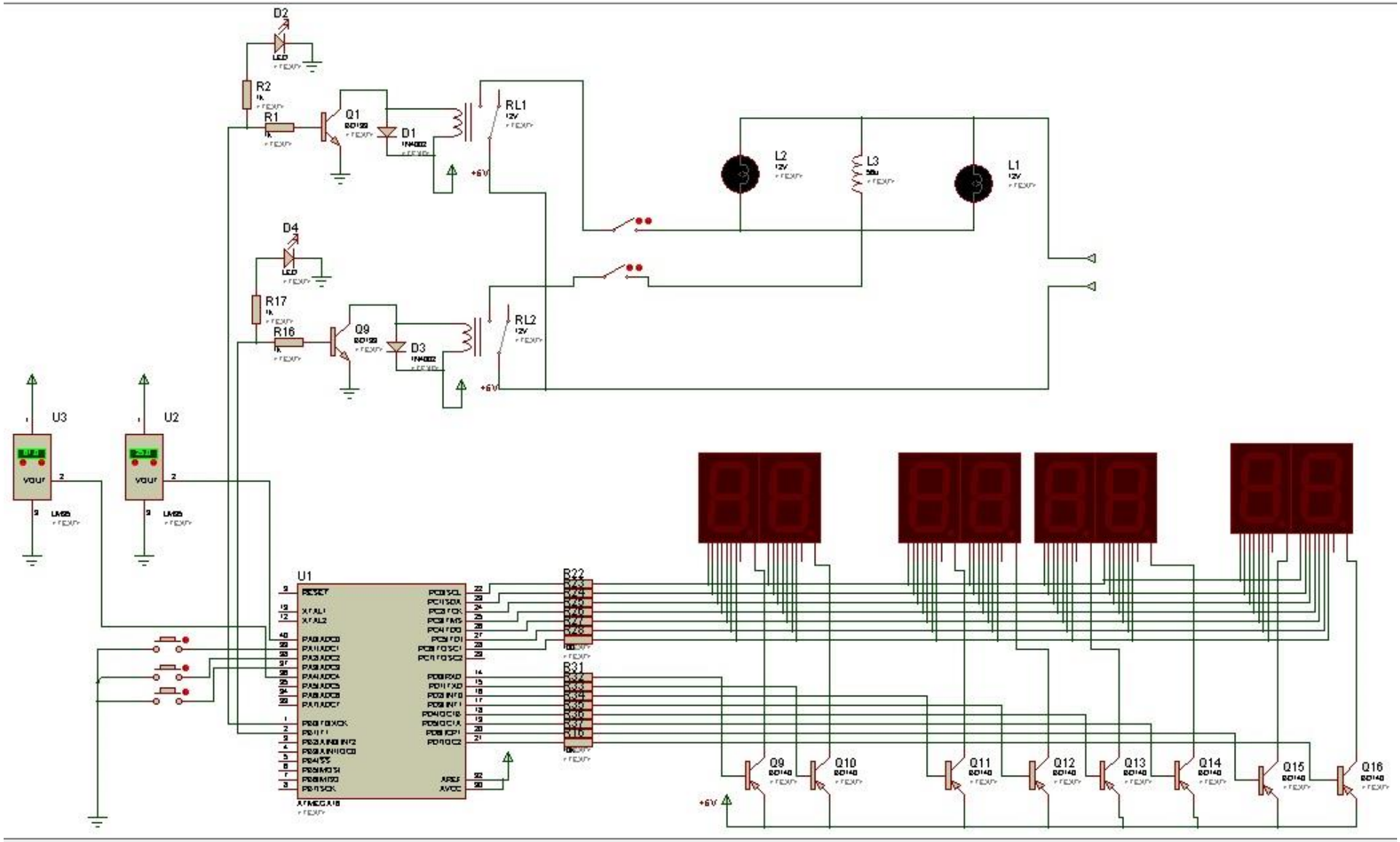
LAMPIRAN


D2
LED
R2
1k
R1
1k
Q1
BD139
D1
1N4002
RL1
12V
+6V
L2
12V
L3
50u
L1
12V
D4
LED
R17
1k
R16
1k
Q9
BD139
D3
1N4002
RL2
12V
+6V
U3
U2
volt
volt
U1
RESET
XTAL1
XTAL2
PA0/ADC0
PA1/ADC1
PA2/ADC2
PA3/ADC3
PA4/ADC4
PA5/ADC5
PA6/ADC6
PA7/ADC7
PB0/T0/XCK
PB1/T1
PB2/AIN0/INT2
PB3/AIN1/OC0
PB4/SS
PB5/MOSI
PB6/MISO
PB7/SCK
PC0/SCL
PC1/SDA
PC2/TCK
PC3/TMS
PC4/TDO
PC5/TDI
PC6/TOSC1
PC7/TOSC2
PD0/RXD
PD1/TXD
PD2/INT0
PD3/INT1
PD4/OC1B
PD5/OC1A
PD6/ICP1
PD7/OC2
AREF
AVCC
ATMEGA16
B22
B31
Q9
BD140
Q10
BD140
Q11
BD140
Q12
BD140
Q13
BD140
Q14
BD140
Q15
BD140
Q16
BD140
+6V

LISTING PROGRAM

```
#include <mega32.h>

#include <delay.h>
#define up PINA.1
#define ok PINA.2
#define ops PINA.3
float suhu_c1,suhu_c2;
int suhu1,suhu2;
unsigned char settemp,sets,set,settime,menu;
int pul4,sat4,pul3,sat3,pul2,sat2,pul1,sat1;

flash unsigned char bil []={
0b11000000,//0
0b11111001,//1
0b10100100,//2
0b10110000,//3
0b10011001,//4
0b10010010,//5
0b10000010,//6
0b11111000,//7
0b10000000,//8
0b10010000,//9
0b01111111 //dot
};
int detik,menit;

#define ADC_VREF_TYPE 0x00
```

```
// 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;
}
```

```
interrupt [TIM1_OVF] void timer1_ovf_isr(void)
{
// Reinitialize Timer1 value
TCNT1H=0xD23A >> 8;
TCNT1L=0xD23A & 0xff;
detik++;
// Place your code here

}
```

```
void settemps(unsigned char angka){
char puls3,sats3;

puls3=angka/10;
sats3=angka%10;
if(!up){delay_ms(200);settemp=settemp+1;}
if(settemp>2){settemp=0;}
if(settemp==0){set=32;}
if(settemp==1){set=34;}
if(settemp==2){set=36;}



//

{

delay_ms(1);
PORTD=0b11111111;
delay_ms(1);
PORTD=0b11111101;
PORTC=bil[sats3];

delay_ms(1);
PORTD=0b11111111;
delay_ms(1);
PORTD=0b11111110;
PORTC=bil[puls3];
delay_ms(1);
```

```
}

}
void settimes(){

char pul2,sat2,pul1,sat1;
pul2=0;
pul1=0;
sat1=0;

if(!up){delay_ms(200);settime=settime+1;}
if(settime>5 ){settime=0;}
if(settime==0){sat2=1;}
if(settime==1){sat2=2;}
if(settime==2){sat2=3;}
if(settime==3){sat2=4;}
if(settime==4){sat2=5;}
if(settime==5){sat2=6;}
{




delay_ms(1);
PORTD=0b11111111;
delay_ms(1);
PORTD=0b11011111;
PORTC=bil[sat1];
```

```
delay_ms(1);
PORTD=0b11111111;
delay_ms(1);
PORTD=0b11101111;
PORTC=bil[pul1];


delay_ms(1);
PORTD=0b11111111;
delay_ms(1);
PORTD=0b11110111;
PORTC=bil[sat2];



delay_ms(1);
PORTD=0b11111111;
delay_ms(1);
PORTD=0b11111011;
PORTC=bil[pul2];
}

}



void lihat3(unsigned char angka )
```

```
{

//
pul4=angka/10;
sat4=angka%10;

{

delay_ms(1);
PORTD=0b11111111;
delay_ms(1);
PORTD=0b01111111;
PORTC=bil[sat4];

delay_ms(1);
PORTD=0b11111111;
delay_ms(1);
PORTD=0b10111111;
PORTC=bil[pul4];
delay_ms(1);




}

}
```

```
void lihat2(unsigned char angka )

{

//
pul3=angka/10;
sat3=angka%10;

{

delay_ms(1);
PORTD=0b11111111;
delay_ms(1);
PORTD=0b11111101;
PORTC=bil[sat3];

delay_ms(1);
PORTD=0b11111111;
delay_ms(1);
PORTD=0b11111110;
PORTC=bil[pul3];
delay_ms(1);




}
```

```
}

void lihat1(  )

{


if(detik==60){sat1=sat1-1;detik=0;}
if(sat1<0){pul1=pul1-1;sat1=9;}
if(pul1<0){sat2=sat2-1;pul1=5;}
//if(sat2>9){pul2=pul2-1;}



{


delay_ms(1);
PORTD=0b11111111;
delay_ms(1);
PORTD=0b11011111;
PORTC=bil[sat1];


delay_ms(1);
PORTD=0b11111111;
delay_ms(1);
PORTD=0b11101111;
PORTC=bil[pul1];
```

```
delay_ms(1);
PORTD=0b11111111;
delay_ms(1);
PORTD=0b11110111;
PORTC=bil[sat2];



delay_ms(1);
PORTD=0b11111111;
delay_ms(1);
PORTD=0b11111011;
PORTC=bil[pul2];

}

}


void run(){
menit=0;
detik=58;
settimes();
sat2=settime+1;

while(1){
PORTB.0=0;
```

```
    suhu1=read_adc(0);
    suhu_c1=(float)suhu1*500/1023;
    if(sat2==-1){break;}
    suhu2=read_adc(4);
    suhu_c2=((   (float)suhu2*500/1023)+5);
      while(1){


lihat1();
lihat2(suhu_c2);
lihat3(set);

if(suhu_c1>set) {PORTB.1=1;}
if(suhu_c1<=set){PORTB.1=0;}

if(suhu_c2>set) {PORTB.1=1;}
if(suhu_c2<=set){PORTB.1=0;}
if(suhu_c1>=70)
if(sat2==-1){break;}
if(detik==10 ||detik==20 ||detik==30||detik==40||detik==50||detik==60){break;}
}
}
}
void main(void)
{
// Declare your local variables here

// Input/Output Ports initialization
```

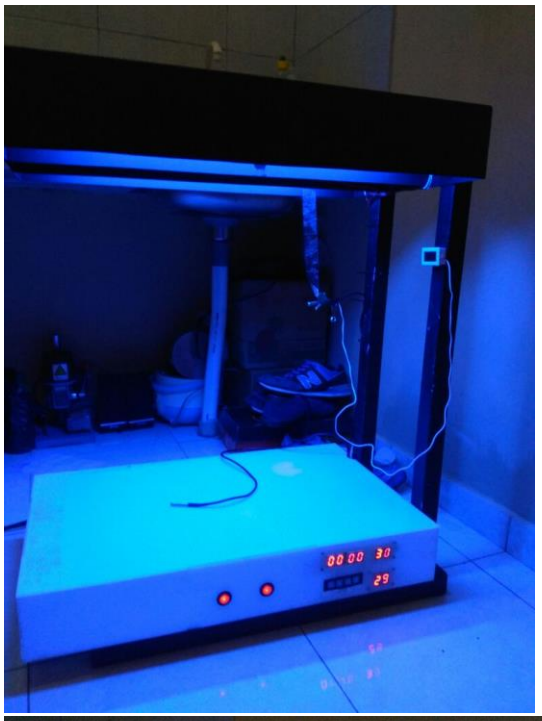
```
// Port A initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTA=0x0E;
DDRA=0x00;

// Port B initialization
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out Func1=Out Func0=Out
// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=0 State0=0
PORTB=0x00;
DDRB=0xFF;

// Port C initialization
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out Func1=Out Func0=Out
// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=0 State0=0
PORTC=0x00;
DDRC=0xFF;

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

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
```

```
// Mode: Normal top=0xFF
// OC0 output: Disconnected
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: 11.719 kHz
// Mode: Normal top=0xFFFF
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer1 Overflow Interrupt: On
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x05;
TCNT1H=0xD2;
TCNT1L=0x3A;
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
// INT2: Off
MCUCR=0x00;
MCUCSR=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x04;

// 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: 750.000 kHz
// ADC Voltage Reference: AREF pin
ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0x84;

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

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

// Global enable interrupts
#asm("sei")

while (1)
    {
        detik=0;    // Place your code here
        if(!ops){delay_ms(200);menu=menu+1;}
        if(menu>1){menu=0;}
        if(menu==0){settemps(set);}
        if(menu==1){settimes();}
        PORTB.1=1;
        PORTB.0=1;
        if(!ok){run();}
```

```
    }
}
```


00:00 30
29