ORGANIZATION PROJECT

System Control percentage of Oxygen in the body using IOT

DOCTOR

TAMER

PREPARED BY

محمد عبد اللطيف عباس رمضان

سعيد عبد الحسيب سعيد حمد

أمير مسلم محمد على

حسام النبوى أحمد الزمزمى

عبد الله محمد السيد حسين

أحمد عبد الرحمن أحمد الحمود

أحمد عزت شحته

آلاء أشرف فوزى السيد

ABSTRACT

The project is to design an IOT App for measuring the percentage of oxygen in the blood and control the operation of the oxygen device "ventilator" remotely using IOT_based technology. So we build our design on implementing a system for sensing the percentage of oxygen then sending the readings remotely by Wi-Fi and allow another responsible technical people from another places to turn on/off the oxygen device which is implemented in the design as LED to indicate to the state of the devices.

INTRODUCTION

The start of the internet was by connecting number of computers together with server. The server applies services to all computers connected to it. After that more different devices get connected to the network such as phones and laptops. Then it is progressed to a matter that all different things_ not required electronic devices_ can get connected to the internet. From that appeared the definition of internet of things which leaded various things to be able to be connected to the internet, contact with each other, be smarter and interactive. Nowadays the field of internet of things is mostly existed in five categories as the following [1]:

(1) home appliances

(2) smart cities

(3) industry

(4) transportation

(5) medical applications

Our target is to design a medical App for measuring the percentage of oxygen in the blood. Based on that we operate the oxygen device remotely. This operation consists of sub_processes as the following:

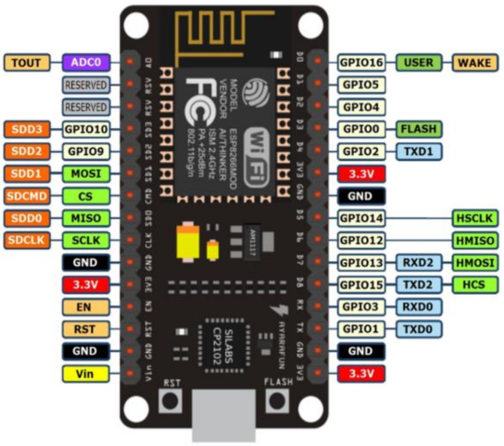
– measure the percentage of oxygen using sensor Max 30100

– send readings and contact with the oxygen device using Wi-Fi

– use "Blynk" App of phones to interface remotely with the ventilator.

This was fulfilled by using the following components:

(1) Node_mcu: the programmable controller kit on which we apply the code and wire other components.



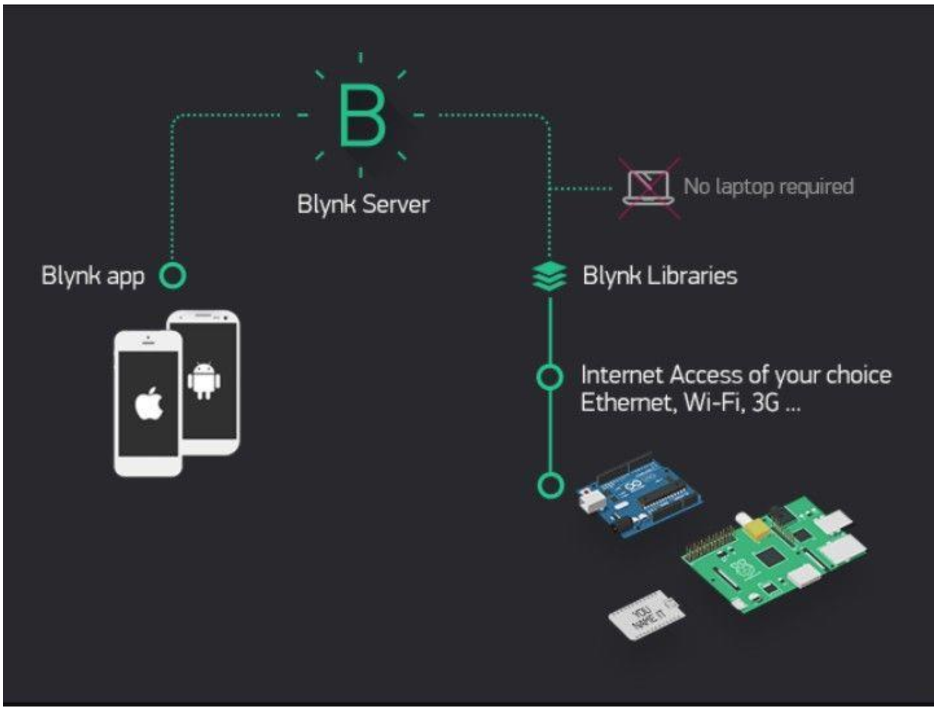
[2]

(2) Max 30100 Pulse Oximeter Sensor: it is an oxygen and pulse rate sensor.

(3) Blynk App: it is a platform designed for IOT Apps. It is available as android and IOS application. Within it we can control hardware and devices remotely which is essential for IOT Apps. The app provides the

ability to display sensor data, store and visualize it. It also allows users to create interactive interfaces in the app using different widges.

Blynk platform provides Blynk server to manage communications between hardware and devices and blynk app in other devices using their open source Blynk cloud [3].



(4) LEDs

(5) connection wires.

DISCUSSION

- In Hardware

After connecting sensor MAX30100 with Node MCU, we used a led in hardware to refer to the ventilator as when it turns on that means the ventilator is on and when it turns off that means the ventilator is off. And we used Blynk app to control the operation of the ventilator remotely using a swtich. In this app, we used two gauges, one to display the percentage of oxygen in blood and the another one to display heart rate. And also, we used a virtual led in Blynk app to refer to the state of the ventilator [5].

- In Code

1) sensor code

```
// Creating instance of PulseOximeter
PulseOximeter pox;
// Getting data from pulse oximeter
SpO2 = pox.getSpO2();
BPM  = pox.getHeartRate();
```

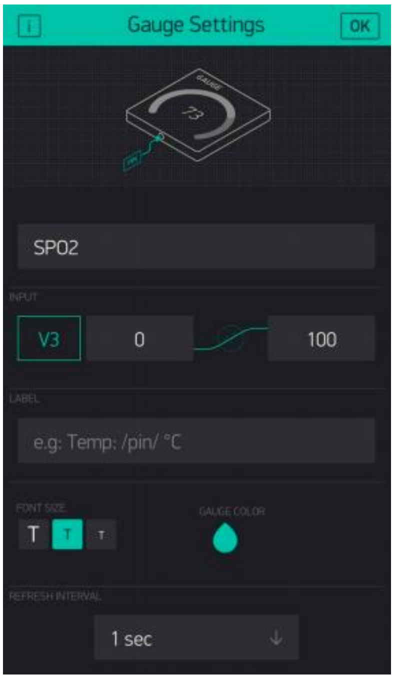
Pulse Oximeter class included in `MAX30100_PulseOximeter.h`
Has two helpful methods such as *getSpO2()* , *getHeartRate()* to get the values of heart rate and oxygen percentage [6].

2) Displaying code the value on gauges

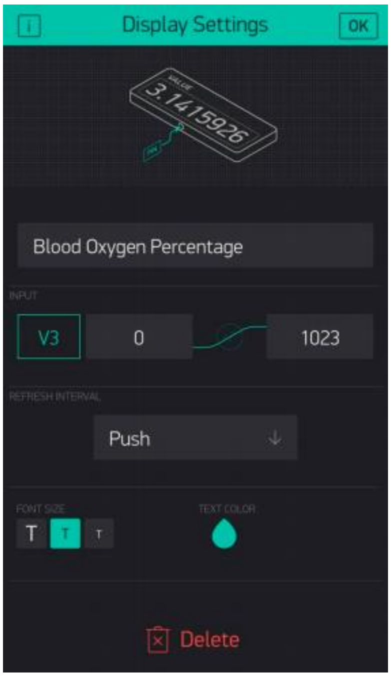
```
        // Arduino Serial Port
        Serial.print("BPM: ");
        Serial.print(BPM);
        Serial.print("  ||   SpO2: ");
        Serial.print(SpO2);
        Serial.print("%");
        Serial.println("\n");

       // Blynk Gauges
       Blynk.virtualWrite(V2, BPM);
        Blynk.virtualWrite(V3, SpO2);
```

In this project there is two ways to display the values we get from Pulse Oximeter sensor. The first used in debug in Arduino IDE Monitor Serial Port, And the second used in Blynk app using gauges [7].



SPO2
INPUT
V3 0 100
LABEL
e.g: Temp: /pin/ °C
FONT SIZE T T T
GAUGE COLOR
REFRESH INTERVAL
1 sec



Blood Oxygen Percentage
INPUT
V3 0 1023
REFRESH INTERVAL
Push
FONT SIZE T T T
TEXT COLOR

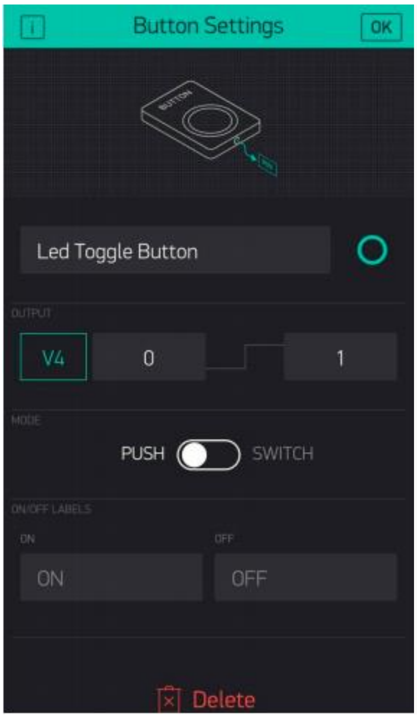
2) Manual control code

```
BLYNK_WRITE(V4){

  int pinValue = param.asInt();
  // Get the Button Value from Blynk App
  if(pinValue == 1){
    spo2StatusLed.on();
    digitalWrite(LED, HIGH);
  }
  if(pinValue == 0){
    spo2StatusLed.off();
    digitalWrite(LED, LOW);
  }

}
```

This code controls at hardware led and virtual led according to the state of the switch (V4) that in Blynk app. As when the state of the switch is high, these leds turn on. And when the state of the switch is low, these leds turn off.



Led Toggle Button

OUTPUT

V4 0 1

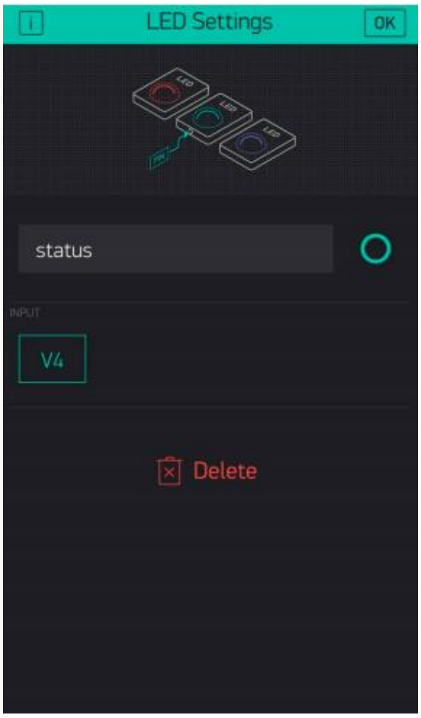
MODE

PUSH SWITCH

ON/OFF LABELS

ON: ON

OFF: OFF



status

INPUT

V4

3) Automatic code

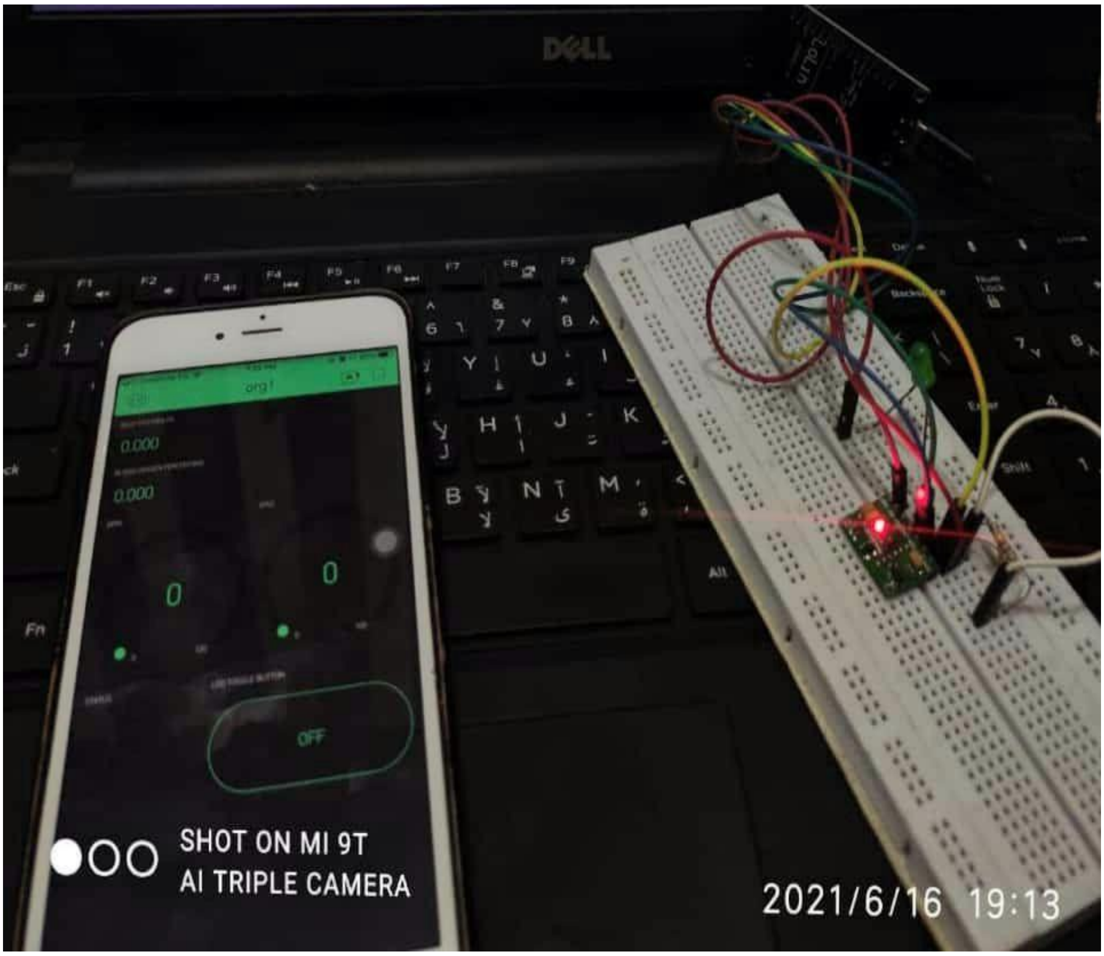
```
if (SpO2 < 95 && SpO2 != 0) {
    spo2StatusLed.on();
    digitalWrite(LED, HIGH);
} else {
    spo2StatusLed.off();
    digitalWrite(LED, LOW);}
```

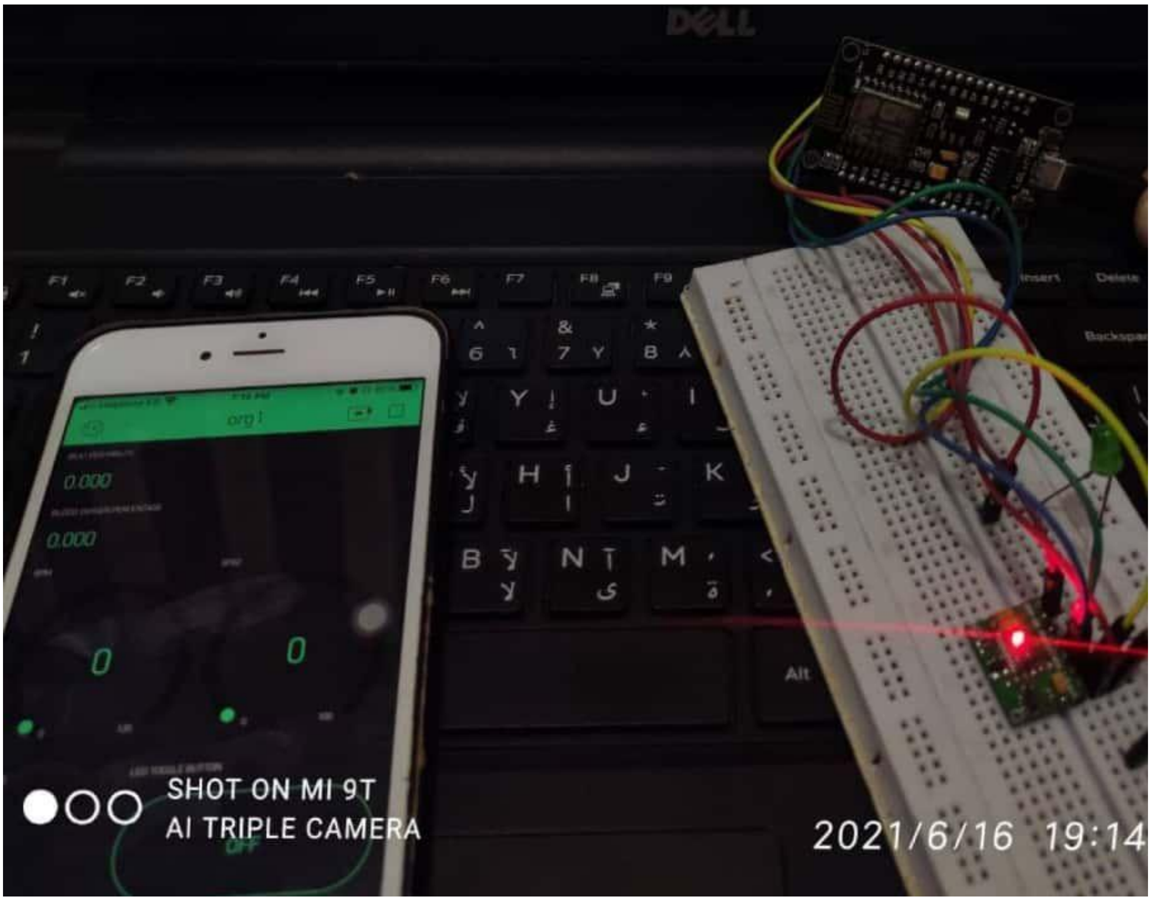
This code controls automatically at hardware led and virtual led according to the value of the sensor. As when the value is less than 95, but make sure that is not zero (which means that the sensor in this case does not measure any value), these leds turn on. And when the value is larger than 95, these leds turn off.

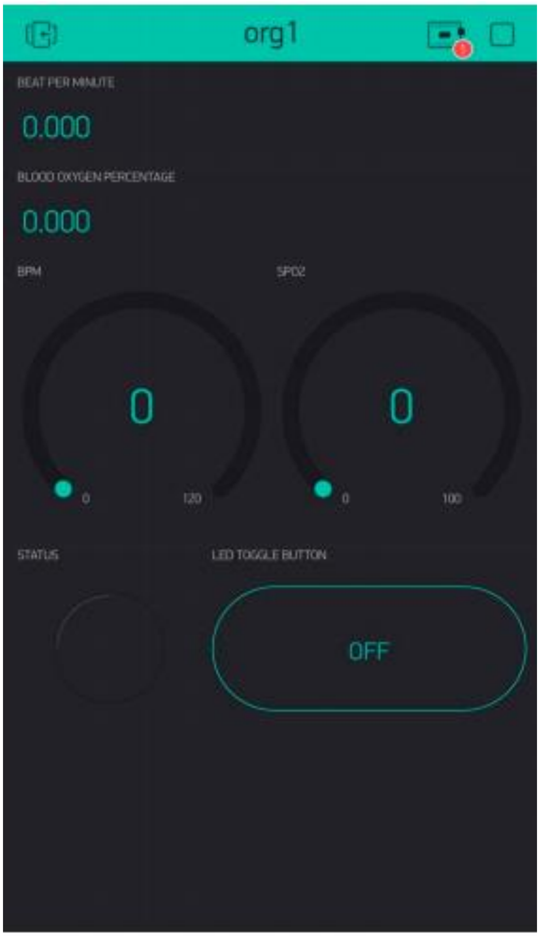
RESULT

Video Link

Photos







BEAT PER MINUTE
0.000

BLOOD OXYGEN PERCENTAGE
0.000

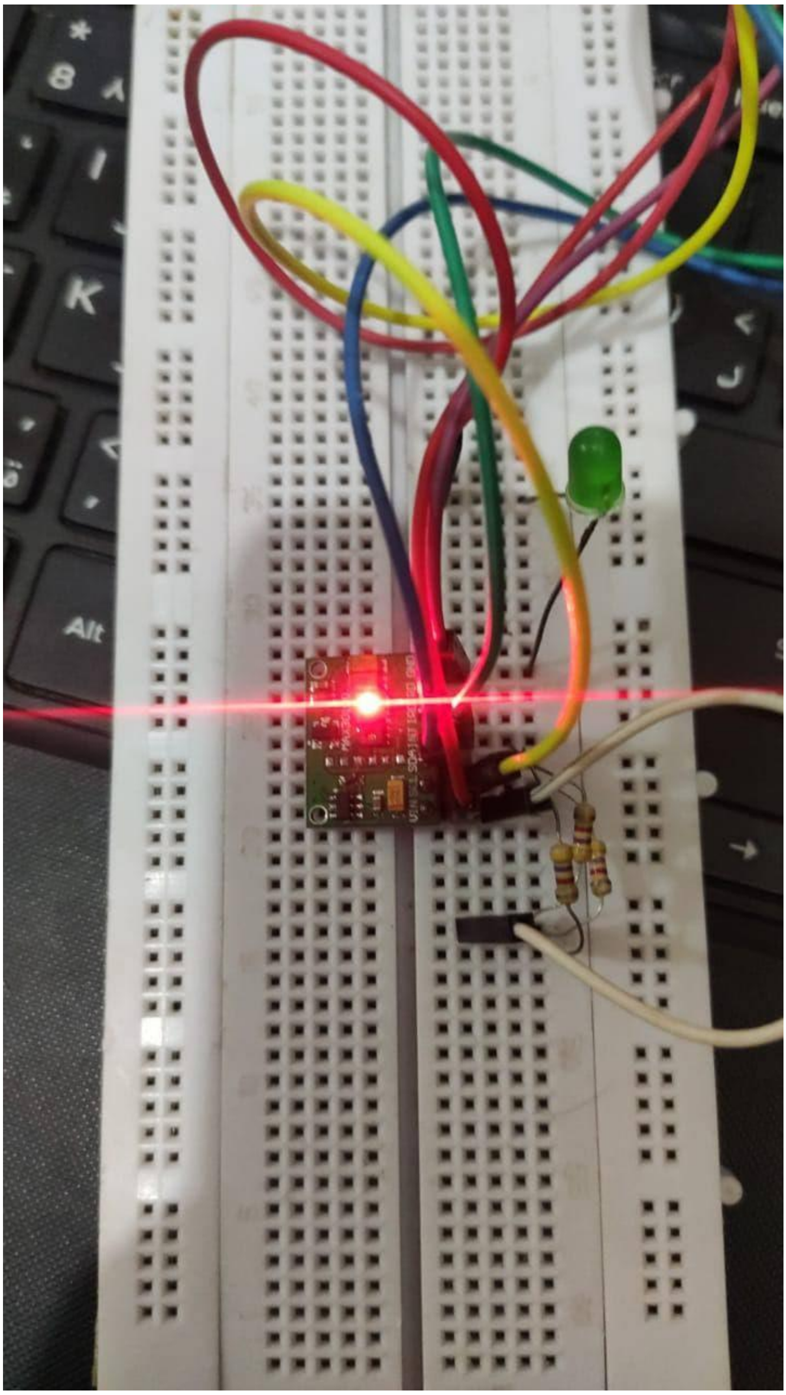
BPM
0
0 120

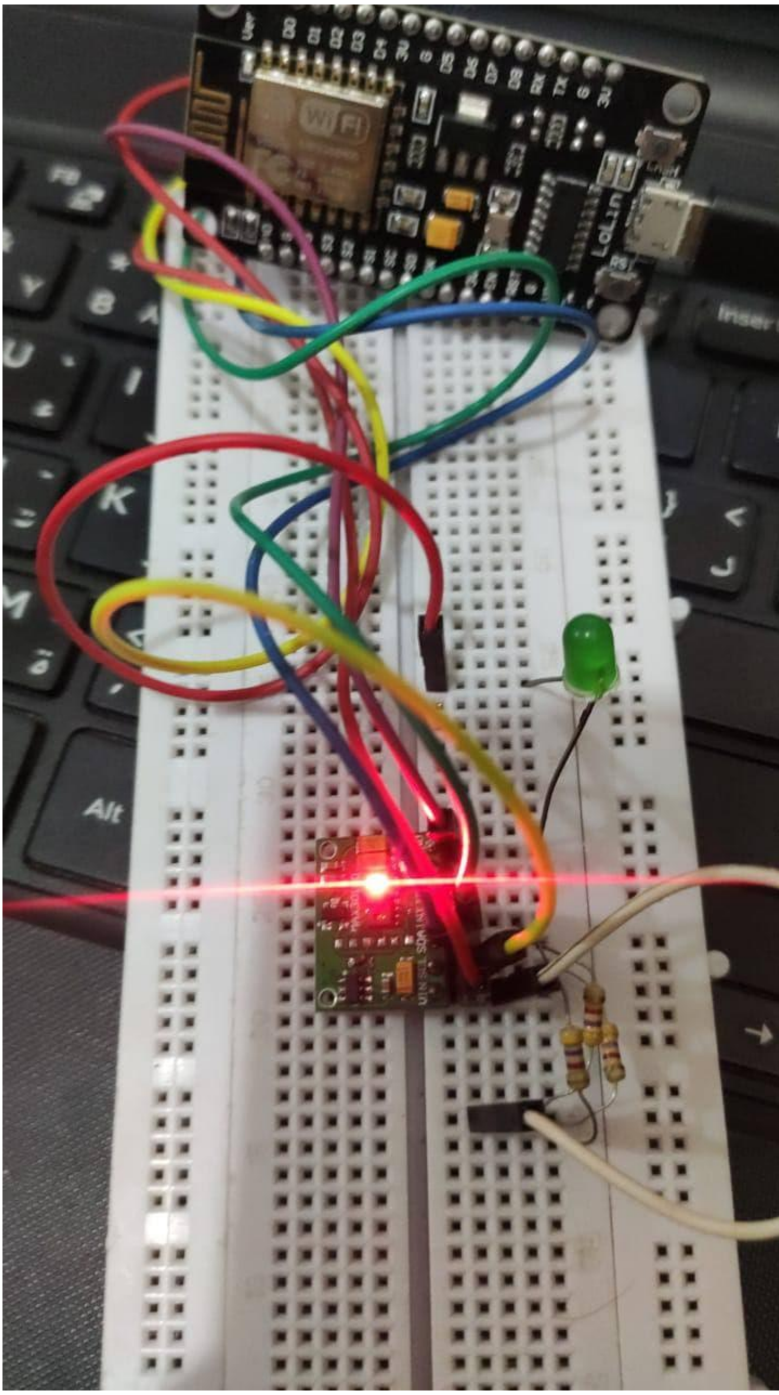
SPO2
0
0 100

STATUS

LED TOGGLE BUTTON
OFF

Finally, after connecting the circuit and linking it with Blynk app we managed to make the NodeMCU work with no errors and connect to the WIFI and this is the base idea to make sensor reading alive in the cloud.





CONCLUSION

Before **Internet of Things**, patients' interactions with doctors were limited to visits, and phones and text communications. Doctors or hospitals couldn't monitor patients' health continuously and make recommendations according to patient health.

Thanks to Internet of Things (IoT), it is possible to have a remote monitoring of patient in the healthcare sector, to keep patients safe and healthy, and empowering physicians to deliver superlative care. Furthermore, remote monitoring of patient's health helps in reducing the length of the length of recovery [1].

Our project will make it possible to trace patient oxygen percentage, and in Circumstances like that we are living now -corona virus disease- it is important do doctors to trace oxygen percentage remotely.

PROBLEMS

(1) In the beginning we use Bluetooth to send readings and operate the device remotely but its range was small so we use Wi-Fi module.

(2) the input voltage required for the sensor max30100 is 3.3 volt in minimum but it didn't work on the kit so we removed some additional resistors imbedded in the sensor.

REFERENCES

[1] HTTPS://MAHARATECH.GOV.EG/COURSE/VIEW.PHP?ID=15

[2] NODEMCU REFERENCE

[3] https://docs.blynk.cc/

[4] MAX30100 SHEET

[5] circuit connection

[6] MAX30100 Pulse Oximeter Webserver using NodeMCU ESP8266 (theiotprojects.com)

[7] Arduino Reference - Arduino Reference