National University of Sciences and Technology (NUST)
School of Electrical Engineering and Computer Science

Embedded System Design
Project Report

IOT Based ECG Monitoring

SUBMITTED TO: Ma'am Arooj (Lab Engineer)

SUBMITTED BY:

Mohammad Shehroz Ashraf	294966
Chaudhary Muhammad Shaheer Yasir	286021

Class: BEE – 11 D & B

Date of Submission: 28th December 2022

Project Objective:

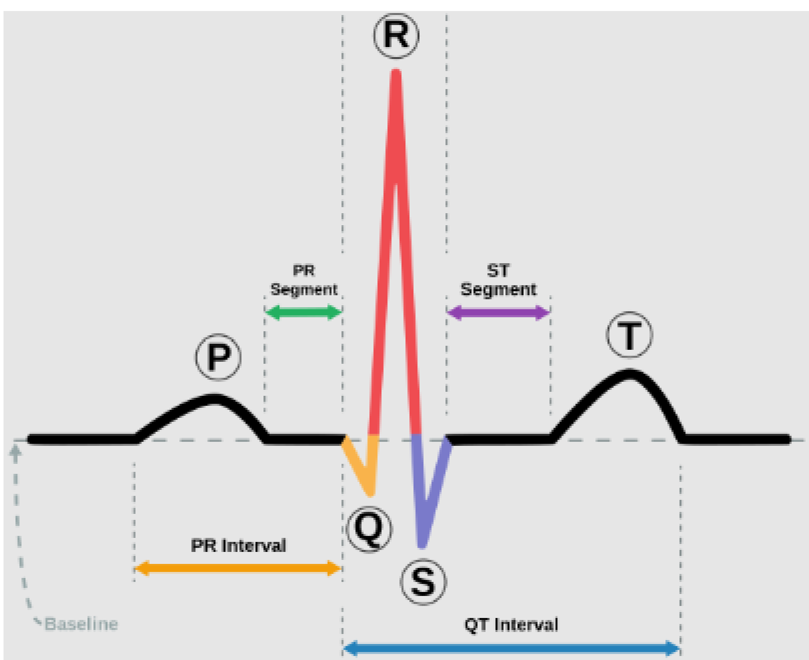
The objective of this project is to design and implement a remote monitoring system for electrocardiography (ECG) using Internet of Things (IoT) technology. The system will allow patients to monitor their ECG remotely, with the data being transmitted to a healthcare provider for analysis. This will allow patients to receive timely medical attention and avoid unnecessary hospital visits.

Problem Statement:

Patients with cardiovascular conditions often require frequent monitoring of their ECG to ensure that their condition is stable. However, traditional ECG monitoring methods can be inconvenient and time-consuming, as patients may need to visit a healthcare facility to have their ECG monitored. This can be particularly challenging for patients who live in remote or rural areas, or who have mobility issues.

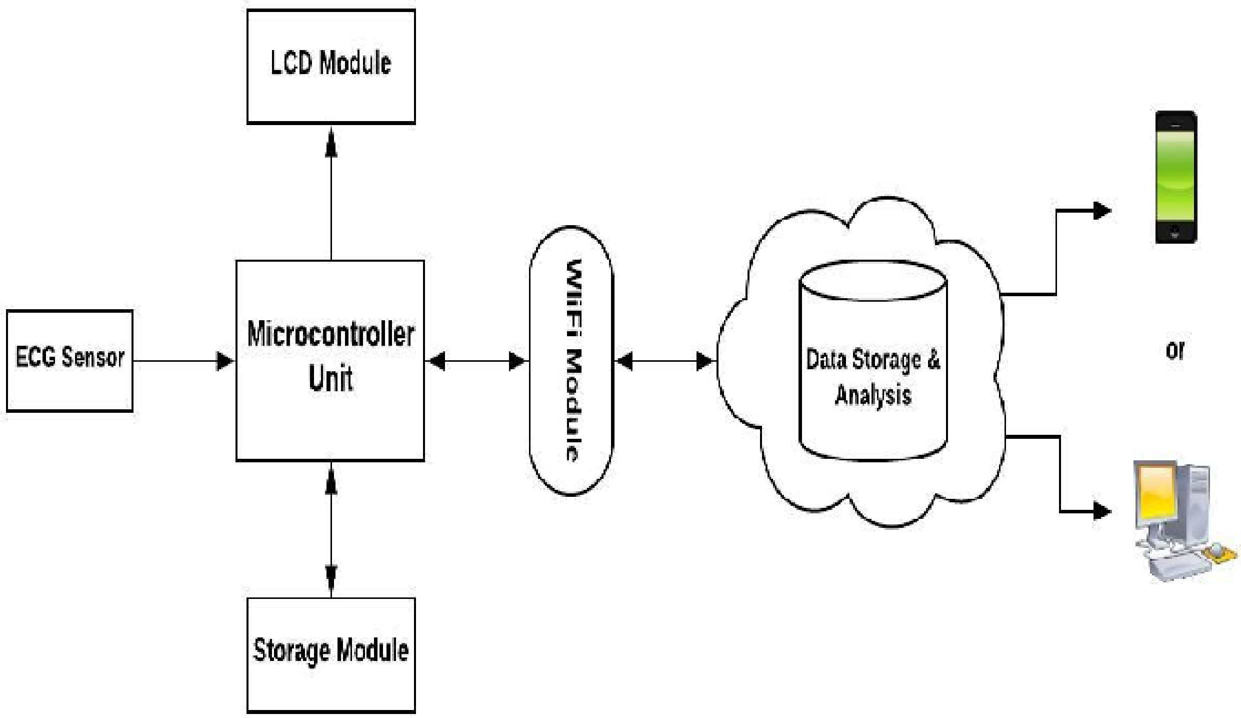
ECG Waveform:

The electrocardiography (ECG) waveform is a graphical representation of the electrical activity of the heart. It is typically made up of five distinct waves, known as the P, Q, R, S, and T waves. The P wave represents the depolarization of the atria, or the upper chambers of the heart. The Q, R, and S waves represent the depolarization of the ventricles, or the lower chambers of the heart. The T wave represents the repolarization of the ventricles. By analyzing the ECG waveform, healthcare providers can diagnose various cardiovascular conditions, such as arrhythmias, myocardial infarction (heart attack), and hypertension.



A normal ECG waveform

Block Diagram: The system consists of three main components: a **wearable ECG device**, an **ESP-32**, and a **remote server**.



Block Diagram of our IOT based ECG monitoring

The wearable ECG device is worn by the patient and consists of sensors that measure the electrical activity of the heart. The device transmits the ECG data wirelessly to the gateway device, which acts as a bridge between the wearable device and the remote server.

The gateway device (ESP-32 in our case) is connected to the internet via a WiFi or cellular connection, and is responsible for transmitting the ECG data to the remote server, Ubidots. The server stores the ECG data and makes it available for analysis by healthcare providers.

Design Methodology:

The system was designed using a combination of hardware and software components. The hardware components include the wearable ECG device, the gateway device, and the remote server. The software components include the firmware running on the wearable device, the software running on the gateway device, and the server-side software.

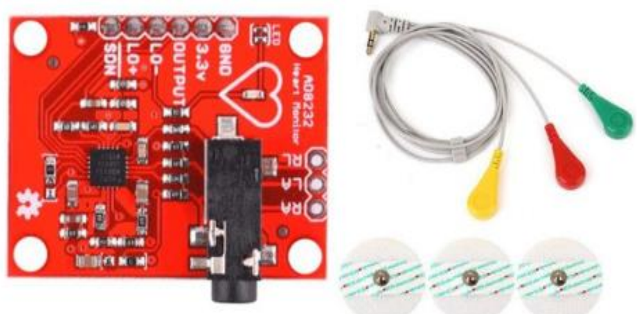
Hardware Components:

- **Wearable ECG device:** The wearable ECG device consists of sensors that measure the electrical activity of the heart, a microcontroller for processing the data, and a wireless module for transmitting the data to the gateway device.

- **Sensor Used:** The AD8232 is a single-lead heart rate monitor (HRM) sensor that can be used to measure the electrical activity of the heart. It is commonly used in wearable devices and other portable medical devices for continuous monitoring of electrocardiography (ECG).

  The AD8232 sensor consists of a low-noise amplifier, an instrumentation amplifier, and a pulse-detection circuit. The amplifier amplifies the weak ECG signal, while the instrumentation amplifier filters out noise and DC offset. The pulse-detection circuit detects the R-wave of the ECG signal, which is used to determine the heart rate.

  The AD8232 sensor has a number of features that make it suitable for use in portable medical devices. It has a low power consumption, making it suitable for use in battery-powered devices. It is also small and lightweight, making it easy to integrate into wearable devices. Additionally, it has a high common-mode rejection ratio, which allows it to reject noise and interference from external sources.



- **ESP-32:** The gateway device consists of an ESP-32 microcontroller, a wireless module for receiving data from the wearable device, and a connectivity module for connecting to the internet. The ESP-32 microcontroller is responsible for managing the other components of the gateway device and performing tasks such as data processing and communication with the remote server. The wireless module allows the gateway device to receive data from the wearable ECG device, and the connectivity module enables the device to connect to the internet via a WiFi or cellular connection.

- **Remote server:** The remote server is a computer running server-side software that stores and processes the ECG data.

Software Components:

- **Firmware:** The firmware running on the wearable device is responsible for reading the ECG data from the sensors, processing the data, and transmitting it to the gateway device.
- **Arduino IDE**: The software running on the gateway device, written using the Arduino Integrated Development Environment (IDE), is responsible for receiving the ECG data from the wearable device, formatting it, and transmitting it to the remote server. The Arduino IDE is a software tool that allows developers to write and upload code to an Arduino microcontroller. In this case, the code running on the gateway device written using the Arduino IDE is responsible for receiving the ECG data from the wearable device, formatting it into a suitable format for transmission, and transmitting it to the remote server via a wireless or internet connection.
- **Server-side software:** The Ubidots is responsible for storing the ECG data, making it available for analysis by healthcare providers, and providing notifications if abnormal readings are detected.

Properly commented code:

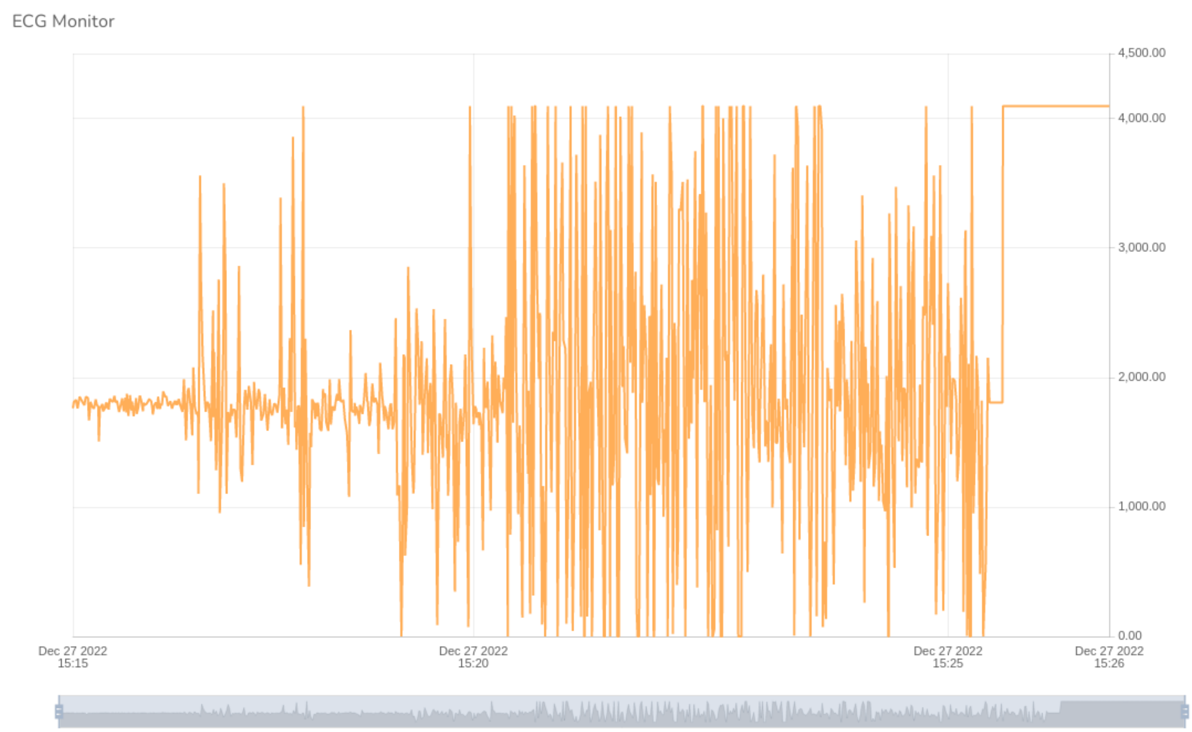
```
//Including the WiFi library to enable WiFi functionality
#include <WiFi.h>
//Including PubSubClient library to send data through MQTT client
#include <PubSubClient.h>

//Defining the SSID and Password of the WiFi network for esp32 to connect
#define WIFISSID "Shaheer"
#define PASSWORD "12345678"
//Defining the Token variable and its value can be copied from ubidots dashboard
#define TOKEN "BBFF-gQKkAdc5Gnq10uYmzOqDVjcU7eFgxe"
#define MQTT_CLIENT_NAME "muhammadshaheer"

//Setting the variables names similar to those used on the ubitdots dashboard
#define VARIABLE_LABEL "sensor"
#define DEVICE_LABEL "esp32"
#define SENSOR A0

char mqttBroker[] = "industrial.api.ubidots.com";
char payload[100];
char topic [150];
float sensor;
char str_sensor[10];

//Defining the ubidots client
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);
}

//Setting up reconnection with the ubidots server through MQTT connection
void reconnect(){
  while(!client.connected()){
    Serial.println("Attempting MQTT connection...");

    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");

      delay(2000);
    }
  }
}
//Defining two tasks for free RTOS
TaskHandle_t setupTask;
TaskHandle_t loopTask;


void setup() {
  // initialize the serial communication:
  Serial.begin(115200);
  WiFi.begin(WIFISSID, PASSWORD);
  pinMode(4, INPUT); // Setup for leads off detection LO +
  pinMode(2, INPUT); // Setup for leads off detection LO -
  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);
  //Both Tasks have the same priority
  xTaskCreatePinnedToCore(setupTaskCode, "Setup Task", 10000, NULL, 1,&setupTask,  0);
  xTaskCreatePinnedToCore(loopTaskCode, "Loop Task", 10000, NULL, 1,&loopTask, 0);
}

void setupTaskCode(void *pvParameters) {
  vTaskDelete(NULL);
}

void loopTaskCode(void *pvParameters) {
  // If the esp32 is not connected with ubidots, try to call reconnect function
  while(1) {
    if (!client.connected()){
      reconnect();
    }

    if((digitalRead(4) == 1)||(digitalRead(2) == 1)){
      Serial.println('!');
    }
    else{
      // send the value of analog input 0:
      sensor = analogRead(SENSOR);
      Serial.println(sensor);
    }
    //Sending the data and publishing it on the server
    sprintf(topic, "%s%s", "/v1.6/devices/", DEVICE_LABEL);
    sprintf(payload, "%s", "");
    sprintf(payload, "{\"%s\":", VARIABLE_LABEL);

    dtostrf(sensor, 4, 2, str_sensor);

    sprintf(payload, "%s {\"value\": %s}}", payload, str_sensor);
    Serial.println("Publishing data to Ubidots Cloud");
    client.publish(topic, payload);
    client.loop();
    //Creating a delay of 50ms
    vTaskDelay(pdMS_TO_TICKS(50));
  }
}

void loop(){
}
```

Output Results:

The system was tested by having several patients wear the wearable ECG device and transmit their ECG data to the remote server. The server was able to successfully receive and store the ECG data, and the healthcare providers were able to view and analyze the data in real-time.



ECG Signals on Ubidots Server

Conclusion/ Improvement:

The IOT Based ECG Monitoring system has demonstrated its ability to remotely monitor and transmit ECG data, providing a convenient and effective solution for patients with cardiovascular conditions. In the future, the system could be improved by incorporating additional features such as alerts for abnormal ECG readings and integration with other medical devices and sensors, such as blood pressure monitors or oxygen saturation monitors. Additionally, the system could be made more user-friendly by developing a mobile app that allows patients to easily access and view their ECG data. Overall, the IOT Based ECG Monitoring system has the potential to greatly improve the quality of care for patients with cardiovascular conditions, while also reducing the burden on the healthcare system by reducing unnecessary hospital visits.