CPE301 – SPRING 2019

Design Assignment 4A

Student Name: Michael Johnson

Student #: 2000878717

Student Email: johnsm17@unlv.nevada.edu

Primary Github address: https://github.com/miggnuggets/submissions.git

Directory: Repository\_301

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

List of Components used

Block diagram with pins used in the Atmega328P

Motor Driver

Potentiometer

Switch

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

**ALL CODE IN NOTEPAD SO ITS CLEANER**

#define F\_CPU 16000000UL

#include <avr/io.h>

#include <stdio.h>

#include <util/delay.h>

#include <avr/interrupt.h>

// Define Baudrate for UART

#define BAUDRATE 9600

#define BAUD\_PRESCALLER (((F\_CPU / (BAUDRATE \* 16UL))) - 1)

// Function Declarations

void read\_adc(void);

void adc\_init(void);

volatile unsigned int adc\_temp;

char outs[20];

int main(void)

{

adc\_init(); // Initialize the ADC

USART\_init(BAUD\_PRESCALLER); // Initialize the USART

USART\_tx\_string("Connected!\r\n"); // we're alive!

\_delay\_ms(125); // wait a bit

DDRD = 0xFF; //DDRD = (1<<3); // Set Port D as an output for PWM

TCCR2A = \_BV(COM2A1) | \_BV(COM2B1) | \_BV(WGM21) | \_BV(WGM20);

TCCR2B = \_BV(WGM22) | \_BV(CS20) | \_BV(CS21) | \_BV(CS22);

OCR2A = 255;

OCR2B = 0;

PCMSK1 |= (1<<PCINT10);

PCICR |= (1<<PCIE1);

sei();

while (1)

{

read\_adc();

\_delay\_ms(250);

adc\_temp = adc\_temp/4;

OCR2B = adc\_temp; // Potentiometer controlling PWM

}

}

/\* INIT ADC \*/

void adc\_init(void)

{

/\*\* Setup and enable 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 PIN27)

(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 Prescaler Select Bits

(0<<ADPS1)|

(1<<ADPS0);

}

/\* READ ADC PINS \*/

void read\_adc(void)

{

unsigned char i = 4;

adc\_temp = 0;

while (i--)

{

ADCSRA |= (1<<ADSC);

while(ADCSRA & (1<<ADSC));

adc\_temp+= ADC;

\_delay\_ms(50);

}

adc\_temp = adc\_temp / 4; // Average a few samples

}

ISR(PCINT1\_vect){

DDRD ^= 0xFF;

\_delay\_ms(1000); //Ground Bounce

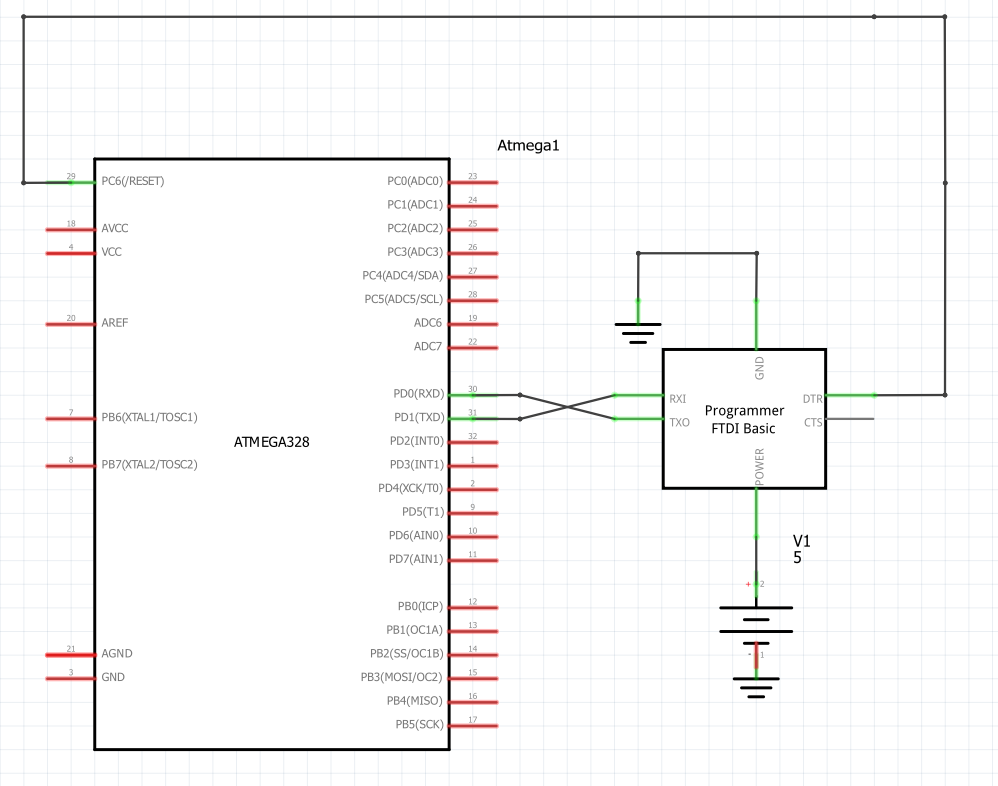
char check = "Pin Change ";

USART\_tx\_string(check); // Print random integer number

// Check if I bit is set in IREG

}

1. **SCHEMATICS**

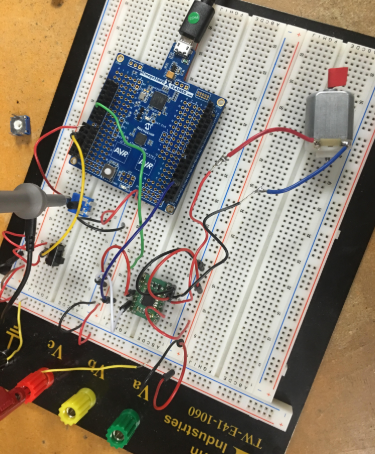


Use fritzing.org

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

**ALL OUTPUT TO MOTOR ON SWITCH SO NO NEED FOR ATMEL OUTPUT**

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



1. **VIDEO LINKS OF EACH DEMO**

**https://youtu.be/OCu\_xX1W\_ks**

1. **GITHUB LINK OF THIS DA**

**https://github.com/miggnuggets/submissions.git**

**Student Academic Misconduct Policy**

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

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

Michael Johnson