TEAM 20

**Remote Health Monitoring System**

Project Report

* **Group member details:**

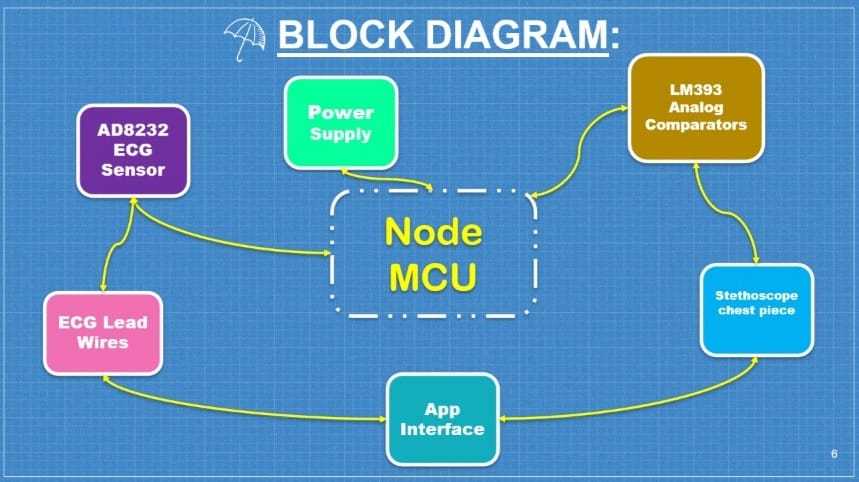
PAMIDI MOHAMMAD ASHRAF: S20210020303

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

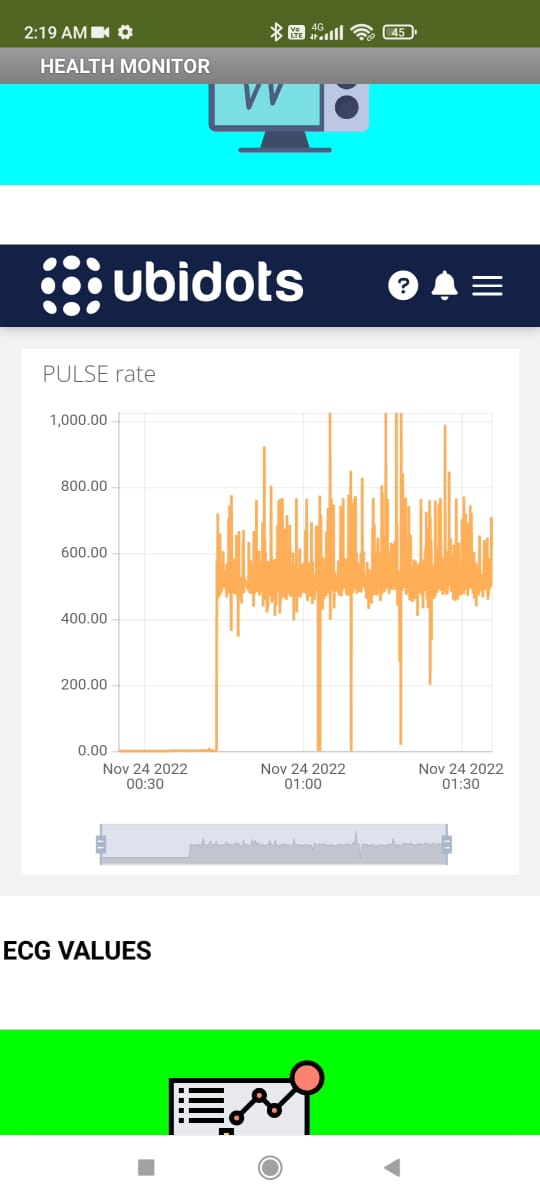
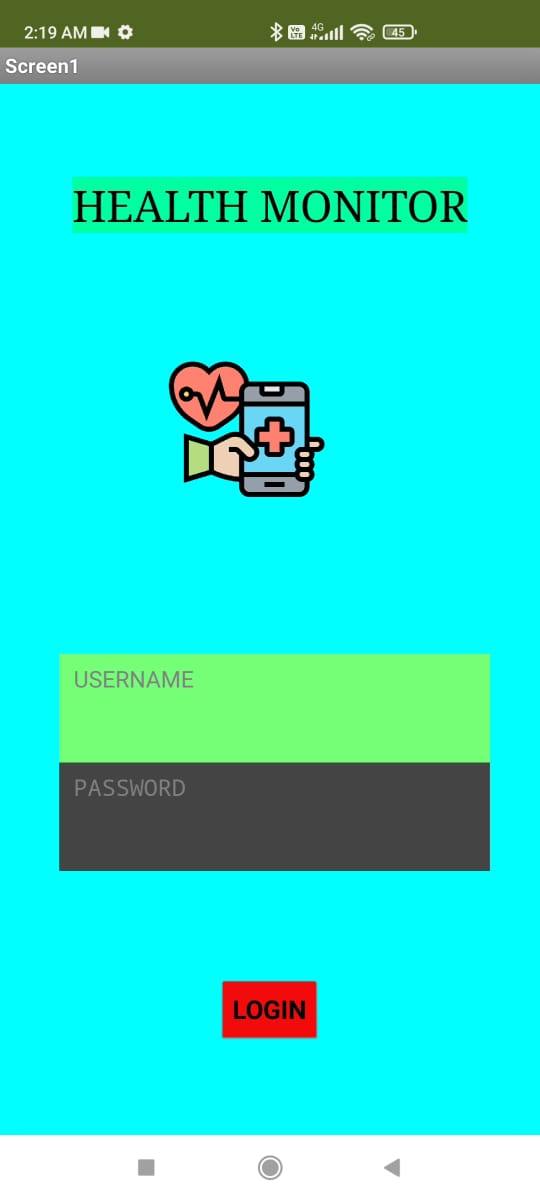
ECG Lead Wires

SOFTWARE

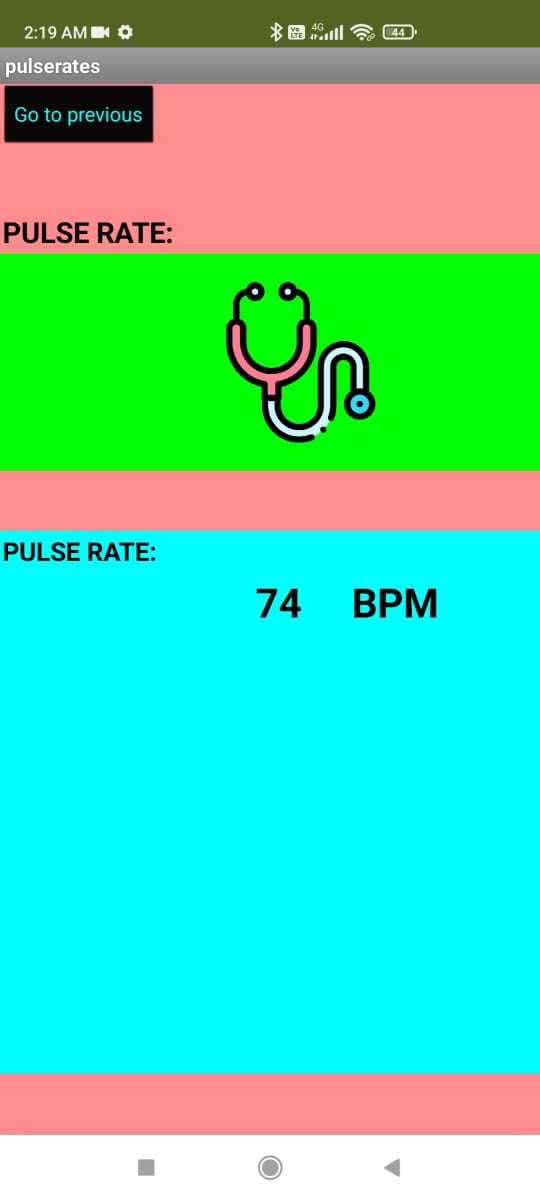
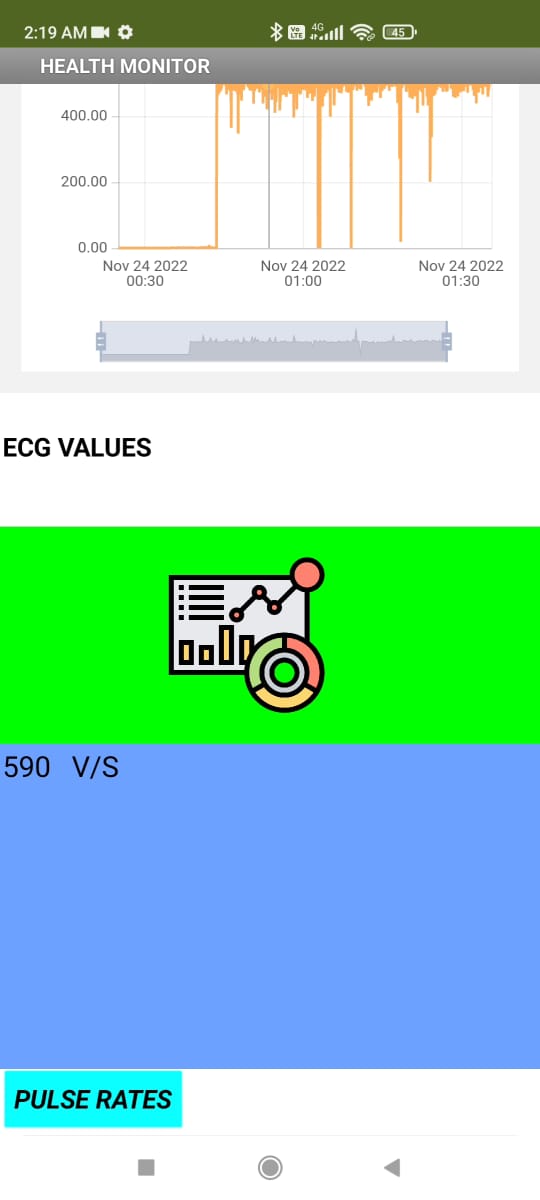
* Arduino IDE
* MIT App Interface/Web framework interface
* Firebase

**Outputs:**

App Outcomes:

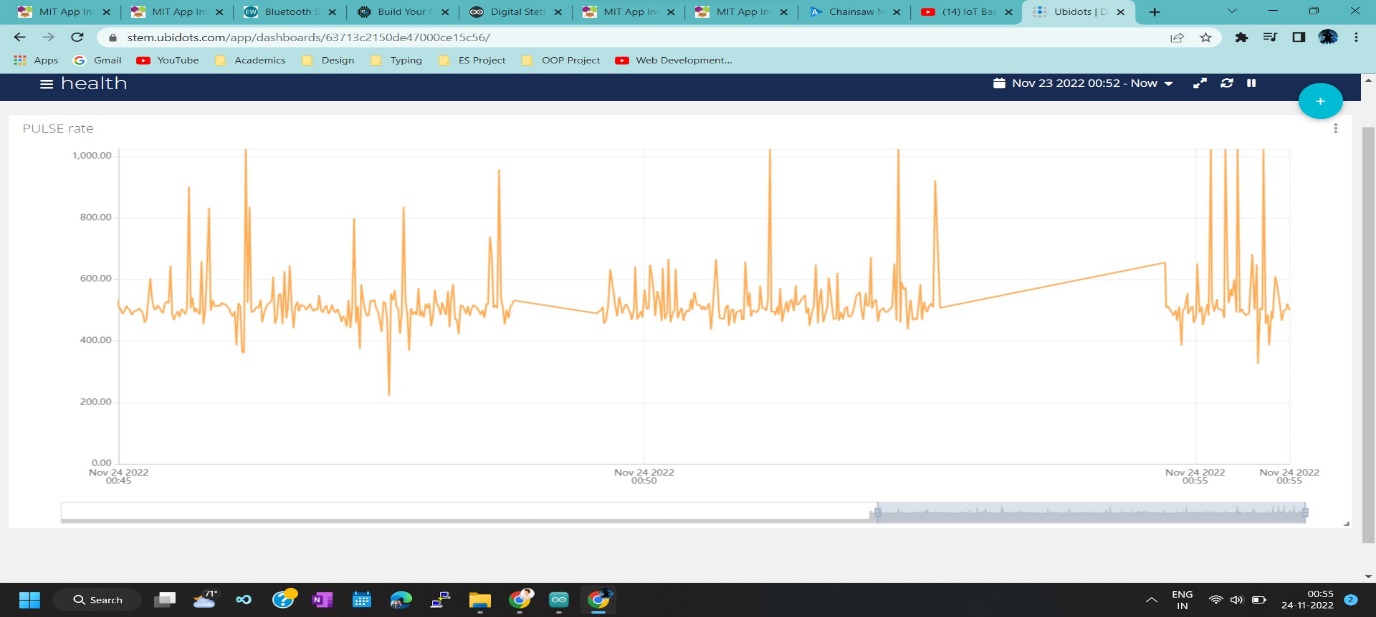


Login Page Introduction Page ECG plotting



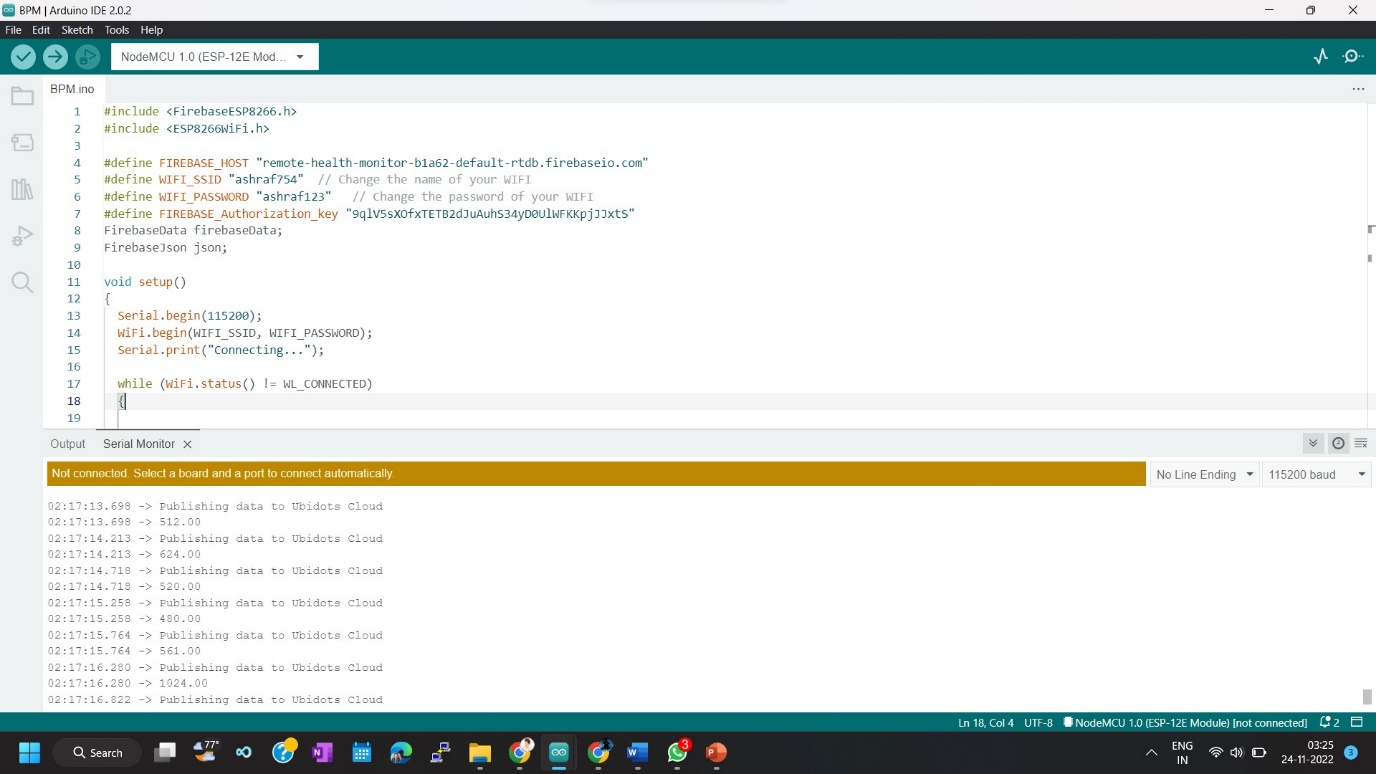
Pulse Rate ECG values

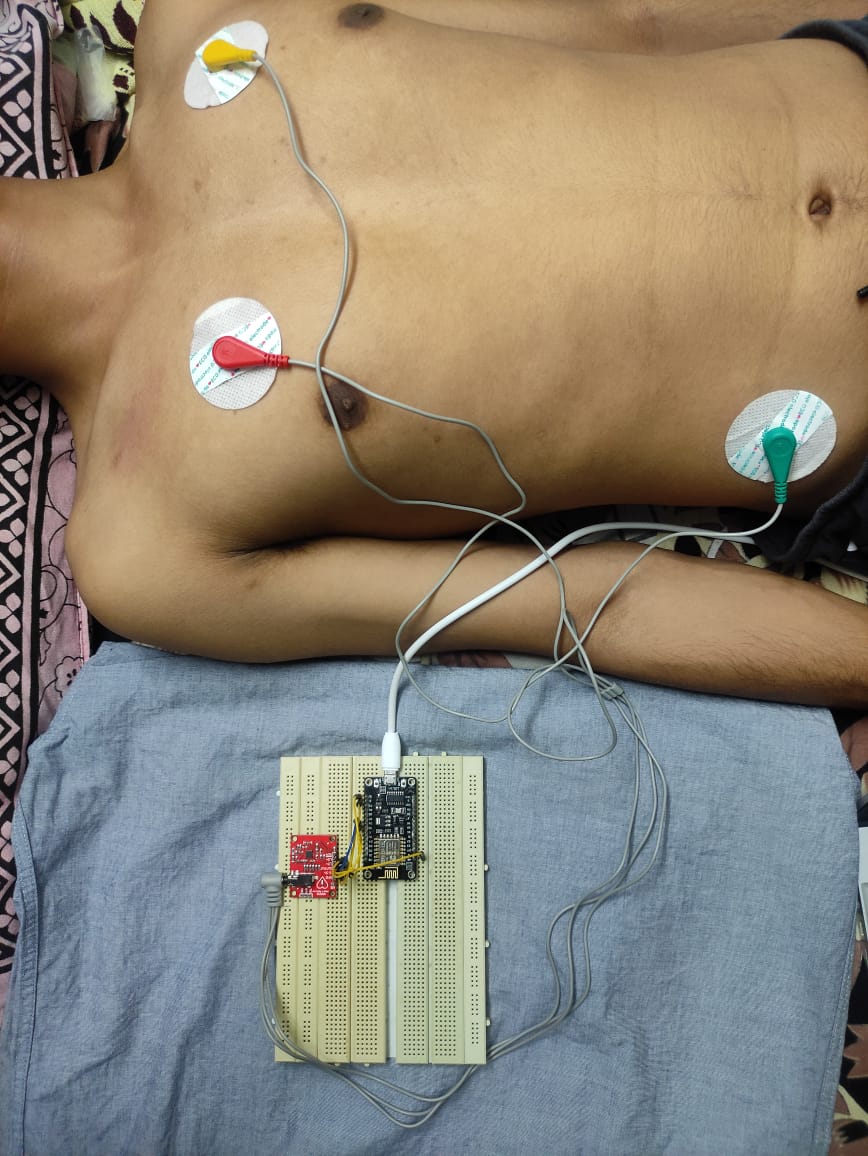
ECG of candidate 1:



ECG of candidate 2:

Pulse rate of the candidate:



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 "9qlV5sXOfxTETB2dJuAuhS34yD0UlWFKKpjJJxtS"

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-cyPGOc5hADPAeRt5VvDXd96doGQpI8"  // 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 "9qlV5sXOfxTETB2dJuAuhS34yD0UlWFKKpjJJxtS"

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...");

    // Attemp 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 mininum 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:**

