

# Design Assignment DA4a

---

Student Name: Daniel Senda

Student #: 5002362633

Student Email: sendad1@unlv.nevada.edu

Primary Github address: <https://github.com/dsenda/Smiles>

Directory: DA4a

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

*List of Components used:*

Atmel Studio 7

ATmega328P Xplained Mini

DC motor driver module (TB6612FNG)

DC motor

Multi-function shield

Potentiometer (Multi-function shield)

Breadboard

Power Supply

*Block diagram about the transmission connections:*



## 2. INITIAL/MODIFIED CODE OF TASK DA4a

Not applicable, there is no modified code.

## 3. DEVELOPED MODIFIED CODE OF TASK DA4a

The following is the code that accomplishes the desired tasks that are required.

```
// c_code_dc_motor_on_off_duty_cycle.c
// Daniel Senda
// Turn DC motor on and off with pushbutton and
// also changes speed with change of potentiometer.

#define F_CPU 16000000UL
#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>

// Declares global variables.
int motor = 0; // On_Off variable for motor.
int poten = 0; // Potentiometer variable.

// Declares function.
void adc_init(void);

int main(void)
{
    DDRD = 0b00100000; // Sets PD5 as an output.
    DDRC = 0b00000000; // Sets all pins in PORTC as inputs.
    TCCR2A = 0x00; // Sets normal mode on timer0.
    TCCR2B = 0x01; // Sets no pre-scaling.
    TIMSK2 = 0x01; // Enables timer0 overflow interrupt.
    sei (); // Enables global interrupts.
```

```

TCCR0A = 0b10100011; // Sets Fast PWM.
TCCR0B = 0x01;        // Sets no pre-scaling.
OCR0B = 0xCC;         // Sets 80% or 20% duty cycle.
adc_init();           // Initializes ADC.

while (1)
{
    ADCSRA |= (1<<ADSC); // Starts conversion.
    while((ADCSRA & (1<<ADIF)) != 0); // Waits for conversion to finish.
    ADCSRA |= (1<<ADIF); // Resets conversion finished flag.
    poten = ADCL;        // Records potentiometer data.
    poten = poten | (ADCH<<8); // Data calculations.
    poten = (poten/1024.0) * 5000/10;
    OCR0B = poten;       // Sets OCR0B to poten value.
}

}

ISR (TIMER2_OVF_vect) // timer0 overflow interrupt vector
{
    if ((PINC & 0b10) == 0) // If pushbutton is pressed.
    {
        motor ^= 1; // Toggles motor variable.
        if (motor == 1) // Turns motor on.
        {
            TCCR0A = 0b00100011;
        }
        else // Turns motor off.
        {
            TCCR0A = 0b00000011;
        }
        _delay_ms(500); // Delay for de-bouncing.
    }
}

void adc_init (void) // Sets up and enables ADC.
{
    ADMUX = (0<<REFS1) | // Reference Selection Bits.
    (1<<REFS0) | // AVcc - external cap at AREF.
    (0<<ADLAR) | // ADC Left Adjust Result.
    (0<<MUX2) | // Analog Channel Selection Bits.
    (0<<MUX1) | // ADC0 (PC0).
    (0<<MUX0);
    ADCSRA = (1<<ADEN) | // ADC Enable.
    (0<<ADSC) | // ADC Start Conversion.
    (0<<ADATE) | // ADC Auto Trigger Enable.
    (0<<ADIF) | // ADC Interrupt Flag.
    (0<<ADIE) | // ADC Interrupt Enable.
    (1<<ADPS2) | // ADC Pre-scaler Select Bits.
    (0<<ADPS1) |
    (1<<ADPS0);
}

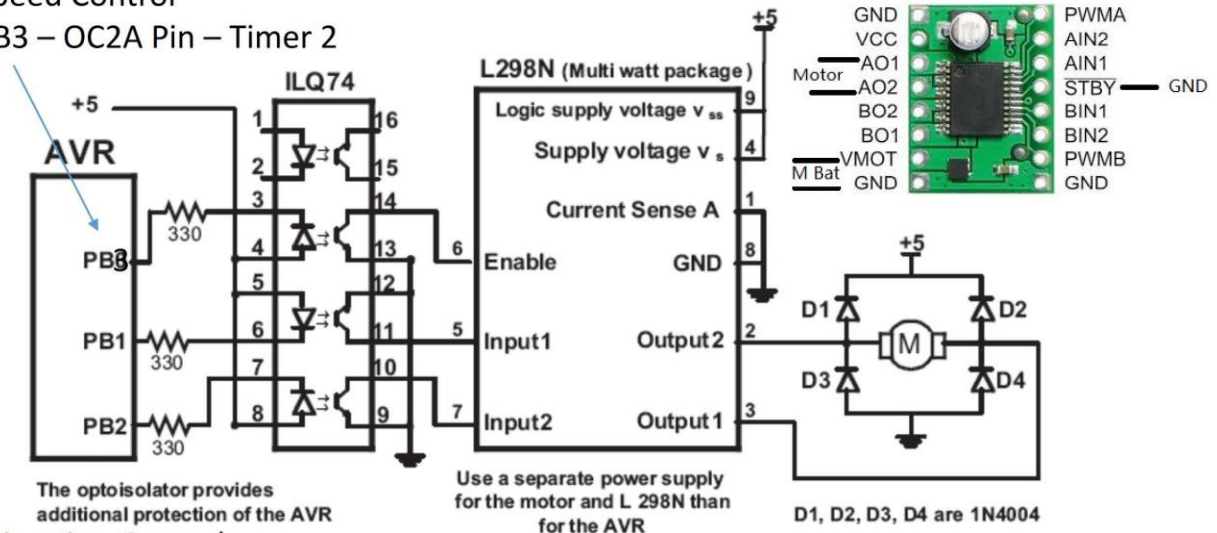
```

#### 4. SCHEMATICS

The following schematics from the slides describe the connections that are made relatively well.

##### Speed Control

PB3 – OC2A Pin – Timer 2



##### Direction Control

PB 1&2 – Digital Output

#### 5. SCREENSHOTS OF EACH TASK OUTPUT (ATMEL STUDIO OUTPUT)

There are no required Atmel studio outputs, but below is an included image of Atmel.

```
main.c
// c_code_dc_motor_on_off_duty_cycle.c
// Daniel Senda
// Turn DC motor on and off with pushbutton and
// also changes speed with change of potentiometer.

#define F_CPU 16000000UL
#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>

// Declares global variables.
int motor = 0; // On_Off variable for motor.
int poten = 0; // Potentiometer variable.

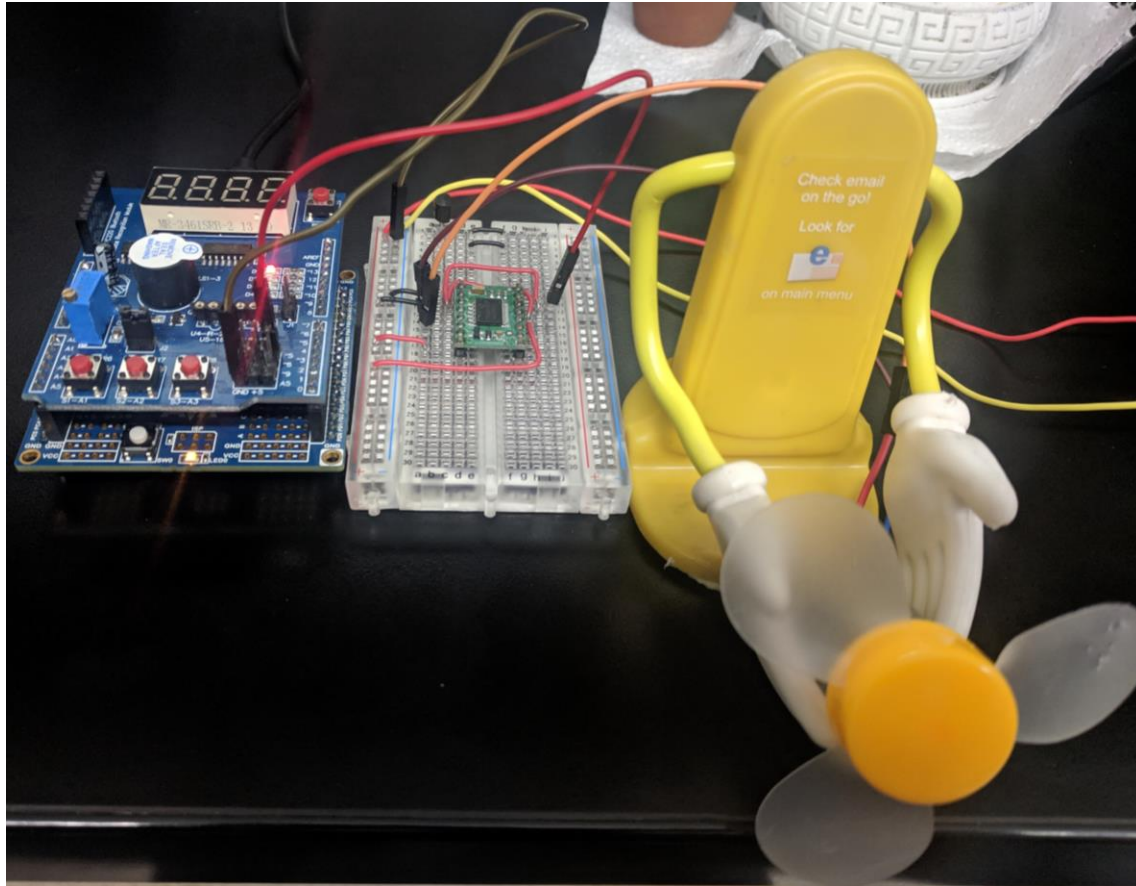
// Declares function.
void adc_init(void);

int main(void)
{
    DDRC = 0x00100000; // Sets PDS as an output.
    DDRC = 0x00000000; // Sets all pins in PORTC as inputs.
    TCCR2A = 0x00; // Sets normal mode on timer0.
    TCCR2B = 0x01; // Sets no prescaling.
    TIMSK2 = 0x01; // Enables timer0 overflow interrupt.
    sei (); // Enables global interrupts.

    TCCR0A = 0b10100011; // Sets Fast PWM.
}
```

## 6. SCREENSHOT OF EACH DEMO (BOARD SETUP)

The following is a picture of the board setup:



## 7. VIDEO LINKS OF EACH DEMO

<https://youtu.be/II7SXncMUe>

## 8. GITHUB LINK OF THIS DA

<https://github.com/dsenda/Smiles/tree/master/DA4a>

**Student Academic Misconduct Policy**

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

*"This assignment submission is my own, original work".*

Daniel Senda