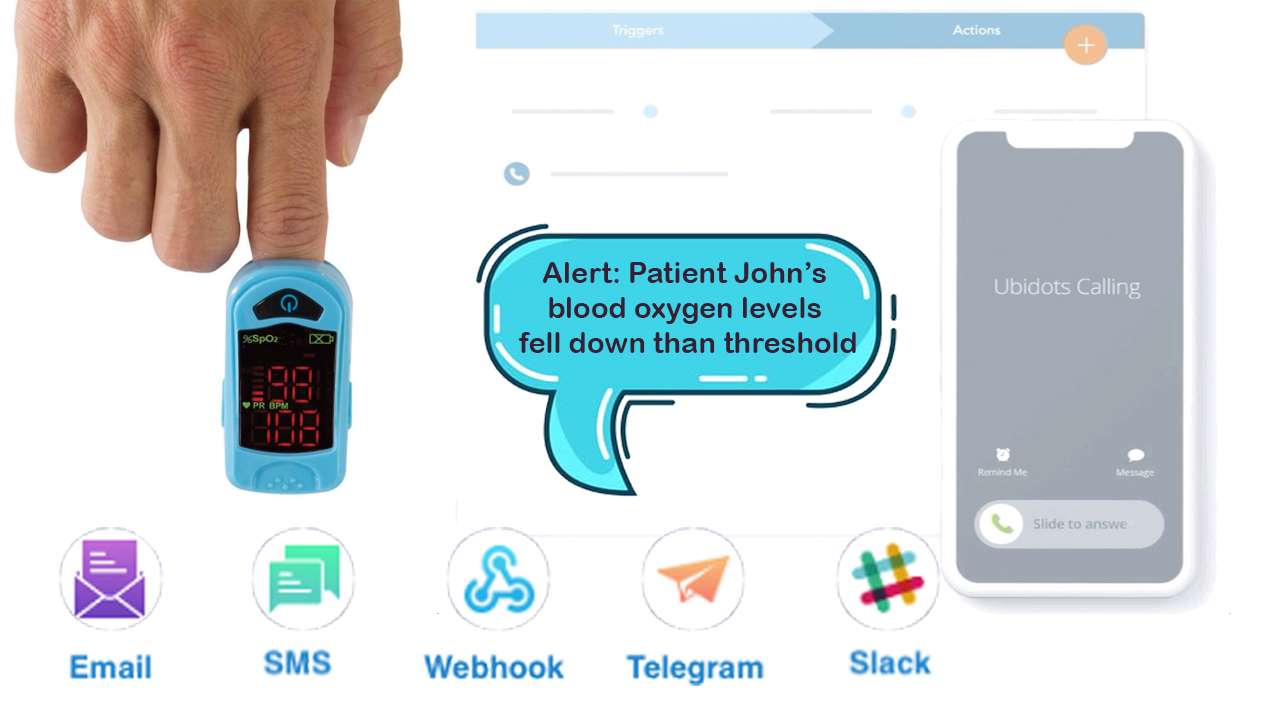
**IoT based Smart Patient Health Monitor with Critical Medical alerts using NodeMCU & Ubidots**



Introduction: Smart Patient Blood Oxygen monitor

With tons of new healthcare technology start-ups, IoT is rapidly revolutionizing the healthcare industry. Keeping track of the **health status** of your **patient** at home is a difficult task because of the busy schedules and our daily life work. Especially **old age patients** should be periodically monitored. So, we propose an **innovative system** that automated this task with ease. Our device puts forward a smart **patient health tracking system** using **Web Server** so that the **Patient health** parameters like **Heart Rate** and **Blood Oxygen level** along with **body temperature** can be monitored.

So, in this project we will learn how to make **IoT Based Patient Health Monitoring System Project**. We will use the **MAX30100/102 Pulse Oximeter sensor** to measure the Heart Rate/Pulse(BPM) as well as the Blood Oxygen Level(SpO2). We will use a **DS18B20 Temperature Sensor** to measure the temperature of the body. Similarly Patient needs to be placed in a room with a certain temperature and humidity level so that he doesn’t feel uncomfortable. To do that we need to monitor the room temperature and humidity as well So, we will be using **DHT11 Humidity & Temperature Sensor**.

”Smart Patient Blood Oxygen monitor”, we are going to talk all about “IoT based Smart Patient Blood Oxygen monitor with ESP8266 and sending critical alerts to the hospital management and patient loved ones” This project is truly designed for patients whose blood oxygen levels need to be monitored 24×7, for example, covid patients or any other patients who are on the ventilator. This project continuously monitors the blood oxygen levels and logs them to the ubidots IoT platform. also, it will send critical medical alerts like voice calls, SMS, Emails, and telegram message to the Hospital management and patient loved ones with the help of Ubidots events and alerts features.

Required Components:

**1.**[**Breadboard**](https://amzn.to/2DZsesr)  
**2.**[**ESP8266 Module**](https://amzn.to/37IsJ50)  
**3.**[**MAX30100 pulse oximeter sensor**](https://amzn.to/37AVhON)  
4. [**4.7k ohm resistors**](https://amzn.to/30ZBuX1)

5. [**DHT**](https://amzn.to/30ZBuX1)**11 sensor**  6. [**Connecting wires.**](https://amzn.to/2ZlUD47)

7. DS18B20 Temperature Sensor

Overview: Pulse Oximeter and Heart-Rate Sensor (MAX30100)

### Sensor Description:

The MAX30100 is complete pulse oximetry and heart-rate sensor system solution designed for the demanding requirements of wearable devices. The MAX30100 provides very small total solution size without sacrificing optical or electrical performance. Minimal external hardware components are needed for integration into a wearable device. The MAX30100 is fully configurable through software registers, and the digital output data is stored in a 16-deep FIFO within the device. The FIFO allows the MAX30100 to be connected to a microcontroller or microprocessor on a shared bus, where the data is not being read continuously from the device’s registers.

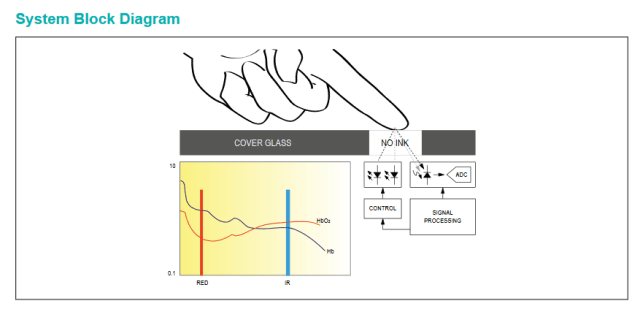


#### SpO2 Subsystem

The SpO2 subsystem in the MAX30100 is composed of ambient light cancellation (ALC), 16-bit sigma-delta ADC, and a proprietary discrete-time filter. The SpO2 ADC is a continuous-time oversampling sigma-delta converter with up to 16-bit resolution. The ADC output data rate can be programmed from 50Hz to 1kHz. The MAX30100 includes a proprietary discrete-time filter to reject 50Hz/60Hz interference and low-frequency residual ambient noise

#### LED Driver

The MAX30100 integrates red and IR LED drivers to drive LED pulses for SpO2 and HR measurements. The LED current can be programmed from 0mA to 50mA (typical only) with proper supply voltage. The LED pulse width can be programmed from 200μs to 1.6ms to optimize measurement accuracy and power consumption based on use cases.



Benefits and Features

* Complete Pulse Oximeter and Heart-Rate Sensor Solution Simplifies Design Integrated LEDs, Photo Sensor, and High-Performance Analog Front -End Tiny 5.6mm x 2.8mm x 1.2mm 14-Pin Optically Enhanced System-in-Package.
* Ultra-Low-Power Operation Increases Battery Life for Wearable Devices
* Programmable Sample Rate and LED Current for Power Savings
* Ultra-Low Shutdown Current (0.7μA, typ)
* Advanced Functionality Improves Measurement Performance
* High SNR Provides Robust Motion Artifact Resilience
* Integrated Ambient Light Cancellation
* High Sample Rate Capability• Fast Data Output Capability

Applications

* Wearable Devices
* Fitness Assistant Devices
* Medical Monitoring Devices

**DHT11 Humidity & Temperature Sensor**

The **DHT11** is a basic, ultra low-cost digital temperature and humidity sensor. It uses a capacitive **humidity sensor** and a **thermistor** to measure the surrounding air and spits out a digital signal on the data pin (no analog input pins needed).

[](https://how2electronics.com/wp-content/uploads/2020/08/DHT11-Humidity-Temperature-Sensor.jpg)

It’s fairly simple to use but requires careful timing to grab data. The only real downside of this sensor is you can only get new data from it once every 2 seconds, so when using the library, sensor readings can be up to 2 seconds old.

**DS18B20 Temperature Sensor**

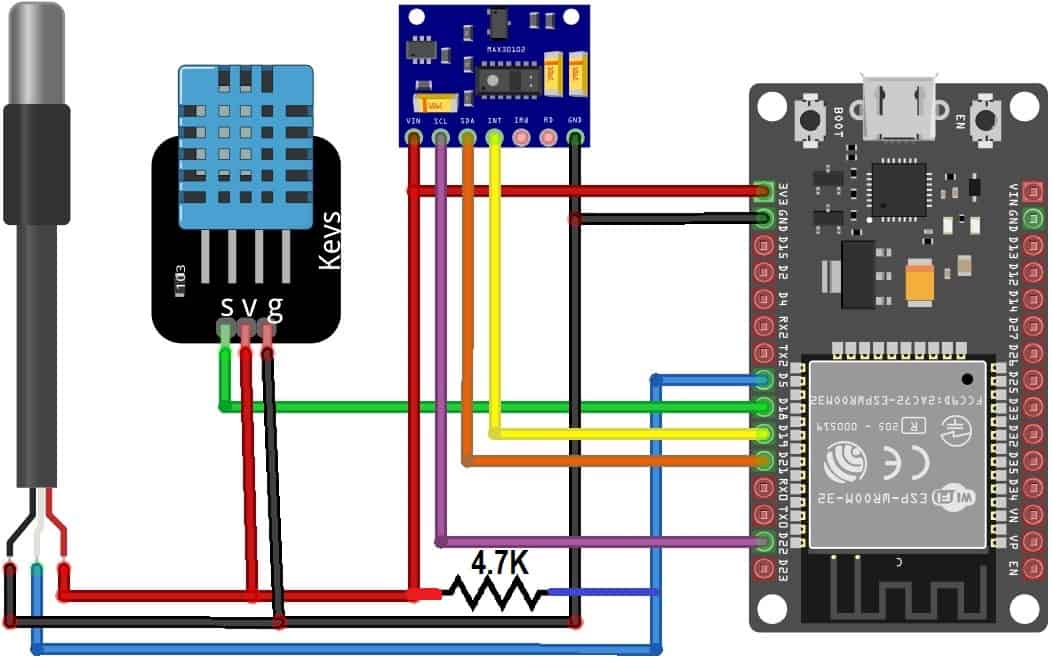
This is a pre-wired and **waterproofed version of the DS18B20 sensor**. Handy for when you need to measure something far away, or in wet conditions. The Sensor can measure the temperature between **-55 to 125°C** (-67°F to +257°F). The cable is jacketed in PVC.

Because it is digital, there is no signal degradation even over long distances. These 1-wire digital temperature sensors are fairly precise, i.e **±0.5°C** over much of the range. It can give up to 12 bits of precision from the onboard digital-to-analog converter. They work great with any microcontroller using a single digital pin.

[](https://how2electronics.com/wp-content/uploads/2018/12/waterproof-temperature-sensor-ds18b20.jpg)

The only downside is they use the **Dallas 1-Wire protocol**, which is somewhat complex and requires a bunch of code to parse out the communication. We toss in a 4.7k resistor, which is required as a pullup from the DATA to the VCC line when using the sensor.

Circuit Diagram: Smart Patient Health Monitor System

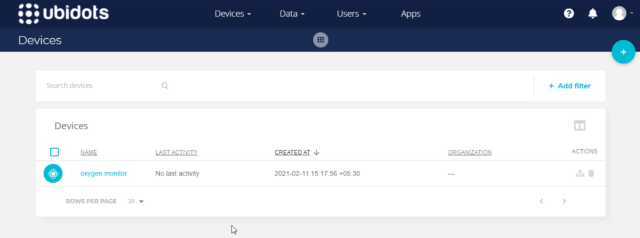


Creating device in Ubidots:

Before jump into the programming, part let’s create a device in the Ubidots Platform for our solution. follow the below steps.

1. Visit [ubidots.com](https://ubidots.com/),
2. Go to Dashboard,
3. Click on devices,
4. create a new device, select a blank device,
5. Enter the name for the device then click on this green tick mark to create the device.

That’s it the device is successfully created.



Programming part: Smart Patient Blood Oxygen monitor

Coming to the programming part, Here Is the code for the IoT-based Smart Patient Blood Oxygen meter with ESP8266. We need to provide the below details, to make this code work with your Ubidots device.

#### Wi-Fi Credentials:

Provide WiFi SSID and password for Internet Connection.

#define WIFISSID “XXXXXXXXXXXXXXXXX” // Put your WifiSSID here  
#define PASSWORD “XXXXXXXXXXXXX” // Put your wifi password here

#### Ubidots Credentials:

Provide Ubidots token, we can get this from the Ubidots itself. Go to API Credentials. Copy the token and past it over here.

Enter MQTT Client name, this can be any alphanumeric string.

Then enter the Variable label & Device label, which we have created on Ubidots earlier.

#define TOKEN “XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX” // Put your Ubidots’ TOKEN  
#define DEVICE\_LABEL “XXXXXXXXXXXXXXX” // Put the device label  
#define VARIABLE\_LABEL\_1 “heartrate” // Put the variable label  
#define VARIABLE\_LABEL\_2 “SPo2” // Put the variable label  
#define MQTT\_CLIENT\_NAME “EI\_OXMO” // MQTT client Name, put a Random ASCII

Now, Go to Ubidots and reload it. If you click on the device, you can see the variables and their data.

The best part of the Ubidots is if it receives data from any Module, it will automatically create those variables and start storing data for us. here you can see the same variables are created automatically.



Ubidots Data Visualization:

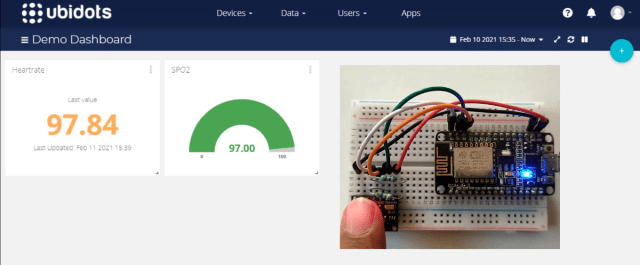
Currently, we are receiving and storing it on the ubidots, let’s visualize the same data. for that Click on Data and then choose Dashboard.

Click on add new widget, you will be prompted with a bunch of data visualization options, choose one among them, I will use a metric to display the Heart rate. Click on add variables, then select the device, and select the variable heart rate, Add a name for the metric. keep the rest of the options default. Later create it.

Now, let’s repeat the same process for SPo2. For spo2 I would like to choose the gauge option to visualize it, click on add variables, then select the device, and select the variable SPo2, Add a name for the gauge. for Spo2 I would like to add color logic to differentiate the safe or critical blood oxygen levels.

we can add that easily from ubidots, select color logic, add color logic. if the blood oxygen levels are 95% or above, they are said to be safe, below 95 % are critical. So, I will put green color if the oxygen levels are greater than 95 and put red color if the oxygen levels are less than 95%. Then click on accept.

After creating you can see the live data in green color on the gauge. because it is greater than 95. if that value is lesser than 95 you can see the red color.



Configuring Critical Medical Alerts using Ubidots Event Function:

Now, let’s move to the most important thing in the episode, Creating alerts when abnormal blood oxygen % is noticed. Ubidots supports integrated events to allow us to send Alerts and notifications to those who need to know when they need to know. so in our case the hospital management and the caretaker of the patient whenever the blood oxygen levels go down. so, let’s configure it now.

Follow the below steps to configure Critical Medical alerts using Ubidots Events Engine.

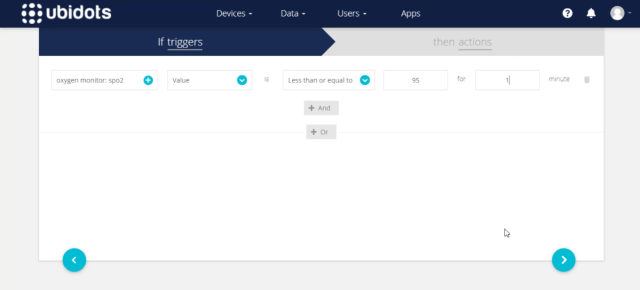
* Select Events from the Data dropdown.
* Click on the create event button.
* Select the If triggers tab to organize your event logic.

First, we will create a condition to trigger the event and then call appropriate action to take place. I wanted to use SPo2 data for conditional expression so, I have selected SPo2,

* Click on Select variable,
* Select the device: Oxygen monitor,
* Select the variable SPo2.

Establish event logic if the value is less than or equals to 95%, for 1 minute.

That’s it, event logic for the If triggers tab is completed.



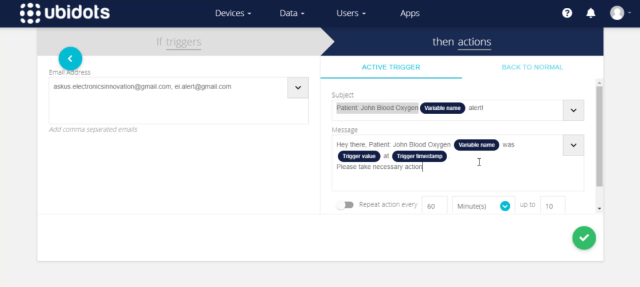
Select the Then actions tab to execute the planned Event or Alert such as Email, SMS, Telegram, Voice Call, Slack, or Webhook.

Click on add new action, Select and Configure which actions are to be executed and the message for the receiver. Available actions include Email, SMS, Telegrams, Voice Call, Set Variable, SLACK, or WebHooks.



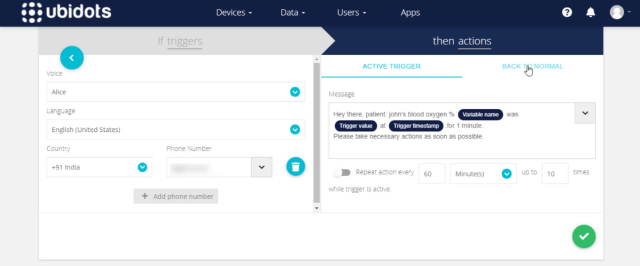
For this solution I would like to configure Email, SMS, Telegrams, Voice Call, first of all, I would like to configure email.

Enter the Email address, we can add 2 or more emails by simply separating them with a comma. then type the alert mail subject and message as you wish. you can also Add Specific Data Inputs like Variable Name, Device Label, Trigger Value, Trigger Timestamp into the Event for the immediate info after configuring the message create it.



likewise, we can configure and create the rest of the alerts like SMS, Telegram Message, and Voice call.  
All you have to do is providing the target mobile not along with the country code and the alert message.

For voice call, you can select the voice of the alert message between John and Alice whereas john is a male voice and Alice is a female voice. then the rest of the details are as same as SMS and telegram message.



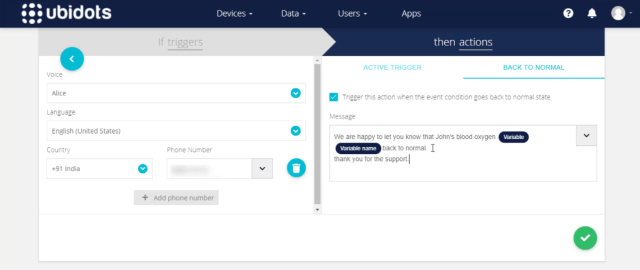
So, all these alerts are triggered when the Spo2 value is less than or equals to 95%, for 1 minute.

After creating an alert, it is inevitable that the event will be triggered at some point;  
one great feature of Ubidots Events Engine is the “Back to Normal” conditional notification. it would send us another alert if the triggered value came back to normal.  
To activate a Back to Normal notification, simply follow the below three steps.

Select the “BACK TO NORMAL” tab from the event’s actions tab.

Select the “Trigger this action” box to verify the desire to receive a “back to normal” notification.

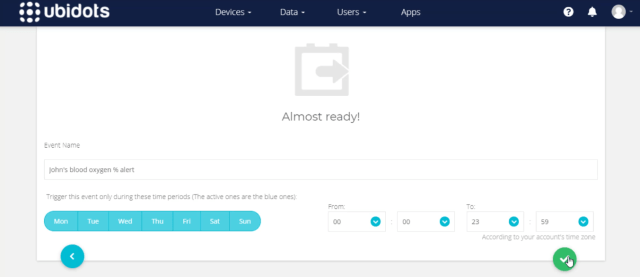
Add the Custom Notification Text as you see fit, then save action.



likewise enable the back to the normal notification to the rest of the Telegram message, SMS, and Email alerts.

After configuring all alerts click on next.

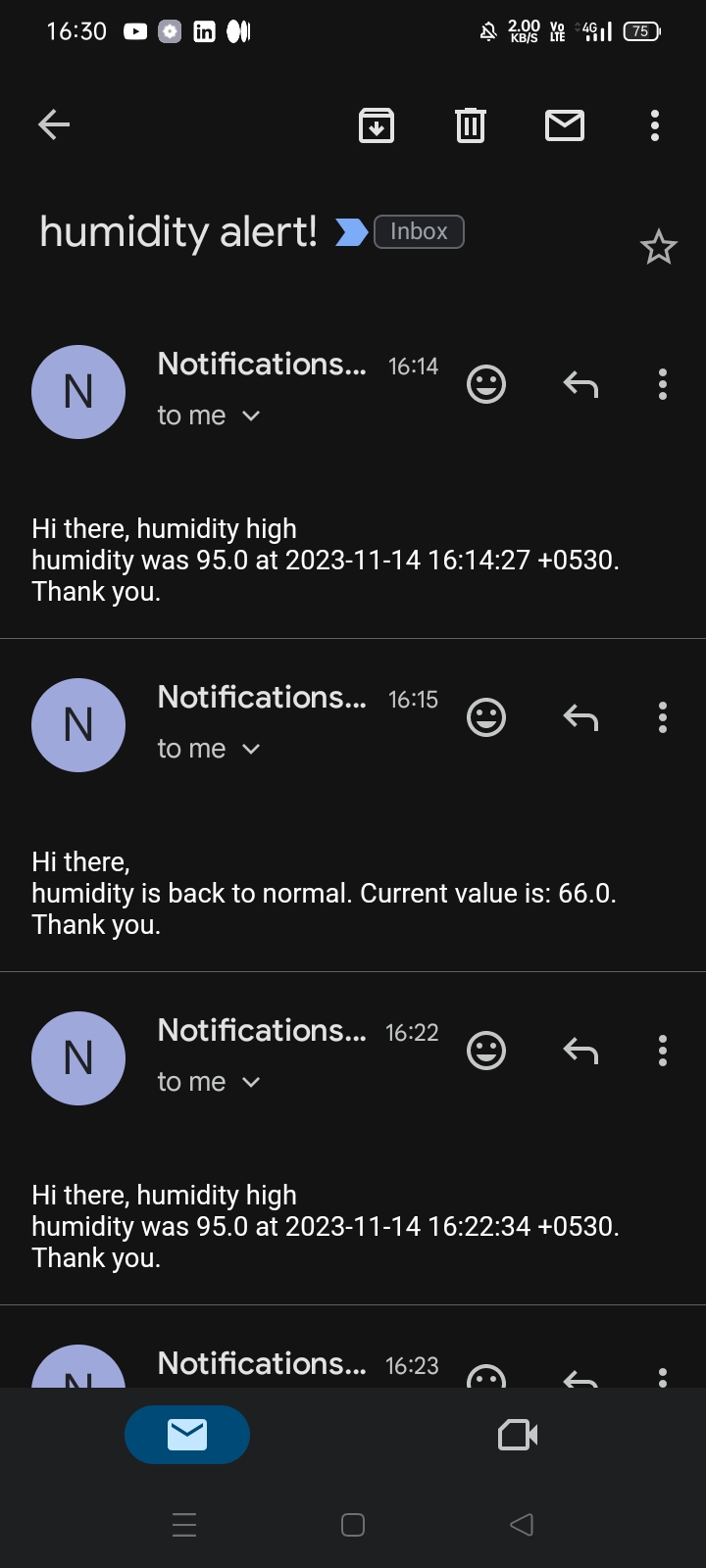
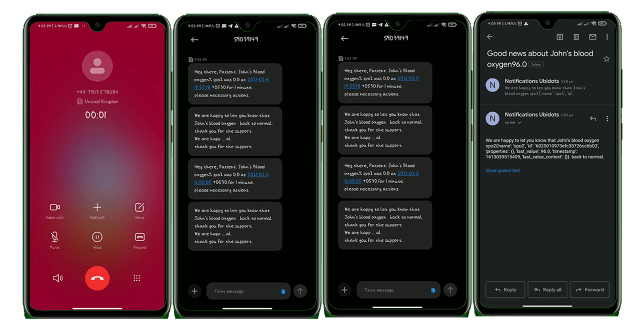
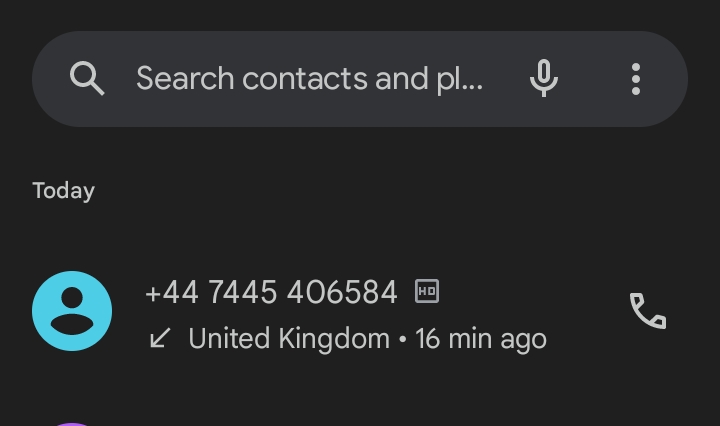
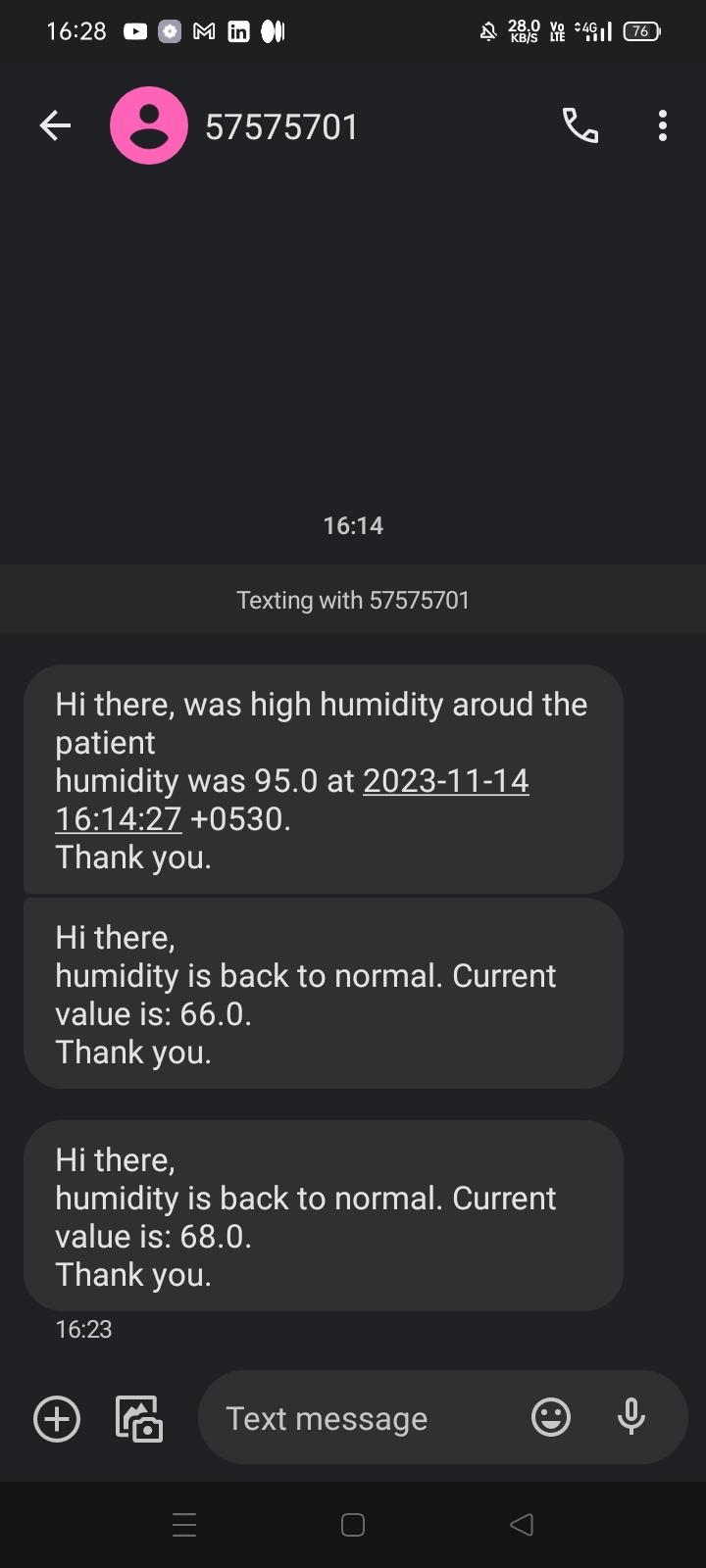
Here you can give a meaningful name to your event and determine when the events should to be executed and not executed.  
For our solution, we have to monitor the patient oxygen levels 24×7 so keep it default and save it.



Now, it’s time to test this event. So come back to the dashboard. here the data can be updated seamlessly.

To make Spo2 less than 95% remove the finger from the top of the sensor, so it will send 0 reading which will satisfy the event condition. then wait for 1 minute, to get event alerts.

Yessss….. After 1 minute, we have received a voice call from ubidots, Also received the SMS. you can read the text in the below picture which is the same as we configured. We also received telegram message from IoT notifications, maybe IoT notifications is the bot created by ubidots. anyway, we have received the Telegram alert. Mail also received from notifications ubidots, with the Subject and mail body we have configured on the event.

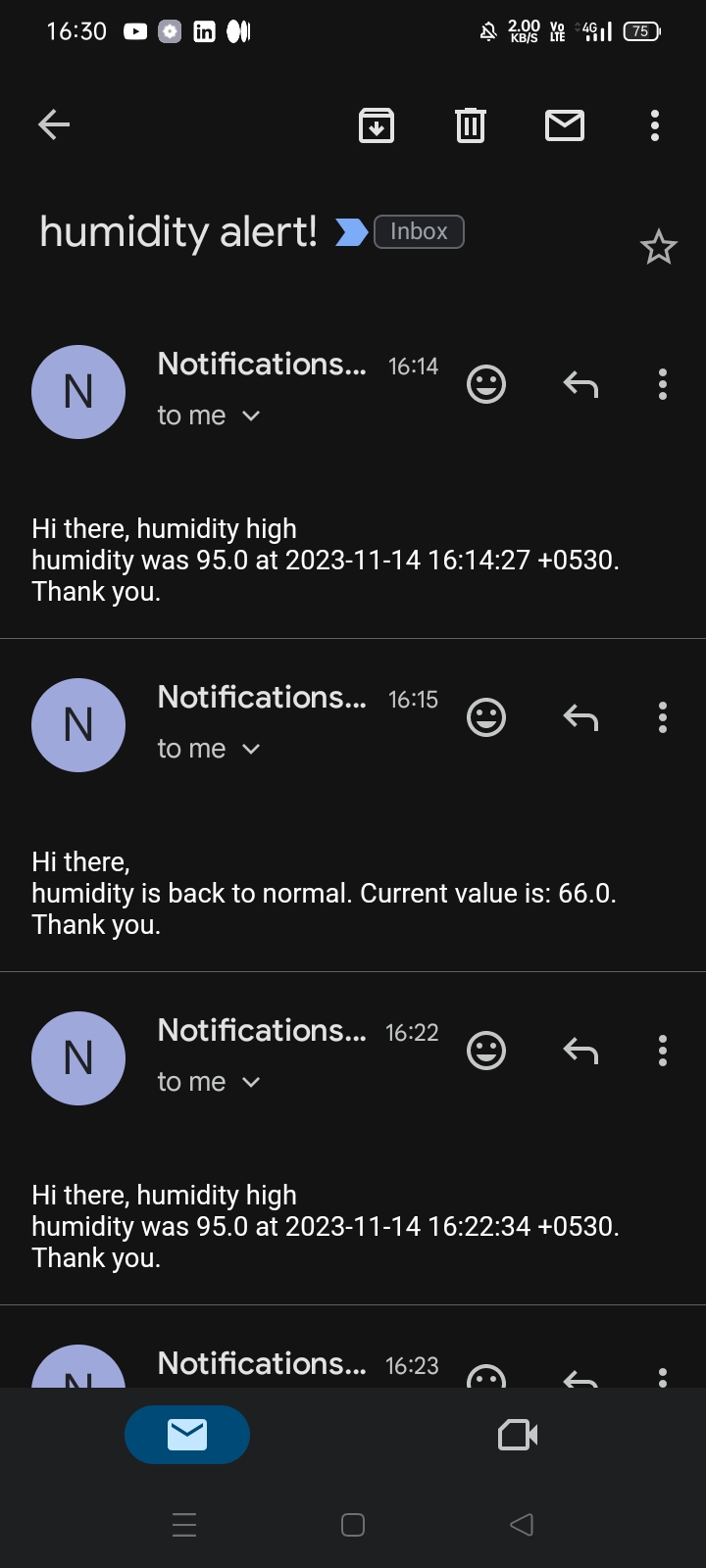
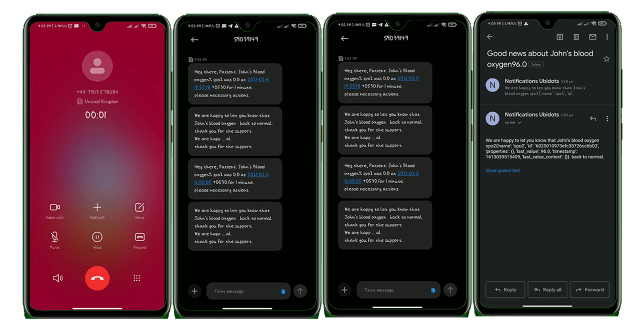
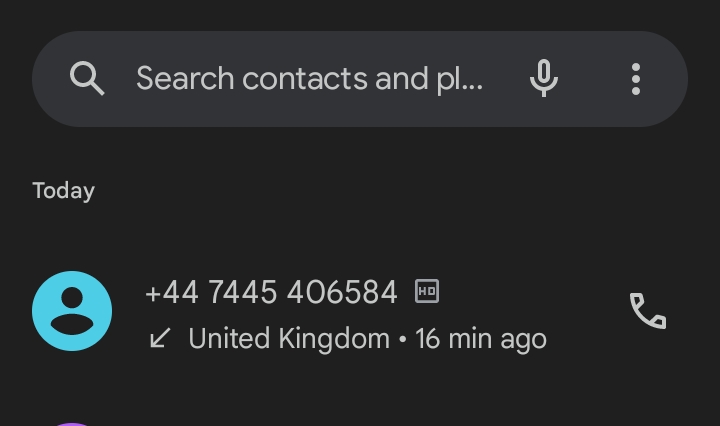
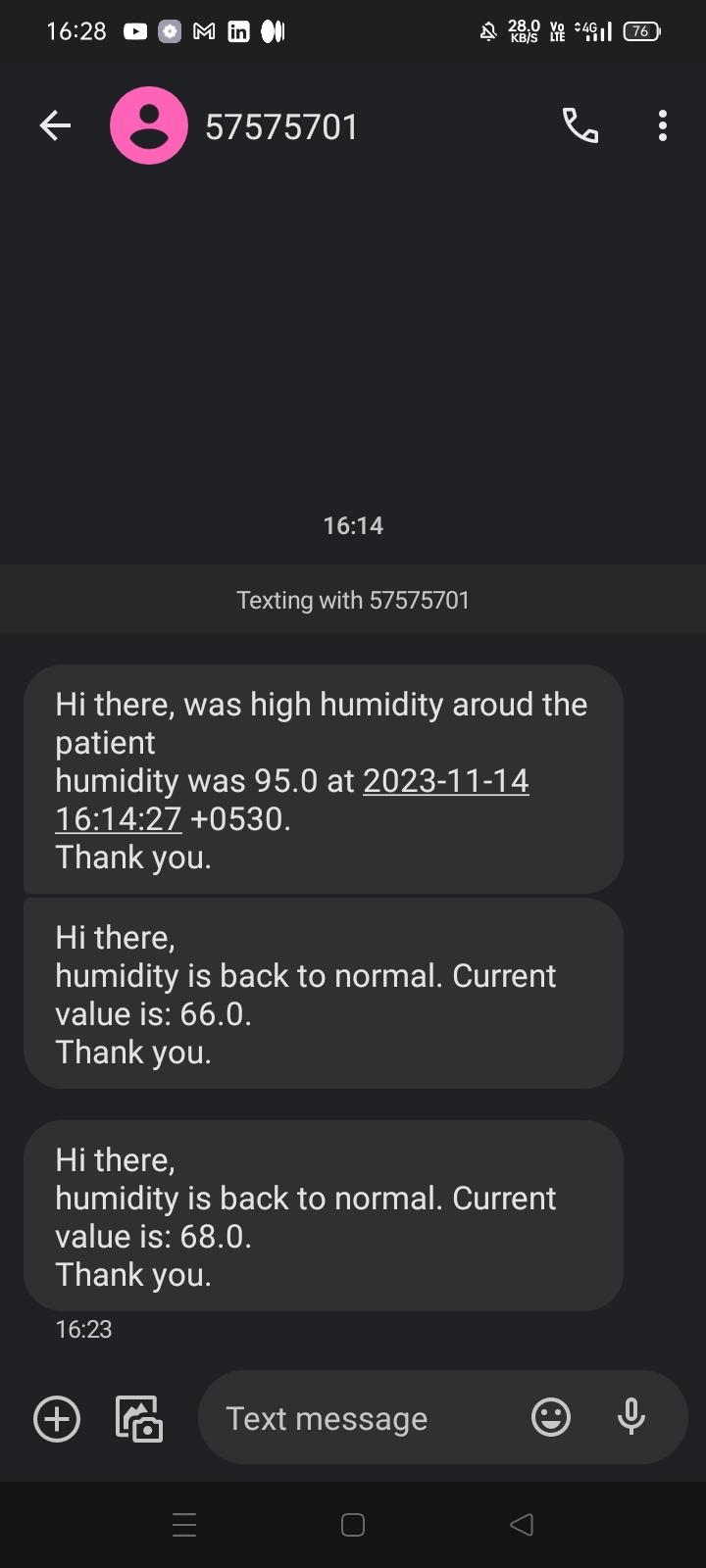


So, all 4 alerts were triggered and sent to the configured receiver. For demonstration, I have provided my mobile no, for real-time scenarios we can simply replace my no with hospital management, patient care takes, or patient loved ones.

Now, it’s time to test the “**back to the normal**” alert feature.

To make this Spo2 greater than 95% I am gone place my finger on top of the sensor, so it will send the exact readings of my body which will satisfy the event condition to trigger back to normal alert.

Yessss….. After 1 minute, we have received a voice call from ubidots again, amazing. likewise, we have received SMS alert, and Telegram message alert from the IoT notification, also we have received the Mail from notifications ubidots, with the Subject and mail body we have configured on the back to the normal event.



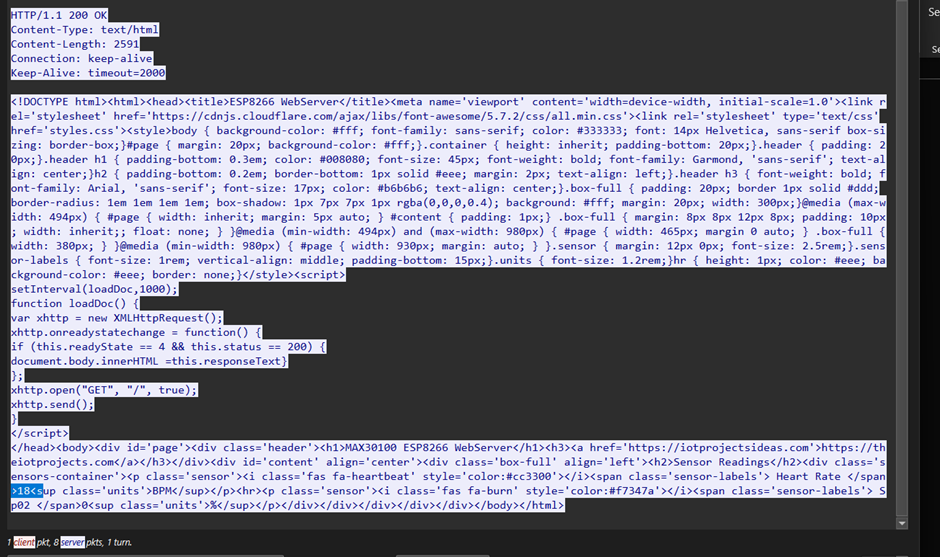
That’s all about developing IoT based Smart Patient Blood Oxygen meter with ESP8266 and sending critical alerts to the hospital management and patient loved ones.

NOW Security aspect’s

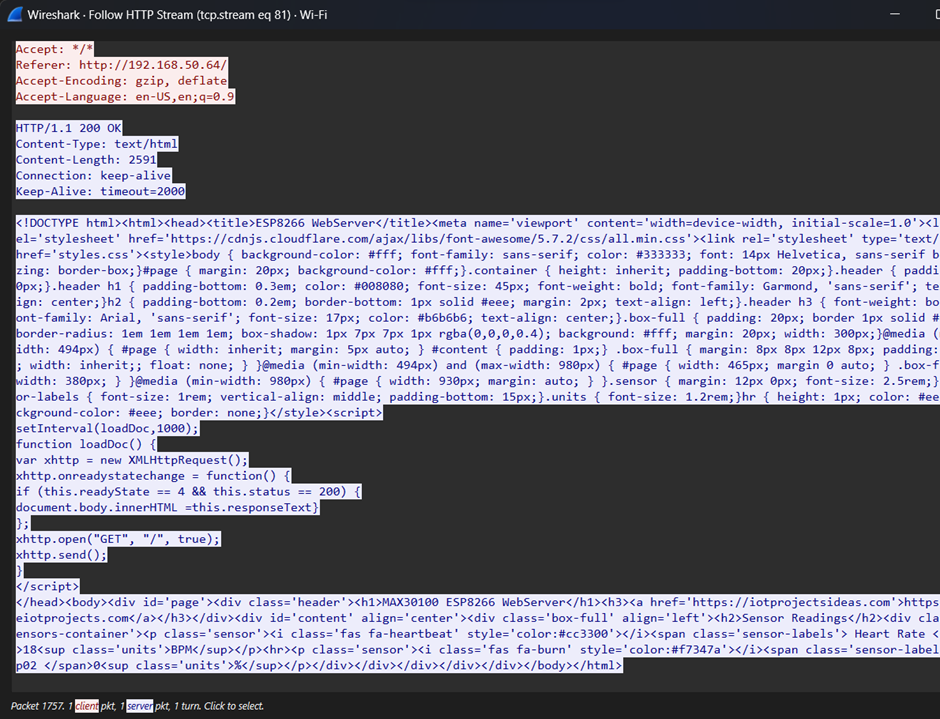
This type of project contains critical information so, here we can check the security in two way:

1. On esp8266 webserver
2. On cloud server

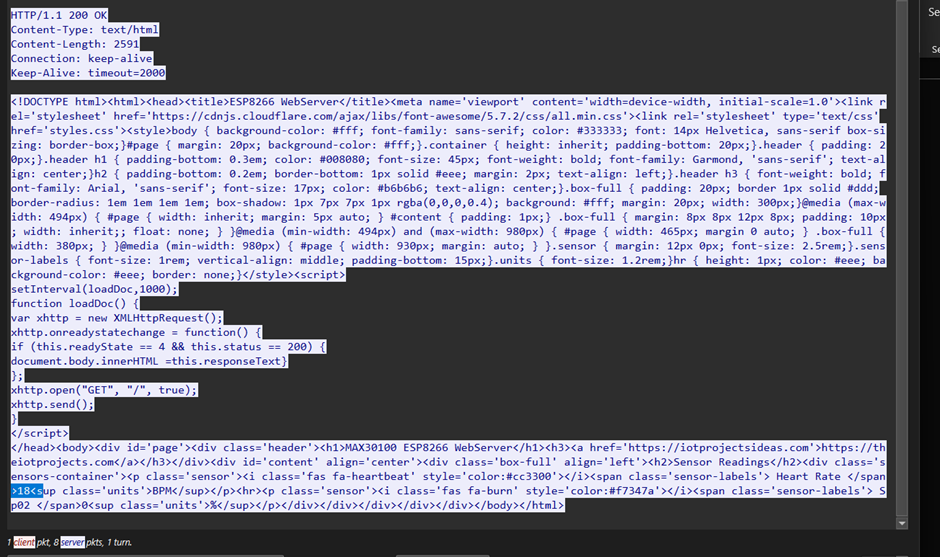
Esp8266 webserver:



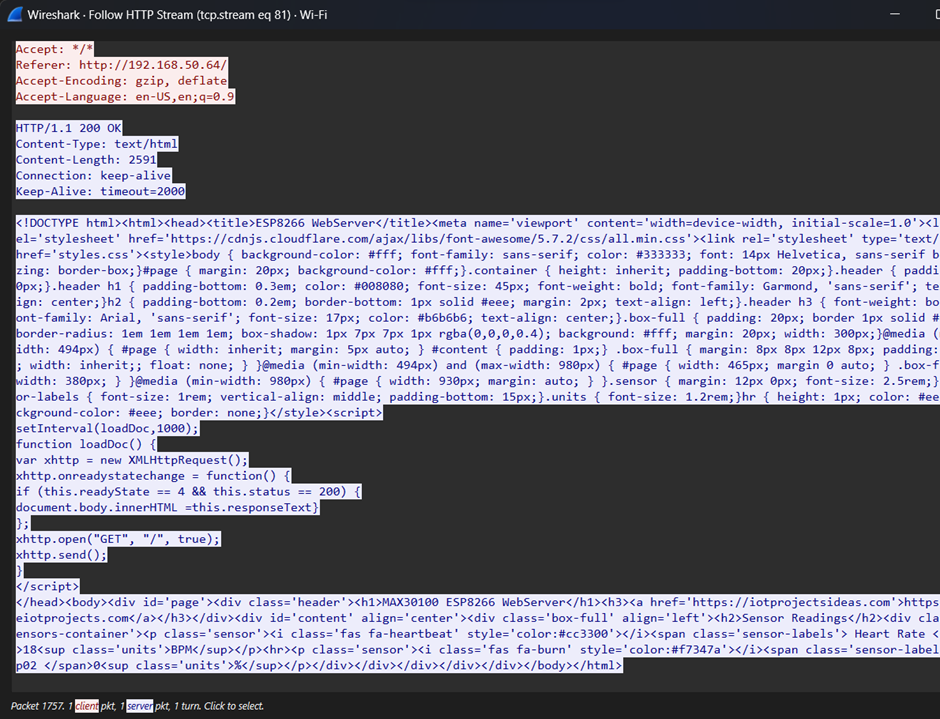
HTTP Stream:



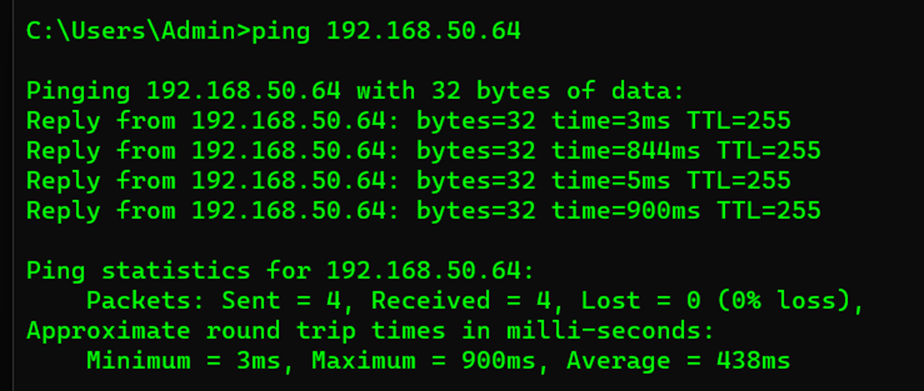
Esp8266 webserver:

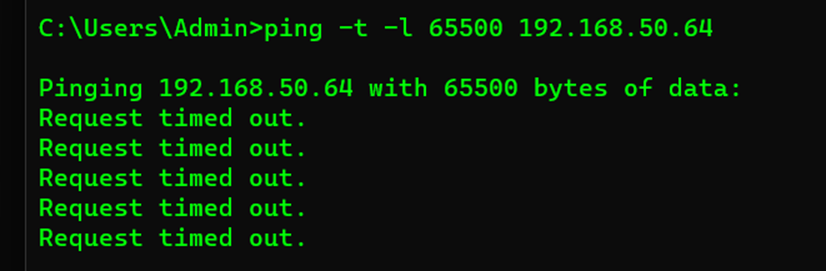


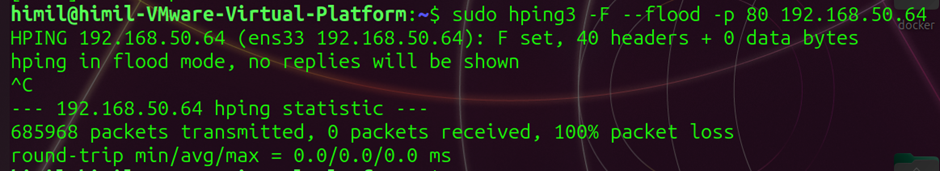
HTTP Stream



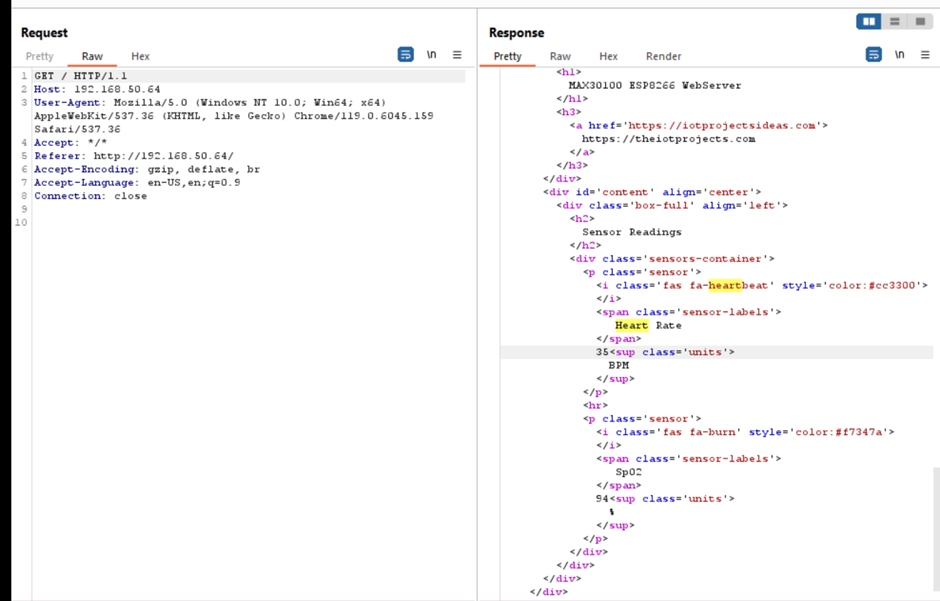
Ping check



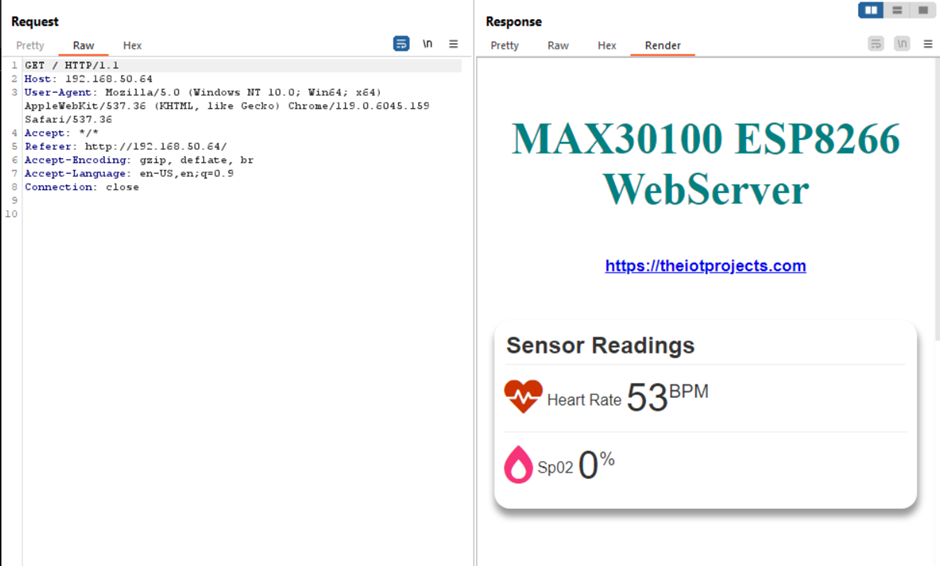




Burp suite

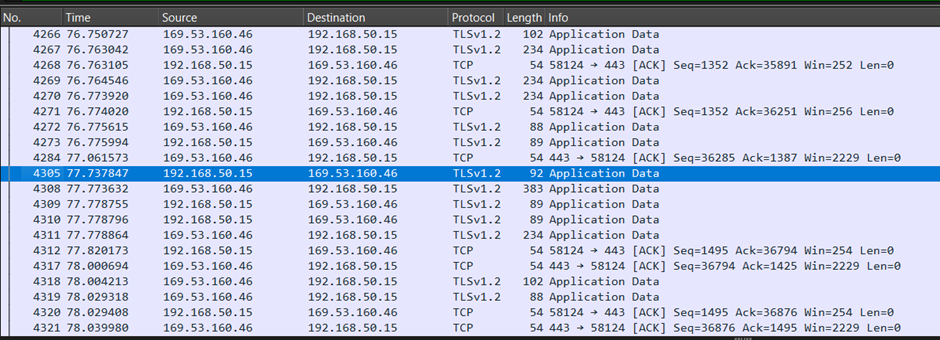


Render View:

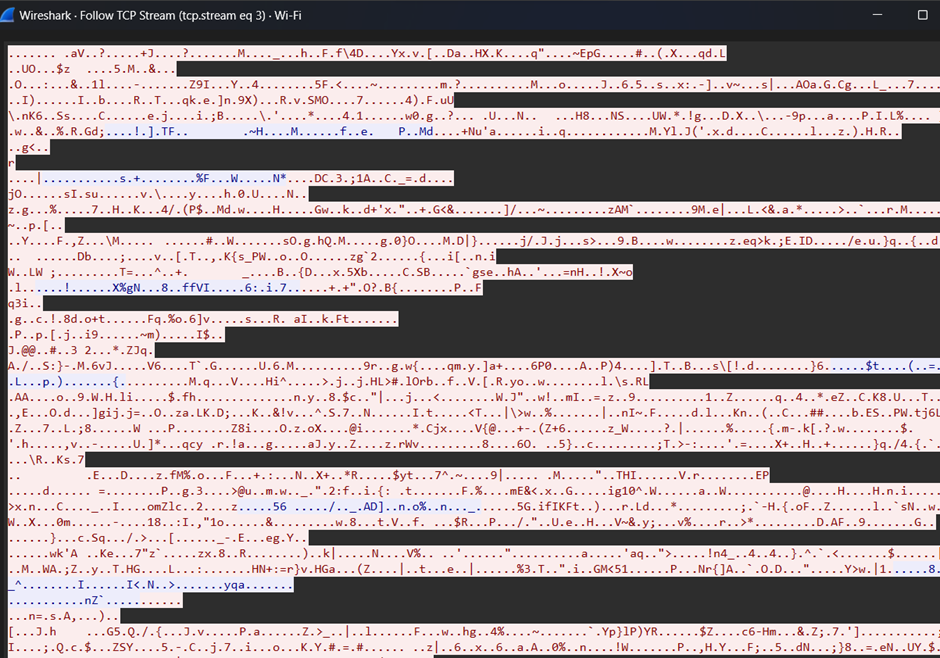


2. Cloud Server Security Check

Wireshark’s



TCP Strem Encrypted by TLS2.01 Protocol:



Burp suit:



Findings

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sro.no** | **Name** | **DATA VALUES (visible or not)** | **Encrypted** | **Ping(DDOS)** | **Manipulation** |
| 1. | Cloud integration | × | ✓ | × | × |
| 2. | ON ESP8266 Server | ✓ | × | ✓ | × |

Future Scope

* In future we can add the AI model into this so that AI can give you a predication of the human that he will suffering the heart base disease or not.
* This will help into the pandemic situation to monitor the patient remotely.

\* \* \*