



# **SNS COLLEGE OF TECHNOLOGY**

**Coimbatore-35**

**An Autonomous Institution**

Accredited by NBA – AICTE and Accredited by NAAC – UGC with  
'A++' Grade

Approved by AICTE, New Delhi & Affiliated to Anna University,  
Chennai



## **DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING**

### **19ECT213- IoT SYSTEM ARCHITECTURE**

II ECE / IV SEMESTER

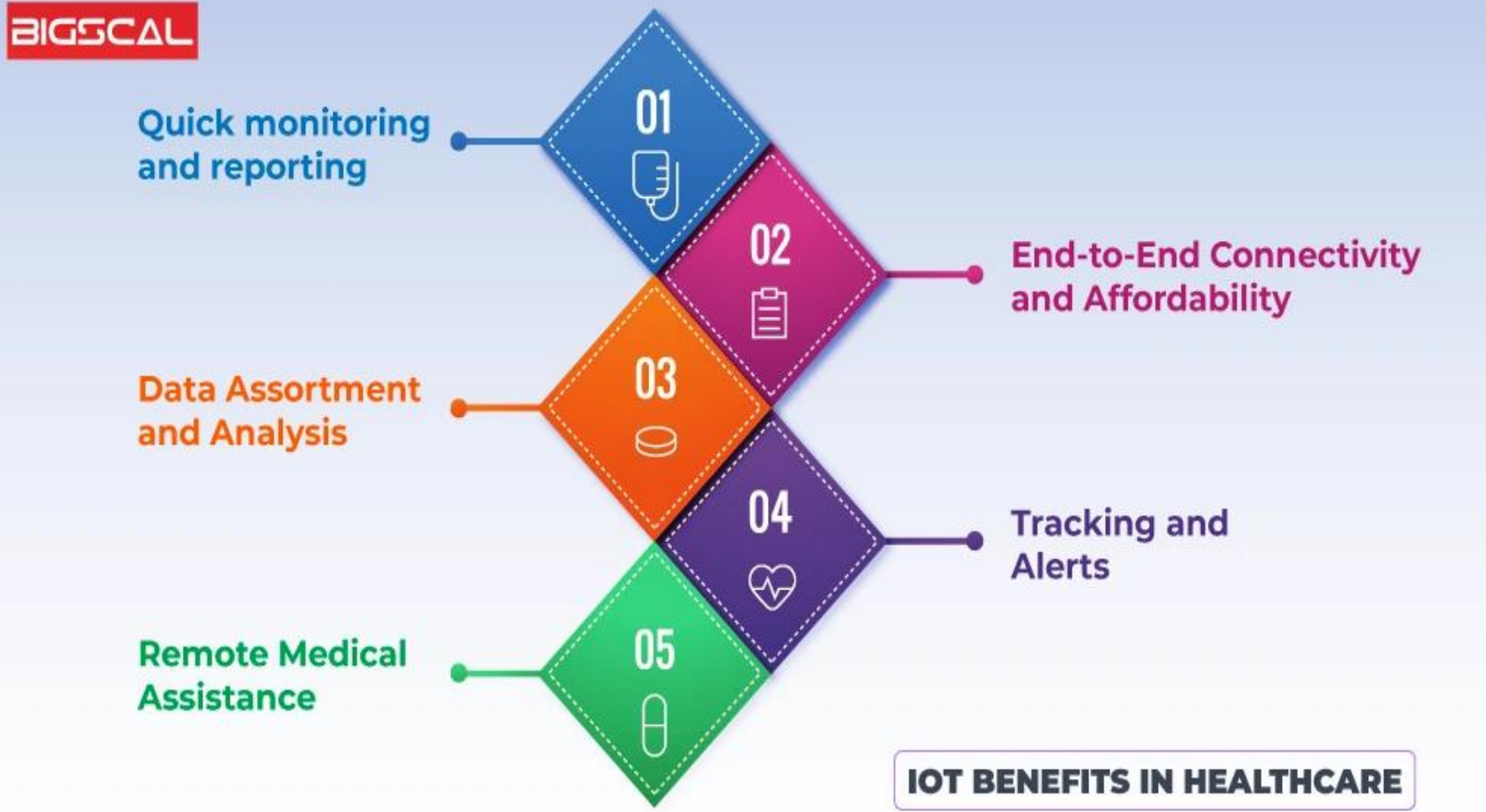
UNIT 1 – OVERVIEW OF INTERNET OF THINGS

**TOPIC 6 –IoT Applications- Healthcare**



# IOT Simple Healthcare System Architecture

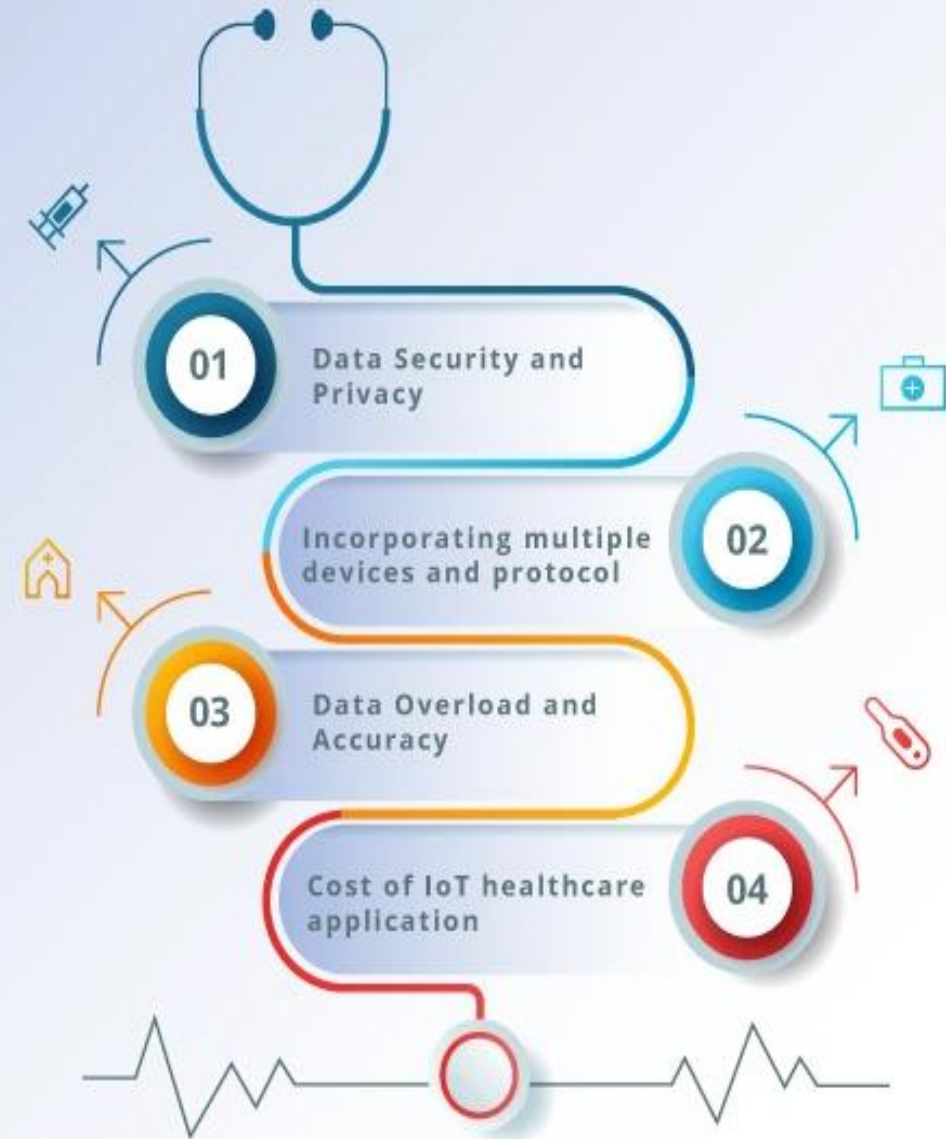




<https://www.bigscal.com/blogs/healthcare-industry/iot-in-healthcare-applications-benefits-challenges-and-future/>



# CHALLENGES OF IOT IN HEALTHCARE





SCAL

# IOT APPLICATIONS IN HEALTHCARE

HEARABLES



1



6

SMART VIDEO PILLS

INGESTIBLE SENSORS



2



5

HEALTHCARE CHARTING

MOODABLES



3



4

COMPUTER VISION TECHNOLOGY



## Hearable



## Chipsets

AI / Processor / Memory / DSP / PMIC  
Low power consumption  
Hi-Res Audio / Low Latency  
mmWave Hand-Gesture Radar  
mmWave distance detection Radar



## Antenna/PCB

High Efficiency Antenna  
mmWave Radar Antenna  
PCB / FPC Designs  
Crosstalk noise



## Passive Components

Resistor / Capacitor / Inductor  
Connector / LED indicator  
LED light source / Magnet  
Conductive Copper Post



## Passive Components

Traditional Speaker /  
MEMS Speaker /  
Microphone



(L)19.7mm  
(W)15.35mm  
(H)13.43mm



## Algorithm Cloud Service

Acoustics / AI Algorithms  
Sensing Device Algorithm  
Product Application Algorithm  
APP Software / Cloud Service

## Battery



Higher capacity battery with FAST charging  
/ Battery protection circuit

## Materials



Skin-friendly Material /  
Resonance cavity materials /  
Sound resonance cavity structure

## Mechanism



Airtight structure treatment  
/ wind noise resistance structure  
/ assembly consistency  
/ Ergonomics ID design  
/ Optical mechanism

## Sensors









Proximity / Touch / Force /  
IMUx6 axis / GMR / Gyroscope  
Temperature / PPG / SpO2 /  
Skin detector / .....



## Benefits of IoT in Healthcare



 <p>Cost savings</p>	 <p>Enhances the capabilities of preventive medicine</p>	 <p>Increased hospital staff versatility and alertness</p>
 <p>Drug and Medical Equipment Management</p>	 <p>Reducing Errors</p>	 <p>Patient data collection is sped up</p>





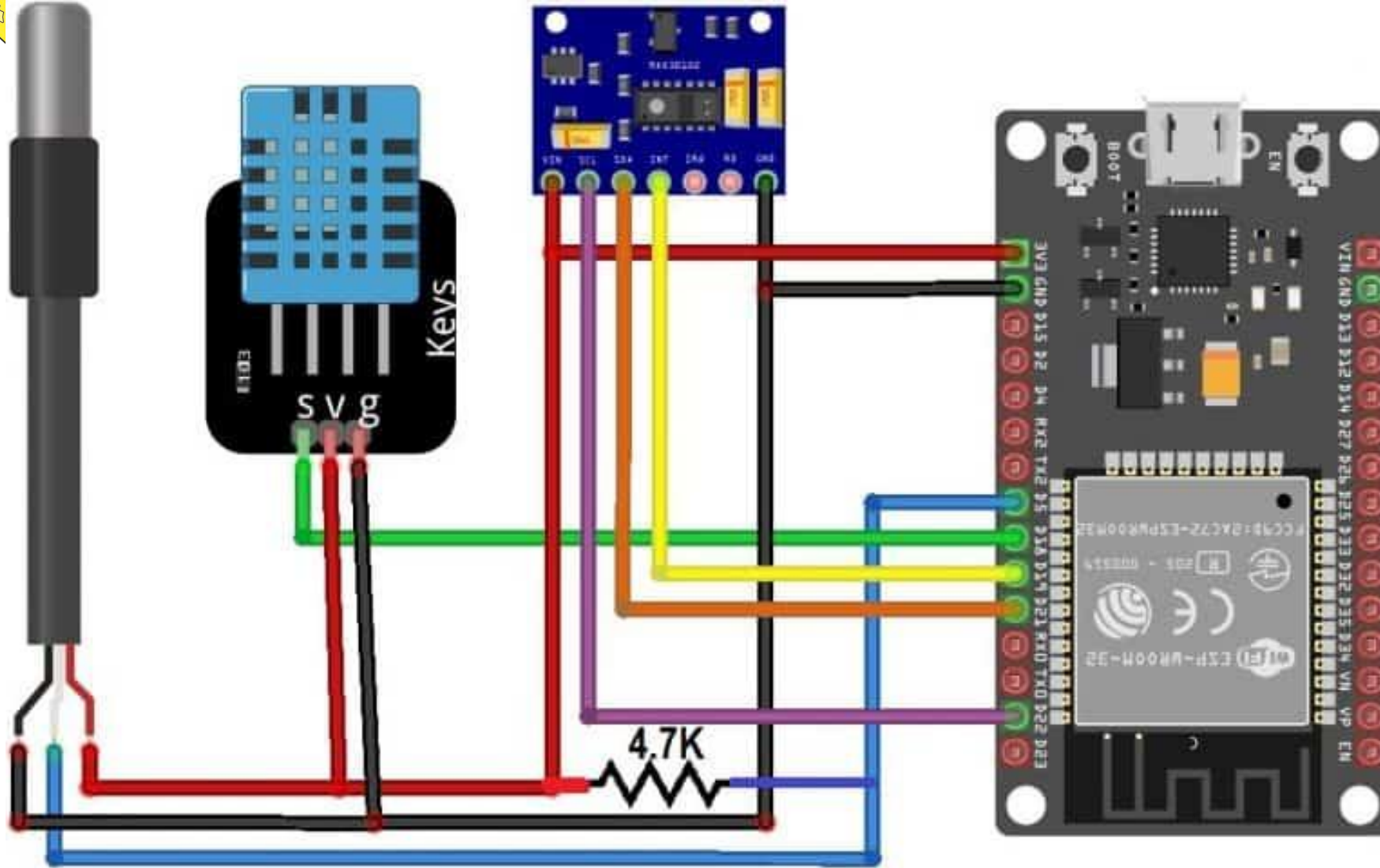
# IoT Based Patient Health Monitoring on ESP32 Web Server



S.N.	Components Name	Quantity
1	ESP32 Board	1
2	MAX30100 Pulse Oximeter Sensor	1
3	DS18B20 Sensor	1
4	DHT11 Sensor	1
5	Resistor 4.7K	1
6	Connecting Wires	10
7	Breadboard	1

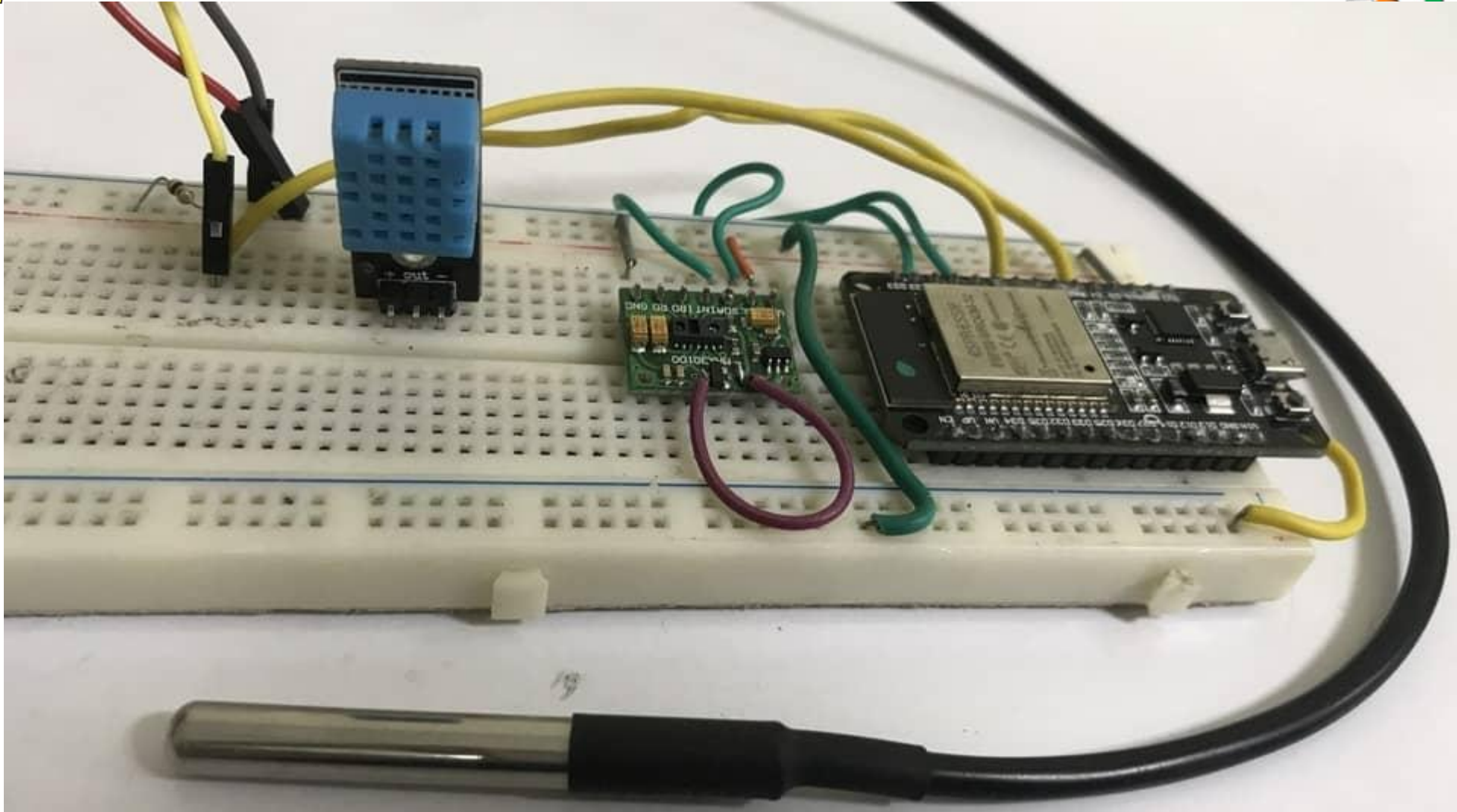
<https://how2electronics.com/iot-based-patient-health-monitoring-esp32-web-server/>

# IoT Based Patient Health Monitoring on ESP32 Web Server



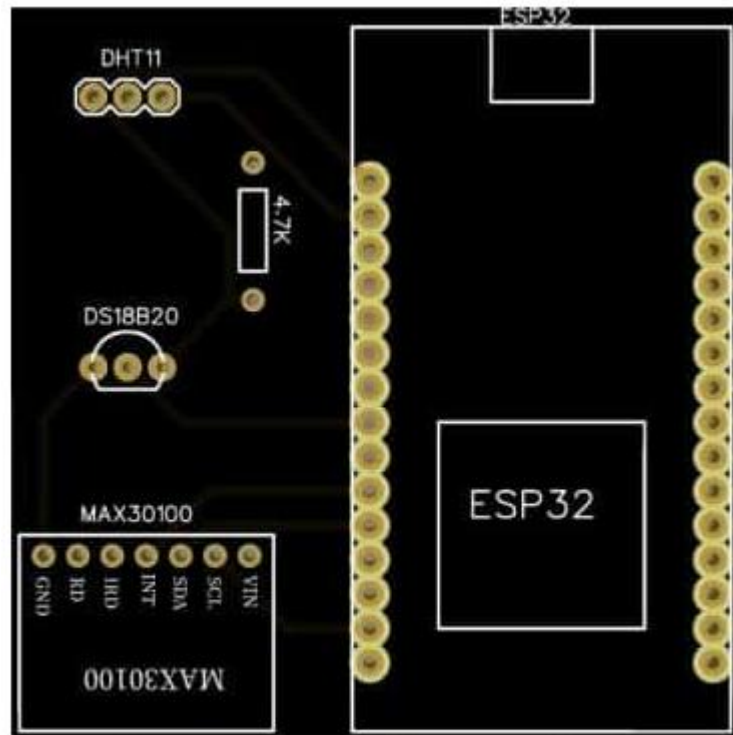


## IoT Based Patient Health Monitoring on ESP32 Web Server

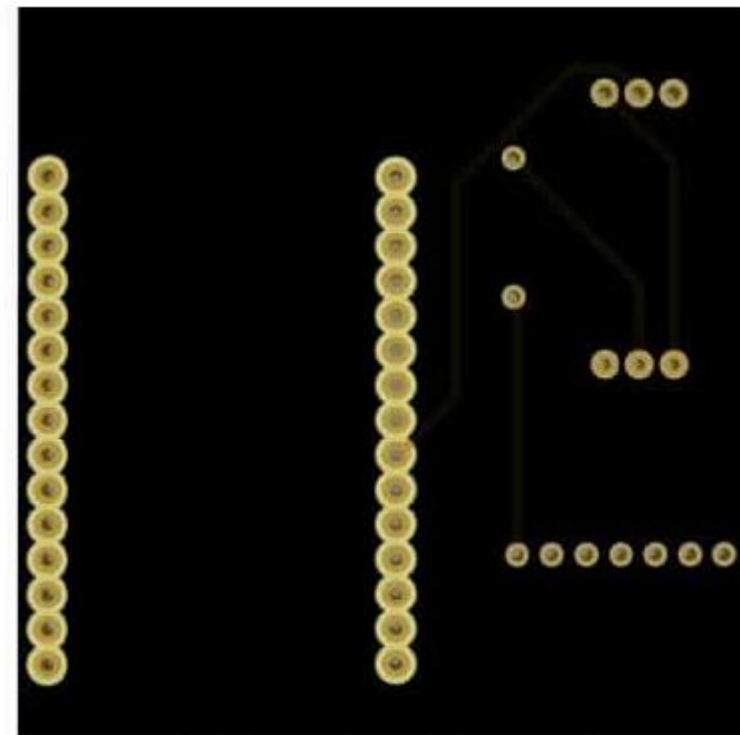




## PCB Designing, Gerber File & PCB Ordering



Front View



Back View



```
1 #include <WiFi.h>
2 #include <WebServer.h>
3 #include <Wire.h>
4 #include "MAX30100_PulseOximeter.h"
5 #include <OneWire.h>
6 #include <DallasTemperature.h>
7 #include <dht.h>
8
9 #define DHT11_PIN 18
10 #define DS18B20 5
11 #define REPORTING_PERIOD_MS    1000
12
13 float temperature, humidity, BPM, SpO2, bodytemperature;
14
15 /*Put your SSID & Password*/
16 const char* ssid = "Alexahome"; // Enter SSID here
17 const char* password = "12345678"; //Enter Password here
18
19 dht DHT;
20 PulseOximeter pox;
21 uint32_t tsLastReport = 0;
22 OneWire oneWire(DS18B20);
23 DallasTemperature sensors(&oneWire);
24
25
26 WebServer server(80);
27
28 void onBeatDetected()
29 {
30   Serial.println("Beat!");
31 }
```

```

void setup() {
  Serial.begin(115200);
  pinMode(19, OUTPUT);
  delay(100);

  Serial.println("Connecting to ");
  Serial.println(ssid);

  //connect to your local wi-fi network
  WiFi.begin(ssid, password);

  //check wi-fi is connected to wi-fi network
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected..!");
  Serial.print("Got IP: "); Serial.println(WiFi.localIP());

  server.on("/", handle_OnConnect);
  server.onNotFound(handle_NotFound);

  server.begin();
  Serial.println("HTTP server started");

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

```

```

if (!pox.begin()) {
  Serial.println("FAILED");
  for (;;)
} else {
  Serial.println("SUCCESS");
  pox.setOnBeatDetectedCallback(onBeatDetected);
}

pox.setIRLedCurrent(MAX30100_LED_CURR_7_6MA);

// Register a callback for the beat detection
}

void loop() {
  server.handleClient();
  pox.update();
  sensors.requestTemperatures();
  int chk = DHT.read11(DHT11_PIN);

  temperature = DHT.temperature;
  humidity = DHT.humidity;
  BPM = pox.getHeartRate();
  SpO2 = pox.getSpO2();
  bodytemperature = sensors.getTempCByIndex(0);

```

```

if (millis() - tsLastReport > REPORTING_PERIOD_MS)
{
  Serial.print("Room Temperature: ");
  Serial.print(DHT.temperature);
  Serial.println("°C");

  Serial.print("Room Humidity: ");
  Serial.print(DHT.humidity);
  Serial.println("%");

  Serial.print("BPM: ");
  Serial.println(BPM);

  Serial.print("SpO2: ");
  Serial.print(SpO2);
  Serial.println("%");

  Serial.print("Body Temperature: ");
  Serial.print(bodytemperature);
  Serial.println("°C");

  Serial.println("*****");
  Serial.println();

  tsLastReport = millis();
}

```

```

116 void handle_OnConnect() {
117
118   server.send(200, "text/html", SendHTML(temperature, humidity, BPM, SpO2, bodytemperat
119 }
120
121 void handle_NotFound(){
122   server.send(404, "text/plain", "Not found");
123 }
124
125 String SendHTML(float temperature, float humidity, float BPM, float SpO2, float bodytemp
126 String ptr = "<!DOCTYPE html>";
127 ptr += "<html>";
128 ptr += "<head>";
129 ptr += "<title>ESP32 Patient Health Monitoring</title>";
130 ptr += "<meta name='viewport' content='width=device-width, initial-scale=1.0'>";
131 ptr += "<link href='https://fonts.googleapis.com/css?family=Open+Sans:300,400,600' rel
132 ptr += "<style>";
133 ptr += "html { font-family: 'Open Sans', sans-serif; display: block; margin: 0px auto;
134 ptr += "body{margin: 0px;} ";
135 ptr += "h1 {margin: 50px auto 30px;} ";
136 ptr += ".side-by-side{display: table-cell;vertical-align: middle;position: relative;}";
137 ptr += ".text{font-weight: 600;font-size: 19px;width: 200px;}";
138 ptr += ".reading{font-weight: 300;font-size: 50px;padding-right: 25px;}";
139 ptr += ".temperature .reading{color: #F29C1F;}";
140 ptr += ".humidity .reading{color: #3B97D3;}";
141 ptr += ".BPM .reading{color: #FF0000;}";
142 ptr += ".SpO2 .reading{color: #955BA5;}";
143 ptr += ".bodytemperature .reading{color: #F29C1F;}";
144 ptr += ".superscript{font-size: 17px;font-weight: 600;position: absolute;top: 10px;}";
145 ptr += ".data{padding: 10px;}";
146 ptr += ".container{display: table;margin: 0 auto;}";
147 ptr += ".icon{width:65px}";
148 ptr += "</style>";

```

```
149 ptr += "</head>";
150 ptr += "<body>";
151 ptr += "<h1>ESP32 Patient Health Monitoring</h1>";
152 ptr += "<h3>www.how2electronics.com</h3>";
153 ptr += "<div class='container'>";
154
155 ptr += "<div class='data temperature'>";
156 ptr += "<div class='side-by-side icon'>";
157 ptr += "<svg enable-background='new 0 0 19.438 54.003'height=54.003px id=Layer_1 versi";
158 ptr += "C1.261,38.825,0,41.403,0,44.286c0,5.367,4.351,9.718,9.719,9.718c5.368,0,9.719-";
159 ptr += "c0-2.943-1.312-5.574-3.378-7.355V18.436h-3.914v-2h3.914v-2.808h-4.084v-2h4.084";
160 ptr += "c0,3.083-2.5,5.583-5.583,5.583s-5.583-2.5-5.583-5.583c0-2.279,1.368-4.236,3.32";
161 ptr += "s2.257,1.01,2.257,2.257V39.73C13.934,40.597,15.302,42.554,15.302,44.833z'fill=";
162 ptr += "</div>";
163 ptr += "<div class='side-by-side text'>Room Temperature</div>";
164 ptr += "<div class='side-by-side reading'>";
165 ptr += (int)temperature;
166 ptr += "<span class='superscript'>°C</span></div>";
167 ptr += "</div>";
168
169 ptr += "<div class='data humidity'>";
170 ptr += "<div class='side-by-side icon'>";
171 ptr += "<svg enable-background='new 0 0 29.235 40.64'height=40.64px id=Layer_1 version";
172 ptr += "C29.235,17.95,14.618,0,14.618,0z M13.667,37.135c-5.604,0-10.162-4.56-10.162-10";
173 ptr += "c0.787,0,1.425,0.639,1.425,1.426c0,4.031,3.28,7.312,7.311,7.312c0.787,0,1.425,";
174 ptr += "C15.093,36.497,14.455,37.135,13.667,37.135z'fill=#3C97D3 /></svg>";
175 ptr += "</div>";
176 ptr += "<div class='side-by-side text'>Room Humidity</div>";
```

```
177 ptr += "<div class='side-by-side reading'>";
178 ptr += (int)humidity;
179 ptr += "<span class='superscript'>%</span></div>";
180 ptr += "</div>";
181
182 ptr += "<div class='data Heart Rate'>";
183 ptr += "<div class='side-by-side icon'>";
184 ptr += "<svg enable-background='new 0 0 40.542 40.541'height=40.541px id=Layer_1 versi";
185 ptr += "c-0.195,0.195-0.451,0.293-0.707,0.293s-0.512-0.098-0.707-0.293c-0.391-0.391-0.";
186 ptr += "c-3.375-3.059-7.776-4.987-12.634-5.215c0.015,0.067,0.041,0.13,0.041,0.202v4.68";
187 ptr += "c0-0.071,0.026-0.134,0.041-0.202C14.39,0.279,9.936,2.256,6.544,6.38513.576,3.5";
188 ptr += "c-0.195,0.195-0.451,0.293-0.707,0.293s-0.512-0.098-0.707-0.293L5.142,6.812c-2.";
189 ptr += "c0.552,0,1,0.448,1,1s-0.448,1-1,1H0.05c0.525,10.728,9.362,19.271,20.22,19.271c";
190 ptr += "C34.76,21.271,34.313,20.823,34.313,20.271z M23.084,22.037c-0.559,1.561-2.274,2";
191 ptr += "c-1.561-0.557-2.373-2.272-1.815-3.833c0.372-1.041,1.263-1.737,2.277-1.928L25.2";
192 ptr += "C23.196,19.843,23.464,20.973,23.084,22.037z'fill=#26B999 /></g></svg>";
193 ptr += "</div>";
194 ptr += "<div class='side-by-side text'>Heart Rate</div>";
195 ptr += "<div class='side-by-side reading'>";
196 ptr += (int)BPM;
197 ptr += "<span class='superscript'>BPM</span></div>";
198 ptr += "</div>";
199
200 ptr += "<div class='data Blood Oxygen'>";
201 ptr += "<div class='side-by-side icon'>";
202 ptr += "<svg enable-background='new 0 0 58.422 40.639'height=40.639px id=Layer_1 versi";
203 ptr += "c-0.655,0-1.231,0.32-1.595,0.8081-0.011-0.0071-0.039,0.067c-0.021,0.03-0.035,0";
204 ptr += "c-0.149,0.28-0.242,0.594-0.242,0.934c0,1.102,0.894,1.995,1.994,1.995v0.015h31.";
205 ptr += "C58.422,38.323,58.339,38.024,58.203,37.754z'fill=#955BA5 /><path d='M19.704,38";
206 ptr += "c-0.84,0-1.582,0.41-2.051,1.0381-0.016-0.01L20.87,1.114c-0.025,0.039-0.046,0.0";
207 ptr += "C0.117,37.215,0,37.62,0,38.059c0,1.412,1.147,2.565,2.565,2.565v0.015h16.989c-0";
208 ptr += "C19.405,39.407,19.518,39.019,19.704,38.674z'fill=#955BA5 /></g></svg>";
209 ptr += "</div>";
210 ptr += "<div class='side-by-side text'>Blood Oxygen</div>";
211 ptr += "<div class='side-by-side reading'>";
```






```
210 ptr += "<div class='side-by-side text'>Blood Oxygen</div>";
211 ptr += "<div class='side-by-side reading'>";
212 ptr += (int) SpO2;
213 ptr += "<span class='superscript'>%</span></div>";
214 ptr += "</div>";
215
216 ptr += "<div class='data Body Temperature'>";
217 ptr += "<div class='side-by-side icon'>";
218 ptr += "<svg enable-background='new 0 0 19.438 54.003' height=54.003px id=Layer_1 versi
219 ptr += "C1.261,38.825,0,41.403,0,44.286c0,5.367,4.351,9.718,9.719,9.718c5.368,0,9.719-
220 ptr += "c0-2.943-1.312-5.574-3.378-7.355V18.436h-3.914v-2h3.914v-2.808h-4.084v-2h4.084
221 ptr += "c0,3.083-2.5,5.583-5.583,5.583s-5.583-2.5-5.583-5.583c0-2.279,1.368-4.236,3.32
222 ptr += "s2.257,1.01,2.257,2.257V39.73C13.934,40.597,15.302,42.554,15.302,44.833z' fill=
223 ptr += "</div>";
224 ptr += "<div class='side-by-side text'>Body Temperature</div>";
225 ptr += "<div class='side-by-side reading'>";
226 ptr += (int) bodytemperature;
227 ptr += "<span class='superscript'>°C</span></div>";
228 ptr += "</div>";
229
230 ptr += "</div>";
231 ptr += "</body>";
232 ptr += "</html>";
233 return ptr;
234 }
```

# ESP32 Patient Health Monitoring

www.how2electronics.com

 Room Temperature 26°C

 Room Humidity 90%

 Heart Rate 91 BPM

 Blood Oxygen 96%

 Body Temperature 36°C

```
ESP32_Patient
1 #include <Arduino.h>
2 #include <Wire.h>
3 #include <Adafruit_BME280.h>
4 #include <Adafruit_SHT40.h>
5 #include <Arduino.h>
6 #include <Adafruit_BME280.h>
7 #include <Adafruit_SHT40.h>
8
9 #define BME280_I2C_ADDR 0x76
10 #define SHT40_I2C_ADDR 0x70
11 #define BME280_I2C_ADDR 0x76
12 #define SHT40_I2C_ADDR 0x70
13 #define SHT40_I2C_ADDR 0x70
14 #define SHT40_I2C_ADDR 0x70
15 #define SHT40_I2C_ADDR 0x70
16 #define SHT40_I2C_ADDR 0x70
17 #define SHT40_I2C_ADDR 0x70
18
19 #define SHT40_I2C_ADDR 0x70
20 #define SHT40_I2C_ADDR 0x70
21 #define SHT40_I2C_ADDR 0x70
22 #define SHT40_I2C_ADDR 0x70
23 #define SHT40_I2C_ADDR 0x70
24 #define SHT40_I2C_ADDR 0x70
25 #define SHT40_I2C_ADDR 0x70
26 #define SHT40_I2C_ADDR 0x70
27 #define SHT40_I2C_ADDR 0x70
28 #define SHT40_I2C_ADDR 0x70
29 #define SHT40_I2C_ADDR 0x70
30 #define SHT40_I2C_ADDR 0x70
31 #define SHT40_I2C_ADDR 0x70
32 #define SHT40_I2C_ADDR 0x70
33 #define SHT40_I2C_ADDR 0x70
34 #define SHT40_I2C_ADDR 0x70
35 #define SHT40_I2C_ADDR 0x70
36 #define SHT40_I2C_ADDR 0x70
37 #define SHT40_I2C_ADDR 0x70
38 #define SHT40_I2C_ADDR 0x70
39 #define SHT40_I2C_ADDR 0x70
40 #define SHT40_I2C_ADDR 0x70
41 #define SHT40_I2C_ADDR 0x70
42 #define SHT40_I2C_ADDR 0x70
43 #define SHT40_I2C_ADDR 0x70
44 #define SHT40_I2C_ADDR 0x70
45 #define SHT40_I2C_ADDR 0x70
46 #define SHT40_I2C_ADDR 0x70
47 #define SHT40_I2C_ADDR 0x70
48 #define SHT40_I2C_ADDR 0x70
49 #define SHT40_I2C_ADDR 0x70
50 #define SHT40_I2C_ADDR 0x70
51 #define SHT40_I2C_ADDR 0x70
52 #define SHT40_I2C_ADDR 0x70
53 #define SHT40_I2C_ADDR 0x70
54 #define SHT40_I2C_ADDR 0x70
55 #define SHT40_I2C_ADDR 0x70
56 #define SHT40_I2C_ADDR 0x70
57 #define SHT40_I2C_ADDR 0x70
58 #define SHT40_I2C_ADDR 0x70
59 #define SHT40_I2C_ADDR 0x70
60 #define SHT40_I2C_ADDR 0x70
61 #define SHT40_I2C_ADDR 0x70
62 #define SHT40_I2C_ADDR 0x70
63 #define SHT40_I2C_ADDR 0x70
64 #define SHT40_I2C_ADDR 0x70
65 #define SHT40_I2C_ADDR 0x70
66 #define SHT40_I2C_ADDR 0x70
67 #define SHT40_I2C_ADDR 0x70
68 #define SHT40_I2C_ADDR 0x70
69 #define SHT40_I2C_ADDR 0x70
70 #define SHT40_I2C_ADDR 0x70
71 #define SHT40_I2C_ADDR 0x70
72 #define SHT40_I2C_ADDR 0x70
73 #define SHT40_I2C_ADDR 0x70
74 #define SHT40_I2C_ADDR 0x70
75 #define SHT40_I2C_ADDR 0x70
76 #define SHT40_I2C_ADDR 0x70
77 #define SHT40_I2C_ADDR 0x70
78 #define SHT40_I2C_ADDR 0x70
79 #define SHT40_I2C_ADDR 0x70
80 #define SHT40_I2C_ADDR 0x70
81 #define SHT40_I2C_ADDR 0x70
82 #define SHT40_I2C_ADDR 0x70
83 #define SHT40_I2C_ADDR 0x70
84 #define SHT40_I2C_ADDR 0x70
85 #define SHT40_I2C_ADDR 0x70
86 #define SHT40_I2C_ADDR 0x70
87 #define SHT40_I2C_ADDR 0x70
88 #define SHT40_I2C_ADDR 0x70
89 #define SHT40_I2C_ADDR 0x70
90 #define SHT40_I2C_ADDR 0x70
91 #define SHT40_I2C_ADDR 0x70
92 #define SHT40_I2C_ADDR 0x70
93 #define SHT40_I2C_ADDR 0x70
94 #define SHT40_I2C_ADDR 0x70
95 #define SHT40_I2C_ADDR 0x70
96 #define SHT40_I2C_ADDR 0x70
97 #define SHT40_I2C_ADDR 0x70
98 #define SHT40_I2C_ADDR 0x70
99 #define SHT40_I2C_ADDR 0x70
100 #define SHT40_I2C_ADDR 0x70
```