Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

ANAESTHESIA CONTROL SYSTEM BASED ON MULTISENSORAL SYSTEM USING ARDUINO UNO

Project Report

SUBMITTED TO

Dr. KANIMOZHI.G

BY

MANGOLIK KUNDU – 19BEE1101

SOUVIK DATTA – 19BEE1213

TRITOY MOHANTY – 19BEE1007

SCHOOL OF ELECTRICAL ENGINEERING
VELLORE INSTITUTE OF TECHNOLOGY
CHENNAI, TAMILNADU-600127

April, 2020

TABLE OF CONTENTS

LIST OF FIGURES

1. ACKNOWLEDGEMENT

We would like to thank Dr. Kanimozhi.G Ma'am for her help, suggestions and constant guidance that we have received over the course of development of the project. The various knowledge that we have gained from our theory courses as well as our lab courses have also helped develop a clear idea on the course of development of the project. Dr. Kanimozhi.G Ma'am's methodology of teaching and imparting knowledge on the course Bio-medical Instrumentation (EEE-1008) have really helped us gain an idea on the topic of the project and its understanding and has been an integral part of the whole project. Last but not the least it was a great learning experience doing this project as a team.

2. INTRODUCTION

General anaesthesia is an anaesthetic used to induce unconsciousness during surgery. The medicine is either inhaled through a breathing mask or tube, or given through an intravenous (IV) line. In the developed world, the most frequent type in use is the continuous-flow anaesthetic machine, which is designed to provide an accurate supply of medical gases mixed with an accurate concentration of anaesthetic vapour, and to deliver this continuously to the patient at a safe pressure and flow.

The original concept of continuous-flow anaesthesia machines was popularized by Boyle's machines, invented by the British anaesthetist Henry Boyle at St. Bartholomew's Hospital, UK in 1917. Prior to this time, anesthesiologists often carried all their equipment's with them, but the development of heavy, bulky cylinder storage and increasingly elaborate airway equipment meant that this was no longer practical for most circumstances.

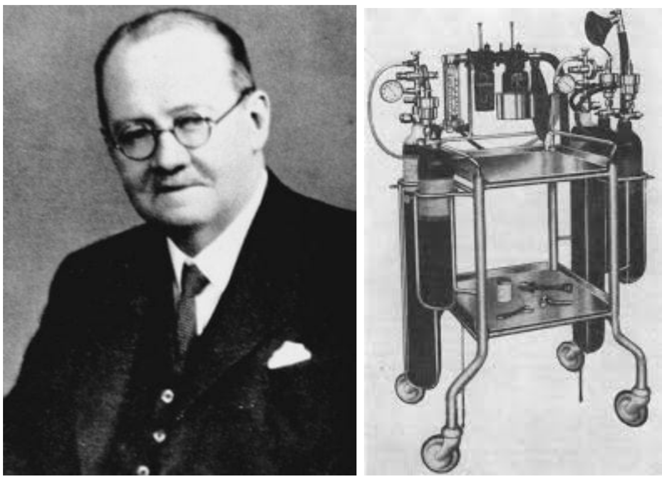

Fig 2.1 Boyle's Anaesthesia Machine

Anaesthesia equipment problems may and have previously contributed to anaesthetic morbidity and mortality.
The administration of high/low dose of anaesthesia during surgery may cause lethal effect to the patient. To avoid such situation, the anaesthetist administers few millilitres of anaesthesia at regular intervals to the patient. But this comes with its own share of

problems, according to the **American Research Society,** "about one quarter of anaesthesia related problems were considered to be related to human error on the part of the users".

So, there was a need to develop an **automatic anaesthesia control system** to minimize the problems or errors faced. We wanted to analyse the frequency, type and severity of equipment related problems and come out with a solution to improve safety standards. ***The primary use of the invention is to propose an improved automatic anaesthesia delivery system which overcomes disadvantages of the manual one.***

3. OBJECTIVES

For any operation, the patient being in an anaesthetic condition is a must. This way the patient won't feel any discomfort or pain during the medical procedure. When the dosage of anaesthesia isn't properly monitored it causes serious health hazards.

- We strive to build a device which will help surgeons in effortlessly administering the dosage of anaesthesia to a patient, as per requirement, by studying its physiological signals.
- During surgery, a patient's condition is primarily influenced by its *blood pressure, temperature, oxygen and pulse activity*. The microcontroller used in this device is designed to automatically analyse these physiological signals and inject the accurate dose of anaesthesia accordingly.

The following parameters dictate how anaesthesia affects body performance –

1. **Heart Rate** - The induction of anaesthesia with propofol is often associated with a significant decrease in arterial blood pressure and heart rate (HR).
   Optimal Level during surgery - 75.2 ± 13.8

2. **Blood Pressure** - On the application of anaesthesia, there is a decrease in blood pressure.
   Optimal Level during surgery - 160/100 > BP > 130/80 mm of Hg

3. **Temperature** - Anaesthesia should be induced only after core body temperature is more than 36°C. The major cause of hypothermia in most patients given general anaesthesia is an internal core-to-peripheral redistribution of body heat that usually reduces core temperature by 0.5°–1.5°C in the first 30 min after induction of anaesthesia.
   Hyperthermia - T > 40 °C
   Optimal Level during surgery - 40 > T > 36 °C

4. **Oxygen Level** - General anaesthesia and mechanical ventilation impair pulmonary function, even in normal individuals, and result in decreased oxygenation in the postanaesthetic period. At 94% administer high flow oxygen.
   Optimal Level during surgery - 100% > Oxy. > 90%.

4. ADVANTAGES OF AUTOMATIC MACHINE OVER MANUAL ANAESTHESIA ADMINISTRATION

A study conducted by **D.N. Thrush**, Department of Anaesthesiology, University of South Florida, College of Medicine -

Objective: To determine whether data recorded by an automatic information management system is significantly better from that recorded manually.

Design: A comparison was made between 13 handwritten and 13 computer-generated anaesthesia records by calculating the frequency with which recorded variables were outside predetermined acceptable ranges. Five physiologic variables (systolic blood pressure (SBP), diastolic blood pressure (DBP), heart rate (HR), end-tidal partial pressure of carbon dioxide ($P_{ET}CO_2$), and oxygen saturation by pulse oximeter (SpO_2) were compared during the initial 112 hours of operation.

Patients: Thirteen adult patients scheduled for operations that required general anaesthesia for longer than 112 hours.

Intervention: In addition to the traditional handwritten anaesthesia records, an information management system (ARKIVE Patient Management System, DIATEK, San Diego, CA) was used to collect data from each case.

Results: No significant differences were found between the methods in the frequency of elevated SBP, elevated DBP, and tachycardia.
However, the manual records showed *"low SBP, DBP, and HR with a significantly lower frequency (2%, 11 %, 1%, respectively) than the automated records (6%, 26%, 5%, respectively; $p < 0.01$)".*
The automated $P_{ET}CO_2$ readings were higher than the upper limit (40 mmHg) with a higher frequency (18%) than the manual records (3%; $p < 0.01$).

Conclusions: Observer bias, missed readings, and errors of memory, which affect manual anesthetic records, may cause significant inaccuracy and may be avoided by using automated records generated by information management systems.

Reference Link - https://pubmed.ncbi.nlm.nih.gov/1389193/

5. BLOCK DIAGRAM OF THE PROJECT (BLOCK DIAGRAM) :

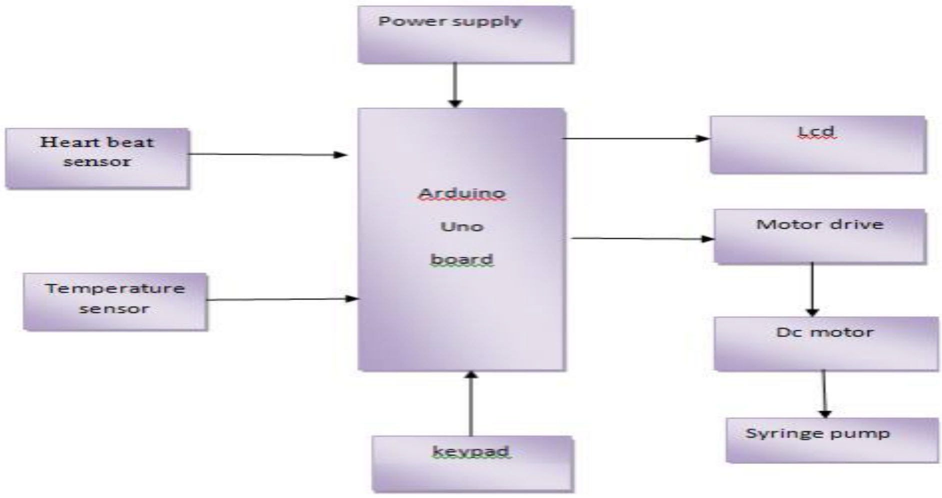

Fig. 5.1 Block diagram depicting operation of the device

5. CIRCUIT DIAGRAM OF THE PROJECT

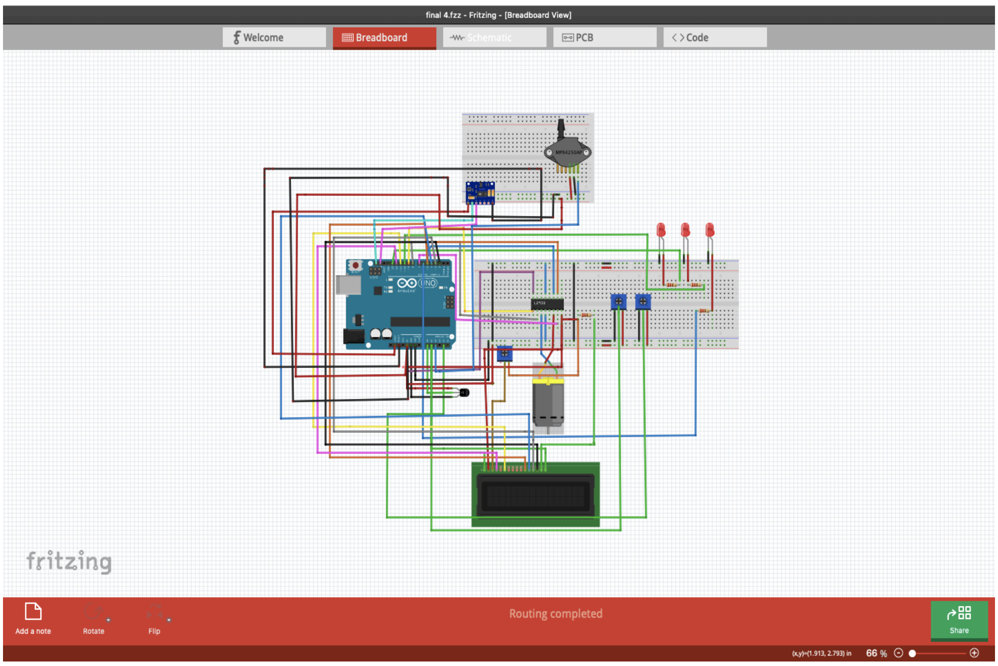

Fig. 6.1 Circuit Diagram using Fritzing

7. HARDWARE COMPONENTS USED IN THE PROJECT :

COMPONENTS USED	QUANTITY
Arduino UNO	1
100K Potentiometer	1
10K Potentiometer	2
Resistors	4
Heart Rate and SpO_2 Sensor-MAX30100	1
LCD Display-16x2	1
Adafruit 0.96''OLED Display	1
L293D - Motor Driver	1
DC Motor	1
Temperature Sensor - LM35	1
LEDs	2
HC-05 Bluetooth Module	1
Jumper Wires	-

8. FIGURE OF HARDWARE COMPONENTS USED (FIGURE-3):

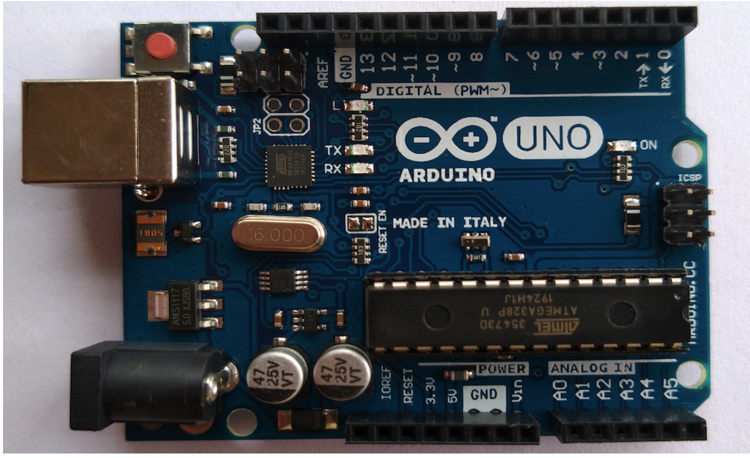

Fig 8.1 Arduino UNO

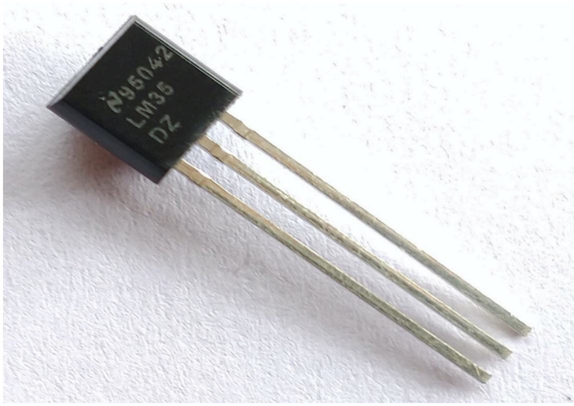

Fig 8.2 LM-35 Temperature Sensor

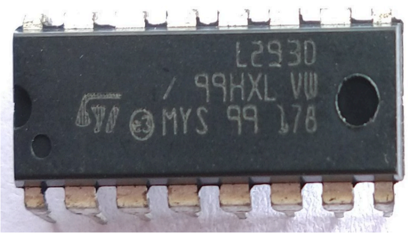

Fig. 8.3 L293D Motor Driver

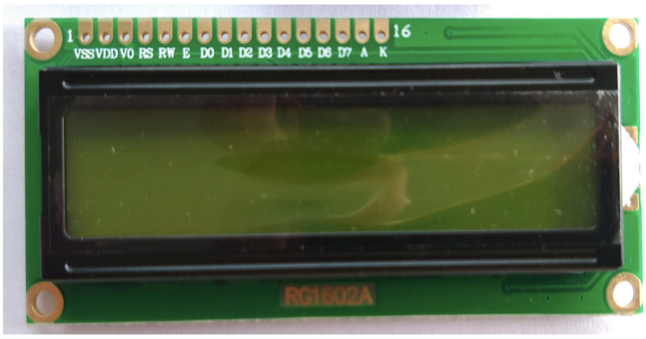

Fig. 8.4 LCD Display 16X2

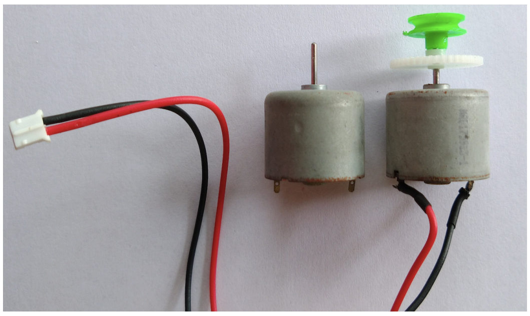

Fig. 8.5 DC Motor

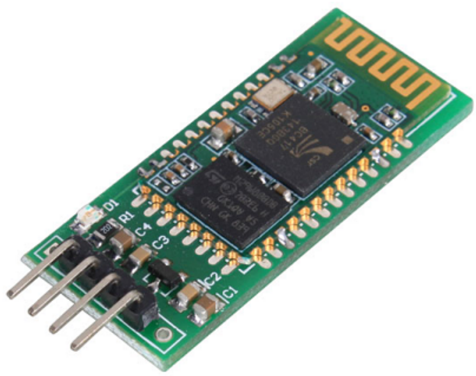

Fig 8.6 HC-05 Bluetooth Module

Fig. 8.7 MAX30100 Heart Rate and Pulse Oximeter

9. IMAGE OF THE HARDWARE –

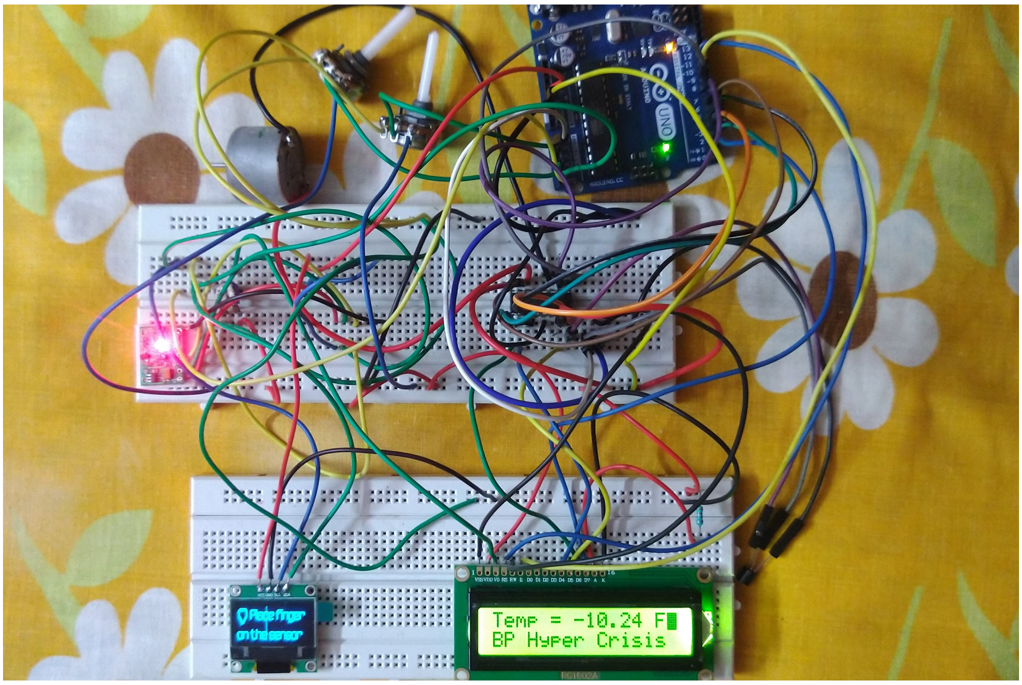

Fig. 9.1 An image of the hardware over the course of development of the project

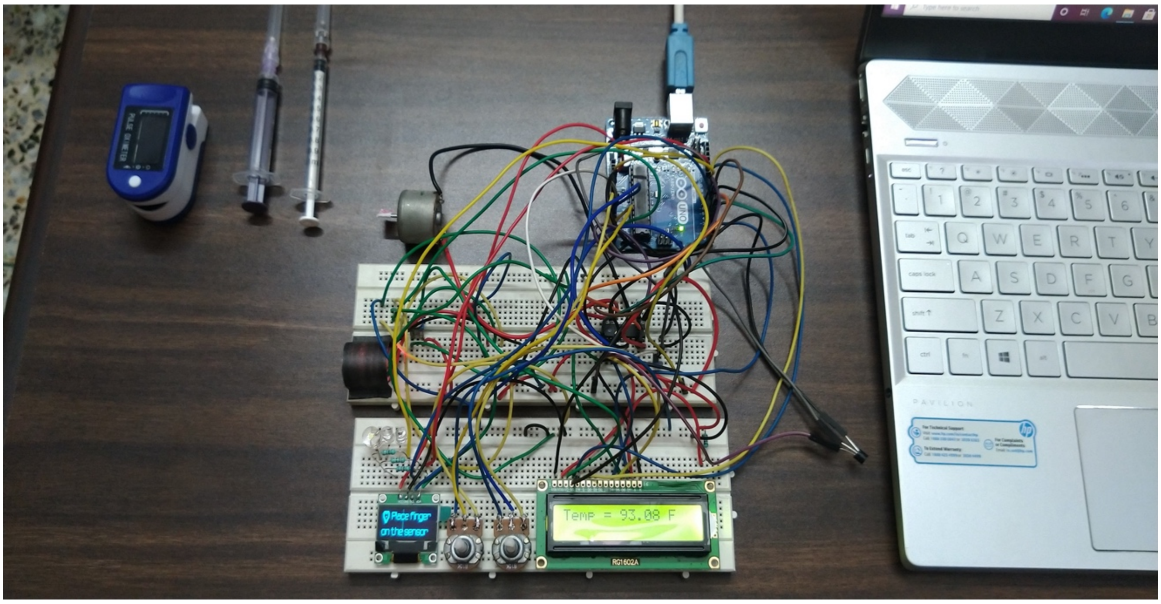

Fig. 9.2 Final Hardware with all the sensors, microcontroller and syringe

10. CODE USED FOR OUR PROJECT:

<u>CODE FOR BLOOD PRESSURE AND TEMPERATURE MEASUREMENT ALONG WITH HC-05 BLUETOOTH MODULE</u>

```
#include <SoftwareSerial.h>

SoftwareSerial BTserial(0, 1); // RX | TX

#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

const int tmp = A0;
const int p = 8;
int enA = 9;
int in1 = 8;
int in2 = 7;

int value = 0;

const int ledPin = 13;//led is connected to digital pin 13
const int ledPin1 = 10;
const int ledPin2 = 6;
int pMin = 0;
int pMax = 1023;
int sen1=0;
int sen2=0;



void setup() {

BTserial.begin(9600);
lcd.begin(16, 2);
     //Serial.begin(9600);
     pinMode(11, OUTPUT);
     pinMode(p, OUTPUT);
    // put your setup code here, to run once:
    // Set all the motor control pins to outputs
    pinMode(enA, OUTPUT);
    pinMode(in1, OUTPUT);
    pinMode(in2, OUTPUT);

    // Turn off motors - Initial state
    digitalWrite(in1, LOW);
    digitalWrite(in2, LOW);

    pinMode(A3,INPUT);//initialize sensor1 as input
```

```
    pinMode(A1 ,INPUT);
    pinMode(ledPin,OUTPUT);//initialize led as output
    pinMode(ledPin1,OUTPUT);
    pinMode(ledPin2,OUTPUT);}

void loop() {

   digitalWrite(p,LOW);
  int Temp = analogRead(tmp);
  float volts = (Temp / 965.0) * 5;
  float c = (volts - .5) * 100;
  float f = (c * 9 / 5 + 32)-20;
  lcd.setCursor(0, 0);
  lcd.print("Temp = ");
  lcd.print(f);
  lcd.print(" F");
  delay(1000);
  lcd.clear();

  BTserial.print(f);

BTserial.print(",");

BTserial.print(sen1);

BTserial.print(",");

BTserial.print(sen2);

BTserial.print(";");

//message to the receiving device

delay(1000);
  if(f > 96.8 && f <= 104)
  {
  motor_run();
  }
  else
  {
    // Now turn off motors
    digitalWrite(in1, LOW);
    digitalWrite(in2, LOW);
  }

sen1 = analogRead(A3);
 sen2 = analogRead(A1);
 sen1= map(sen1, pMin, pMax, 0,250);
 sen2= map(sen2, pMin, pMax, 0,250);
```

```
  lcd.setCursor(0, 0);
  lcd.print("Sys=");
  lcd.print(sen1);
  lcd.setCursor(9, 0);
  lcd.print("Dia=");
  lcd.print(sen2);
   delay(1000);

 if ((sen1 > 130 && sen1 < 160) && (sen2 > 80 && sen2 <
100))
  {
  motor_run();
  }
  else
  {
    // Now turn off motors
    digitalWrite(in1, LOW);
    digitalWrite(in2, LOW);
  }

    if((sen1>130 && sen1<=135)&&(sen2<100 && sen2>=50))//25
psi to 35 psi is normal pressure
   {

  lcd.setCursor(0,1);
  lcd.print("BP Pre-hyper");
  delay(1500);
  digitalWrite(ledPin2, HIGH);
  delay(1000);

   }
      if((sen1<=130 && sen1>=110)&&(sen2<90 && sen2>=50))
   {

  lcd.setCursor(0,1);
  lcd.print("BP Normal");

  delay(1500);
  digitalWrite(ledPin2, HIGH);
  delay(1000);
  }
     if((sen1>135 && sen1<145) && (sen2>=50 && sen2<110))
   {

  lcd.setCursor(0,1);
  lcd.print("BP S1 Hyper");

  delay(1500);
  digitalWrite(ledPin, HIGH);
```

```
  delay(1000);
  }

  if((sen1<80 && sen2<65)||(sen2<=49))
   {
     lcd.setCursor(0,1);
  lcd.print("BP Hypo");

  delay(1500);
  digitalWrite(ledPin1, HIGH);
  delay(1000);
        }
  if((sen1>=145 && sen1<180) && (sen2>=60 && sen2<120))
      {
         lcd.setCursor(0,1);
  lcd.print("BP S2 Hyper");

  delay(1500);
  digitalWrite(ledPin, HIGH);
  delay(1000);
        }
  if(sen1>=180 || sen2>=120)
      {
         lcd.setCursor(0,1);
  lcd.print("BP Hyper Crisis");
    delay(1500);
  digitalWrite(ledPin, HIGH);
  delay(1000);
        }

  else
  {

   lcd.clear();
   digitalWrite(ledPin, LOW);
  }
  digitalWrite(ledPin, LOW);
  digitalWrite(ledPin1, LOW);
  digitalWrite(ledPin2, LOW);

//IMPORTANT: The complete String has to be of the Form:
1234,1234,1234,1234;

//(every Value has to be seperated through a comma (',') and
the message has to

//end with a semikolon (';'))

}
void motor_run()
```

```
{

  // For PWM maximum possible values are 0 to 255
  analogWrite(enA, 255);

  // Turn on motors
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
}
```

CODE FOR MAX30100 PULSE OXIMETER AND OLED DISPLAY USING I2C CONNECTION

```
#include "MAX30100_PulseOximeter.h"
#include <U8g2lib.h>
#include <Wire.h>

#define REPORTING_PERIOD_MS     500
U8G2_SSD1306_128X32_UNIVISION_F_HW_I2C u8g2(U8G2_R0);

// PulseOximeter is the higher level interface to the sensor
// it offers:
//  * beat detection reporting
//  * heart rate calculation
//  * SpO2 (oxidation level) calculation
PulseOximeter pox;

const int numReadings=10;
float filterweight=0.5;
uint32_t tsLastReport = 0;
uint32_t last_beat=0;
int readIndex=0;
int average_beat=0;
int average_SpO2=0;
bool calculation_complete=false;
bool calculating=false;
bool initialized=false;
byte beat=0;
int enA = 9;
int in1 = 8;
int in2 = 7;

// Callback (registered below) fired when a pulse is
detected
void onBeatDetected()
{
  show_beat();
```

```
  last_beat=millis();
}

void show_beat()
{
  u8g2.setFont(u8g2_font_cursor_tf);
  u8g2.setCursor(8,10);
  if (beat==0) {
    u8g2.print("_");
    beat=1;
  }
  else
  {
    u8g2.print("^");
    beat=0;
  }
  u8g2.sendBuffer();
}

void initial_display()
{
  if (not initialized)
  {
    u8g2.clearBuffer();
    show_beat();
    u8g2.setCursor(24,12);
    u8g2.setFont(u8g2_font_smart_patrol_nbp_tf);
    u8g2.print("Place finger");
    u8g2.setCursor(0,30);
    u8g2.print("on the sensor");
    u8g2.sendBuffer();
    initialized=true;
  }
}

void display_calculating(int j)
{
  if (not calculating) {
    u8g2.clearBuffer();
    calculating=true;
    initialized=false;
  }
  show_beat();
  u8g2.setCursor(24,12);
  u8g2.setFont(u8g2_font_smart_patrol_nbp_tf);
  u8g2.print("Measuring");
  u8g2.setCursor(0,30);
  for (int i=0;i<=j;i++) {
    u8g2.print(". ");
  }
```

```
  u8g2.sendBuffer();
}
void display_values()
{
  u8g2.clearBuffer();
  u8g2.setFont(u8g2_font_smart_patrol_nbp_tf);

  u8g2.setCursor(65,12);
  u8g2.print(average_beat);
  u8g2.print(" Bpm");
  u8g2.setCursor(0,30);
  u8g2.print("SpO2 ");
  u8g2.setCursor(65,30);
  u8g2.print(average_SpO2);
  u8g2.print("%");
  u8g2.sendBuffer();

  if ((average_beat > 61 && average_beat < 89) &&
(average_SpO2 >=90 && average_SpO2 <=100 ))
  {
  motor_run();
  }
  else
  {
    // Now turn off motors
    digitalWrite(in1, LOW);
    digitalWrite(in2, LOW);
  }
}

void calculate_average(int beat, int SpO2)
{
  if (readIndex==numReadings) {
    calculation_complete=true;
    calculating=false;
    initialized=false;
    readIndex=0;
    display_values();
  }

  if (not calculation_complete and beat>30 and beat<220 and
SpO2>50) {
    average_beat = filterweight * (beat) + (1 - filterweight
) * average_beat;
    average_SpO2 = filterweight * (SpO2) + (1 - filterweight
) * average_SpO2;
    readIndex++;
    display_calculating(readIndex);
  }
}
```

```
void setup()
{
    //Serial.begin(9600);
    u8g2.begin();
    pox.begin();
    pox.setOnBeatDetectedCallback(onBeatDetected);
    initial_display();
}

void loop()
{
    // Make sure to call update as fast as possible
    pox.update();
    if ((millis() - tsLastReport > REPORTING_PERIOD_MS) and
(not calculation_complete)) {
        calculate_average(pox.getHeartRate(),pox.getSpO2());
        tsLastReport = millis();
    }
    if ((millis()-last_beat>2000)) {
      calculation_complete=false;
      average_beat=0;
      average_SpO2=0;
      initial_display();
    }
}

void motor_run()
{

  // For PWM maximum possible values are 0 to 255
  analogWrite(enA, 255);

  // Turn on motors
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
}
```

11. HARDWARE RESULTS AND DISCUSSIONS:

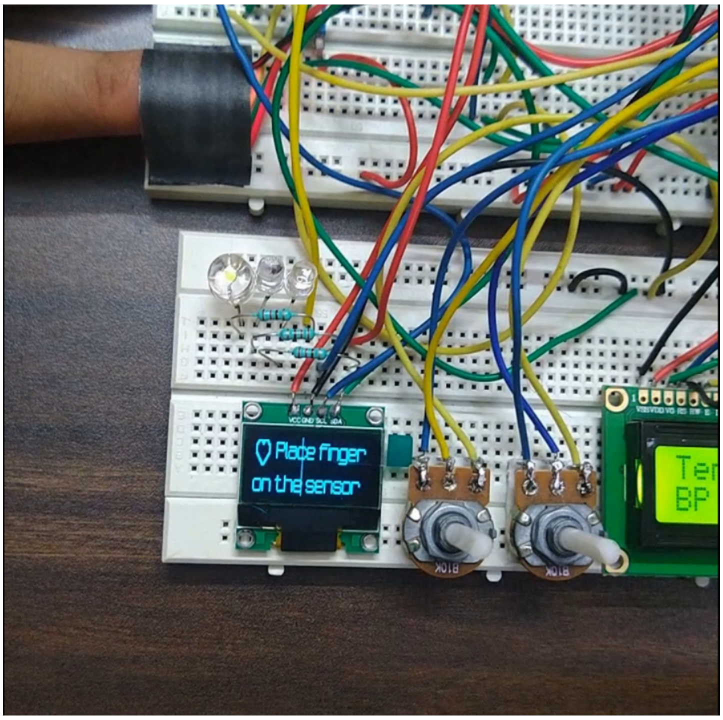

Fig. 11.1 Reading data from pulse oximeter MAX30100

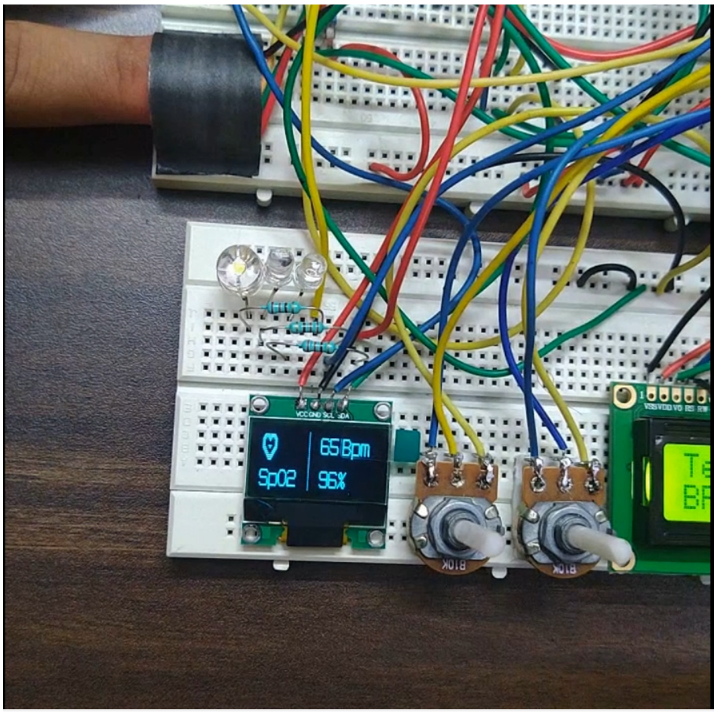

Fig. 11.2 Displaying heart rate and blood oxygen level on OLED Display

Fig. 11.3 Verifying the heart rate and oxygen level with a readymade pulse oximeter

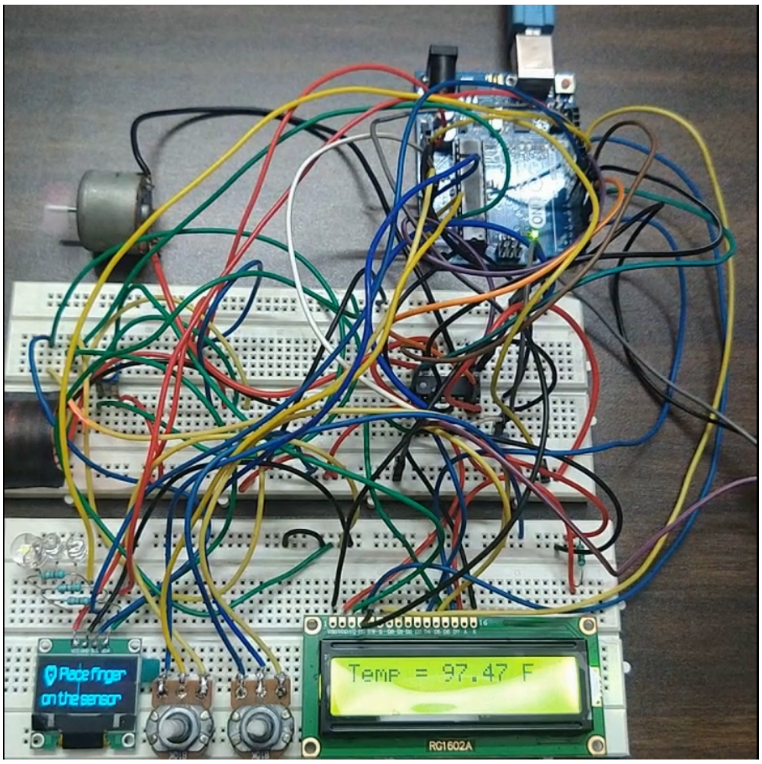

Fig. 11.4 Displaying Temperature on LCD Display

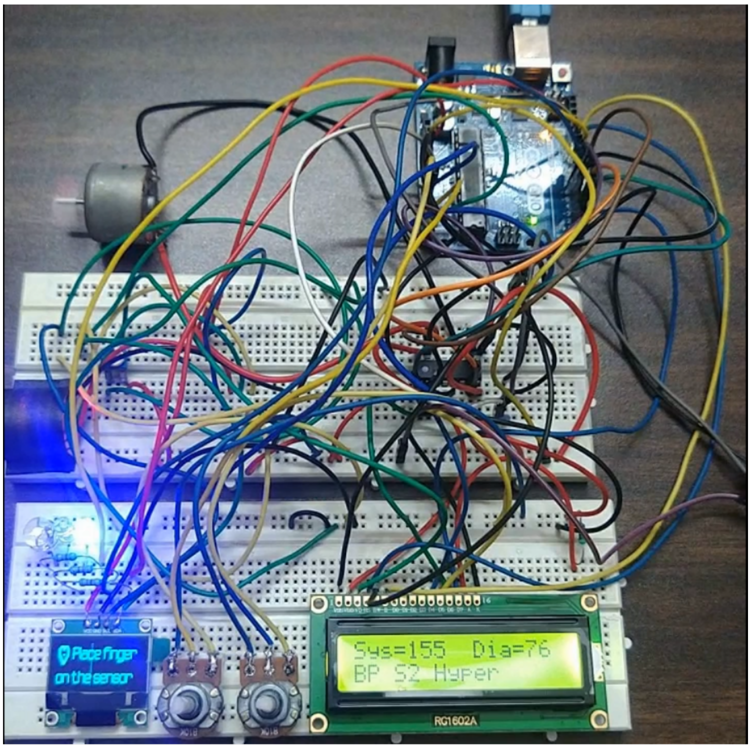

Fig. 11.5 Displaying Blood Pressure values on LCD Display

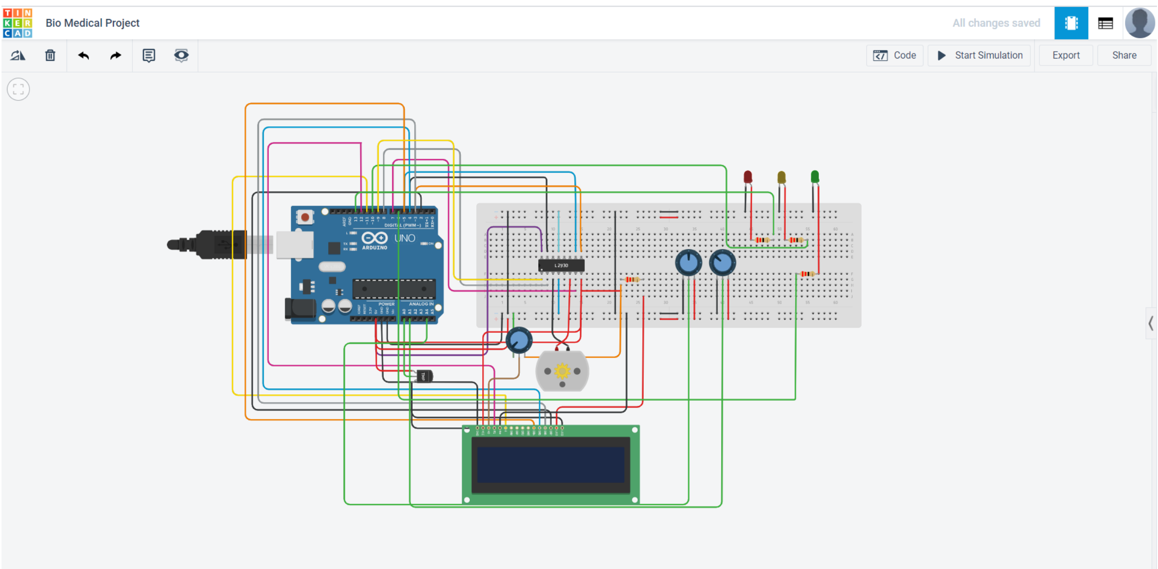

Fig. 11.6 Circuit simulation on Tinkercad - I

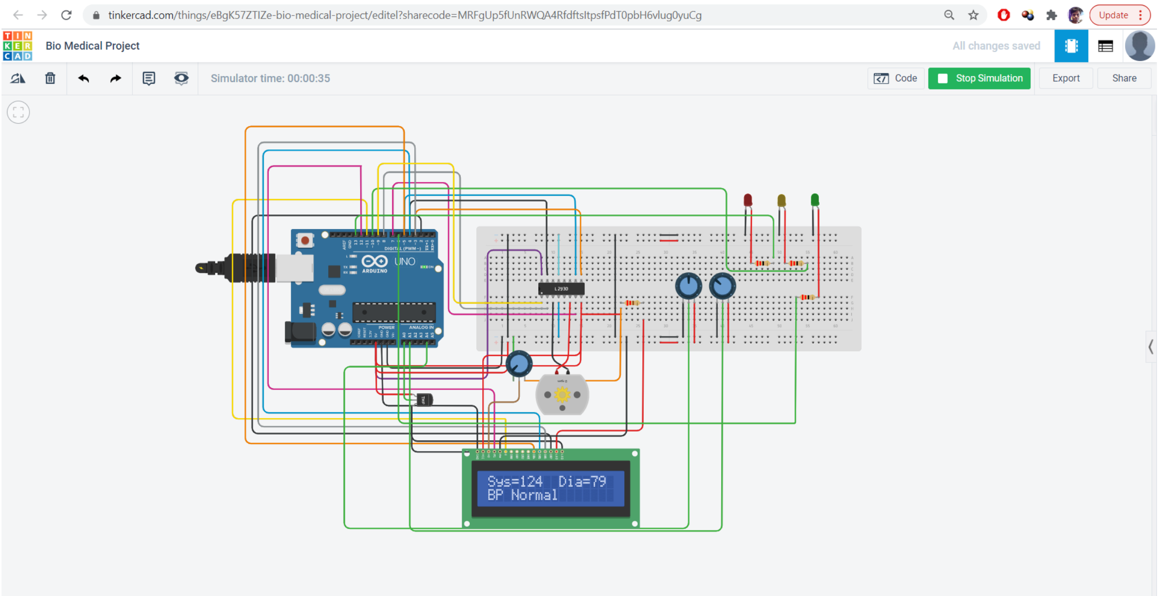

Fig. 11.7 Circuit simulation on Tinkercad - II

(NOTE: The Tinkercad circuit simulation doesn't include the MAX30100 pulse oximeter due to unavailability of component on the software.)

Link for Video: https://drive.google.com/file/d/1-R_k_hb-HyKKpbYtPq-VnVPQSHRS7w0X/view

Link for Circuit Simulation: https://www.tinkercad.com/things/eBgK57ZTIZe-bio-medical-project/editel?sharecode=MRFgUp5fUnRWQA4RfdftsItpsfPdT0pbH6vlug0yuCg

20. CONCLUSION:

On running the circuit, we can successfully control a motor that can be in turn used to administer anaesthesia in the patient's body using 4 physiological factors including body temperature, blood pressure, heart rate and blood oxygen level of the patient. The device can show the real-time data on two screens and also work as an IOT based device with the help of a HC-05 Bluetooth module or an ESP-8266 Wi-Fi module.

This project has the potential to find a future scope into a large field and is a project of significant importance, as this integrated device can prevent human errors and at the same time can be over-ridden by anaesthesiologists in case of an emergency or a critical condition.

As this is our first Arduino based project, thus in the development of the hardware as well as the software simulation part of the project we have learnt working on various components, their working principle, methodology of working and various troubleshooting techniques. The project helped us to grasp the working principle of the MAX30100 pulse oximeter sensor, Arduino UNO, LM35 Temperature sensor, various methods of interfacing the Arduino UNO with a motor, interfacing the Arduino UNO with a L293D motor driver. We have also learned on the two methods of interfacing the LCD Display and the OLED Display with the Arduino UNO – The normal 16 pin interfacing of the lcd display and the I2C connection of the MAX30100 and OLED Display with the Arduino UNO. Overall, it was a great learning experience for all of us.

13. REFERENCE:

[1] Metz S., *Legal implications of automated records*. *APSF Newsletter* 2001;16(4):59.

[2] Morris C., *Legal aspects of monitoring. In: Gravenstein JS*, ed. The Automated Anaesthesia Record and Alarm Systems. Boston: Butterworths, 1987:269-75.

[3] Whitcher C., *Advantages of automated record keeping*. In: Gravenstein JS, Holzer JF, eds. Safety and Cost Containment in Anaesthesia. Boston: Butterworths, 1988:207-21.

[4] Eichhorn JH., *Disadvantages of automated Anaesthesia records.* In: Gravenstein JS, Holzer JF, eds. *Safety and Cost Containment in Anaesthesia*. Boston: Butterworths, 1988:223-32.

[5] S. Fasting and S. E. Gisvold, *"Equipment problems during anaesthesia - are they a quality problem?"*, British Journal of Anaesthesia 89 (6): 825 - 31 (2002)