IoT Based Heart Monitoring System Using ECG

Name - Pooja

Roll no – 204161012

Affiliation – IIT Guwahati, MTech Data Science

Project Report

a. Main Objective:

Heart diseases are becoming a big issue for the last few decades and many people die because of certain health problems. Therefore, heart disease cannot be taken lightly. So, there should be a technology that can monitor the heart rate and heart behavior of the patient regularly. By analyzing or monitoring the **ECG signal** at the initial stage the various heart disease can be prevented. This is the reason why I am presenting you with this **IoT project**.

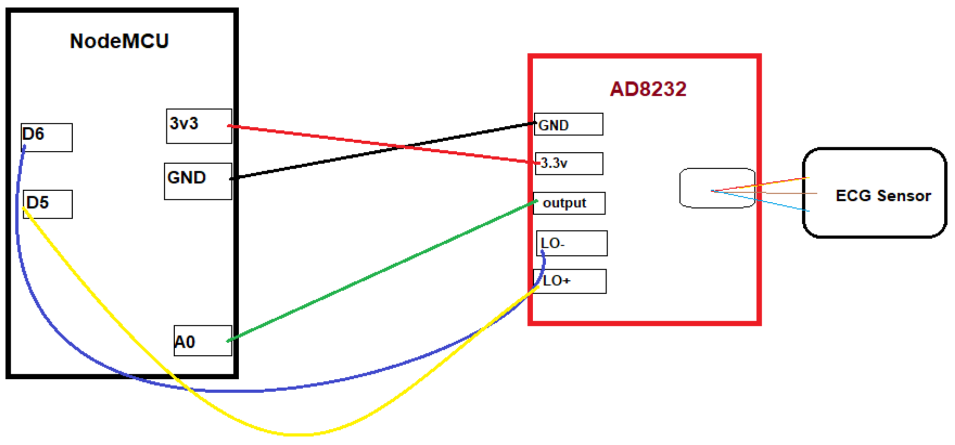
In this project, I will show you how we can interface **AD8232 ECG Sensor with NodeMCU ESP8266** Board and monitor the **ECG Waveform** on Serial Plotter Screen. Similarly, we can send the ECG waveform over the IoT Cloud platform and monitor the signal online from any part of the world using the PC or simply using the Smartphone. There is no need for staying in the Hospital to monitor heart activity/behavior just because you can monitor it online from anywhere. Thus, it can be said it is advancement in **Patient Health Monitoring System**.

Ubidots: The IoT platform that I am going to use here is **Ubidots**. Ubidots is an IoT Platform empowering innovators and industries to prototype and scale **IoT projects** to production. I am using the Ubidots platform to send data to the cloud from any Internet-enabled device.

b. Implemented Attributes:

I have used two attributes to show the output:
1). Line Chart
2). Gauge widget

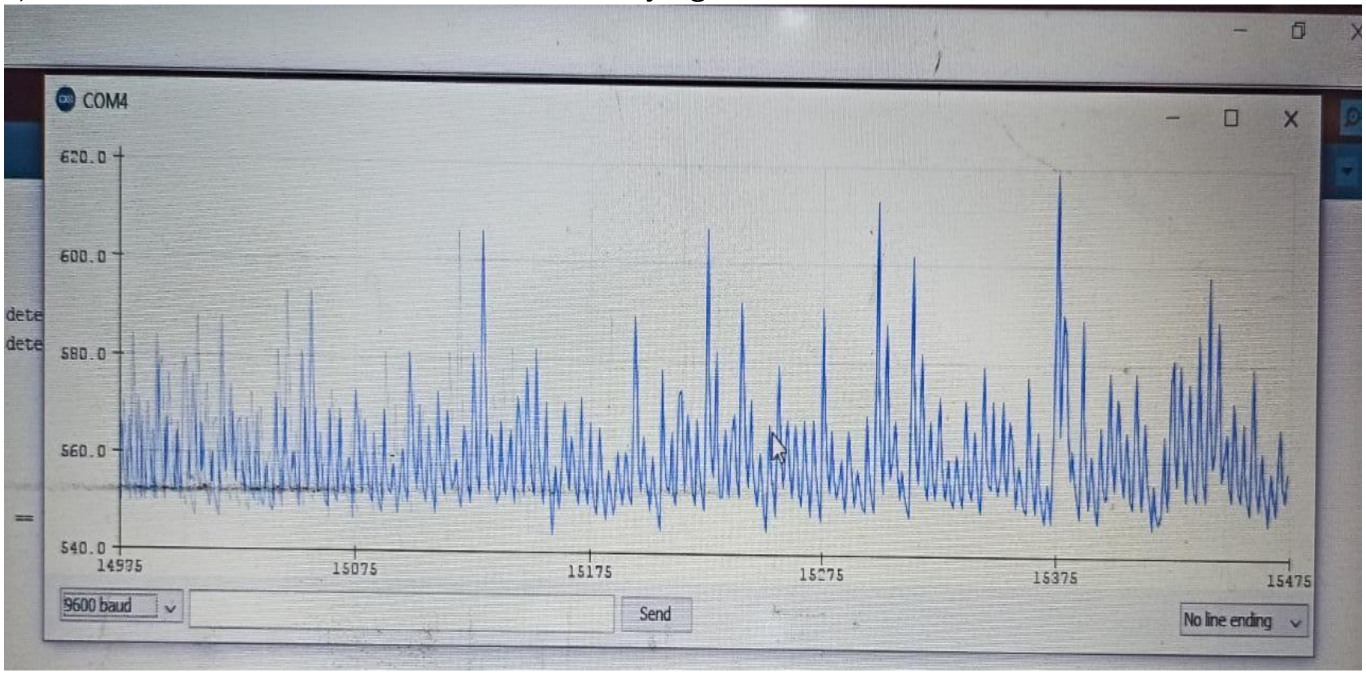
c. Configuration Diagram:



d. Sample Outputs:

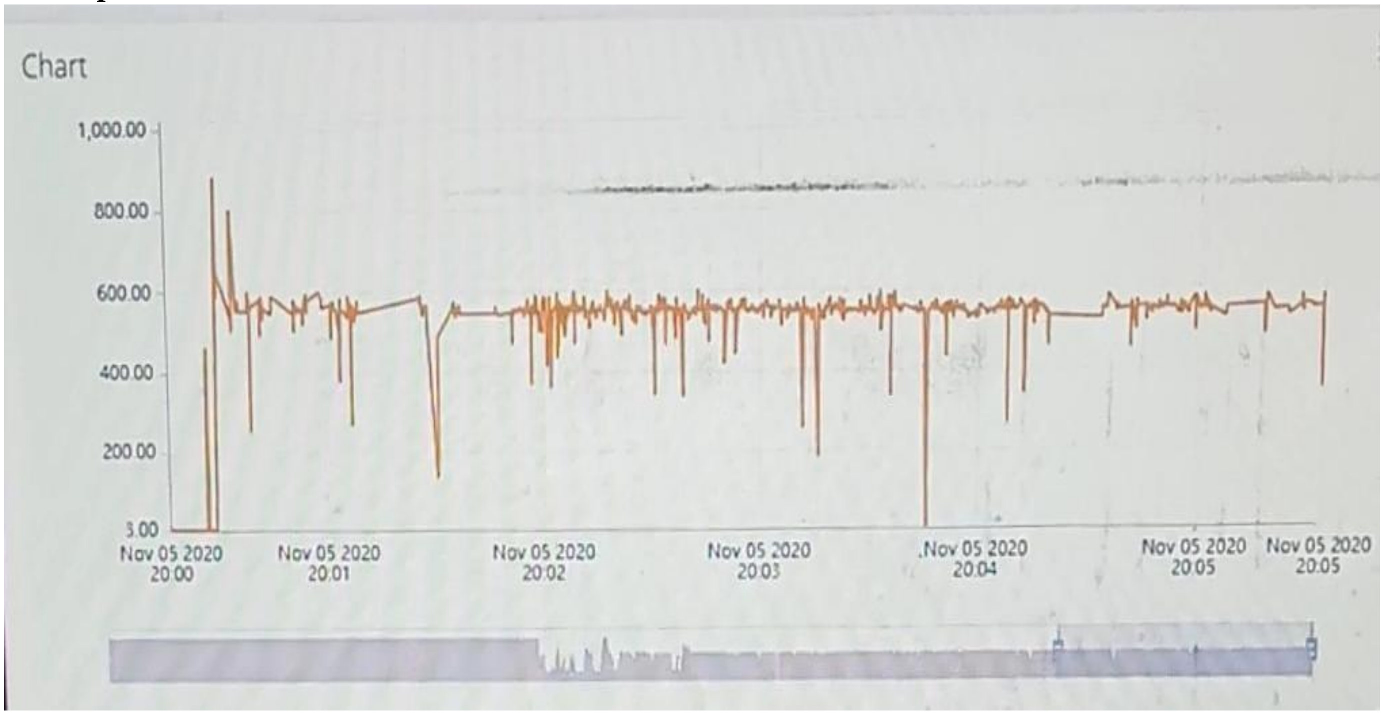
There are two .ino programs:

1). myecg.ino :

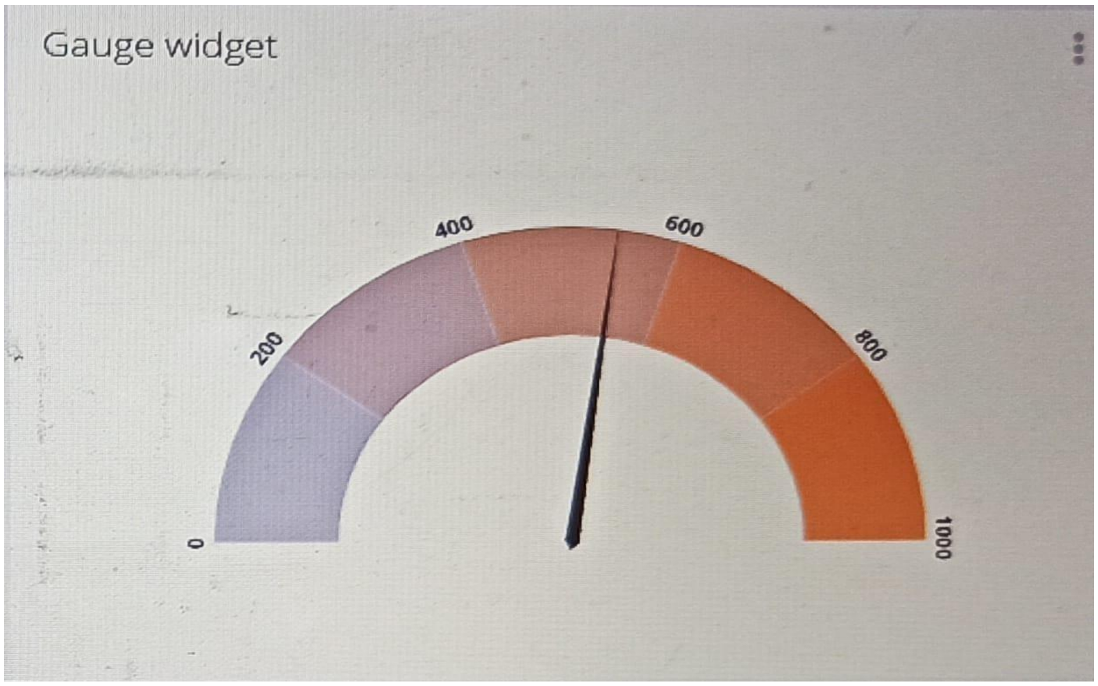


2). Heart_Monitoring_System.ino :

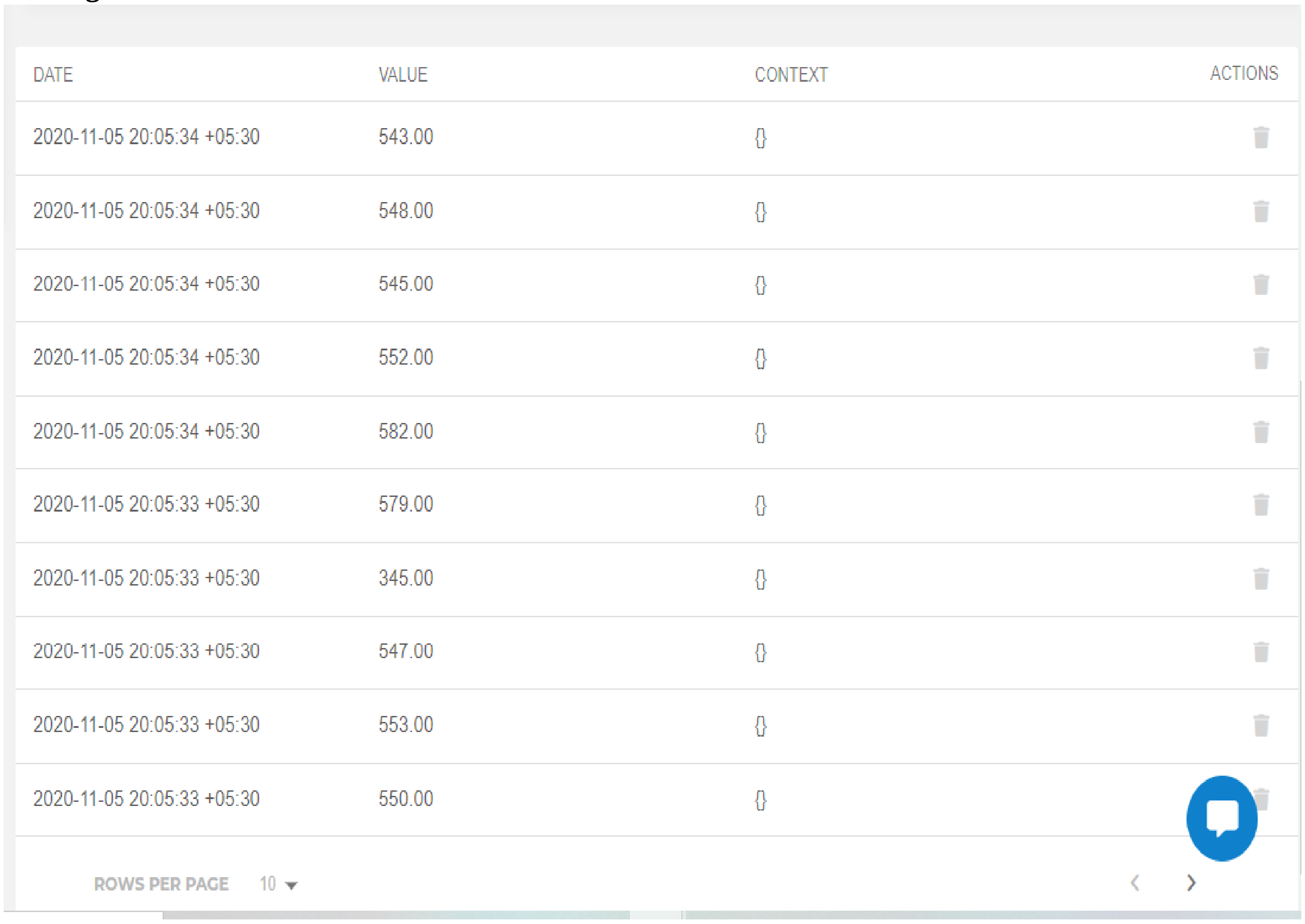
output_1:



output_2 :



readings :



DATE	VALUE	CONTEXT	ACTIONS
2020-11-05 20:05:34 +05:30	543.00	{}	
2020-11-05 20:05:34 +05:30	548.00	{}	
2020-11-05 20:05:34 +05:30	545.00	{}	
2020-11-05 20:05:34 +05:30	552.00	{}	
2020-11-05 20:05:34 +05:30	582.00	{}	
2020-11-05 20:05:33 +05:30	579.00	{}	
2020-11-05 20:05:33 +05:30	345.00	{}	
2020-11-05 20:05:33 +05:30	547.00	{}	
2020-11-05 20:05:33 +05:30	553.00	{}	
2020-11-05 20:05:33 +05:30	550.00	{}	

e. Codes:

1). myecg.ino :

```
void setup()
```

```
{
// initialize the serial communication:
Serial.begin(9600);
pinMode(14, INPUT); // Setup for leads off detection LO +
pinMode(12, INPUT); // Setup for leads off detection LO -
}
void loop() {
if((digitalRead(10) == 1)||(digitalRead(11) == 1)){
Serial.println('!');
}
else{
// send the value of analog input 0:
Serial.println(analogRead(A0));
}
//Wait for a bit to keep serial data from saturating
delay(1);
}
```

2). Heart_Monitoring_System.ino :

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

#define WIFISSID "Pooh"                              // WifiSSID
#define PASSWORD "hulkbuster"                              // wifi password
#define TOKEN "BBFF-8GERINm7dMjRnq1n2yvoELeRIwGL1R"         // Ubidots' TOKEN
#define MQTT_CLIENT_NAME "My ECG"                      // MQTT client Name

///// Define Constants
#define VARIABLE_LABEL "My ECG" // Assign the variable label
#define DEVICE_LABEL "204161012" // Assign 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);
}
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("Data uploaded to Ubidots Cloud");
  client.publish(topic, payload);
  client.loop();
  delay(10);
}
```

f. User Manual:

There are some steps to run the above codes:

1). Connect the circuit according to the **configuration diagram** given above. And connect to the laptop.

2). Open the Arduino. Copy the given codes in the .ino file. And save the file.

3). Change the WIFISSID as your wifi name, and PASSWORD as your password. And change the **Ubidots token.** Then save the code and hit the upload button.

4). Check the **serial plotter** for the first program.

5). And check the **serial monitor** for second program. Serial monitor will show the output "Data uploaded to Ubidots Cloud". Then go to the Ubidots. And add the variables line chart and Gauge widget to see the output.

Then click on the device then follow the given sequence:

device -> 204161012 -> my-ecg -> variable (my-ecg report)