#define *F\_CPU* 16000000UL

#include <avr/io.h>

#include <avr/interrupt.h>

#include <util/delay.h>

#include <string.h>

#include <stdlib.h >

int ADC\_value[8];

int ADC\_position;

int adc\_flag;

int timer\_flag;

int timer\_value;

int counter;

int i;

char buffer[200];

int position;

char new\_line[10]=" \n";

void ADC\_begin()

{

DDRF=0x00;

ADCSRA|=(1<<ADEN);

ADMUX=(1<<REFS0);//set voltage reference

ADCSRA|=(1<<ADPS2)|(1<<ADPS1);//set prescaler of 64

ADCSRA|=(1<<ADIE);

}

void \_uart0\_begin()

{

UBRR0H=0;

UBRR0L=103;

UCSR0A|=(1<<RXC0);

UCSR0B|=(1<<RXEN0)|(1<<TXEN0)|(1<<RXCIE0);

UCSR0C|=(1<<UCSZ01)|(1<<UCSZ00);

}

void UART0\_TRANSMIT(char\*request)

{

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

{

while(!(UCSR0A&(1<<UDRE0))){}

UDR0=\*request;

request++;

}

}

void TIMER\_BEGIN()

{

TCNT0=0;

OCR0A=75;//for 300us

TCCR0A=(1<<WGM12);//CTC mode

TCCR0A=(1<<COM0A1)|(1<<COM0A0);

TIMSK0=(1<<OCIE0A);//enable timer interrupt

TCCR0B|=(1<<CS01)|(1<<CS00);//prescaler of 64

}

int main(void)

{

DDRJ|=(1<<PJ0);

\_uart0\_begin();

TIMER\_BEGIN();

timer\_value=0;

ADC\_position=0;

adc\_flag=0;

ADC\_begin();

sei();

ADCSRA|=(1<<ADSC);

while (1)

{

*\_delay\_ms*(1);

if (ADC\_value[0]<500 && adc\_flag == 0)

{

PORTJ|=(1<<PJ0);

adc\_flag=1;

timer\_value=0;

timer\_flag=0;

}

if ((ADC\_value[0] == 500) && (adc\_flag == 1) && (timer\_flag==1))

{

PORTJ&=~(1<<PJ0);

counter++;

adc\_flag=0;

timer\_flag=0;

}

*itoa*(ADC\_value[0], buffer,10);

*\_delay\_ms*(1);

UART0\_TRANSMIT(buffer);

UART0\_TRANSMIT(new\_line);

ADCSRA|=(1<<ADSC);

}

}

ISR(TIMER0\_COMPA\_vect)

{

timer\_value++;

if (timer\_value>75)

{

timer\_flag=1;

timer\_value=0;

}

}

ISR(ADC\_vect)

{

ADC\_value[ADC\_position]=(ADCL)|(ADCH<<8);

ADC\_position++;

if(ADC\_position==8)

{

ADC\_position=0;

}

ADMUX=(1<<REFS0)|(ADC\_position);

ADCSRA|=(1<<ADSC);

}