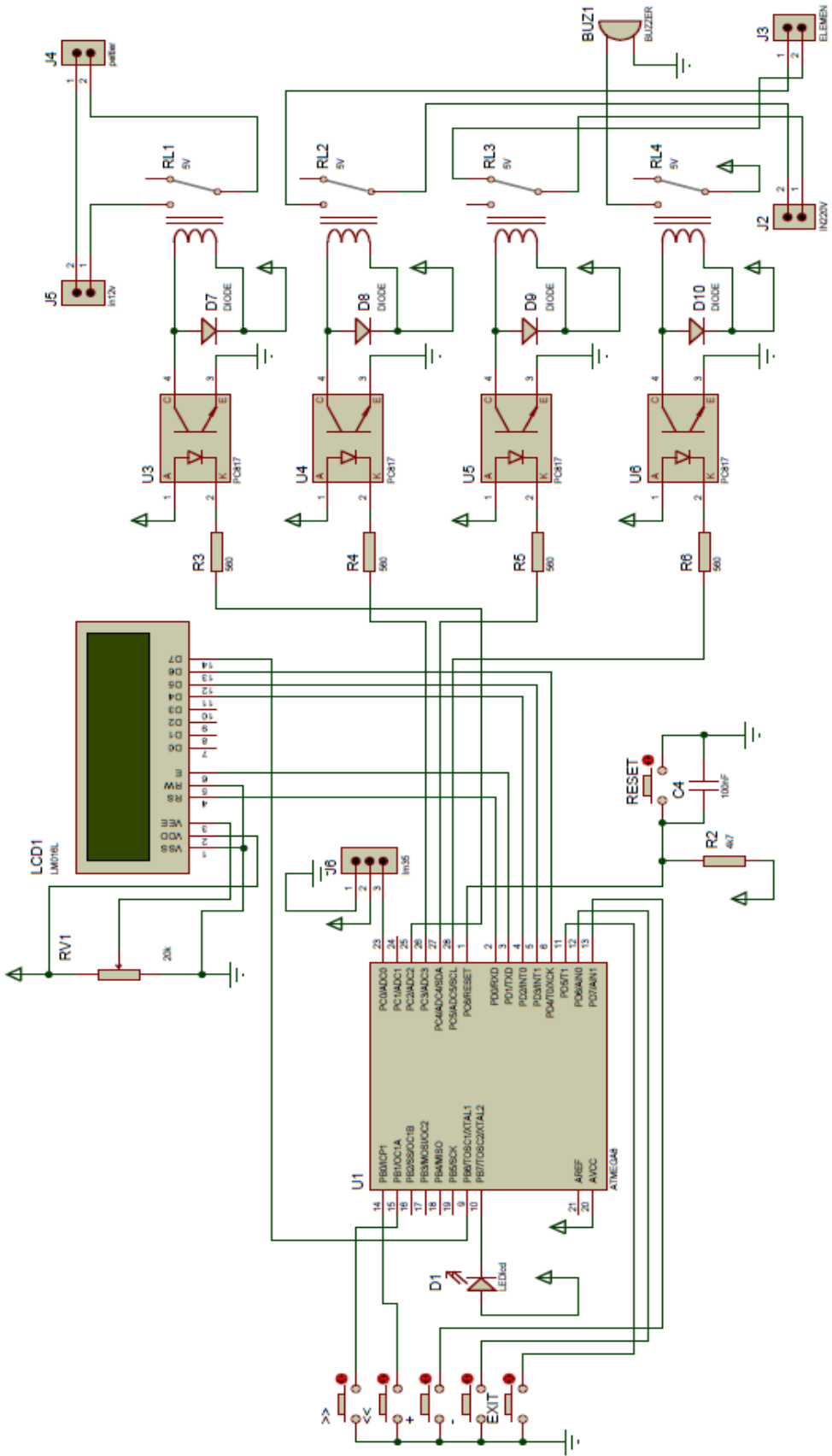


LAMPIRAN

SKEMATIK KESELURUHAN



```

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

#include <delay.h>

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

#define sw1 PINB.1
#define sw2 PINB.0
#define sw3 PIND.7
#define sw4 PIND.6
#define sw5 PIND.5

#define peltier PORTC.1

#define elemen1 PORTC.2
#define elemen2 PORTC.3

#define buzzer PORTC.4

#define lm35 0

int menit=0, detik=0;
float suhu;
eeprom unsigned char timer=1, timer1=1;
unsigned char pendingin=18, pemanas=40;
unsigned char mode=1, tanda=0, menu=0, menu1=0;
/*
rumus timer:
(16bit+1)-(1detik*(xtal/prescaller)
TCNT: (65535+1)+(1*(8mhz/1024))
TCNT: 57723
jadikan HEXADESIMAL
TCNT: E17B
*/

// Timer1 overflow interrupt service routine
interrupt [TIM1_OVF] void timer1_ovf_isr(void)
{
// Reinitialize Timer1 value
TCNT1H=0xE17B >> 8;

```

```

TCNT1L=0xE17B & 0xff;
// Place your code here
if(tanda==1){
if(menit<timer){
detik++;
if(detik>59){menit++;detik=0;}
}
}

if(tanda==2){
if(menit<timer1){
detik++;
if(detik>59){menit++;detik=0;}
}
}

}

// 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 baca_suhu(){
// nilai 500 di dapat dari 5000mv(5v)/10mv : 500
// ketelitian lm35 10mv/derajat
// nilai 1023 adalah nilai tertinggi (5v) pada pembacaan adc
float factor_error=3.2;
long int index,avg=0;
float temp,average;
for(index=0;index<100;index++){avg=avg+read_adc(lm35);delay_ms(1);}
average=(float)avg/100-factor_error;
temp=(float)average*500/1023; //rumus untuk mengubah kedalam derajat celcius
return temp;
}

```

```

void program_utama(){
suhu=baca_suhu();

if(sw4==0&&tanda==0){
if(mode==0)menu++;
if(menu1==0){
mode=0;
}
}
if(sw5==0&&tanda==0){
if(mode==1)menu1++;
if(menu==0){
mode=1;
}
}

if(menu>1)menu=0;
if(menu1>1)menu1=0;

if(tanda==1){

if(menit<timer){
lcd_clear();
lcd_gotoxy(0,0);
sprintf(lcd_buff,"T: %d Menit",timer);
lcd_puts(lcd_buff);
lcd_gotoxy(0,1);
sprintf(lcd_buff,"    %d:%d",menit,detik);
lcd_puts(lcd_buff);
}
else{

lcd_clear();
lcd_gotoxy(0,0);
lcd_putsf("Timer Selesai");
lcd_gotoxy(0,1);
lcd_putsf("Tekan Reset!");

while(1){
if(buzer==1)buzer=0;
else buzer=1;
delay_ms(300);
}
}
}
}

```

```

}

if(tanda==2){

if(menit<timer1){
lcd_clear();
lcd_gotoxy(0,0);
sprintf(lcd_buff,"T: %d Menit",timer1);
lcd_puts(lcd_buff);
lcd_gotoxy(0,1);
sprintf(lcd_buff,"    %d:%d",menit,detik);
lcd_puts(lcd_buff);
}
else{

lcd_clear();
lcd_gotoxy(0,0);
lcd_putsf("Timer Selesai");
lcd_gotoxy(0,1);
lcd_putsf("Tekan Reset!");

while(1){
if(buzer==1)buzer=0;
else buzer=1;
delay_ms(300);
}
}

}

if(tanda==0){
if(mode==0&&menu==0){

lcd_clear();
lcd_gotoxy(0,0);
sprintf(lcd_buff,"Suhu: %.1f ",suhu);
lcd_puts(lcd_buff);
lcd_putchar(0xdf);
lcd_putchar('C');
lcd_gotoxy(0,1);
sprintf(lcd_buff,"pendingin: %d",pendingin);
lcd_puts(lcd_buff);
lcd_putchar(0xdf);
lcd_putchar('C');

```

```

if(sw1==0){peltier=0; }

if(suhu<pendingin) { peltier=1; tanda=1; }

elemen1=1;
elemen2=1;

}

if(mode==0&&menu==1){

lcd_clear();
lcd_gotoxy(0,0);
sprintf(lcd_buff,"T: %d Menit",timer);
lcd_puts(lcd_buff);

if(sw2==0)timer++;
if(sw3==0)timer--;
peltier=1;
elemen1=1;
elemen2=1;
}

if(mode==1&&menu1==0){

lcd_clear();
lcd_gotoxy(0,0);
sprintf(lcd_buff,"Suhu: %.1f",suhu);
lcd_puts(lcd_buff);
lcd_putchar(0xdf);
lcd_putchar('C');
lcd_gotoxy(0,1);
sprintf(lcd_buff,"Pemanas: %d",pemanas);
lcd_puts(lcd_buff);
lcd_putchar(0xdf);
lcd_putchar('C');

if(sw1==0){elemen1=0; elemen2=0; }

if(suhu>pemanas) {elemen1=1; elemen2=1; tanda=2; }
peltier=1;
}

if(mode==1&&menu1==1){

lcd_clear();

```

```

lcd_gotoxy(0,0);
sprintf(lcd_buff,"T: %d Menit",timer1);
lcd_puts(lcd_buff);

if(sw2==0)timer1++;
if(sw3==0)timer1--;
peltier=1;
elemen1=1;
elemen2=1;
}

}

delay_ms(200);
}

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=In
Func0=In
// State7=0 State6=T State5=T State4=T State3=T State2=T State1=P State0=P
PORTB=0x03;
DDRB=0x80;

// Port C initialization
// Func6=In Func5=In Func4=Out Func3=Out Func2=Out Func1=Out Func0=In
// State6=T State5=T State4=0 State3=0 State2=0 State1=0 State0=T
PORTC=0x1E;
DDRC=0x1E;

// 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: 7,813 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=0xE1;
TCNT1L=0x7B;
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=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: 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 - PORTB Bit 4
// 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);

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

lcd_clear();
lcd_gotoxy(0,0);
lcd_putsf("WATERBATH");
lcd_gotoxy(0,1);
lcd_putsf("THERAPI");
delay_ms(1500);

while (1)
{
// Place your code here

```

```
program_utama();
```

```
}
```

```
}
```