CPE301 – SPRING 2019

Design Assignment 4A

Student Name: Ali Asadi

Student #: 5004200625

Student Email: aliasadivet@gmail.com

Primary Github address: https://github.com/aliasadivet/Submitions\_DA\_AliAsadi

Directory: DessignAssignments

In this assignment we need to write a C program to control the speed of the DC Motor using a potentiometer connected to any of the analog-in port. An interrupt on a button will stop and start the motor at each click. The minimum speed of the motor should be 0 when pot is minimum and maximum should be 95% of PWM value.

List of Components used:

1. DC motor
2. 1 kOhm Potentiometer
3. Atmega 328p Xplained Mini board
4. Breadboard
5. 22G jumper wires

Connection:



GND (board)

Vcc (board)

PD2

DC motor

12V-DC

PB3

PB1

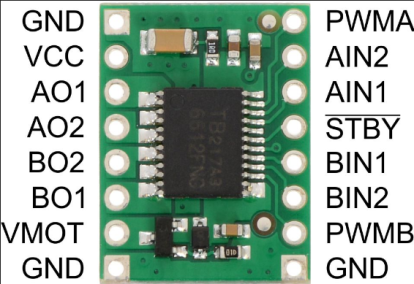
PB0

Vcc (board)

GND (board)

GND (board)

Vcc (board)



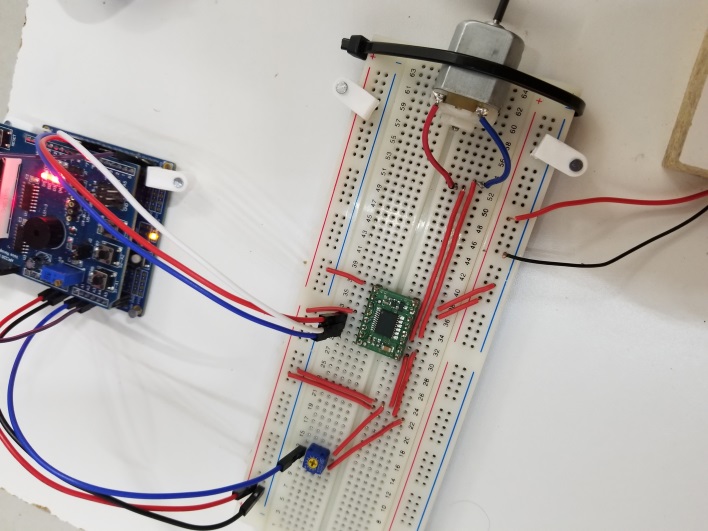
PC0

GND (board)

Vcc (board)



The following is a screenshot of project connections.



Programming:

/\*

\* DA4A.c

\*

\* Created: 4/13/2019 11:49:01 AM

\* Author : Ali Asadi

\*/

#include <inttypes.h>

#include <avr/io.h>

#include <avr/interrupt.h>

#include <util/delay.h>

void Timer2\_init();

void ADC\_init();

void Port\_init();

*uint16\_t* ReadADC(*uint8\_t* );

void INT0\_init();

void Motor();

volatile *uint8\_t* Motor\_ON = 0;

ISR(INT0\_vect)

{

*uint8\_t* tmp = PIND & \_BV(PD2); // read the PIN D2

*\_delay\_ms*(10); // debounce Delay

if(tmp == (PIND & \_BV(PD2))) // check if the PIN is stable

Motor\_ON = Motor\_ON ? 0 : 1 ; // if stable toggle the Motor between ON or OFF state

while (!(PIND & \_BV(PD2)));

}

int main()

{

Port\_init(); // Initiate the PORT

ADC\_init(); // Initiate the analog to digital converter

INT0\_init(); // Initiate the Interrupt 0 for the button

Timer2\_init(); // Using Timer 2 for Motor

sei(); // Enable global interrupt

while (1)

{

Motor();

}

return 0;

}

void Port\_init()

{

DDRB = \_BV(PB0)|\_BV(PB1)|\_BV(PB3); // DC Motor Pin Connection

PORTB = \_BV(PB0); // set high

PORTD = \_BV(PD2); //external interrupt 0

}

void Timer2\_init()

{

TCCR2A = \_BV(COM2A1)|\_BV(WGM20); // non-inverting mode, Phase Correct PWM

TCCR2B = \_BV(CS21)|\_BV(CS20); // Clk / 32 = 8uS

}

void ADC\_init()

{

ADMUX = \_BV(REFS0); // AVcc as Reference

ADCSRA = \_BV(ADEN)|\_BV(ADPS0); // Enable ADC, Clk / 2

}

*uint16\_t* ReadADC(*uint8\_t* Channel)

{

ADMUX = (ADMUX & 0xF0)|(Channel & 0x0F); // Select channel

ADCSRA |= \_BV(ADSC); // start conversion

while(!(ADCSRA & \_BV(ADIF))); // wait for completion

ADCSRA |= \_BV(ADIF); // clear flag

return ADC; // return the converted value

}

void INT0\_init()

{

EICRA = \_BV(ISC01); // The falling edge of INT0 generates an interrupt request.

EIMSK = \_BV(INT0); // External Interrupt Request 0 Enable

}

void Motor()

{

if(Motor\_ON)

OCR2A =(ReadADC(0) / 1024.0) \* 243; // Makes the max speed of the motor 95% of PWM value. (256\*95%=243)

else

OCR2A = 0; // Turn off

}

The execution results are posted on YouTube, and can be found on link below:

<https://www.youtube.com/watch?v=XWCg-334Fzo>