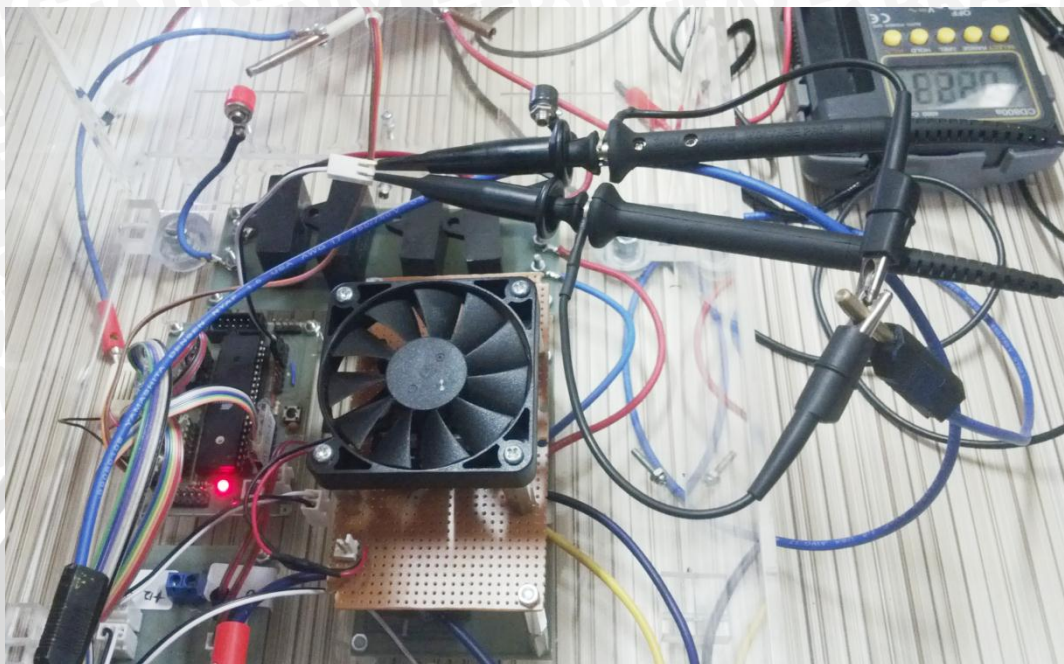
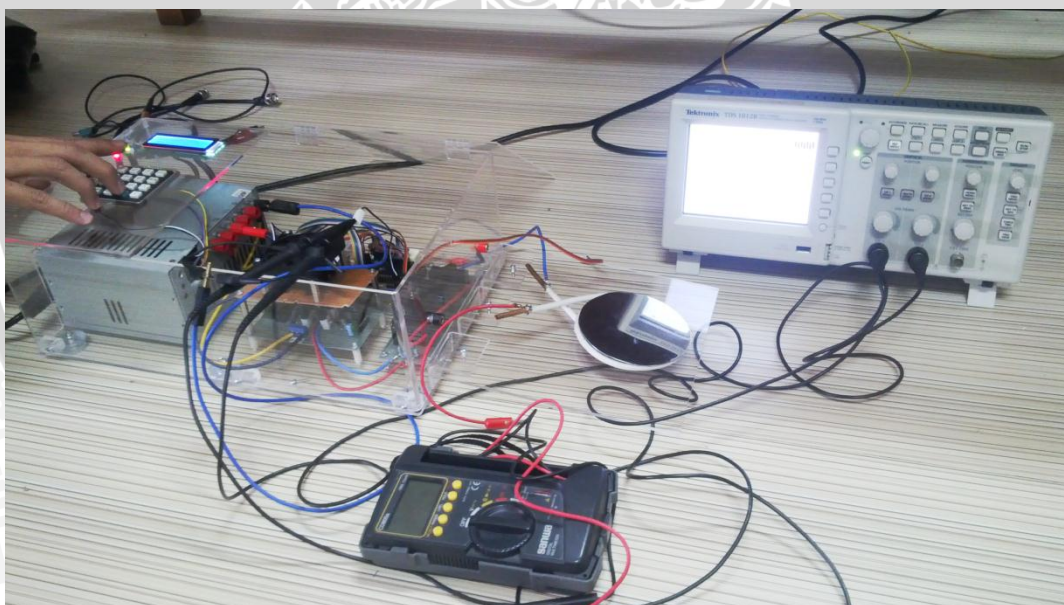


## LAMPIRAN

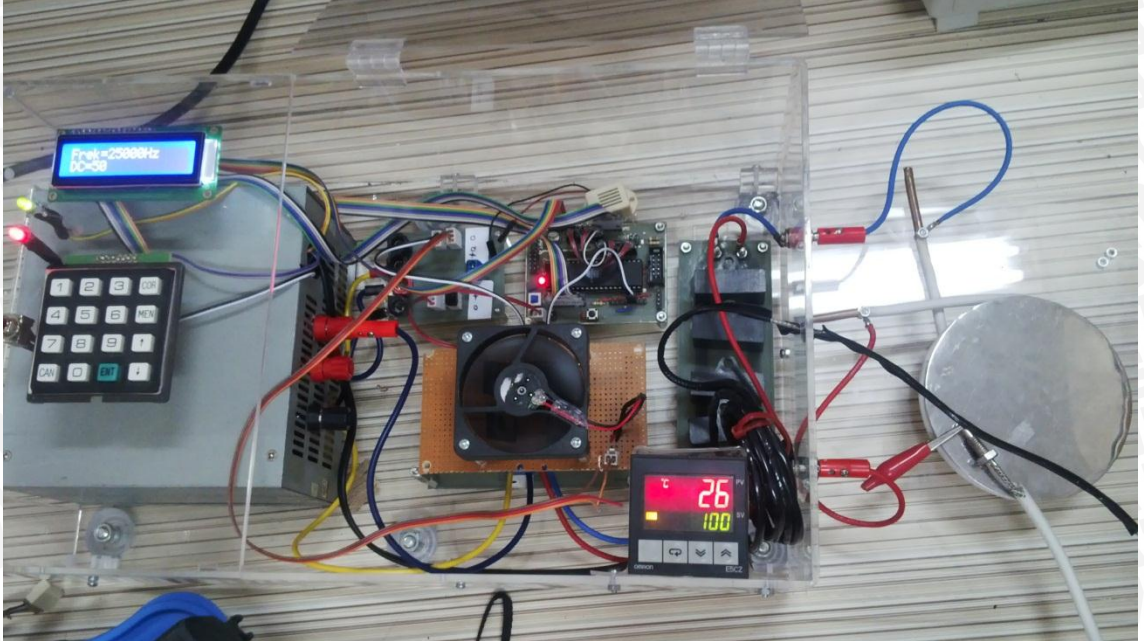
### I. Gambar alat



*Pengujian frekuensi pada masing-masing blok sistem*



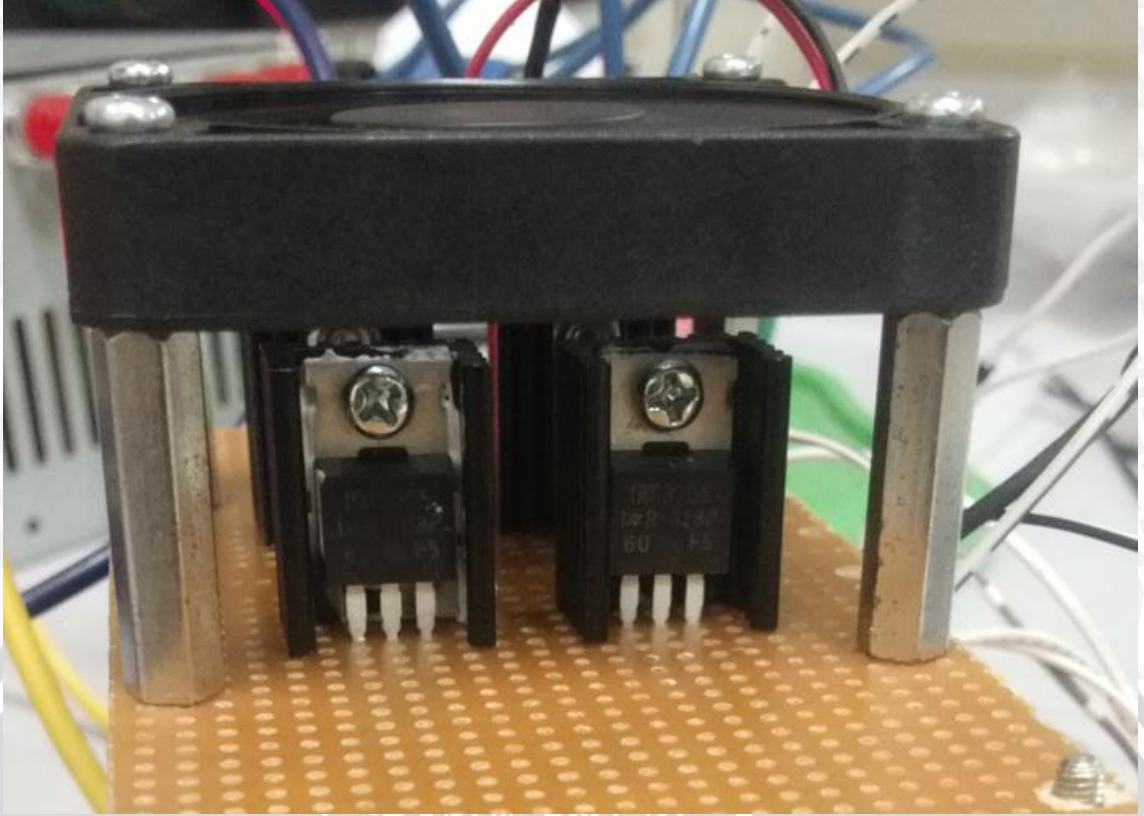
*Pengujian frekuensi pada masing-masing blok sistem*



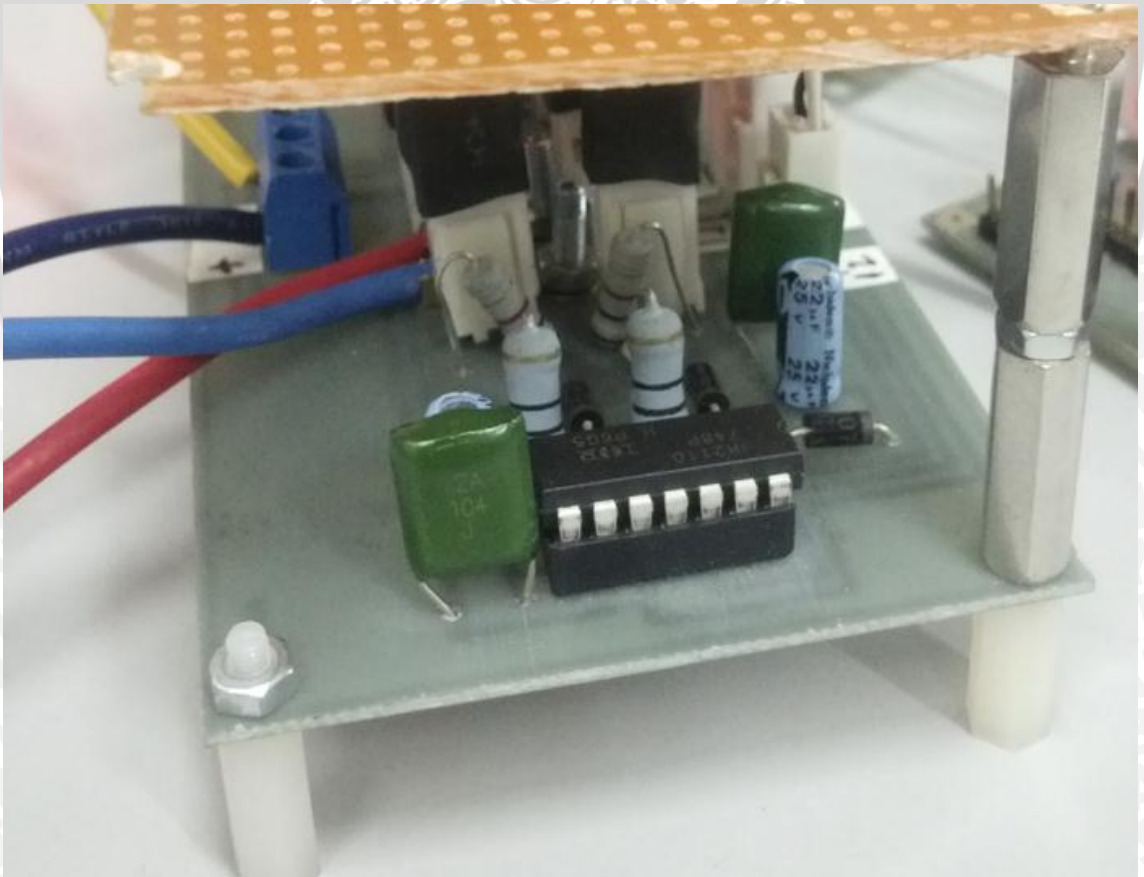
*Pengujian keseluruhan pemanas induksi pada beberapa frekuensi kerja*



*Benda kerja dan thermocouple pada saat pengujian*

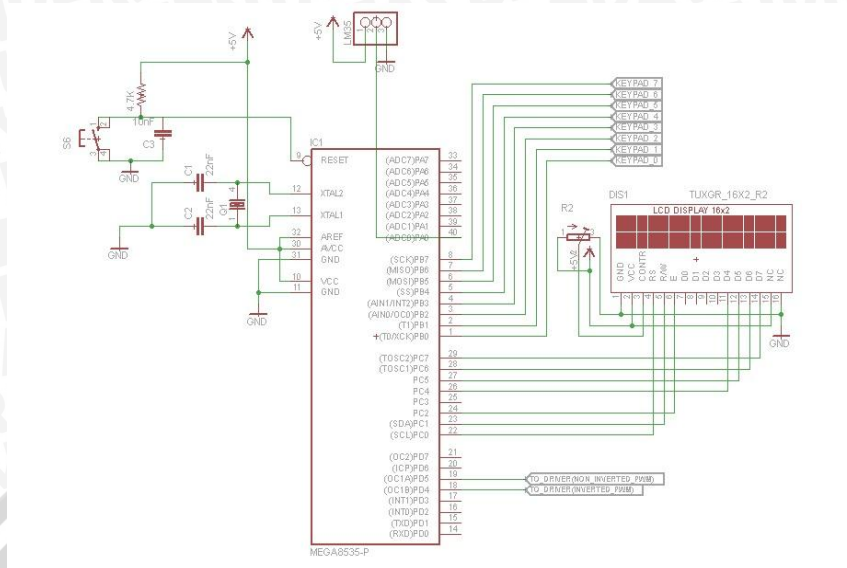


*Mosfet daya IRF3205 untuk switching frekuensi tinggi*

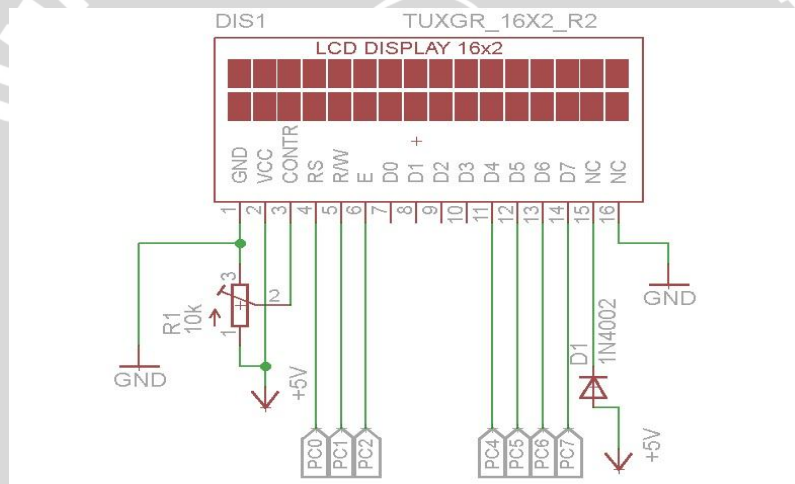


*IC IR2110 sebagai driver mosfet*

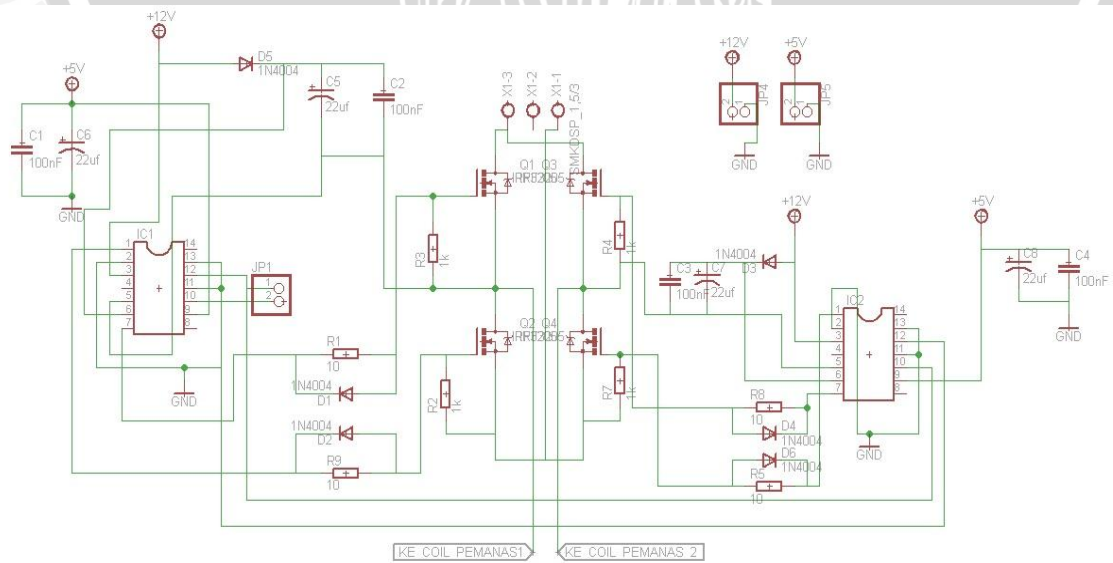
## II. Rangkaian keseluruhan



Rangkaian Mikrokontroler Pemroses Utama



Rangkaian Antarmuka Modul LCD



Schematic Rangkaian Driver Switching

### III. Listing Program

```
/******
```

```
This program was created by the  
CodeWizardAVR V3.04 Evaluation  
Automatic Program Generator  
© Copyright 1998-2013 Pavel Haiduc, HP InfoTech s.r.l.  
http://www.hpinfotech.com
```

```
Project :  
Version :  
Date : 18/08/2013  
Author :  
Company :  
Comments:
```

```
Chip type : ATmega8535  
Program type : Application  
AVR Core Clock frequency: 16,000000 MHz  
Memory model : Small  
External RAM size : 0  
Data Stack size : 128  
*****
```

```
//===== header =====
```

```
#include <mega8535.h>  
#include <delay.h>  
#include <stdio.h>  
#include <stdlib.h>  
#include <alcd.h>
```

```
//===== predefine =====
```

```
//KEYPAD  
#define DDR_keypad DDRB  
#define PORT_keypad PORTB  
#define PIN_keypad PINB  
#define HI_keypad(x) PORT_keypad|(1<<x)  
#define LO_keypad(x) PORT_keypad&=~(1<<x)  
#define IS_HI(x) PIN_keypad&(1<<x)
```

```
//===== FUNGSI PROTOTYPE =====
```

```
void inisialisasi_sistem();  
unsigned char scan_keypad();  
unsigned int read_adc(unsigned char adc_input);  
void validasi();  
void scan_switch();  
void scan(char y,char l);  
void enter_setting();  
void mode_menu();
```

```
//===== GLOBAL VARIABLE =====
```

```

char x;
unsigned char key,keyp,n;
char frek[10],dutc[10],derajat[10],tess[10],de[10];
unsigned int dutc_set;
unsigned char scan_keypad();
unsigned long int ocra,icr;
long int a,frek_set;

```

```

void main(void)
{
inisialisasi_sistem();
lcd_clear();
mode_menu();
}
//===== ANTARMUKA KEYPAD =====
/*
PINB.3 PINB.2 PINB.1 PINB.0
| | | |
-----
1 2 3 cor |-- PORTB.7
|
4 5 6 men |-- PORTB.6
|
7 8 9 up |-- PORTB.5
|
can 0 ent down |-- PORTB.4

*/
unsigned char scan_keypad()
{
unsigned char tombol[16]={'<','e','0','c','>','9','8','7','#','6','5','4','d','3','2','1'};
unsigned char kolom,baris,tombolke=0,key_press;
key_press='$'; //key_press bernilai karakter $ bila tidak ditekan
DDR_keypad=0xF0;
for(baris=4;baris<=7;baris++)
{
PORT_keypad=0xFF;
LO_keypad(baris);
for(kolom=0;kolom<=3;kolom++)
{
//jika ada yang ditekan maka beri nilai key_press sesuai dengan karakter
yang
//tersimpan dalam array tombol[]
if(~IS_HI(kolom)) key_press=tombol[tombolke];
tombolke++;
}
}
return key_press;
}
//===== VALIDASI =====

```

```

void validasi()
{
    key='$';
    while(key!='e'){

key=scan_keypad();keyp=key;while(keyp!='$'){keyp=scan_keypad();delay_ms(50);}
    }
}
//===== SWITCH =====
void scan(char y, char l){
    key='$';
    while(key=='$'||key=='<'||key=='e'||key=='>'||key=='#'||key=='d'){

key=scan_keypad();keyp=key;while(keyp!='$'){keyp=scan_keypad();delay_ms(50);}
    }
    lcd_gotoxy(x+5,y);lcd_putchar(key);if(l==1){frek[x]=key;}else
if(l==2){dutc[x]=key;}else if(l==3){derajat[x]=key;}
    key=keyp;
    x++;
}
void scan_switch(){
    key='$';
    while(key=='$'||key=='<'||key=='e'||key=='c'||key=='>'||key=='#'||key=='d'){

key=scan_keypad();keyp=key;while(keyp!='$'){keyp=scan_keypad();delay_ms(50);}
    }
}
//===== SETTING FREKUENSI =====
void enter_setting()
{
    lcd_clear();
    //masukkan frekuensi
    x=0;
    lcd_gotoxy(0,1);lcd_putsf("Frek: ");
    while(x<=4){scan(1,1);}
    frek[5]='\0';
    frek_set=atol(frek);
    validasi();lcd_clear();
    //masukkan duty cycle
    x=0;
    lcd_gotoxy(0,1);lcd_putsf("Duty Cycle: ");
    while(x<=1){scan(4,2);}
    dutc[2]='\0';
    dutc_set=atoi(dutc);
    validasi();lcd_clear();

    a=(frek_set);
    sprintf(tess,"Frek=%uHz",a);
    lcd_gotoxy(0,0);lcd_puts(tess);
    sprintf(de,"DC=%u%",dutc_set);
}

```

```

lcd_gotoxy(0,1);lcd_puts(de);
validasi();lcd_clear();
//perhitungan nilai ICR dan OCR
icr=((16000000/a)+3)/4;
ocra=((icr*dutc_set)/100);
ICR1H=icr/256;
ICR1L=icr%256;
OCR1A=ocra;
OCR1B=OCR1A;
PORTA.1=1;delay_ms(100);
PORTA.1=0;
lcd_gotoxy(0,0);lcd_putsf("Heating Process");
validasi();
ICR1H=0;ICR1L=0;
OCR1A=0;OCR1B=0;
lcd_clear();
lcd_gotoxy(2,0);lcd_putsf("Stop Process");
lcd_gotoxy(4,1);lcd_putsf("Cooling");
}
//===== TEST LCD =====
void lcd_test()
{
  lcd_clear();
  for(n=0;n<20;n++){
    lcd_gotoxy(4,0);lcd_putsf("LOADING");
    lcd_gotoxy(n,1);
    lcd_putsf("\xff");
    delay_ms(30);
  }
  lcd_clear();
  lcd_gotoxy(3,0);lcd_putsf("-SKRIPSI-");
  lcd_gotoxy(0,1);lcd_putsf("PEMANAS INDUKSI");
  delay_ms(500);lcd_clear();
  lcd_gotoxy(4,0);lcd_putsf("YUKOVANY");
  lcd_gotoxy(3,1);lcd_putsf("0810630105");
  //nim 0810630105
}
//===== MODE MENU =====
void mode_menu()
{
  lcd_clear();
  lcd_gotoxy(0,0);lcd_putsf("MODE MENU");
  lcd_gotoxy(0,1);lcd_putsf("Press Ent to Set");
  validasi();
  enter_setting();
};
//===== KONVERSI ADC =====
// Read the AD conversion result
// Voltage Reference: AREF pin
#define ADC_VREF_TYPE ((0<<REFS1) | (0<<REFS0) | (0<<ADLAR))

```



```

unsigned int read_adc(unsigned char adc_input)
{
  ADMUX=adc_input | ADC_VREF_TYPE;
  // Delay needed for the stabilization of the ADC input voltage
  delay_us(10);
  // Start the AD conversion
  ADCSRA|=(1<<ADSC);
  // Wait for the AD conversion to complete
  while ((ADCSRA & (1<<ADIF))==0);
  ADCSRA|=(1<<ADIF);
  return ADCW;
}
//===== INISIALISASI SISTEM =====
void inisialisasi_sistem()
{
  // Declare your local variables here

  // Input/Output Ports initialization
  // Port A initialization
  // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=Out Bit0=In
  DDRA=(0<<DDA7) | (0<<DDA6) | (0<<DDA5) | (0<<DDA4) | (0<<DDA3) |
(0<<DDA2) | (1<<DDA1) | (0<<DDA0);
  // State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=0 Bit0=T
  PORTA=(0<<PORTA7) | (0<<PORTA6) | (0<<PORTA5) | (0<<PORTA4) |
(0<<PORTA3) | (0<<PORTA2) | (0<<PORTA1) | (0<<PORTA0);

  // Port B initialization
  // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
  DDRB=(0<<DDB7) | (0<<DDB6) | (0<<DDB5) | (0<<DDB4) | (0<<DDB3) |
(0<<DDB2) | (0<<DDB1) | (0<<DDB0);
  // State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
  PORTB=(0<<PORTB7) | (0<<PORTB6) | (0<<PORTB5) | (0<<PORTB4) |
(0<<PORTB3) | (0<<PORTB2) | (0<<PORTB1) | (0<<PORTB0);

  // Port C initialization
  // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
  DDRC=(0<<DDC7) | (0<<DDC6) | (0<<DDC5) | (0<<DDC4) | (0<<DDC3) |
(0<<DDC2) | (0<<DDC1) | (0<<DDC0);
  // State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
  PORTC=(0<<PORTC7) | (0<<PORTC6) | (0<<PORTC5) | (0<<PORTC4) |
(0<<PORTC3) | (0<<PORTC2) | (0<<PORTC1) | (0<<PORTC0);

  // Port D initialization
  // Function: Bit7=In Bit6=In Bit5=Out Bit4=Out Bit3=In Bit2=In Bit1=In
Bit0=In
  DDRD=(0<<DDD7) | (0<<DDD6) | (1<<DDD5) | (1<<DDD4) | (0<<DDD3) |
(0<<DDD2) | (0<<DDD1) | (0<<DDD0);
  // State: Bit7=T Bit6=T Bit5=0 Bit4=0 Bit3=T Bit2=T Bit1=T Bit0=T
  PORTD=(0<<PORTD7) | (0<<PORTD6) | (0<<PORTD5) | (0<<PORTD4) |
(0<<PORTD3) | (0<<PORTD2) | (0<<PORTD1) | (0<<PORTD0);

```

```

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: 16000,000 kHz
// Mode: Fast PWM top=ICR1
// OC1A output: Non-Inverted PWM
// OC1B output: Inverted PWM
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer Period: 0,0625 us
// Output Pulse(s):
// OC1A Period: 0,0625 us
// OC1B Period: 0,0625 us
// Timer1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=(1<<COM1A1) | (0<<COM1A0) | (1<<COM1B1) | (1<<COM1B0) |
(1<<WGM11) | (0<<WGM10);
TCCR1B=(0<<ICNC1) | (0<<ICES1) | (1<<WGM13) | (1<<WGM12) |
(0<<CS12) | (0<<CS11) | (1<<CS10);
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: 16000,000 kHz
// Mode: Fast PWM top=0xFF
// OC2 output: Disconnected
// Timer Period: 0,016 ms
ASSR=0<<AS2;
TCCR2=(1<<WGM20) | (0<<COM21) | (0<<COM20) | (1<<WGM21) |
(0<<CS22) | (0<<CS21) | (1<<CS20); //prescaler 1
TCNT2=0x00;
OCR2=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=(0<<OCIE2) | (0<<TOIE2) | (0<<TICIE1) | (0<<OCIE1A) |
(0<<OCIE1B) | (0<<TOIE1) | (0<<OCIE0) | (0<<TOIE0);

// ADC initialization
// ADC Clock frequency: 1000,000 kHz
// ADC Voltage Reference: AREF pin
// ADC High Speed Mode: Off

```

```
// ADC Auto Trigger Source: ADC Stopped
ADMUX=ADC_VREF_TYPE;
ADCSRA=(1<<ADEN) | (0<<ADSC) | (0<<ADATE) | (0<<ADIF) |
(0<<ADIE) | (1<<ADPS2) | (0<<ADPS1) | (0<<ADPS0);
SFIOR=(1<<ADHSM) | (0<<ADTS2) | (0<<ADTS1) | (0<<ADTS0);
```

```
// Alphanumeric LCD initialization
// Connections are specified in the
// Project|Configure|C Compiler|Libraries|Alphanumeric LCD menu:
// RS - PORTC Bit 0
// RD - PORTC Bit 1
// EN - PORTC Bit 2
// D4 - PORTC Bit 4
// D5 - PORTC Bit 5
// D6 - PORTC Bit 6
// D7 - PORTC Bit 7
// Characters/line: 16
lcd_init(16);
lcd_test();
lcd_clear();
}
```

