#define *F\_CPU* 16000000UL

#define BAUD 9600

#include <avr/io.h>

#include <util/delay.h>

#include <avr/interrupt.h>

#include <util/setbaud.h>

#include <stdio.h>

volatile *uint8\_t* of;

volatile float adc\_temp;

void usart\_init(void)

{

UBRR0H = *UBRRH\_VALUE*; // loads the ubrrh value into high

UBRR0L = *UBRRL\_VALUE*;// loads the ubrr0l value into low

UCSR0C = \_BV(UCSZ01) | \_BV(UCSZ00); // 8-bit data

UCSR0B = \_BV(RXEN0) | \_BV(TXEN0);// enable RX and TX

}

void set()

{

TCCR0A = 0; // normal mode is initiated

TCCR0B = 0x05;// the prescaler is set to 1024

TCNT0 = 0x00;// initiates the counter to 0

TIMSK0 = (1<<TOIE0); // enable interrupt

sei(); // enable global interrupt

}

void usart\_tx\_string(char \* info)

{

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

{

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

UDR0 = \*info;

info++;

}

}

ISR(TIMER0\_OVF\_vect)

{

of++; // this will increment the overflow

}

void adc\_init(void)

{

ADMUX = (0<<REFS1)| // reference selection bit

(1<<REFS0)| //AVCC EXTERNAL CAP AT REF

(0<<ADLAR)| //ADC LEFT ADJUST RESULT

(1<<MUX2)| //ANALOG CHANNEL SELECTION BITS

(0<<MUX1)| //ADC

(0<<MUX0);

ADCSRA = (1<<ADEN)| // ADC ENABLE

(0<<ADSC)| // ADC START CONVERSION

(0<<ADATE)| // ADC AUTO TRIGGER ENABLE

(0<<ADIF)| // ADC INTERRUPT ENABLE

(0<<ADIE)|// ADC PRESCALER SELECT BIT

(1<<ADPS2)|

(1<<ADPS1)|

(1<<ADPS0); // SELECT CHANNEL

}

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);

}

int main(void)

{

usart\_init();

set();

adc\_init();

char c[20];

while(1)

{

while(of < 61);

read\_adc();

*snprintf*(c,sizeof(c), "%f\r\n", adc\_temp);

usart\_tx\_string(c);

usart\_tx\_string("\n");

of = 0;

}

}