CPE301 – SPRING 2019

Design Assignment 4B

Student Name: Tanner Tindall

Student #: 8000733733

Student Email: tindat1@unlv.nevada.edu

Primary Github address: <https://github.com/TannerTindall51>

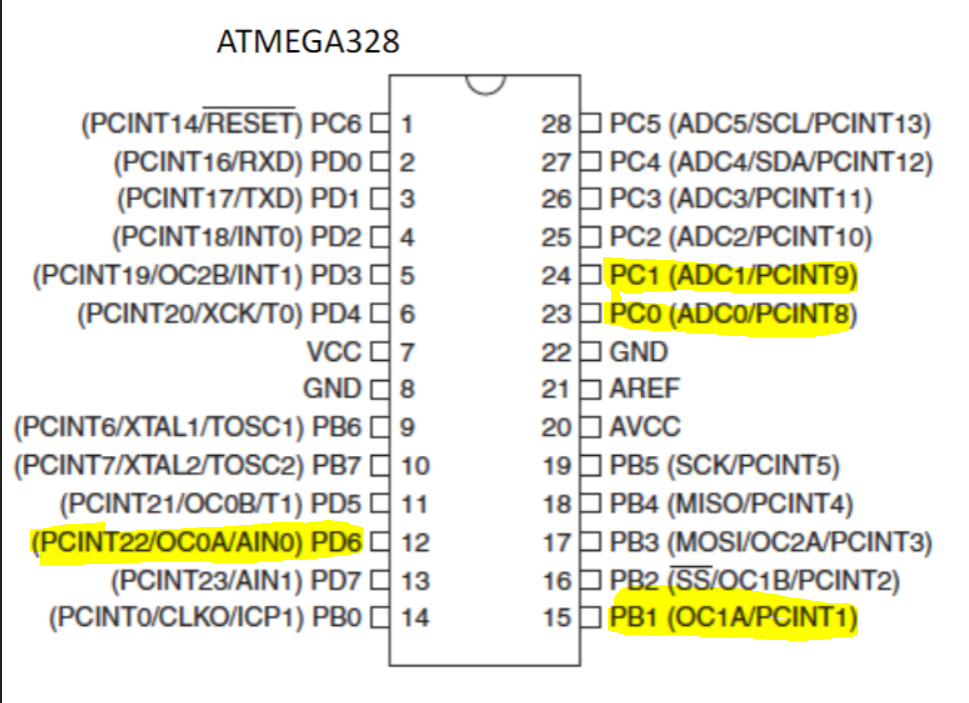
Directory:

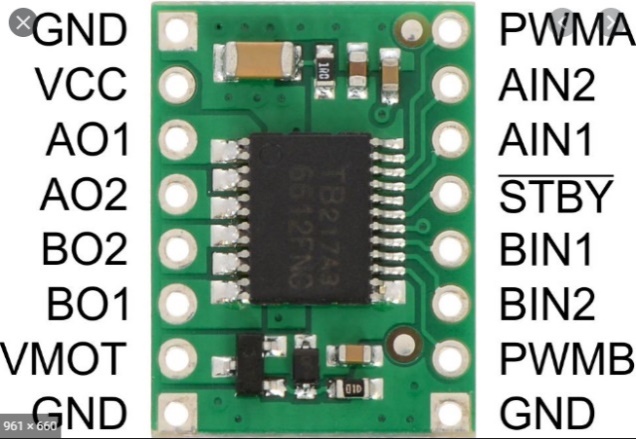
<https://github.com/TannerTindall51/tindalltannerm_submission/tree/master/Design_Assignments/DA4A>

1. Write an AVR C program to control the speed of the DC Motor using a potentiometer connected to PC0. Use an interrupt on a button (PC1/2/3) to stop and start the motor at each click. The minimum speed of the motor should be 0 when pot is at minimum position and at maximum position should be 95% of PWM value.

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

* Atmel Studio 7.0 (Assembler, Simulator, & Debugger)
* Atmega328PB-Xmini
* Micro USB
* LM35 Temperature Sensor
* Multi-Functional Shield
* MD08A
* DC Motor





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

|  |
| --- |
|  |

//

//CPE301 - DA4A

//Tanner Tindall

//

#define F\_CPU 16000000UL

#include <avr/io.h>

#include <avr/interrupt.h>

#include <stdbool.h>

#include <util/delay.h>

volatile unsigned int dutyCycle; //value of duty cycle (changed with potentiometer)

volatile bool state = false; //determines the state of motor

void configPWM()

{

TCCR0A = 0b10000011; //set noninverting, fast PWM

TCCR0B = 0b00000011; //set prescalar 1024

}

void startADC()

{

ADMUX = (1<<REFS0); //set Vref and set ADC0 as input

ADCSRA = (1<<ADEN) | (1<<ADSC) | (1<<ADATE) | (1<<ADPS2) | (1<<ADPS1) | (1<<ADPS0);

} //enable ADC and set prescalar to 128

void configDC()

{

dutyCycle = (adcRead()/1023.0) \* 95; //take value read from ADC, prescale it then take 95% of it

OCR0A = (dutyCycle/100.0)\*255; //convert dutyCycle into corresponding 8 bit value

}

int adcRead()

{

ADCSRA |= (1<<6); //enable ADC conversion

while (!(ADCSRA & (1<<4))); //waits for conversion to complete by ADIF flag

ADCSRA |= (1<<4); //reset when complete

return ADC; //return value once complete

}

int main(void)

{

DDRB |= (1<<2); //set DDRB to output

DDRC &= ~(1<<PINC1|1<<PINC0);//set DDRC TO input

DDRD |= (1<<PIND6); //set DDRD to output (PWM)

PORTB |= (1<<2); //set PB1 to HIGH

PORTC |= (0<<PINC0) | (1<<PINC1);//set PC0 to input (potentiometer) & PC1 to output (interrupt)

PCICR = (1<<PCIE1); //enable pin interrupt

PCMSK1 = (1<<PCINT9); //set interrupt on PC1

sei(); //enable interrupts

configPWM(); //config PWM

startADC(); //initialize ADC

while (1)

{

configDC(); //set duty cycle to corresponding potentiometer value

}

}

ISR(PCINT1\_vect) //PC1 interrupt subroutine

{

if(!(PINC & (1 << PINC1))) //if button is pressed begin

{

if(state == false) //if the DC motor is off

{

TCCR0B = 0x00; // Turns off timer.

DDRD &= ~(1<<PIND6);

}

else if(state == true) //if the DC is on

{

TCCR0B = 0b00000011; //set prescalar 1024

DDRD |= (1<<PIND6);

}

PCIFR = (1<<PCIF1); //clear interrupt

state = !state; //invert current state

}

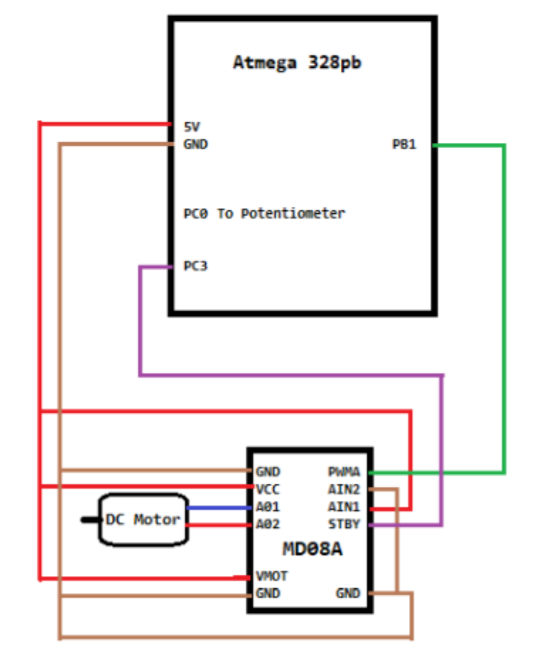
}

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

N/A

1. **SCHEMATICS**

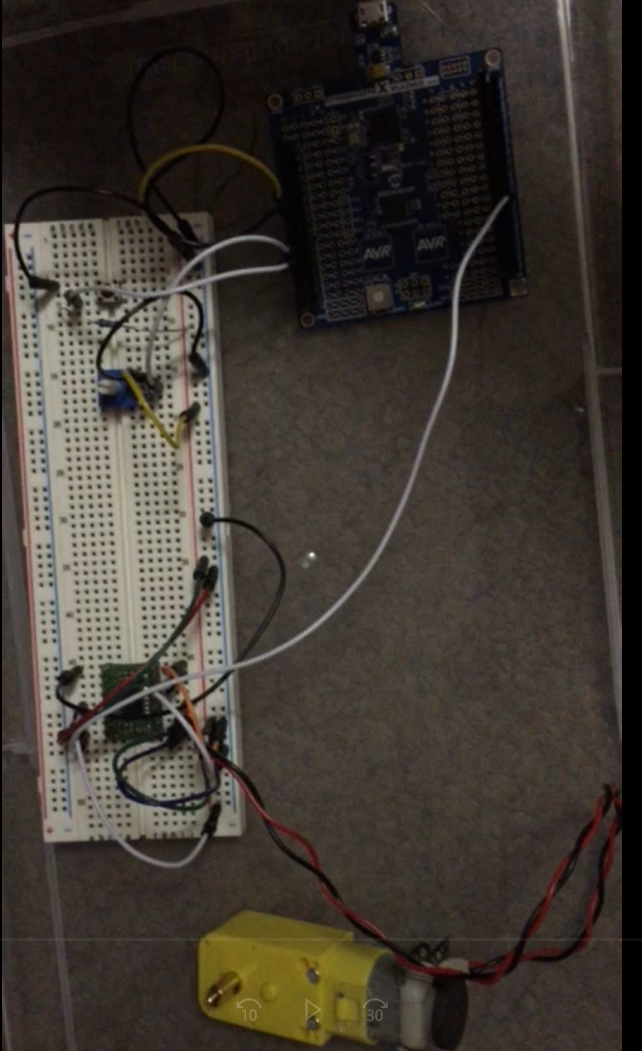
Note: This is the same schematic used in Dr. Venki’s annotations and is how the code this was configured to work.



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

N/A

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



1. **VIDEO LINKS OF EACH DEMO**

Demo Video: https://youtu.be/LhFxsiOmJ7k

1. **GITHUB LINK OF THIS DA**

<https://github.com/TannerTindall51/tindalltannerm_submission/tree/master/Design_Assignments/DA4A>

**Student Academic Misconduct Policy**

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

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

Tanner Tindall