****

**American International University- Bangladesh (AIUB)**

**Department of Computer Science**

**Advanced Operating System**

**Spring 2020-2021**

**Faculty: M. Arifur Rahman**

**Project: Patient Monitoring System**

**Section: A**

|  |  |
| --- | --- |
| **Name** | **ID** |
| **ASIF AL JUBAYER** | **17-35983-3** |
| **MD. WAHIDUL HAQUE** | **17-35453-3** |
| **MD FAHIM ALMAS** | **18-37602-1** |
| **ANIKA TABASSUM** | **17-34982-2** |
| **KHAN MD. AZIMUZZAMAN** | **09-13228-1** |

**Project Idea**

As we all know that the world is going through a very tough time. Covid-19 has done a lot of damage in our daily life. This virus is very contagious. That’s why it spread very fast. It spread form an affected person to another person. This virus attacks our lunge. That’s why the blood oxygen level gets down. So, we decided to build a Blood oxygen measurement device. Which is IOT based. So that the doctor can check the patient’s oxygen level and BPM without touching the person. For this feature we use Blynk application. This app will show the result through internet network.

**Component List**

* ESP8266 Wi-Fi Development Board.
* Max30100 BPM & Oxygen Measure Sensor
* DS1820B Temperature Sensor
* Jumper Wires
* Bread Board

**Project Implementation Description**

Step: 01 (Attaching Breadboard with ESP8266)

* First insert a pin of Male-Male jumper cable into Esp8266’s GND port then insert the other pin into port of the cable into the Breadboard’s negative hole.
* Then insert another Male-Male cable’s one pin into Esp8266’s 3.3V port and other one into positive port of the breadboard.
* After that connect other negative-positive side of the Breadboard with Male-Male jumper cables. (Remember you have to insert cables into positive-positive and negative-negative)

Step: 02 (Attaching the Max30100 Sensor)

* At the beginning you have to connect the sensor’s GND with Breadboard’s negative port and the VCC with Breadboard’s positive port using Male-Male and Female-Female jumper cables or Male-Female jumper cables. In our project we have used Male-Male and Female-Female due to unavailability of Male-Female cable.
* Then connect the sensor’s SCL, SDA & INT pin with breadboard’s any middle portion ports. (The middle portion may indicate by a, b, c, d, e, f, g, h)
* Next you have to insert the pin of jumper cables in any port beside of the port at the same row that you have inserted the signal port of the sensor and after that you have to put other pin of that jumper cable into ESP8266’s D1, D2 & D0 Pin port.
* Finally following the upper instructions connect the sensor.

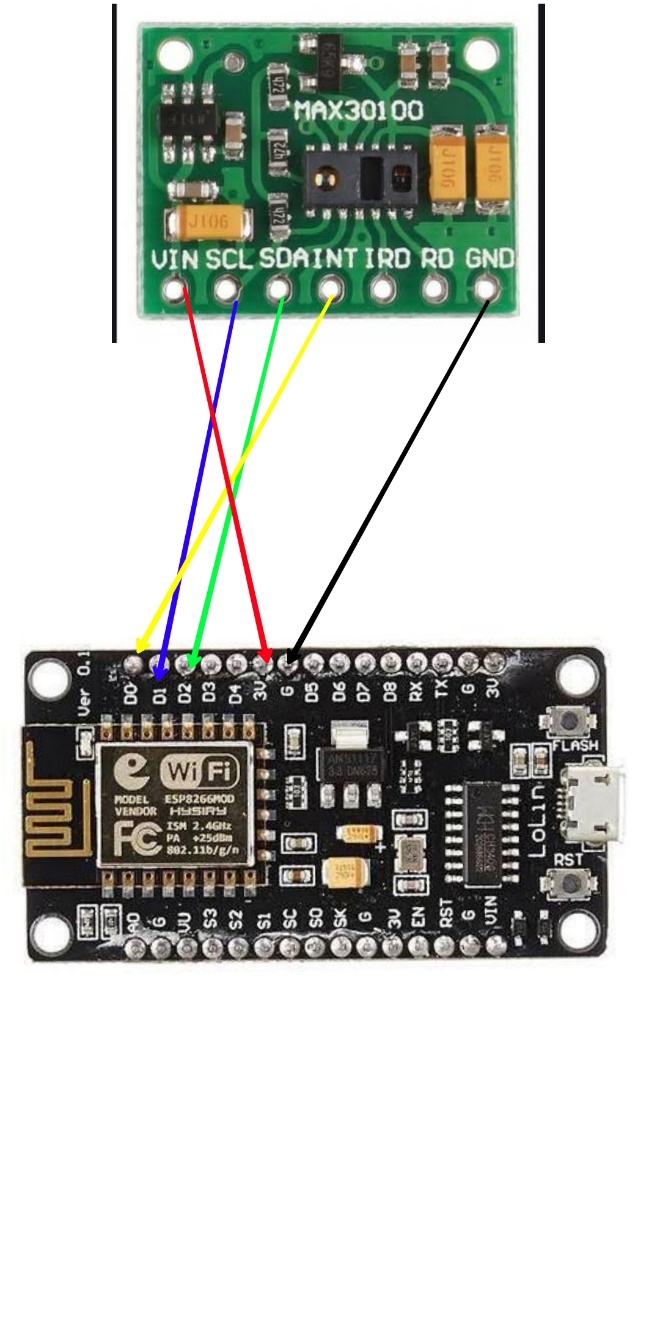
Step: 04 (Connecting Esp8266 with your Laptop/PC)

* Connect Esp8266 with your Laptop/PC using USB cable Type B.
* Open Arduino IDE and make sure you have selected the right board of Esp8266 as well as the right port.

Step 05: (Connecting with Blynk Application)

* Download the blynk app from Playstore. Install it on your phone. Sing-up with a google account.
* Create new project. Select “NodeMcu” as Board and select network type as WiFi. A token will send automatically to your given mail address. Copy the token number and paste it into your code.
* Then take two Gauge from the options. Select one Gauge as SPo2(oxygen level) and another one is BPM

**Circuit Diagram**



**Implemented Code**

#include <Wire.h>

#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>

#include "MAX30100\_PulseOximeter.h"

#include <SimpleTimer.h>

#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>

char auth[] = "6497f4ca052e43e3817c1e7157926730"; // You should get Auth Token in the Blynk App.

char ssid[] = "JustDo"; // Your WiFi credentials.

char pass[] = "par12345";

#define REPORTING\_PERIOD\_MS 3000

SimpleTimer timer;

PulseOximeter pox;

uint32\_t tsLastReport = 0;

void onBeatDetected()

{

;

}

void setup()

{

Serial.begin(115200);

Blynk.begin(auth, ssid, pass);

Serial.print("Initializing pulse oximeter..");

if (!pox.begin()) {

Serial.println("FAILED");

for (;;);

} else {

Serial.println("SUCCESS");

digitalWrite(1, HIGH);

}

pox.setIRLedCurrent(MAX30100\_LED\_CURR\_24MA);

// Register a callback for the beat detection

pox.setOnBeatDetectedCallback(onBeatDetected);

timer.setInterval(3000L, getSendData);

}

void loop()

{

timer.run(); // Initiates SimpleTimer

Blynk.run();

// Make sure to call update as fast as possible

pox.update();

if (millis() - tsLastReport > REPORTING\_PERIOD\_MS) {

// to android cell phone application

Serial.print("BPM: ");

Serial.print(pox.getHeartRate());

//blue.println("\n");

Serial.print(" SpO2: ");

Serial.print(pox.getSpO2());

Serial.print("%");

Serial.println("\n");

Blynk.virtualWrite(V2, pox.getHeartRate() );

Blynk.virtualWrite(V3, pox.getSpO2());

tsLastReport = millis();

}

}

void getSendData()

{

}

**Reference**

[1]<https://components101.com/development-boards/nodemcu-esp8266-pinout-features-and-datasheet>

[2]<https://www.instructables.com/NodeMCU-ESP8266-Details-and-Pinout/>

[3]<https://www.espressif.com/en/products/socs/esp8266>

[4]<https://datasheets.maximintegrated.com/en/ds/MAX30100.pdf>

[5]<https://www.maximintegrated.com/en/products/sensors/MAX30100.html#:~:text=Product%20Details,-Package%20%7C%20Pin%20%7C%20Size&text=The%20MAX30100%20is%20an%20integrated,oximetry%20and%20heart%2Drate%20signals>.

[6]<https://circuitdigest.com/microcontroller-projects/iot-based-heart-rate-monitor-using-max30100-pulse-oximeter-and-esp32>

[7] <https://ieeexplore.ieee.org/document/8554425>

[8] <https://www.researchgate.net/publication/324798616_Internet_of_Things_Low_Cost_and_Wearable_SpO2_Device_for_Health_Monitoring>