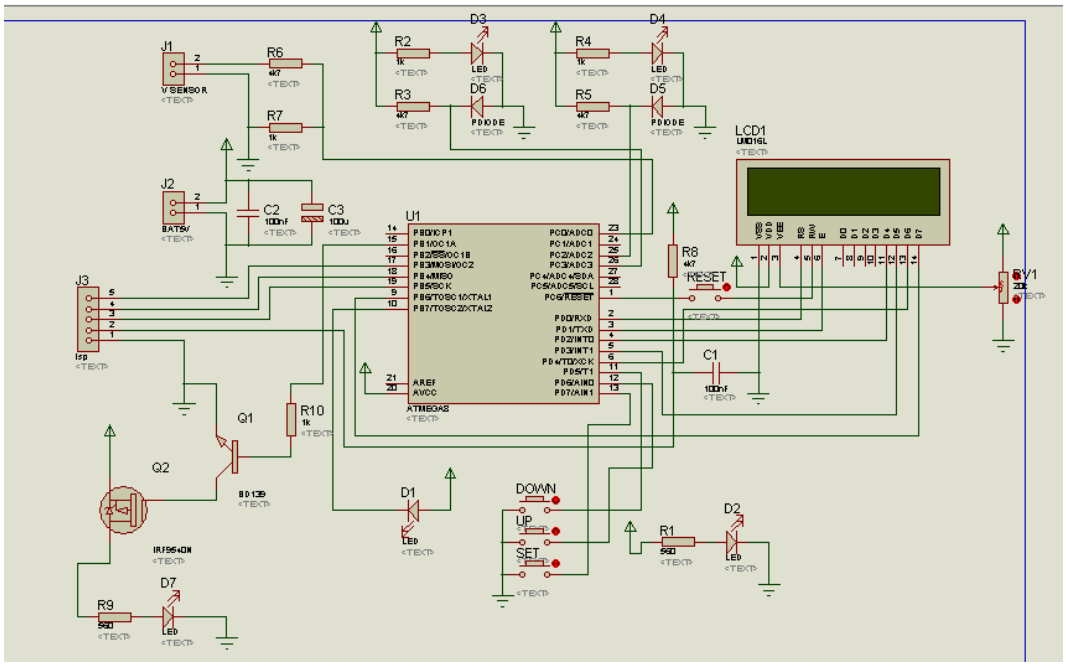
LAMPIRAN

Rangkaian Keseluruhan



Program

```
#include <mega8.h>
#include <stdio.h>
#include <delay.h>

// Alphanumeric LCD functions
#include <alcd.h>
char buff[33];
#define ADC_VREF_TYPE 0x40

#define lampu OCR1A
#define ledlcd PORTB.7

#define set  PINB.0
#define up   PIND.6
#define down PIND.7

#define sensor_v  0
#define sensor_a  2
#define sensor_b  3


// Read the AD conversion result
```

```
unsigned int read_adc(unsigned char adc_input)
{
ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
// Delay needed for the stabilization of the ADC input voltage
delay_us(10);
// Start the AD conversion
ADCSRA|=0x40;
// Wait for the AD conversion to complete
while ((ADCSRA & 0x10)==0);
ADCSRA|=0x10;
return ADCW;
}

// Declare your global variables here

float vbat;
eeprom unsigned char pwm=1;
eeprom int pekaa=30,pekab=30;
int out1,out2;

float read_volt(){
 float volt,voltout;
 float voltmax=3.5,involtmax=20.0;
 int data=read_adc(sensor_v);
 // rumus sensor tegangan
 // r1= 4,7k r2= 1k
 // involt max= 20.0 volt (sesuai kebutuhan yang peting hasil perhitungan voltmax
tidak lebih dari 5v)
 // mencari voltmax ( max 5V), voltmax=r2/(r1+r2)*involtmax
 // voltmax=1k/(4,7+1)*20.0
 // voltmax= 3.50 volt

 volt=data*((float)5/1023);  // mengubah nilai adc ke tegangan
 voltout=volt*((float)involtmax/voltmax); // mengubah nilai tegangan kecil ke
tegangan sensor
return voltout;
}

void lamp(int pulsa){
if(pulsa>0){
TCCR1A=0x81;
TCCR1B=0x0B;
lampu=pulsa;
}
else{
TCCR1A=0x00;
```

```
TCCR1B=0x00;
}
}



int menu=0,dim;
void go_program(){

vbat=read_volt();

lcd_clear();
lcd_gotoxy(0,0);
sprintf(buff,"LED:%d B:%.2f V",dim,vbat);
lcd_puts(buff);
lcd_gotoxy(0,1);
sprintf(buff,"SA:%d SB:%d",read_adc(sensor_a),read_adc(sensor_b));
lcd_puts(buff);

if(read_adc(sensor_a)>pekaa)out1=1;
else out1=0;

if(read_adc(sensor_b)>pekab)out2=1;
else out2=0;

if(out1==1||out2==1)lamp(dim);
else lamp(0);

if(pwm==1) dim=100;
if(pwm==2) dim=150;
if(pwm==3) dim=200;

if(!set){
lcd_clear();
delay_ms(200);
menu=0;
while(1){

if(!set)menu++;
if(menu>2)break;

lamp(pwm);

if(pwm==1) dim=100;
if(pwm==2) dim=150;
if(pwm==3) dim=200;
```

```
if(menu==0){
lcd_clear();
lcd_gotoxy(0,0);
if(pwm==1)lcd_putsf("1.LOW");
if(pwm==2)lcd_putsf("1.MEDIUM");
if(pwm==3)lcd_putsf("1.HIGH");
if(!up)pwm++;
if(!down)pwm--;

if(pwm>3) pwm=1;
if(pwm<1) pwm=3;

}

if(menu==1){
lcd_clear();
lcd_gotoxy(0,0);
sprintf(buff,"2.ADC A:%d ",read_adc(sensor_a));
lcd_puts(buff);
lcd_gotoxy(0,1);
sprintf(buff,"PEKA A:%d %d",pekaa,out1);
lcd_puts(buff);
if(!up)pekaa++;
if(!down)pekaa--;
if(pekaa>1023)pekaa=0;
if(pekaa<0)pekaa=1023;
if(read_adc(sensor_a)>pekaa)out1=1;
else out1=0;
}

if(menu==2){
lcd_clear();
lcd_gotoxy(0,0);
sprintf(buff,"3.ADC B:%d ",read_adc(sensor_b));
lcd_puts(buff);
lcd_gotoxy(0,1);
sprintf(buff,"PEKA B:%d %d",pekab,out2);
lcd_puts(buff);
if(!up)pekab++;
if(!down)pekab--;
if(pekab>1023)pekab=0;
if(pekab<0)pekab=1023;
if(read_adc(sensor_b)>pekab)out2=1;
else out2=0;
}
```

```
delay_ms(150);
}
lamp(0);
lcd_clear();
delay_ms(200);
}


delay_ms(150);
}

void main(void)
{
// Declare your local variables here

// Input/Output Ports initialization
// Port B initialization
// Func7=Out Func6=In Func5=In Func4=In Func3=In Func2=In Func1=Out
Func0=In
// State7=0 State6=T State5=T State4=T State3=T State2=T State1=0 State0=T
PORTB=0x01;
DDRB=0x82;

// Port C initialization
// Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTC=0x00;
DDRC=0x00;

// Port D initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
Func0=In
// State7=P State6=P State5=P State4=T State3=T State2=T State1=T State0=T
PORTD=0xE0;
DDRD=0x00;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
TCCR0=0x00;
TCNT0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: 125,000 kHz
// Mode: Fast PWM top=0x00FF
```

```
// OC1A output: Non-Inv.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0x81;
TCCR1B=0x0B;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer2 Stopped
// Mode: Normal top=0xFF
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
MCUCR=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;

// USART initialization
// USART disabled
UCSRB=0x00;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;
```

```
// ADC initialization
// ADC Clock frequency: 1000,000 kHz
// ADC Voltage Reference: AVCC pin
ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0x83;

// SPI initialization
// SPI disabled
SPCR=0x00;

// TWI initialization
// TWI disabled
TWCR=0x00;

// Alphanumeric LCD initialization
// Connections are specified in the
// Project|Configure|C Compiler|Libraries|Alphanumeric LCD menu:
// RS - PORTD Bit 0
// RD - PORTC Bit 6
// EN - PORTD Bit 1
// D4 - PORTD Bit 2
// D5 - PORTD Bit 3
// D6 - PORTD Bit 4
// D7 - PORTB Bit 6
// Characters/line: 16
lcd_init(16);

lcd_clear();
lcd_gotoxy(0,0);
lcd_putsf("Film Wiewer");
delay_ms(1000);

while (1)
      {
      // Place your code here
      go_program();
      }
}
```