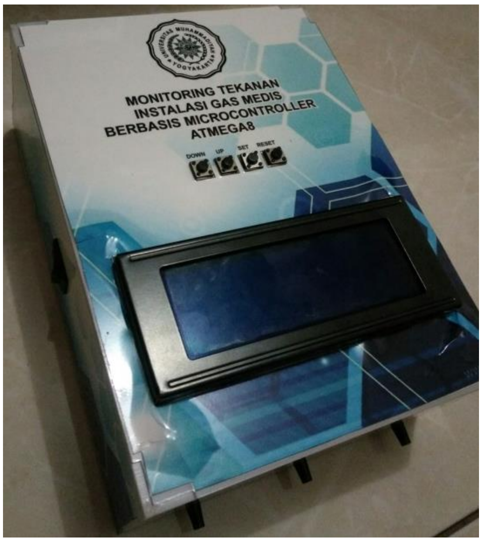
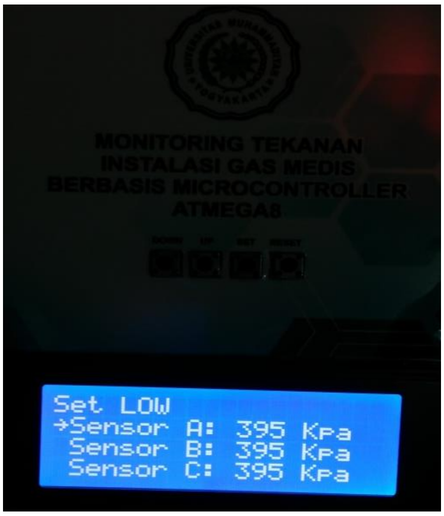
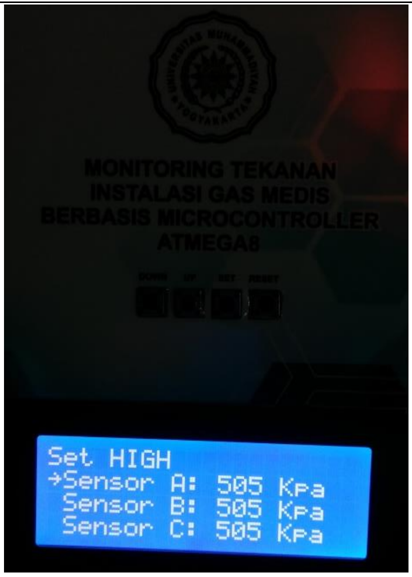
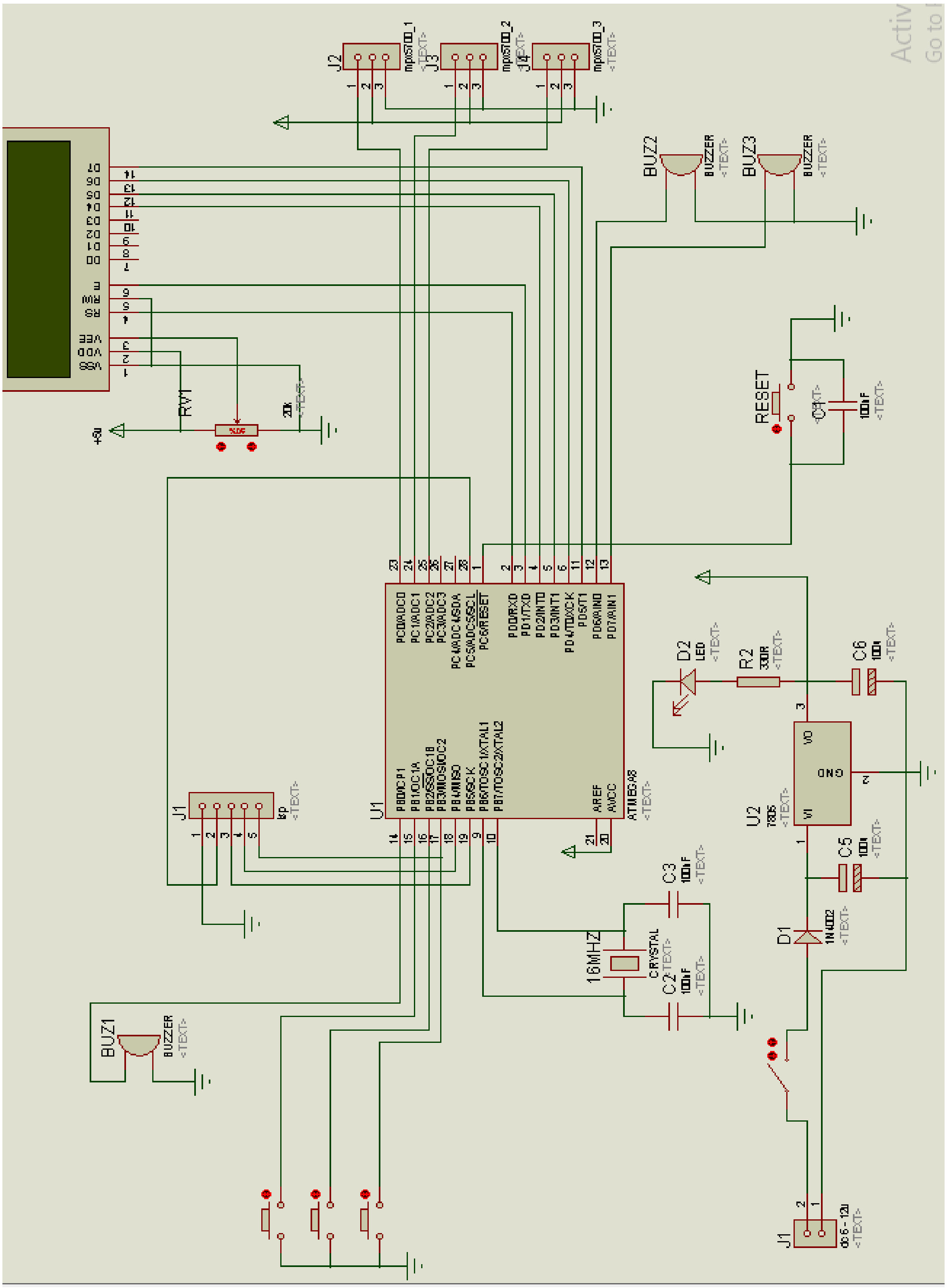
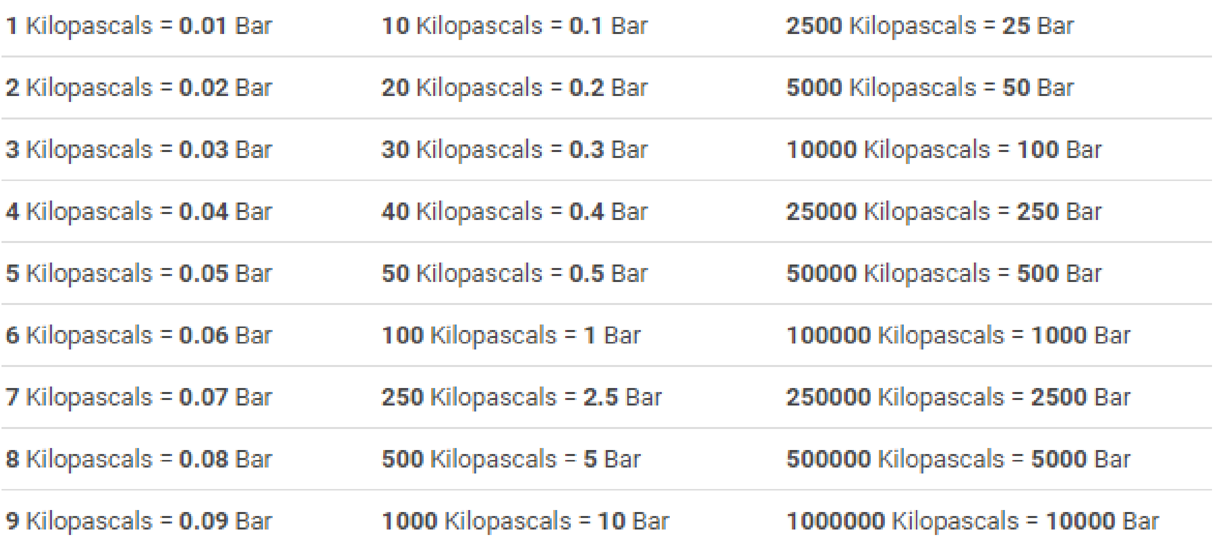
LAMPIRAN

 MONITORING TEKANAN INSTALASI GAS MEDIS BERBASIS MICROCONTROLLER ATMEGA8 DOWN UP SET RESET Foto alat	 Set LOW →Sensor A: 395 Kpa Sensor B: 395 Kpa Sensor C: 395 Kpa Setting low pressure
 Set HIGH →Sensor A: 505 Kpa Sensor B: 505 Kpa Sensor C: 505 Kpa Setting High pressure	 Proses pengambilan data

RANGKAIAN KESELURUHAN



KONVERSI KPA KE BAR



1 Kilopascals = **0.01** Bar	**10** Kilopascals = **0.1** Bar	**2500** Kilopascals = **25** Bar
2 Kilopascals = **0.02** Bar	**20** Kilopascals = **0.2** Bar	**5000** Kilopascals = **50** Bar
3 Kilopascals = **0.03** Bar	**30** Kilopascals = **0.3** Bar	**10000** Kilopascals = **100** Bar
4 Kilopascals = **0.04** Bar	**40** Kilopascals = **0.4** Bar	**25000** Kilopascals = **250** Bar
5 Kilopascals = **0.05** Bar	**50** Kilopascals = **0.5** Bar	**50000** Kilopascals = **500** Bar
6 Kilopascals = **0.06** Bar	**100** Kilopascals = **1** Bar	**100000** Kilopascals = **1000** Bar
7 Kilopascals = **0.07** Bar	**250** Kilopascals = **2.5** Bar	**250000** Kilopascals = **2500** Bar
8 Kilopascals = **0.08** Bar	**500** Kilopascals = **5** Bar	**500000** Kilopascals = **5000** Bar
9 Kilopascals = **0.09** Bar	**1000** Kilopascals = **10** Bar	**1000000** Kilopascals = **10000** Bar

PERHITUNGAN STATISTIK

a. Rata – rata

Rata-rata adalah nilai atau hasil pembagian dari jumlah data yang diambil atau diukur dengan banyaknya pengambilan data atau banyaknya pengukuran.

$$\boxed{\text{Rata – Rata } (\overline{X}) = \frac{\Sigma Xi}{n}}$$

Dimana : $\overline{X}$ = rata-rata

$\sum$Xi = Jumlah nilai data

n = Banyak data (1,2,3,…,n)

1. Pengukuran tekanan 300 kPa

$$\frac{285+286+286+287+287+287+287+287+286+287+286+287+287+287+286+287+287+287+287+287}{20}$$

= 286,6 kPa

2. Pengukuran tekanan 400 kPa

$$\frac{396+396+397+398+396+396+396+396+398+398+398+396+397+398+396+397+397+397+396+396}{20}$$

=396,75 kPa

3. Pengukuran tekanan 500 kPa

$$\frac{492+493+494+491+492+491+491+492+492+492+492+494+491+492+491+492+491+492+492+492}{20}$$

= 491,95 kPa

4. Pengukuran tekanan 550 kPa

$$\frac{539+539+540+540+540+540+540+540+539+540+540+540+540+540+540+540+539+539+541+539}{20}$$

= 539,75 kPa

b. Simpangan

Simpangan adalah selisih dari rata– rata nilai harga yang dikehendaki dengan nilai yang diukur. Berikut rumus dari simpangan :

$$\boxed{\text{Simpangan} = \mathrm{Y} - \overline{X}}$$

Dimana : Y = suhu *setting*

$\overline{X}$ = rata –rata

1. Pengukuran tekanan 300 kPa

   Simpangan = 300 - 286,6

   = 13,4

2. Pengukuan tekanan 400 kPa

   Simpangan 400 - 396,75

   = 3,25

3. Pengukuran tekanan 500 kPa

   Simpangan 500 - 491,95

   = 8,05

4. Pengukuran 550 kPa

   Simpangan 550 - 539,75

   = 10,25

c. *Error* (%)

Error (kesalahan) adalah selisih antara *mean* terhadap masing-masing data. Rumus *error* adalah:

$$\boxed{\begin{array}{c} \text{\textit{Error} \%=} \\ \left(\frac{DataSetting - \mathrm{Re}\,rata}{Datasetting}\right) x100\% \end{array}}$$

1. Pengukuran tekanan 300 kPa

$$\frac{300 - 286{,}6}{300} x100$$

= 4,46%

2. Pengukuran tekanan 400 kPa

$$\frac{400 - 396{,}75}{400} x100$$

= 0,81%

3. Pengukuran tekanan 500 kPa

$$\frac{500 - 491{,}95}{500} x100$$

= 1,61%

4. Pengukuran tekanan 550 kPa

$$\frac{550 - 539{,}75}{550} x100$$

= 1,86%

PROGRAM ALAT

```
#include <mega8.h>
#include <stdio.h>
#include <delay.h>
#define buzzer1 PORTB.0
#define buzzer2 PORTD.7
#define buzzer3 PORTD.6
#define s1      PINB.3
#define s2      PINB.4
#define s3      PINB.5
char buff[33];
// Alphanumeric LCD functions
#include <alcd.h>
#define ADC_VREF_TYPE 0x40
eeprom int h_batas1=10,h_batas2=10,h_batas3=10;
eeprom int l_batas1=5,l_batas2=5,l_batas3=5;
int go=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  v_mpx(int ch){
float sample=0,ratarata=0;
float volt;
int i;
for(i=0; i<50; i++){
sample = sample + read_adc(ch);
delay_ms(1);
}
ratarata =  sample / 50;
volt=ratarata*((float)5/1023);
return volt;
}
float tekanan1(){
 float offset=0.83;
 float volt=v_mpx(0)-offset;  // mengubah nilai adc ke tegangan dikurangi offset
 float pressure=volt*((float)700/((float)4.7-offset)); // mengubah ke kpa
 // memabatasi minimal 0
 if(pressure<0)pressure=0;
// memberi nilai balik
return pressure;
}
```

```
float tekanan2(){
 float offset=0.839;
 float volt=v_mpx(1)-offset;  // mengubah nilai adc ke tegangan dikurangi offset
 float pressure=volt*((float)700/((float)4.7-offset)); // mengubah ke kpa
 // memabatasi minimal 0
 if(pressure<0)pressure=0;
// memberi nilai balik
return pressure;
}
float tekanan3(){
 float offset=0.83;
 float volt=v_mpx(2)-offset;  // mengubah nilai adc ke tegangan dikurangi offset
 float pressure=volt*((float)700/((float)4.7-offset)); // mengubah ke kpa
 // memabatasi minimal 0
 if(pressure<0)pressure=0;
// memberi nilai balik
return pressure;
}
void mpx_test(){
float v1,v2,v3;
v1 = v_mpx(0);
v2 = v_mpx(1);
v3 = v_mpx(2);
lcd_clear();
lcd_gotoxy(0,0);
sprintf(buff,"Sensor A: %.3f V",v1);
lcd_puts(buff);
```

```
lcd_gotoxy(0,1);
sprintf(buff,"Sensor B: %.3f V",v2);
lcd_puts(buff);
lcd_gotoxy(0,2);
sprintf(buff,"Sensor C: %.3f V",v3);
lcd_puts(buff);
delay_ms(100);
}
void atur(){
int menu=0;
buzzer1=0;
buzzer2=0;
buzzer3=0;
lcd_clear();
delay_ms(200);
while(1){
if(s1==0) menu++;
if(menu>5)break;
lcd_clear();
if(menu<=2){
lcd_gotoxy(0,0);
lcd_putsf("Set HIGH");
lcd_gotoxy(0,1);
if(menu==0)lcd_putchar('~');
lcd_gotoxy(1,1);
sprintf(buff,"Sensor A: %d kPa",h_batas1);
lcd_puts(buff);
```

```
lcd_gotoxy(0,2);
if(menu==1)lcd_putchar('~');
lcd_gotoxy(1,2);
sprintf(buff,"Sensor B: %d kPa",h_batas2);
lcd_puts(buff);
lcd_gotoxy(0,3);
if(menu==2)lcd_putchar('~');
lcd_gotoxy(1,3);
sprintf(buff,"Sensor C: %d kPa",h_batas3);
lcd_puts(buff);
if(menu==0){
if(s2==0) h_batas1++;
if(s3==0) h_batas1--;
if(h_batas1>700)h_batas1=1;
if(h_batas1<1)h_batas1=700;
}
if(menu==1){
if(s2==0) h_batas2++;
if(s3==0) h_batas2--;
if(h_batas2>700)h_batas2=1;
if(h_batas2<1)h_batas2=700;
}
if(menu==2){
if(s2==0) h_batas3++;
if(s3==0) h_batas3--;
if(h_batas3>700)h_batas3=1;

if(h_batas3<1)h_batas3=700;

}}

if(menu>2){

lcd_gotoxy(0,0);

lcd_putsf("Set LOW");

lcd_gotoxy(0,1);

if(menu==3)lcd_putchar('~');

lcd_gotoxy(1,1);

sprintf(buff,"Sensor A: %d kPa",l_batas1);

lcd_puts(buff);

lcd_gotoxy(0,2);

if(menu==4)lcd_putchar('~');

lcd_gotoxy(1,2);

sprintf(buff,"Sensor B: %d kPa",l_batas2);

lcd_puts(buff);

lcd_gotoxy(0,3);

if(menu==5)lcd_putchar('~');

lcd_gotoxy(1,3);

sprintf(buff,"Sensor C: %d kPa",l_batas3);

lcd_puts(buff);

if(menu==3){

if(s2==0) l_batas1++;

if(s3==0) l_batas1--;

if(l_batas1>700)l_batas1=1;

if(l_batas1<1)l_batas1=700;

}
```

```
if(menu==4){
if(s2==0) l_batas2++;
if(s3==0) l_batas2--;
if(l_batas2>700)l_batas2=1;
if(l_batas2<1)l_batas2=700;
}
if(menu==5){
if(s2==0) l_batas3++;
if(s3==0) l_batas3--;
if(l_batas3>700)l_batas3=1;
if(l_batas3<1)l_batas3=700;
}}delay_ms(150);
}go=1;
}
void program_run(){
float sensora,sensorb,sensorc;
sensora=tekanan1();
sensorb=tekanan2();
sensorc=tekanan3();
if(s1==0) atur();
lcd_clear();
lcd_gotoxy(2,0);
lcd_putsf("Preasure Monitor");
lcd_gotoxy(0,1);
sprintf(buff,"O2 : %.1f kPa",sensora);
lcd_puts(buff);
```

```
lcd_gotoxy(19,1);
if(buzzer1==1) lcd_putchar(0xFF);
lcd_gotoxy(0,2);
sprintf(buff,"N2O: %.1f kPa",sensorb);
lcd_puts(buff);
lcd_gotoxy(19,2);
if(buzzer1==1) lcd_putchar(0xFF);
lcd_gotoxy(0,3);
sprintf(buff,"MCA: %.1f kPa",sensorc);
lcd_puts(buff);
lcd_gotoxy(19,3);
if(buzzer1==1) lcd_putchar(0xFF);
if(go==1&&(sensora>h_batas1 || sensora<l_batas1)){
buzzer1=1;
delay_ms(20);
buzzer1=0;
lcd_gotoxy(14,1);
if(sensora>h_batas1)lcd_putsf("MAX");
if(sensora<l_batas1)lcd_putsf("MIN");
}
else buzzer1=0;
if(go==1&&(sensorb>h_batas2 || sensorb<l_batas2)){
buzzer2=1;
delay_ms(40);
buzzer2=0;
lcd_gotoxy(14,2);
if(sensorb>h_batas2)lcd_putsf("MAX");
if(sensorb<l_batas2)lcd_putsf("MIN");
}
else buzzer2=0;
if(go==1&&(sensorc>h_batas3 || sensorc<l_batas3)){
buzzer3=1;
delay_ms(60);
buzzer3=0;
lcd_gotoxy(14,3);
if(sensorc>h_batas3)lcd_putsf("MAX");
if(sensorc<l_batas3)lcd_putsf("MIN");
}
else buzzer3=0;
//delay_ms(50);
}
void main(void)
{
// Declare your local variables here
// Input/Output Ports initialization
// Port B initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=Out
// State7=T State6=T State5=P State4=P State3=P State2=T State1=T State0=0
PORTB=0x38;
DDRB=0x01;
// 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=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTD=0x00;
DDRD=0xC0;
// 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: Timer1 Stopped
// Mode: Normal top=0xFFFF
// OC1A output: Discon.
// 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=0x00;
TCCR1B=0x00;
```

```
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=0x84;
// 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 - PORTD Bit 7
// EN - PORTD Bit 1
// D4 - PORTD Bit 2
// D5 - PORTD Bit 3
// D6 - PORTD Bit 4
// D7 - PORTD Bit 5
// Characters/line: 20
```

```
lcd_init(20);
lcd_clear();
lcd_gotoxy(2,0);
lcd_putsf("Preasure monitor");
lcd_gotoxy(0,2);
lcd_putsf("Intivada Khoirunnisa") ;
lcd_gotoxy(4,3);
lcd_putsf("20153010046") ;
delay_ms(1000);
lcd_clear();
while (1)
    {
    // Place your code here
    program_run();
    //mpx_test();
    }
}
```