

# TEAM 20

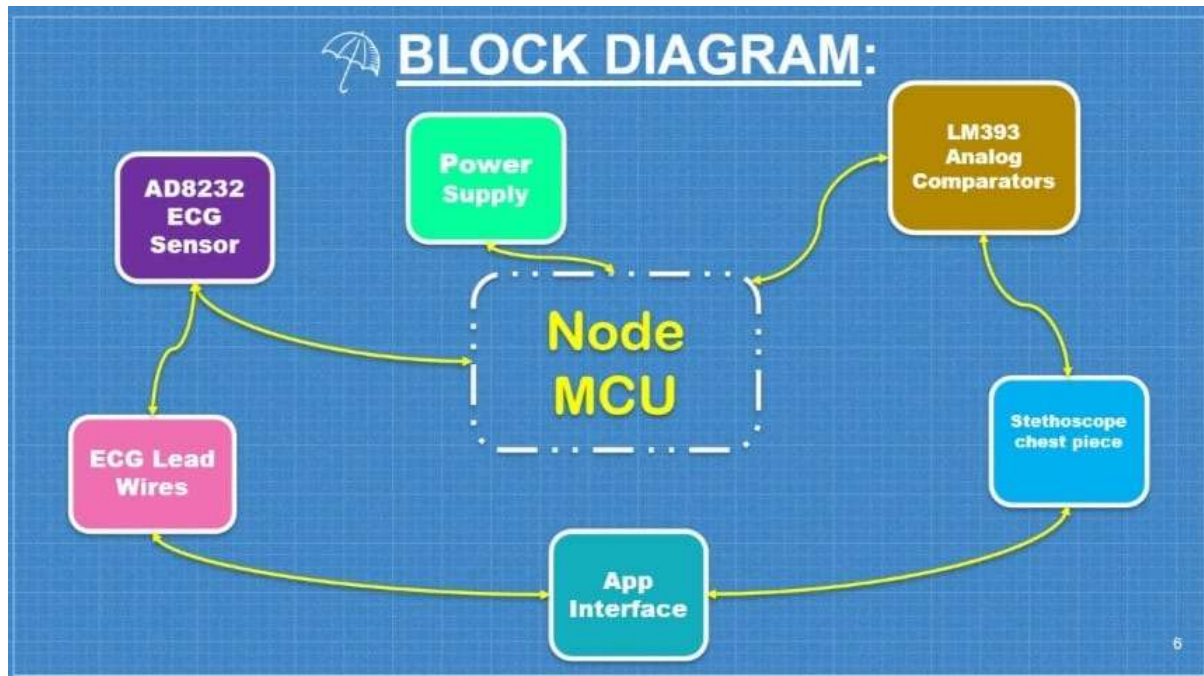
## Remote Health Monitoring System

### Project Report

- **Group member details:**

PAMIDI MOHAMMAD ASHRAF:	S20210020303
SANDEEP NIDAMANURI:	S20210020300
PRAVEEN KUMAR:	S20210020313
J MOHIT NAGA VARDHA GANESH:	S20210020283

## Final Block Diagram:



## COMPONENTS:

### HARDWARE

LM393 Analog Comparators

Breadboard

Stethoscope chest piece

Medical grade tube

Jumper Wires

Node MCU

AD8232 ECG Sensor

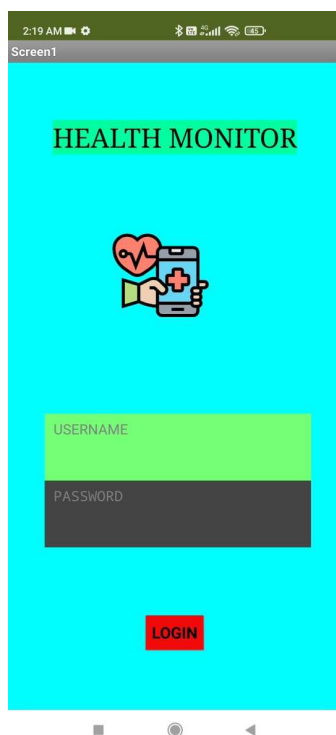
ECG Lead Wires

## SOFTWARE

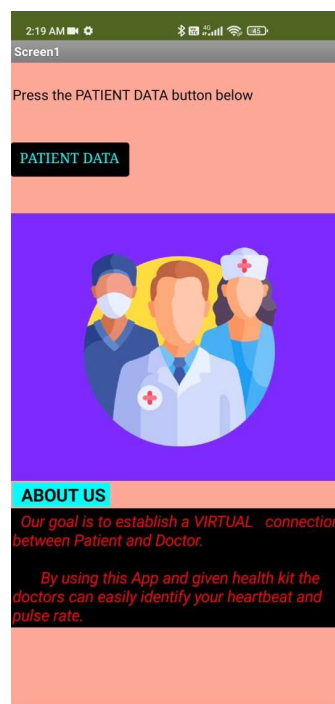
- Arduino IDE
- MIT App Interface/Web framework interface
- Firebase

## Outputs:

### App Outcomes:



Login Page



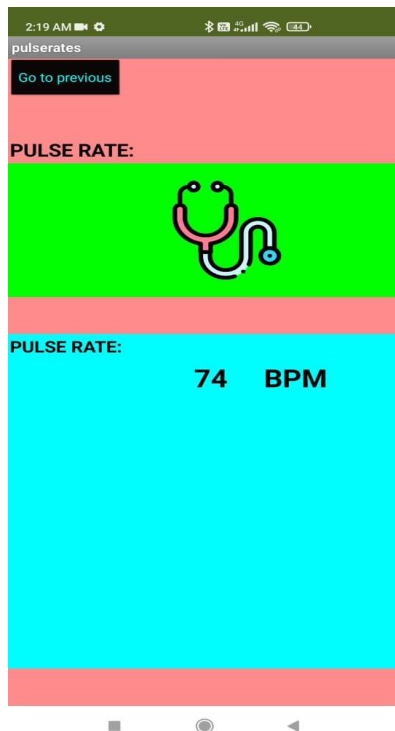
Introduction Page



ECG VALUES



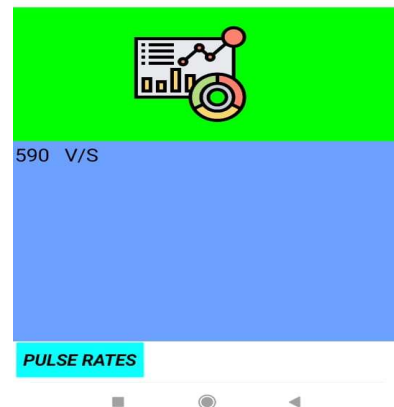
ECG plotting



Pulse Rate



ECG VALUES

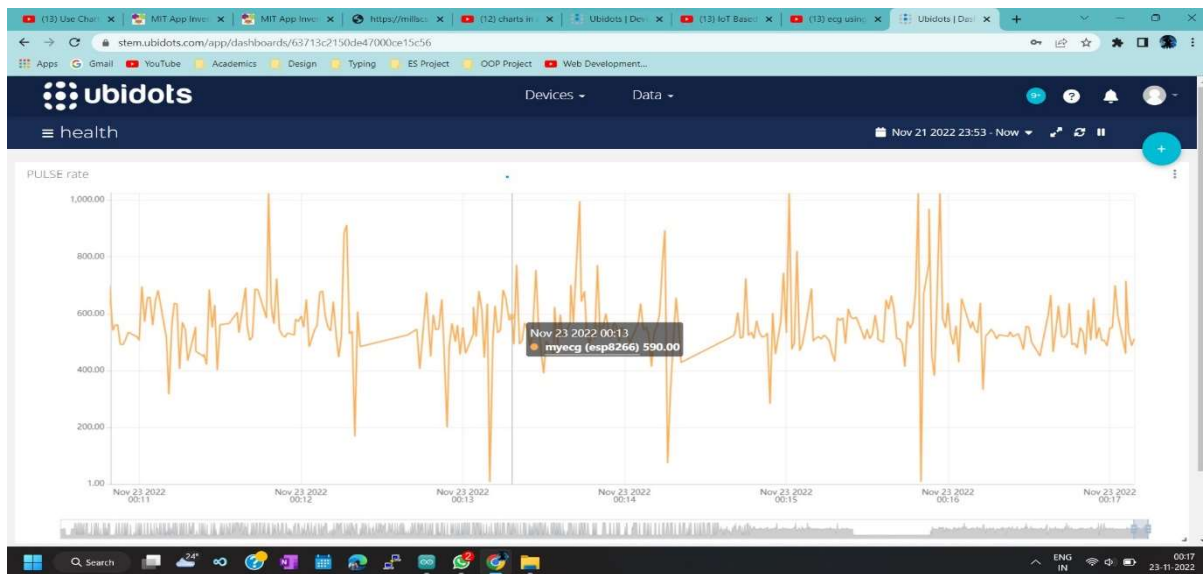


ECG values

## ECG of candidate 1:



## ECG of candidate 2:



## Pulse rate of the candidate:

The screenshot shows the Arduino IDE interface with the 'BPM.ino' file open. The code is written in C++ and includes the following components:

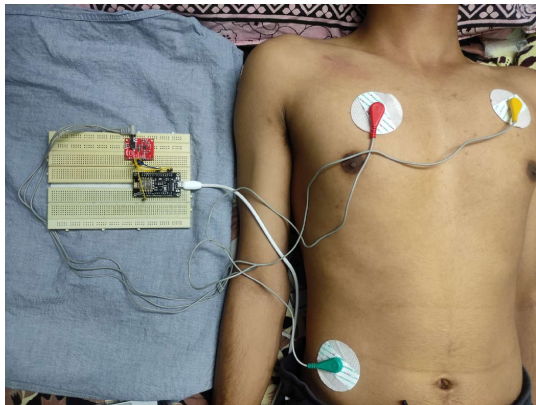
```
1 #include <FirebaseESP8266.h>
2 #include <ESP8266WiFi.h>
3
4 #define FIREBASE_HOST "remote-health-monitor-bia62-default-rtdb.firebaseio.com"
5 #define WIFI_SSID "ashraf794" // Change the name of your WIFI
6 #define WIFI_PASSWORD "ashraf123" // Change the password of your WIFI
7 #define FIREBASE_Authorization_key "9q1V5sX0fxTETB2d1uAHS34y0eUlwFKkpj3xts"
8 FirebaseData firebaseData;
9 FirebaseJson json;
10
11 void setup()
12 {
13   Serial.begin(115200);
14   WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
15   Serial.print("Connecting...");
16
17   while (WiFi.status() != WL_CONNECTED)
18   {
19     // Loop until connected
20   }
21 }
```

The 'Output' window at the bottom shows the following log messages:

```
02:17:13.698 -> Publishing data to Ubidots Cloud
02:17:13.698 -> 512.00
02:17:14.213 -> Publishing data to Ubidots Cloud
02:17:14.213 -> 624.00
02:17:14.710 -> Publishing data to Ubidots Cloud
02:17:14.710 -> 520.00
02:17:15.258 -> Publishing data to Ubidots Cloud
02:17:15.258 -> 480.00
02:17:15.764 -> Publishing data to Ubidots Cloud
02:17:15.764 -> 561.00
02:17:16.280 -> Publishing data to Ubidots Cloud
02:17:16.280 -> 1024.00
02:17:16.822 -> Publishing data to Ubidots Cloud
```

The bottom status bar of the IDE indicates 'Ln 18, Col 4 - UTF-8 - NodeMCU 1.0 (ESP-12E Module) [not connected]' and shows the system clock as '03:25' and '24-11-2022'.

## Project Models:



Remote ECG



Remote Stethoscope

## **Challenges faced:**

- First setting up the project using Arduino UNO made a difficulty to adjust the disturbances and noise then we implemented the whole project with the node MCU.
- Getting the accurate sound waves reducing the noise was a bit of a challenge.
- Plotting the ECG graph was a little challenge since we don't have enough resources.
- And finally integrating all the circuits made a bit difficult.

## Codes:

---

### ➤ For the Stethoscope:

```
#include <FirebaseESP8266.h>
#include <ESP8266WiFi.h>

#define FIREBASE_HOST "remote-health-monitor-b1a62-default-
rtdb.firebaseio.com"
#define WIFI_SSID "ashraf754" // Change the name of your WIFI
#define WIFI_PASSWORD "ashraf123" // Change the password of your WIFI
#define FIREBASE_Authorization_key "9qlV5sX0fxTETB2dJuAuhS34yD0UlwFKKpjJJxtS"
FirebaseData firebaseData;
FirebaseJson json;

void setup()
{
  Serial.begin(115200);
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting...");

  while (WiFi.status() != WL_CONNECTED)
  {
    Serial.print(".");
    delay(300);
  }

  Serial.println();
  Serial.print("IP Address: ");
  Serial.println(WiFi.localIP());
  Serial.println();
  Firebase.begin(FIREBASE_HOST, FIREBASE_Authorization_key);
}

void loop() {
  int bpm = analogRead(A0);
  int noise = 100;
  // float bpm = random(75, 82);
  if (isnan(bpm))
  {
    Serial.println(F("Failed to read from DHT sensor!"));
    return;
  }
  bpm = (bpm - noise)/10;
```

```

    Serial.print(" BPM: ");
    Serial.println(bpm);
    Firebase.setFloat(firebaseData, "BPM", bpm);
    delay(100);
}

```

## ➤ For the ECG:

```

#include <ESP8266WiFi.h>
#include <FirebaseESP8266.h>
#include <PubSubClient.h>

#define FIREBASE_HOST "remote-health-monitor-b1a62-default-
rtdb.firebaseio.com"
#define WIFISSID "ashraf754" // Put your WifiSSID here
#define PASSWORD "ashraf123" // Put your wifi password
here
#define TOKEN "BBFF-cyPG0c5hADPAeRt5VvDXd96doGQpI8" // Ashraf API code:
"BBFF-Fa5BhZKZ0wnRUbCxBKrJm9WMSt3sqD" // Put your Ubidots' TOKEN
#define MQTT_CLIENT_NAME "myecgsensor" // MQTT client Name,
please enter your own 8-12 alphanumeric character ASCII string; \
//it should be a random
and unique ascii string and different from all other devices
#define FIREBASE_Authorization_key "9q1V5sX0fxTETB2dJuAuhS34yD0ULWFKKpjJJxtS"
FirebaseData firebaseData;
FirebaseJson json;

/*****
* Define Constants
*****/
#define VARIABLE_LABEL "myecg" // Assing the variable label
#define DEVICE_LABEL "esp8266" // Assig the device label

#define SENSOR A0 // Set the A0 as SENSOR

char mqttBroker[] = "industrial.api.ubidots.com";
char payload[100];
char topic[150];
// Space to store values to send
char str_sensor[10];

/*****
* Auxiliar Functions
*****/
WiFiClient ubidots;
PubSubClient client(ubidots);

```



```

void callback(char* topic, byte* payload, unsigned int length) {
    char p[length + 1];
    memcpy(p, payload, length);
    p[length] = NULL;
    Serial.write(payload, length);
    Serial.println(topic);
}

void reconnect() {
    // Loop until we're reconnected
    while (!client.connected()) {
        Serial.println("Attempting MQTT connection...");

        // Attempt to connect
        if (client.connect(MQTT_CLIENT_NAME, TOKEN, "")) {
            Serial.println("Connected");
        } else {
            Serial.print("Failed, rc=");
            Serial.print(client.state());
            Serial.println(" try again in 2 seconds");
            // Wait 2 seconds before retrying
            delay(2000);
        }
    }
}

/*****
 * Main Functions
 *****/

void setup() {
    Serial.begin(115200);
    WiFi.begin(WIFISSID, PASSWORD);
    // Assign the pin as INPUT
    pinMode(SENSOR, INPUT);

    Serial.println();
    Serial.print("Waiting for WiFi...");

    while (WiFi.status() != WL_CONNECTED) {
        Serial.print(".");
        delay(500);
    }

    Serial.println("");
    Serial.println("WiFi Connected");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());
    client.setServer(mqttBroker, 1883);
}

```

```

    client.setCallback(callback);
    Firebase.begin(FIREBASE_HOST, FIREBASE_Authorization_key);

}

void loop() {
    if (!client.connected()) {
        reconnect();
    }

    sprintf(topic, "%s%s", "/v1.6/devices/", DEVICE_LABEL);
    sprintf(payload, "%s", ""); // Cleans the payload
    sprintf(payload, "{\"%s\":\"", VARIABLE_LABEL); // Adds the variable label

    float myecg = analogRead(SENSOR);

    /* 4 is minimum width, 2 is precision; float value is copied onto
    str_sensor*/
    dtostrf(myecg, 4, 2, str_sensor);

    sprintf(payload, "%s {\"value\": %s}}", payload, str_sensor); // Adds the
value
    Serial.println("Publishing data to Ubidots Cloud");
    Serial.println(myecg);
    client.publish(topic, payload);
    Firebase.setFloat(firebaseData, "ECG", myecg);
    client.loop();
}

```

## ➤ Block Code for the APP:

The image displays three screenshots of the App Inventor web interface, showing the block code for three different screens: Screen1, patientsdata, and pulserates.

**Screen1:** The code includes three event-driven blocks. The first block, triggered by LOGINbtn.Click, calls FirebaseDB1.GetValue with tag 'TextBox1.Text' and valueIfTagNotThere 'NA'. The second block, triggered by Signupbtn.Click, calls FirebaseDB1.StoreValue with tag 'TextBox1.Text' and valueToStore 'PasswordTextBox1.Text'. The third block, triggered by FirebaseDB1.DataChanged, contains an if-then-else structure. The if condition checks if the tag is 'TextBox1.Text' and the value is 'PasswordTextBox1.Text'. If true, it sets VerticalArrangement1.Visible to false and VerticalArrangement2.Visible to true. If false, it sets Label2.Text to 'NA'. A 'Show Warnings' button is visible at the bottom left.

**patientsdata:** The code includes two event-driven blocks. The first block, triggered by FirebaseDB1.DataChanged, contains an if-then structure. The if condition checks if the tag is 'ECG'. If true, it sets ECG.Text to the value of a join block that concatenates 'V/S' and the value of a variable. The second block, triggered by pulse\_screen.Click, opens another screen named 'pulserates'.

**pulserates:** The code includes two event-driven blocks. The first block, triggered by FirebaseDB1.DataChanged, contains an if-then structure. The if condition checks if the tag is 'BPM'. If true, it sets Label6.Text to the value of a join block that concatenates 'BPM' and the value of a variable. The second block, triggered by Button1.Click, opens another screen named 'patientsdata'.

