CPE301 – SPRING 2019

Design Assignment 3B

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

LM35 Temperature Sensor

FTDI Basic Breakout board (RS232 to TTL)

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

**ALL CODE IN NOTEPAD SO ITS CLEANER**

/\*

\* DA3B.c

\*

\* Created: 3/29/2019 4:47:57 PM

\* Author : Migggy

\*/

#define F\_CPU 16000000UL

#include <avr/io.h>

#include <util/delay.h>

#include <stdio.h>

#define BAUDRATE 9600

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

void read\_adc(void); // Function Declarations

void adc\_init(void);

void USART\_init( unsigned int ubrr );

void USART\_tx\_string( char \*data );

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

while(1)

{

read\_adc();

snprintf(outs,sizeof(outs),"%3d\r\n", adc\_temp); // print it

USART\_tx\_string(outs);

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

}

}

/\* 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

(1<<MUX2)| // Analog Channel Selection Bits

(0<<MUX1)| // ADC4 (PC4 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 = 5;

adc\_temp = 0;

while (i--)

{

ADCSRA |= (1<<ADSC);

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

adc\_temp+= ADC;

\_delay\_ms(50);

}

adc\_temp = adc\_temp / 5; // Average a few samples

}

/\* INIT USART (RS-232) \*/

void USART\_init( unsigned int ubrr )

{

UBRR0H = (unsigned char)(ubrr>>8);

UBRR0L = (unsigned char)ubrr;

UCSR0B = (1 << TXEN0); // Enable RX, TX & RX interrupt

UCSR0C = (3 << UCSZ00); //asynchronous 8 N 1

}

/\* SEND A STRING TO THE RS-232 \*/

void USART\_tx\_string( char \*data )

{

while ((\*data != '\0'))

{

while (!(UCSR0A & (1 <<UDRE0)));

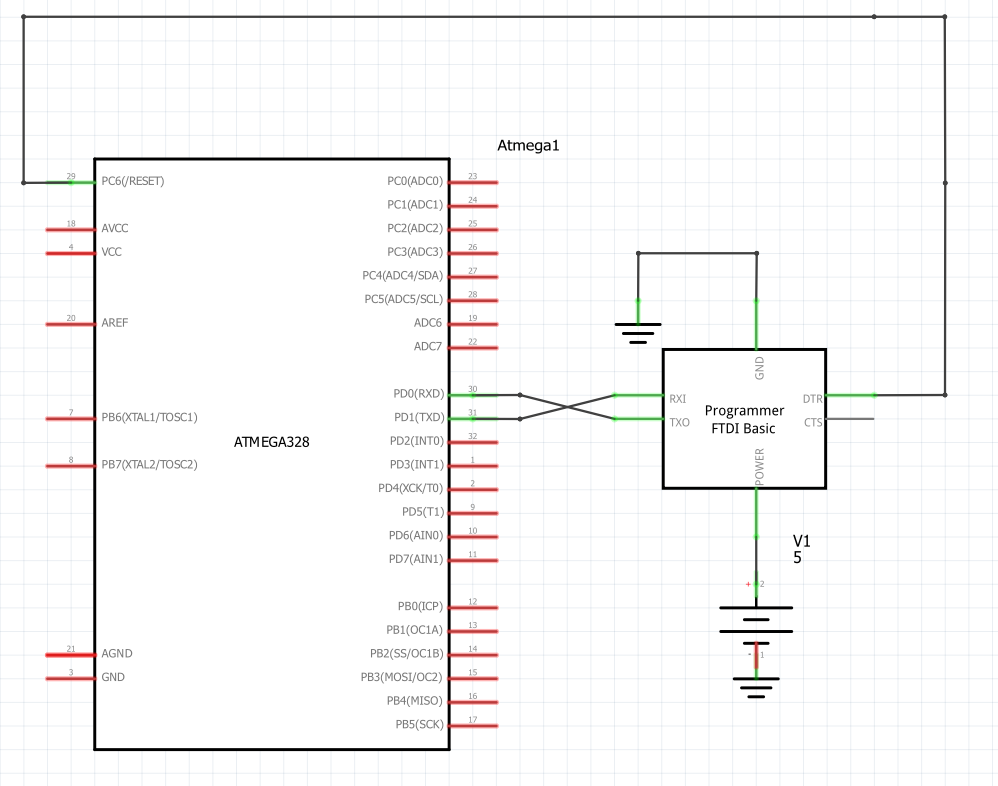
UDR0 = \*data;

data++;

}

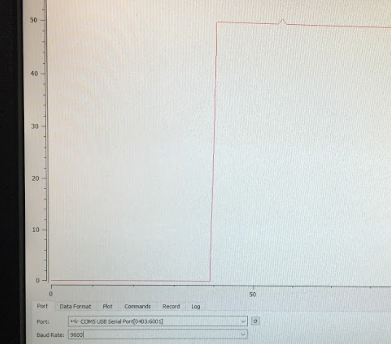
}

1. **SCHEMATICS**

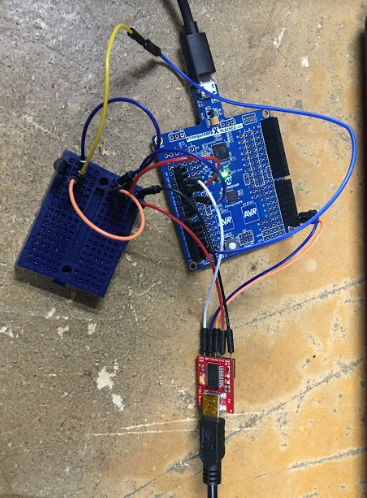


Use fritzing.org

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



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



1. **VIDEO LINKS OF EACH DEMO**

**https://youtu.be/3LTgXjBYT2w**

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