1 << COM2A1) | (0 << COM2A0) | (1 << COM2B1) | (0 << COM2B0) | (1 << WGM21) | (1 << WGM20);

}

// enable ADC on PINC5, free running, interrupt enable

void ADC\_init()

{

ADMUX =

// Reference Selection Bits

// AVcc with external capacitor at AREF

(0 << REFS1) |

(1 << REFS0) |

// ADC Left Adjust Result

(1 << ADLAR) |

// Analog Channel Selection Bits

// ADC5 (PC5 PIN28)

(0 << MUX3) |

(1 << MUX2) |

(0 << MUX1) |

(1 << MUX0);

ADCSRA =

// ADC ENable

(1 << ADEN) |

// ADC Start Conversion

(1 << ADSC) |

// ADC Auto Trigger Enable

(1 << ADATE) |

// ADC Interrupt Flag

(0 << ADIF) |

// ADC Interrupt Enable

(1 << ADIE) |

// ADC Prescaler Select Bits

// prescalar = 128

(1 << ADPS2) |

(1 << ADPS1) |

(1 << ADPS0);

ADCSRB =

// disable analog comparator

(0 << ACME) |

// enable free running mode

(0 << ADTS2) |

(0 << ADTS1) |

(0 << ADTS0);

}

int main(void)

{

// initialize ADC

ADC\_init();

// enable those interrupts

sei();

// initialize PWM TIMER2

PWM\_init();

// the usual infinite loop

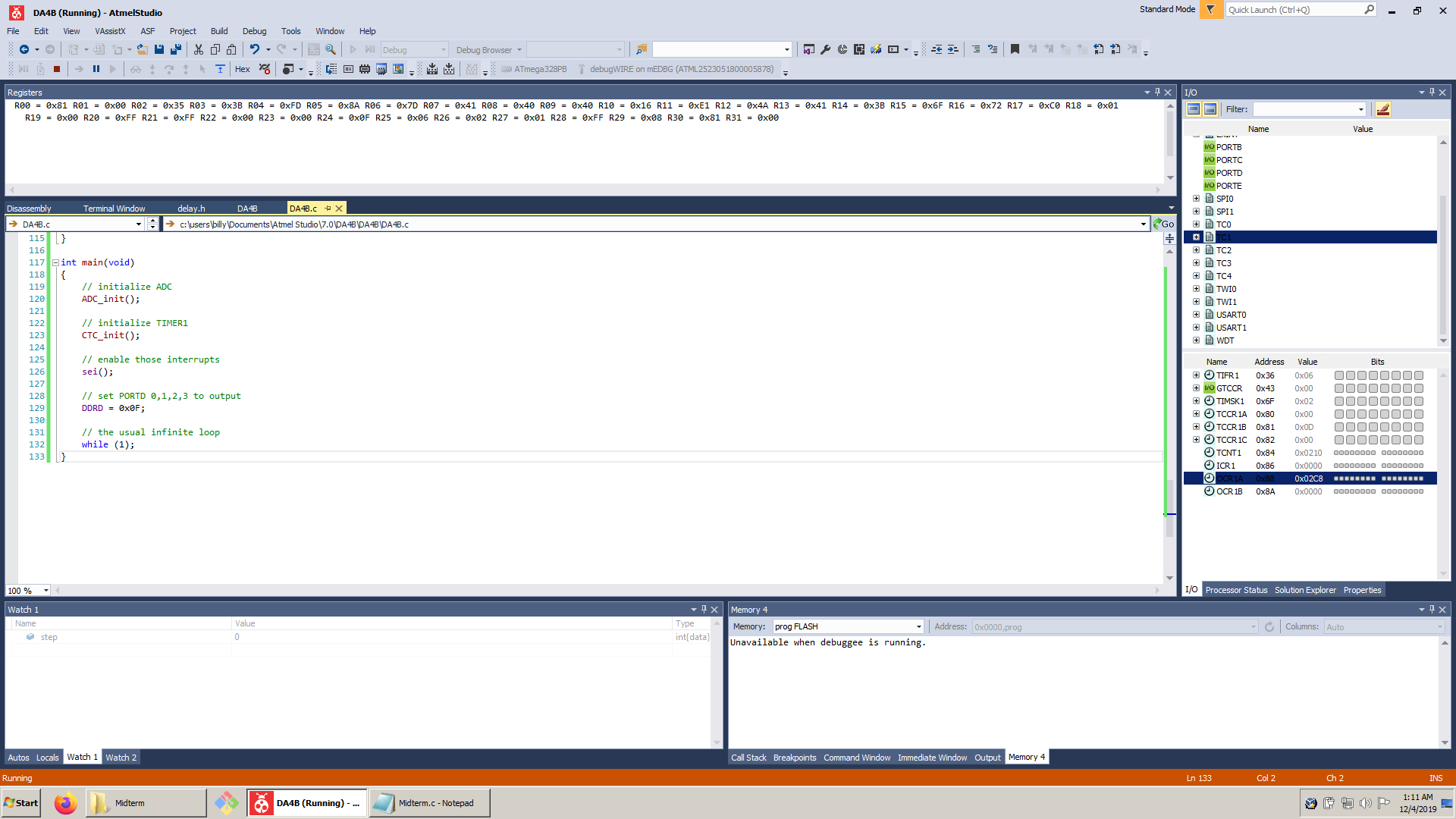
while (1);

}

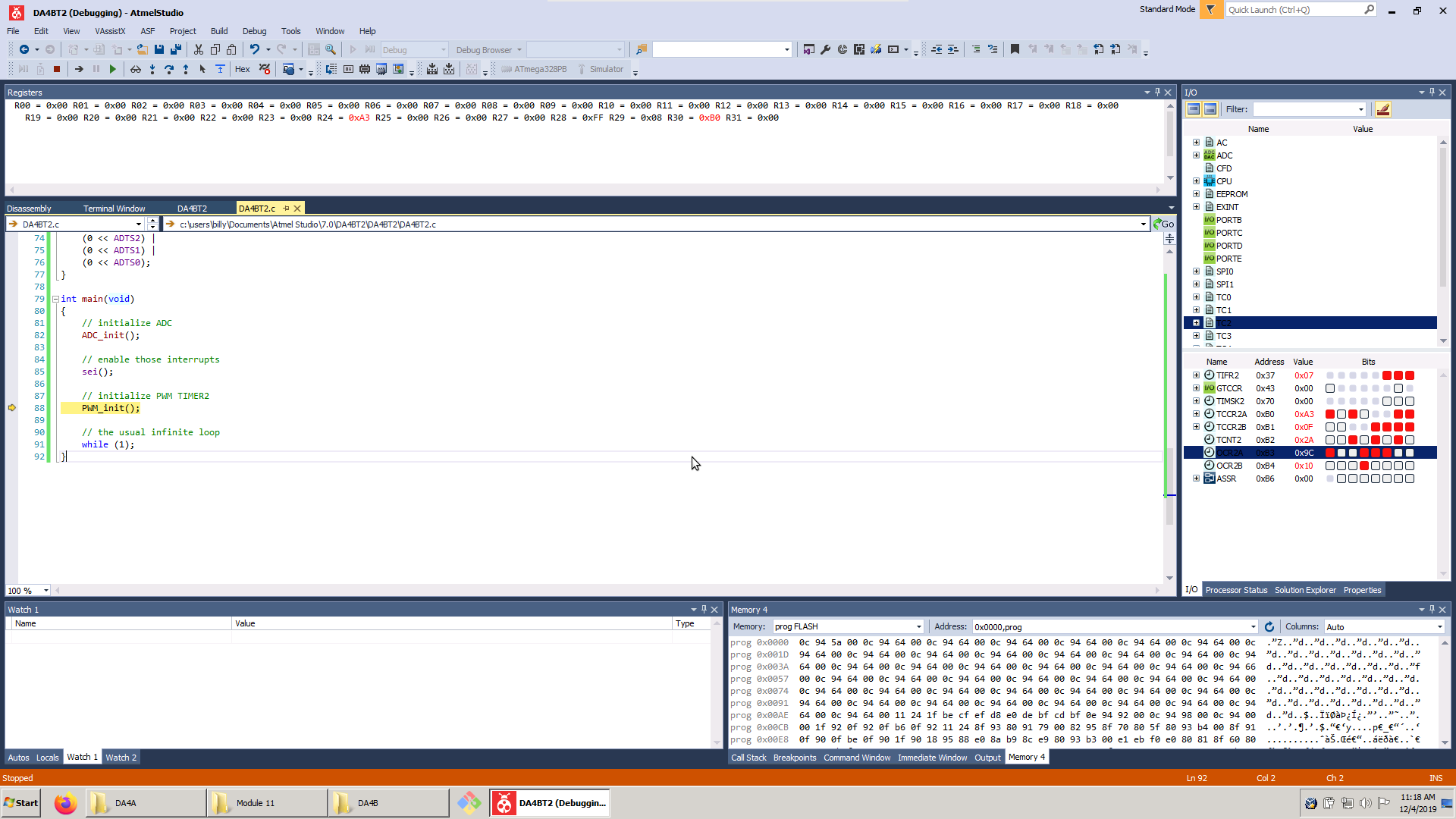
1. **SCHEMATICS**

Use fritzing.org

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

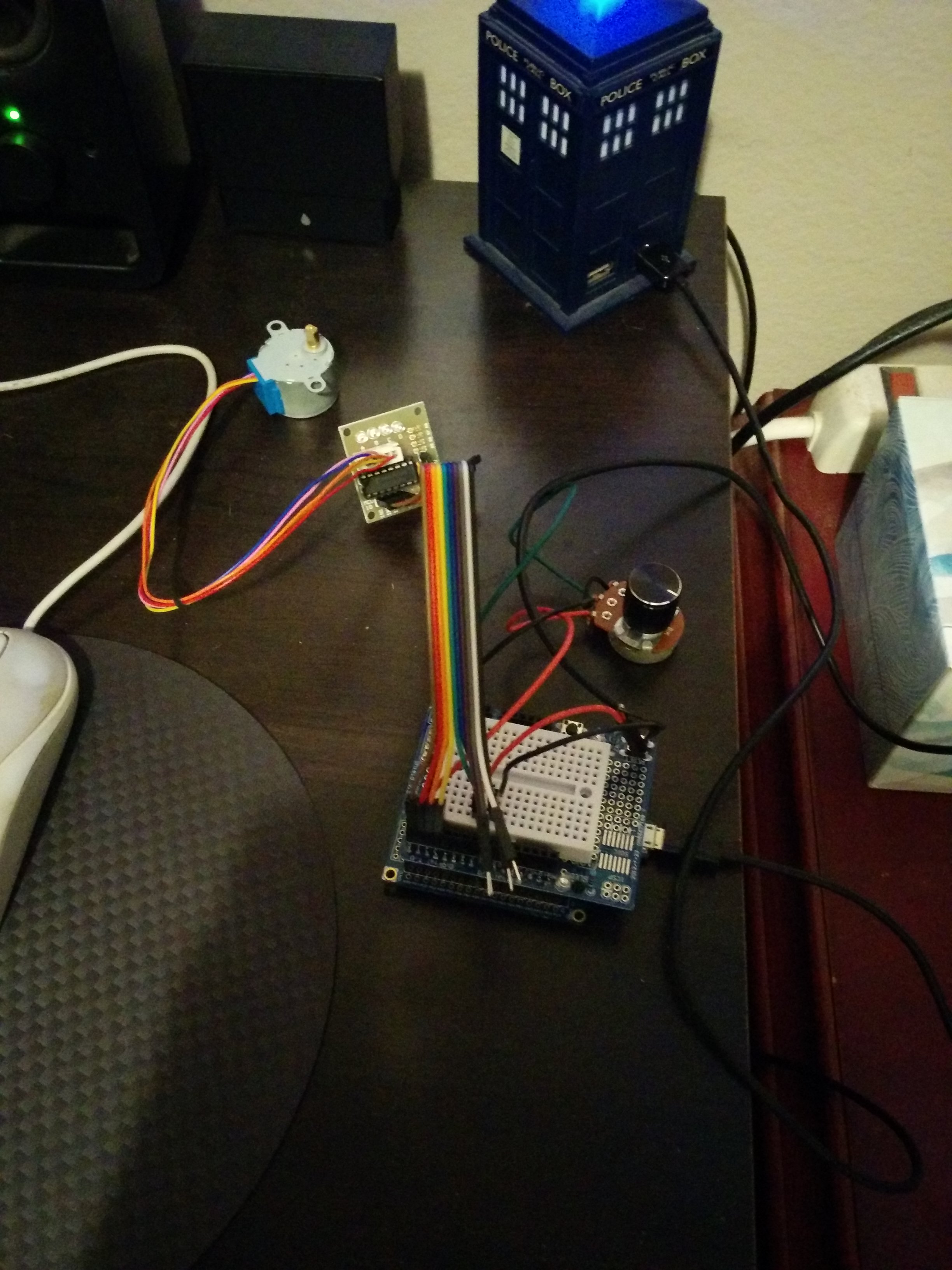


Task 1



Task 2

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



1. **VIDEO LINKS OF EACH DEMO**

<https://youtu.be/ucvMlmPrwss>

1. **GITHUB LINK OF THIS DA**

<https://github.com/billymaddex/fluffy-chainsaw/tree/master/DesignAssignments/DA4B>

**Student Academic Misconduct Policy**

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

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

Billy Maddex