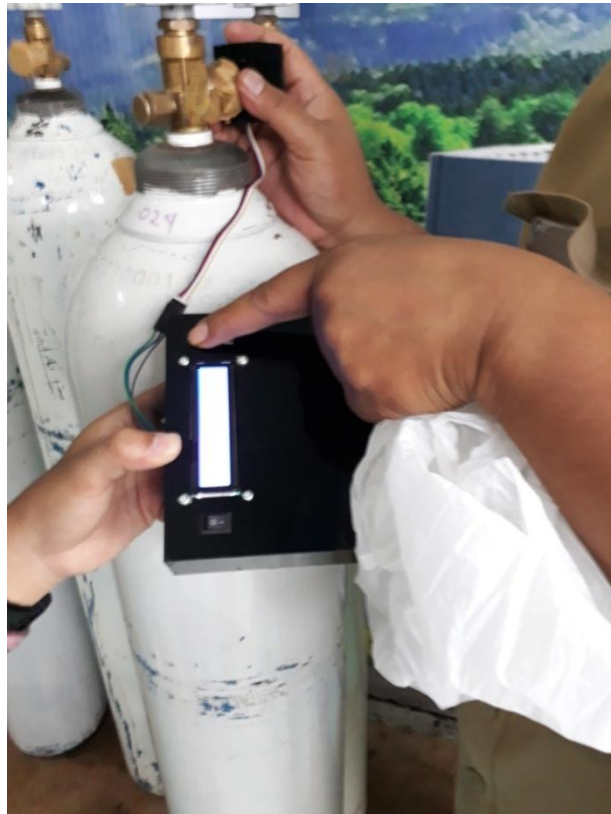


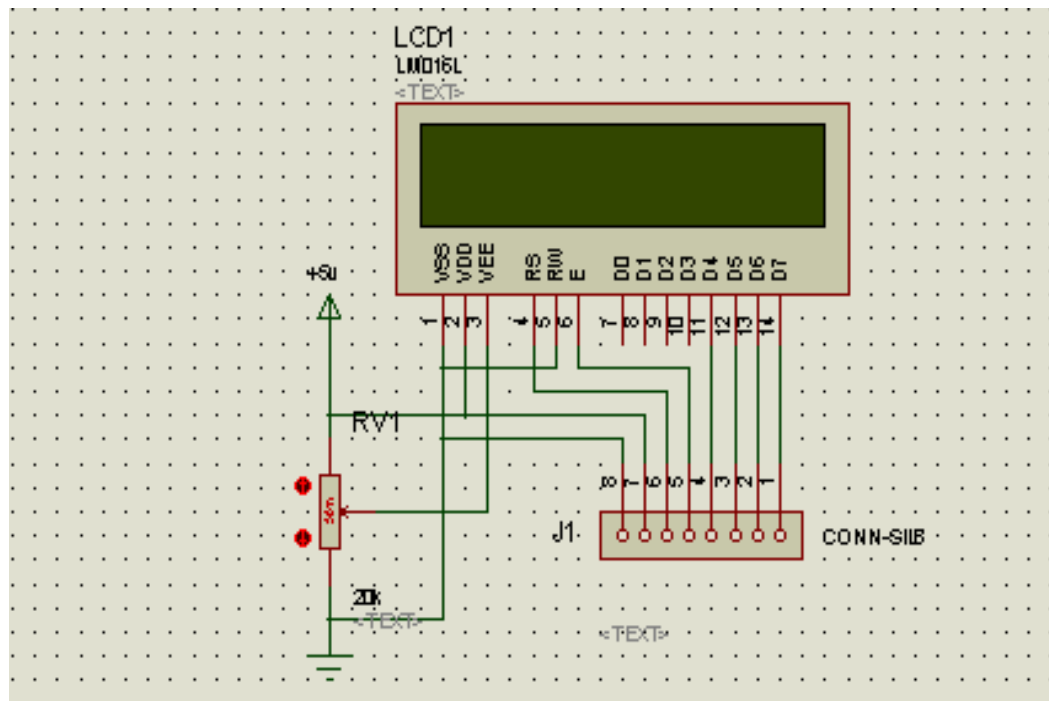
LAMPIRAN

Proses pengambilan data pada sentral gas medis di RSUD Wonosari.

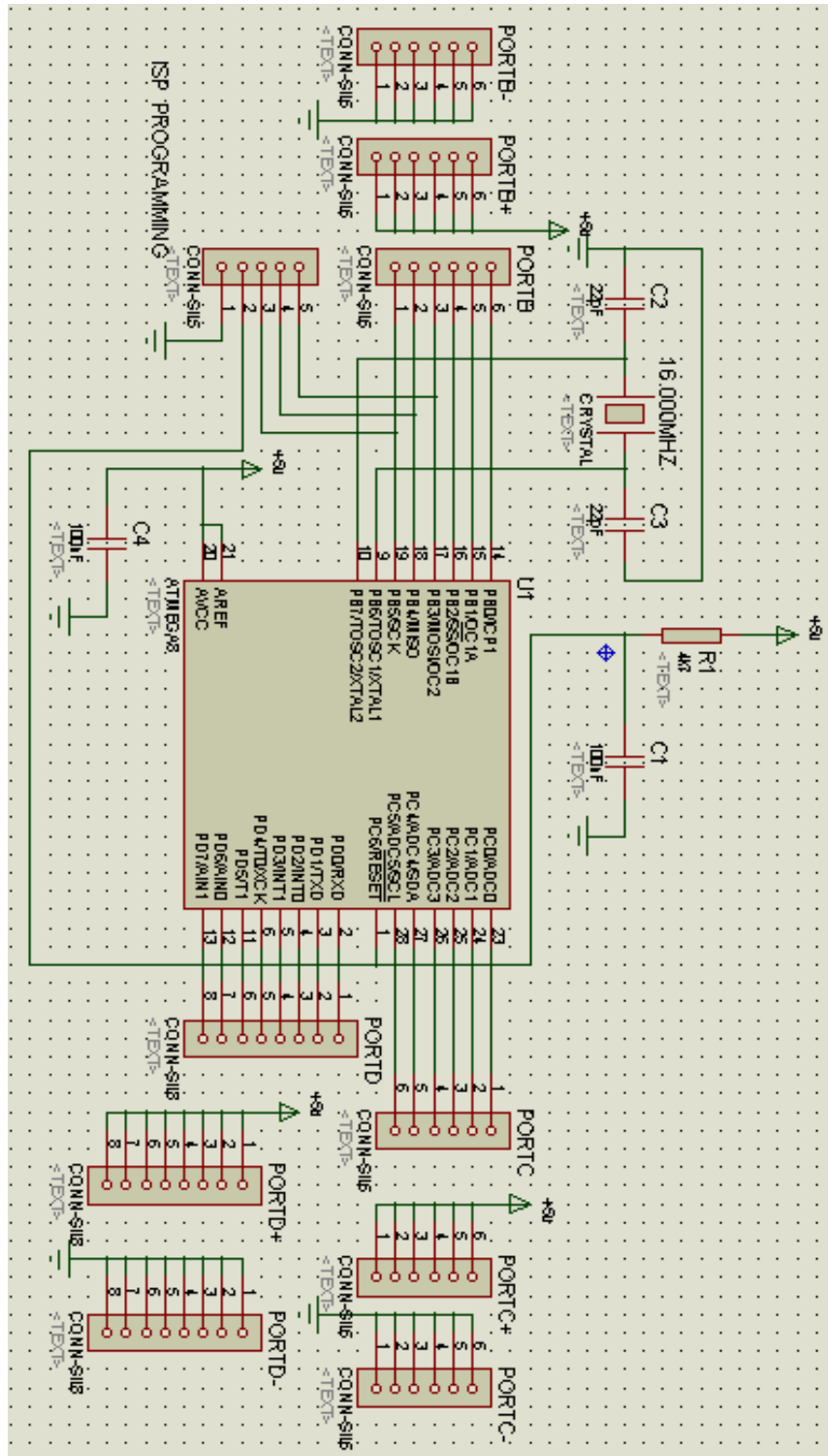




Skematik rangkaian LCD



Skematik rangkaian Minsis



Listing Program

```

/*****
Chip type           : ATMEGA8
Program type       : Application
AVR Core Clock frequency: 8,000000 MHz
Memory model      : Small
External RAM size  : 0
Data Stack size   : 256

http://wiki.seeed.cc/Grove-Gas_Sensor-O2/

*****/
#include <mega8.h> //library atau spesifikasi ATMEGA8
#include <stdio.h> //library input output bilangan(sprintf)
#include <delay.h> //library waktu tunda

// memasukkan library / spesifikasi LCD
#include <alcd.h>

#define ADC_VREF_TYPE 0x40 // mengaktifkan ADC pada portC

char buff [33]; //tipe data char dgn variabel (buff) untuk menampilkan
nilai di lcd
//Pengalamatan input output
#define buzzer PORTB.0 //dialamatkan pada portB.0
#define buton  PINC.4 //save data di alamatkan di pinC.4
#define button  PINC.5 //open data di alamatkan di pinC.5

//tipe data yang digunakan beserta variabelnya
eeprom float save0,save1,save2,save3,save4;
eeprom int mem=0; //mem berfungsi untuk menggeser save1 ke save2
float VRefer = 5.0; //tipe data float dengan variabel VRefer berfungsi
mengatur nilai tegangan referensi yaitu 5v
int pinAdc = 0; //tipe data int dgn variabel pinAdc brfungsi
mengalamatkan pinAdc di adc0 atau kaki 23 portC
int vsensor = 1; //tipe data int dgn variabel vsensor brfungsi
mengalamatkan vsensor di adc1 atau kaki 24 portC
int menu; //membaca menu selanjutnya pada tombol save data
int mems; //int=tipe data yg melihat simpan data selanjutnya

//settingan adc 10bit, 10bit karena memiliki resolusi yg besar, jadi
lebih teliti dalam pembacaan
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;
}

//fungsi memori
void memori()

```

```

{
  lcd_clear(); //menghapus karakter sebelumnya
  menu=0; //menu diberi nilai 0
  delay_ms(200);
  while(1)
  {
    //OPEN DATA
    if(!buton){delay_ms(500);menu=menu+1;} //jika buton ditekan maka menu
bertambah 1
    if(menu>4){menu=0;break;} //jika menu lebih dari 4 maka menu kembali
ke 0 dan break untuk kembali ke program utama di while pertama

    if(menu==0)//jika menu bernilai 0
    {
      lcd_gotoxy(0,0); //letak karakter di 0,0
      sprintf(buff,"Memory 1:%.01f",save0); //%.01f sebagai penampil
bilangan yg menggunakan koma
      lcd_puts(buff); //lcd menampilkan nilai dari save 0
      lcd_putchar('%'); //menambahkan karakter persen
    }

    if(menu==1) //jika menu bernilai 1
    {
      lcd_gotoxy(0,0); //letak karakter di 0,0
      sprintf(buff,"Memory 2:%.01f",save1);
      lcd_puts(buff); //lcd menampilkan nilai dari save 1
      lcd_putchar('%'); //menambahkan karakter persen
    }

    if(menu==2) //jika menu bernilai 2
    {
      lcd_gotoxy(0,0); //letak karakter di 0,0
      sprintf(buff,"Memory 3:%.01f",save2);
      lcd_puts(buff); //lcd menampilkan nilai dari save 2
      lcd_putchar('%'); //menambahkan karakter persen
    }

    if(menu==3) //jika menu bernilai 3
    {
      lcd_gotoxy(0,0); //letak karakter di 0,0
      sprintf(buff,"Memory 4:%.01f",save3);
      lcd_puts(buff); //lcd menampilkan nilai dari save 3
      lcd_putchar('%'); //menambahkan karakter persen
    }

    if(menu==4) //jika menu bernilai 4
    {
      lcd_gotoxy(0,0); //letak karakter di 0,0
      sprintf(buff,"Memory 5:%.01f",save4);
      lcd_puts(buff); //lcd menampilkan nilai dari save 4
      lcd_putchar('%'); //menambahkan karakter persen
    }
  }

  //berfungsi untuk membaca tegangan baterai
  float baca_volt(){

    float volt,voltout; //menamakan variabel volt dan voltout ke tipe data
float karna float bisa baca koma
    float voltmax=3.5,involtmax=20.0; //variabel volt diberi nilai

```

```

tegangan minimal 3.5 dan variabel involt diberi nilai 20
int data=read_adc(vsensor); //rumus sensor tegangan
//knp bisa 3.5 karna di dpt dari rumus di bawah untuk mncari minimal
tegangan masuk
//r1= 4,7k r2= 1k resistor 1 resistor
//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= 3.5 volt (nilai float)

volt=data*((float)5/1023); //mengubah nilai adc ke tegangan. 5 karena
tegangan max referensi adc
voltout=volt*((float)involtmax/voltmax); //mengubah nilai tegangan kecil
ke tegangan sensor
return voltout; //melanjutkan nilai voltout ke fungsi lain
}

float readO2Vout() //fungsi rumus nilai adc ke volt (program pembacaan
sensor)
{
    float MeasuredVout =0; //variabel yg digunakan untuk menampilkan vout
dri sensor oksigen
    long sum = 0; //untuk mencari nilai rata2 dari adc
    int i; //tipe data variabel bernilai awal 0

    //ambil nilai rata-rata adc
    for(i=0; i<32; i++) //i=0 jika i kurang dari 32 maka i selalu
bertambah
    {
        sum +=read_adc(pinAdc); //menjumlahkan nilai adc dalam 32 kali
pngambilan data
    }
    sum = sum / 32; //membagi seluruh hasil penjumlahan dengan 32 untuk
menemukan rata-rata

    // ubah nilai adc ke volt
    MeasuredVout = sum * (VRefer / 1023.0); //rata-rata adc dikali nilai
adc dibagi 10bit

    return MeasuredVout;
    //melanjutkan nilai MeasuredVout ke fungsi yang lain
}

float readConcentration() //fungsi mengubah volt ke O2
{
    float MeasuredVout = readO2Vout(); //nilai dari MeasuredVout
disamadengankan readO2Vout()

    float Concentration = MeasuredVout * 0.21 / 2.0; //rumus sensor
grove dari data sheet
    float Concentration_Percentage=(Concentration*100)*2.7;

    if(Concentration_Percentage<56){Concentration_Percentage=Concentration_Pe
rcentage/2.2;}
    //jika konsentrasi oksigen kurang dari 56 maka konsentrasi di bagi
2.2

```



```

        if(mem==0){save0=Concentration_Percentage;} //jika mem bernilai 0
maka Concentration_Percentage disamadengankan save 0
        if(mem==1){save1=Concentration_Percentage;} //jika mem bernilai 1
maka Concentration_Percentage disamadengankan save 1
        if(mem==2){save2=Concentration_Percentage;} //jika mem bernilai 2
maka Concentration_Percentage disamadengankan save 2
        if(mem==3){save3=Concentration_Percentage;} //jika mem bernilai 3
maka Concentration_Percentage disamadengankan save 3
        if(mem==4){save4=Concentration_Percentage;} //jika mem bernilai 4
maka Concentration_Percentage disamadengankan save 4
        return Concentration_Percentage; //melanjutkan nilai ke fungsi lain
    }

void main(void)
{
int x=0; //x bernilai 0
PORTB=0x00;
DDRB=0x01;
PORTC=0x30;
DDRC=0x00;
PORTD=0x00;
DDRD=0x00;
TCCR0=0x00;
TCNT0=0x00;
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;
MCUCR=0x00;
TIMSK=0x00;
UCSRB=0x00;
ACSR=0x80;
SFIO=0x00;
ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0x83;
SPCR=0x00;
TWCR=0x00;

lcd_init(16);

lcd_clear(); //menghapus karakter sebelumnya
//Di bawah program tampilan awal ketika dihidupkan
lcd_gotoxy(0,0); //karakter pada posisi 0,0
lcd_putsf("Oxygen Analyzer"); //menampilakn tulisan "Oxygen Analyzer"
delay_ms(3000); //waktu jeda 3000ms

lcd_clear(); //menghapus karakter sebelumnya
lcd_gotoxy(0,0); //karakter pada posisi 0,0
lcd_putsf("Bramudya Arya G"); //menampilakn tulisan "Bramudya Arya G"
lcd_gotoxy(0,1); //karakter pada posisi 0,1
lcd_putsf("20143010055"); //menampilakn tulisan "20143010055"

```

```

delay_ms(2000); //waktu jeda 2000ms

for(x=0; x<5; x++) //perulangan x yg bermula dari 0 sselalu bertambah
sampai 5 dan kembali ke 0 lagi
{
    buzzer=1;
    delay_ms(100); //buzzer nyala selama 100ms

    buzzer=0;
    delay_ms(100); //buzzer mati selama 100ms
}

while (1)
{
    //Place your code here

    if(!button) //jika button ditekan (save data)
    {
        delay_ms(200); //jeda 200ms
        mem=mem+1; //mem bertambah 1
    }

    if(mem>4) //jika mem lebih dari 4
    {
        mem=0; //mem diberi nilai 0
    }

    if(!buton) //jika buton ditekan (open data)
    {
        delay_ms(200); //jeda 200 ms
        memori(); //memanggil fungsi memori di atas
    }
    //program tampilan di lcd
    lcd_clear(); //menghapus karakter sebelumnya
    lcd_gotoxy(0,0); //karakter pada posisi x(kolom)0, y(baris)0 atau
pojok kiri atas
    sprintf(buff,"Oxygen:%.01f",readConcentration()); //perintah
menampilkan letak nilai o2 pada lcd
    lcd_puts(buff);
    lcd_putchar('%'); //menampilkan %(persen)

    lcd_gotoxy(11,1); //save pada posisi kolom ke 11,dan baris kel
sprintf(buff,"Sav:%d ",mems); //perintah menampilkan letak nilai
mems (save data)
    lcd_puts(buff);

    lcd_gotoxy(0,1); //karakter pada posisi 0,1
    sprintf(buff,"Batt:%.01fV ",baca_volt()); //perintah menampilkan
letak voltase battery
    lcd_puts(buff);

    mems=mem+2; //untuk membuat tampilan di lcd mulai dari 1 sampai 5
    if(mems==6){mems=1;} //di save data jika mems bernilai 6 maka mem
diberi nilai 1 perintah penyetaraan untuk kembali ke angka 1

    delay_ms(100); //jeda 100ms
}
}

```