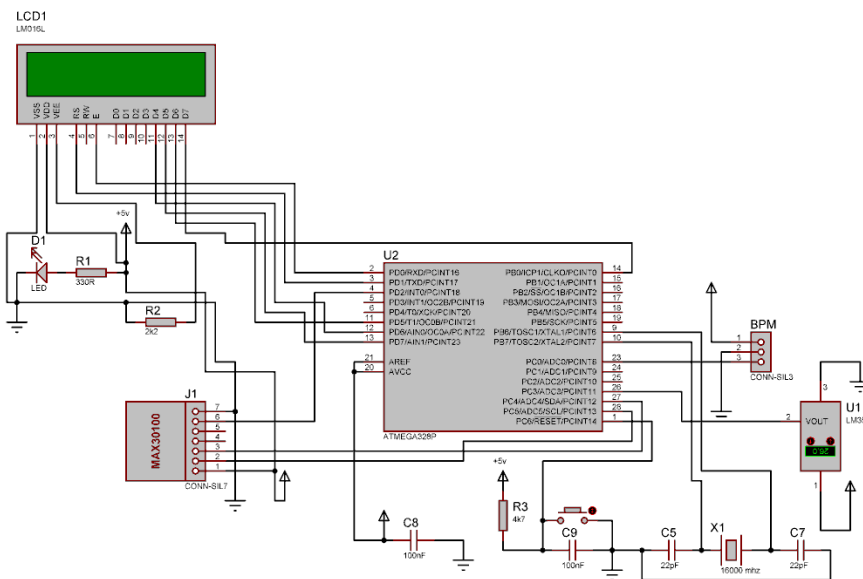


a. Rangkain keseluruhan

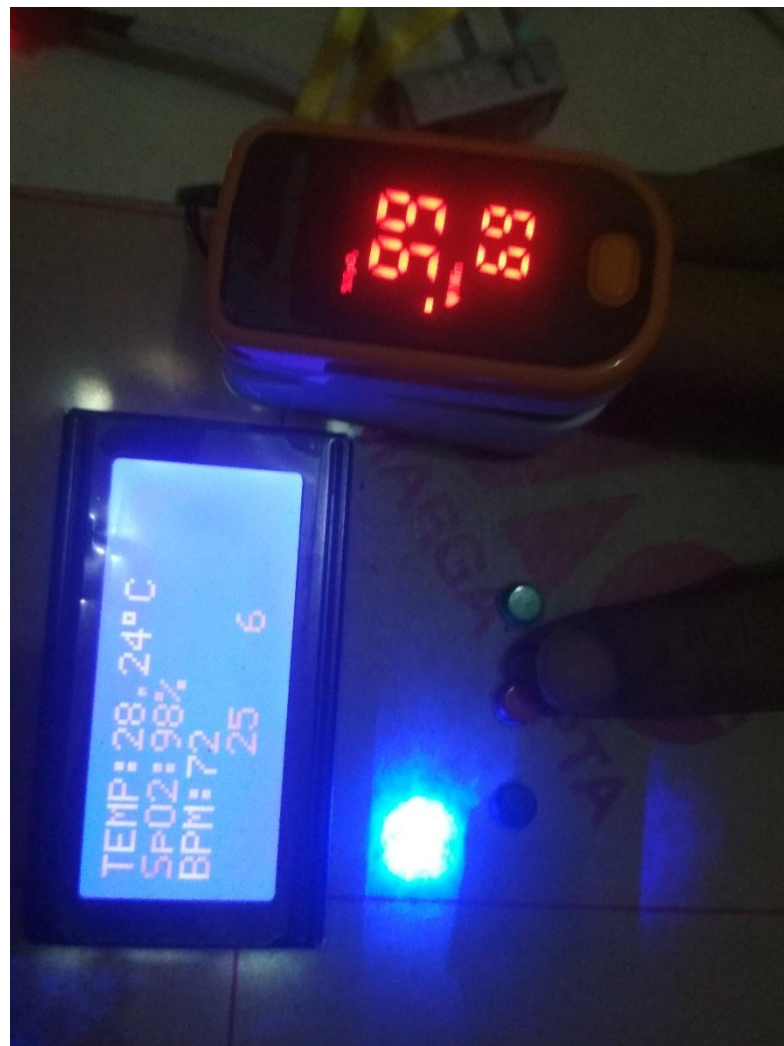


b. Gambar alat



c. Gambar pengambilan data





Dari rangkaian *high pass filter* pasif 1 kemudian diteruskan ke rangkaian penguat 1 *non inverting* untuk menguatkan tegangan dari *output* sensor, besar penguatan dapat dihitung menggunakan rumus sebagai berikut:

$$\begin{aligned}
 \text{Penguatan 1} &= 1 + \frac{R_f}{R_{in}} \\
 &= 1 + \frac{680K}{6K8} \\
 &= 101 \text{ kali.}
 \end{aligned}$$

D. LISTING PROGRAM

Program fungsi celcius

```

void celcius()
{

    int dataadc2 ;//definisi variable dataadc2 ke tipe data int
    long sum = 0; //definisi variable sum ke tipe data long
    int i; //definisi variable i ke tipe data int

    for (i = 0; i < 100; i++)
    {
        sum += analogRead(sensorlm35);
    }
    dataadc2 = sum / 100;
    //mengambil rata rata dari sensorlm35 dari pengambilan data sebanyak 100x

    volt = (dataadc2 * (5.0 / 1023.0)-0.03);
    //mengubah dataadc2 menjadi tegangan

    suhu = volt * 100;
    //mengubah tegangan menjadi suhu
}

```

Program fungsi bpm

```

void bpm()
{
    int dataadc1 ; //definisi variable dataadc1 ke tipe data int
    long sum = 0; //definisi variable sum ke tipe data long
    int i; //definisi variable i ke tipe data int
    for (i = 0; i < 30; i++)

```

```

{
  sum += analogRead(sensorbpm);
}
dataadc1 = sum / 30;
//mengambil rata rata dari sensorbpm dari pengambilan data sebanyak 30x

if(dataadc1>80 && tanda1==1) //jika data adc1>80 dan tanda1 bernilai 1
{
  digitalWrite(buzz,HIGH); //buzzer hidup
  counter1++; //counter1 mulai menghitung
  tanda1=0; //tanda1 diberi nilai 0
  timeout1=0; //timeout1 diberi nilai 0
}
if(dataadc1<80) //jika data adc1<80
{
  digitalWrite(buzz,LOW); //buzzer mati
  timeout1++; //timeout1 mulai menghitung
}

if(timeout1>20){tanda1=1;} //jika timeout lebih dari 20 maka tanda1 diberi nilai
1
if(timeout1<20){tanda1=0;} //jika timeout kurang dari 20 maka tanda1 diberi
nilai 0
counterbpm=(counter1*3); //counterbpm adalah hasil dari counter1 dikali 3

lcd.setCursor(10,3);
lcd.print(counter1);
//menampilkan counter1
}

```

Program fungsi spo

```

void spo()
{

    pox.update();
    if (millis() - tsLastReport > REPORTING_PERIOD_MS)
    {
        if(pox.getSpO2()<100)
        {
            lcd.setCursor(0,1);
            lcd.print("SpO2:");
            lcd.print(pox.getSpO2()+4);
            lcd.print("%");
            if(pox.getSpO2()<20)
            {
                lcd.setCursor(0,1);
                lcd.print("SpO2:0 %");
            }
            tsLastReport = millis();}}}

```

Program fungsi display/Tampilan

```

void display()
{
    lcd.clear(); //menghapus karakter lcd sebelumnya
    lcd.setCursor(0,0);
    lcd.print("TEMP:");
    lcd.print(suhu); // menampilkan suhu
    lcd.print((char)223); // symbol derajat
    lcd.print("C"); // C karakter
    lcd.setCursor(0,1);
    lcd.print("SpO2:");
    lcd.print(pox.getSpO2()); //menampilkan spo2

```

```

lcd.print("%"); //menampilkan persen
lcd.setCursor(0,2);
lcd.print("BPM:");
lcd.print(counterbpm); //menampilkan counterbpm

delay(1000); //jeda 1000ms
counter1=0; //counter1 diberi nilai 0
}

```

Program deklarasi variable dan library

```

#include <Time.h>
#include <TimeLib.h>
//library timer
#include <Wire.h>
#include <LiquidCrystal.h>
LiquidCrystal lcd(8, 7, 6, 5, 1, 0);
#define USE_ARDUINO_INTERRUPTS true
#include <PulseSensorPlayground.h>
#include "MAX30100_PulseOximeter.h"
#define REPORTING_PERIOD_MS 1000
PulseSensorPlayground pulseSensor;
PulseOximeter pox;

uint32_t tsLastReport = 0;

#define buzz 4 //mendefinisikan buzz pada pin 4
#define sensorbpm A0 //mendefinisikan sensorbpm pada pin A0
#define sensorlm35 A3 //mendefinisikan sensorbpm pada pin A3

int tanda1, counter1, timeout1, counterbpm;

```

```
//variabel pembacaan bpm dengan tipe data int  
float volt,suhu;  
//variabel pembacaan suhu dengan tipe data float
```