Smart Healthcare Monitoring System for COVID-19 Quarantine Patients

Mostafa A. Mohamed, Amr Essam Anwar, Ahmed M. Sobh, Hend T. El-Saloussi, Gerges H. Gerges, Moataz E. Fahmy and Selim A. Selim
Egyptian Academy for Engineering and Advanced Technology(EAEAT), Egypt, mostafaahmed169@gmail.com, amressam01@gmail.com, ahmedsobh239@gmail.com, hend.tarek997@gmail.com, gergeshany51@gmail.com, moataz.3ssam96@gmail.com, selimabdou6@gmail.com

Supervisor: M.M.Elsherbini, Assistant Professor
Department of Electrical Engineering, the Egyptian Academy of Engineering and Advanced Technology (EAEAT), Egypt, m.moustafa@eaeat-academy.edu.eg

Abstract*– The recent spread of the new Corona-virus leading to the Covid-19 disease has put tremendous stress on world health sector as corona-virus requires continuous health monitoring and follow-up. Losses of lives in both physicians and patients plagued our communities with no answer to how to monitor patients without putting lives at risk. This emerging new disease imposes a danger to all people suffering from chronic diseases such as heart diseases, respiratory problems, and blood pressure. Furthermore, cross-contamination to physicians became the riskiest factor that restricts their ability to continue working on patients with this highly infectious disease, as they risk both their life and the life of their frequent mainstream patients. Hence the emergence of a new need for new ways to monitor patients with minimal contact and contamination. Thanks to the Internet of things (IoT) development have recently helped us create networks that connect many things through the Internet. Physicians can follow-up with their patients' condition remotely without any physical intervention and contamination. Using wearable devices, specialists have developed at a medium cost so that many people can obtain them with the sole purpose of monitoring the patients' conditions. We can limit if not prevent the losses in physicians and patients altogether. In this paper, we propose a remote health monitoring system to achieve this progression.

Keywords-- IoT, E-Health, MQTT, Flutter, Healthcare.

I. INTRODUCTION

Coronavirus has become a dreadful Pandemic; the world has been trying to stop and limit its spread by applying social distancing obligations, but this is not helping doctors at all as they cannot keep away from their Patients and medical instruments, which is causing contamination and a massive loss of life in citizens as well as great doctors. So, we are trying here to help stop that spread and save our doctors' lives by using wireless technologies in monitoring patients' health remotely. Our real-time multipurpose system is IoT based that can share real-time medical data between the patients and doctors. The data will consist of the room temperature and humidity to ensure ergonomic suitable recovery conditions, and the patients' temperature, heart rate, and Electric cardiograph.
[1] [2].

The aims of the project are: Reduce the direct interaction between doctors and patients, Helps the hospital to improve the treatment process, Limit the spread of dreadful diseases (such as Covid19), protect both medical personnel and patients by safety precautions and provide a multipurpose system that actually does the following [3]:
-Measures body and surrounding atmosphere temperature.
-Measures the oxygen percentage in the blood.
-Measures the heart rate and performs Electrocardiogram "ECG".
The main idea of the system is to monitor the patient continuously online depending on: ESP32 micro-controller, pulse Oximeter (JUMPER 500G), temperature sensors (MLX90614) and (DHT11) and heart rate (AD8232) as well as mood changes (GSR).

Here ESP32 micro-controller collects data from sensors and sent it over the internet, this data can be received by the physicians at any time by the mobile phone application.

II. THEORY

Using the ESP32 Micro Controller Unit (MCU) as our central processing unit, this project aims to collect data from a multitude of sensors such as the oxygen level in blood and heart rate using JPD500G, Electrocardiogram using AD8232, Patient's mood swings using Grove's Seeed GSR, Room temperature and humidity using DHT11 and MLX90614 for the patient's temperature; compile these data on the ESP32 MCU and then establishing a connection to multiple subscribers using MQTT protocol as a baseline form of communication. The subscribers could be either Personal Computer (PC) or Mobile Interface or a cloud server for distribution of data and control access of data to comply with the Health Insurance Portability and Accountability Act (HIPAA).

In the following section we will discuss the main components of our proposed E-Health system as shown in Figure [1].
The system consists of three micro-controllers (ESP32), the first MCU acquires data from three sensors JPD500G (Oxy-

genated blood, Heart rate and Perfusion Index sensor), MLX90614 (Body Temperature sensor) and DHT11 (Room Humidity and Temperature sensor); The second MCU acquires data from AD8232 (Electrocardiogram sensor); the third MCU acquires data from Seeed's Grove GSR (Galvanic Skin Response sensor); The MCUs are connected to the internet using WiFi and send their data to HiveMQ (Online MQTT Broker) using MQTT protocol which forwards the patients data to our mobile app using MQTT protocol.

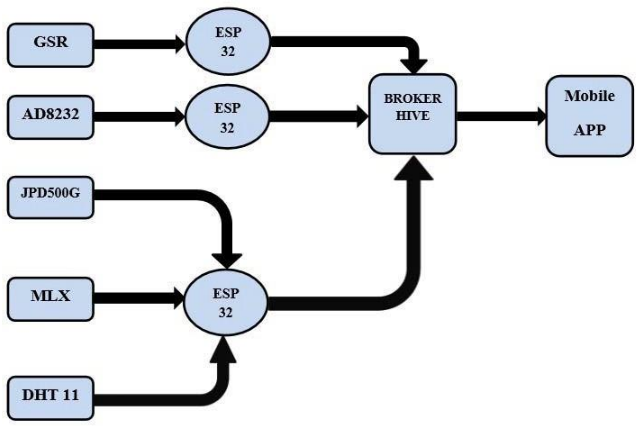

Figure 1: Proposed Block Diagram

III. Components

A. ESP32

The ESP32 Devkit V1 as shown in figure [2] is a microcontroller with a chip capable of using both Wi-Fi and Bluetooth using a 2.4 GHz embedded antenna all while having a Dual-core CPU Xtensa LX6 240 MHz and capable of running on ultra-low-power.
The following section will discuss the features used in the ESP32 [4].

Figure 2: ESP32 Devkit V1 MCU

I. WiFi

The ESP32 is capable of connecting to nearby WLAN connection using its built-in antenna that runs on 2.4 GHz,
Using its open source library WIFI which can be found on Espressif's website, WiFi is used in the project proposed in this paper to connect The ESP32 to the internet hence enabling the usage of MQTT protocol [5].

II. Bluetooth Low Energy (BLE)

BLE is a variation of the Classic Bluetooth supported on the ESP32 it main advantage is its low energy usage hence comes its name and the second advantage is it's need for only the service UUID (Universal Unique Identifier) to connect and the desired characteristics UUID to obtain the desired data, Using the open source library BLEDevice.
This feature is used to connect to JPD500G.

III. MQTT Protocol

Message Queuing Telemetry Transport protocol is a light weight publish/subscribe that uses a broker that pushes published messages in topics subscribed to by other devices in real-time.
This protocol is the main component used for data transfer in our system using its open source library on the ESP32 and the public broker HIVEMQ to send the data from the ESP32 to the broker and from the broker to the subscribed Mobile App [6] [7].

B. JPD500G

JPD500G as shown in figure [3] is a sensor made by Jumper Electronics Which offers a highly calibrated sensor all in one package, it can sense BPM, SpO2 and Perfusion Index. The device prints said data on an LED screen or it can be sent through Bluetooth as it offers Bluetooth low energy BLE as a form of communication.

Figure 3: JUMPER's JPD500G

C. MLX90614

The MLX90614 as shown in figure [4] is a thermometer for non-contact temperature measurements based on infrared technology. This Sensor Continuously transmitting the measured temperature in scale of-20 to 120 C, with a performance resolution of 0.14 C. The Normal precision of ±0.5 C across room temperatures [6].

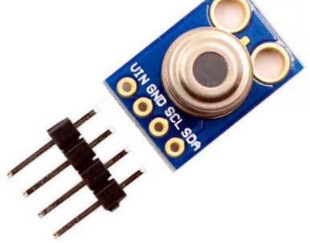

Figure 4: MLX90614

D. DHT11

DHT11 as shown in figure [5] is a sensor used to measure the temperature and humidity, uses inside a thermistor to measure the air temperature and capacitive humidity sensor, and then sends a digital signal on the data pin.
The sensor has an 8-bit micro-controller to make the calculations and comes in either 3 pins or 4 pin configurations. it reads humidity from 0-100% with 2-5% accuracy and reads temperature from -40 to 100°C with ±0.5 °C accuracy [7].

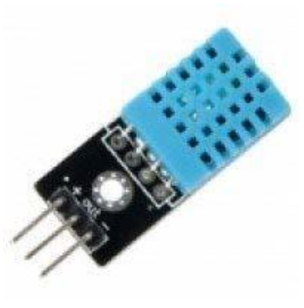

Figure 5: DHT11

E. Grove's Seeed GSR

GSR the galvanic skin response, refers to the sweat gland activity changing, and this reflects the mood and sensory arousal, if the person feels happy/relaxed or sad/tensioned. Emotional arousal level changes according to the surrounding environment. GSR applies a constant voltage of usually 0.5 volts to two electrodes that touch the skin and to a small resistance less than skin resistance series with the voltage source and electrodes. Any fluctuations in the current are caused by a change in the properties of the electric skin, and so in sweat gland activity hence changing the resistivity the sensor shown in figure [6] below.

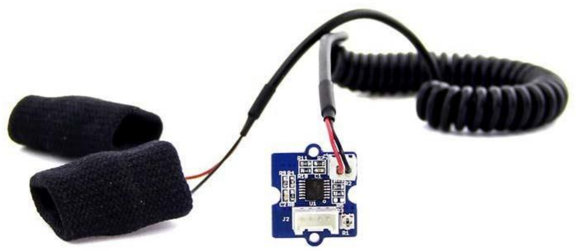

Figure 6: Grove's Seeed GSR

F. AD8232

The AD8232 ECG Module as shown in figure [7], integrated with the Analog Equipment AD8232 IC, which is a single chip for bio-potential measurement applications designed to extract, amplify, and filter bio-potential signals (like ECG and others) [8] [9].
ECGs can be highly noisy to help quickly achieve a consistent pulse from the PR and QT cycles with the AD8232 Single Lead Heart Rate Sensor functioning as an operational amplifier.

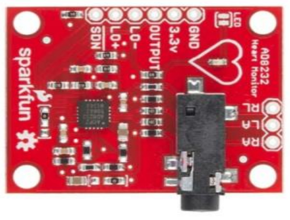

Figure 7: AD8232

IV. DESIGN

In this section we will discuss how we optimized the previously discussed sensors as well as show the circuit design, PCB design and the flow chart of each MCU. Starting with MLX90614 as shown in figures [8], [9], [10].

The MLX90614 is an infrared-based temperature sensor measuring the approximate temperature of a set object while measuring the ambient temperature around that set object. Using I2C Master-Slave communication with the ESP32 assures acceptable accuracy. Using the open-source libraries Wire.h and Adafruit_MLX90614.h.

As discussed in the method chapter this sensor acquires data about current room temperature and humidity using the DHT11 open source library we can initialize the sensor, then we set up the serial path for the serial monitor, later within the full integration process we will omit this part as it won't be required when dealing with MQTT protocol [8].

JPD500G is a sensor as discussed before that measure SpO2, BPM and PI% and it offers BLE (Bluetooth Low Energy) so using this we can interface with the sensor in a wireless fashion. But first we needed a way to obtain the Service UUID and the desired Characteristics UUID that transmits the data. To do so we used the hacking tool BLEAH on Kali Linux which is a nearby BLE Scanner and Hacker tool. After Acquiring the Desired UUID we can begin our work on the ESP32 to connect to the Device and acquire the desired Data using the Characteristics UUID. First, we include the BLEDevice library for ESP32 and we input the Service UUID to connect to the device then we can acquire the desired data using Characteristics UUID [5] [10] [11].

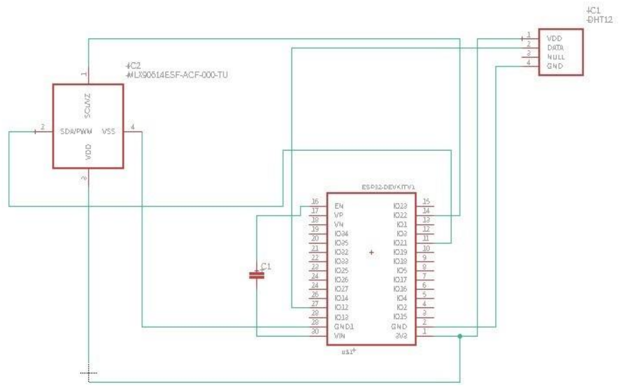

Figure 8: ESP32-1 Connected to DHT11 & MLX90614

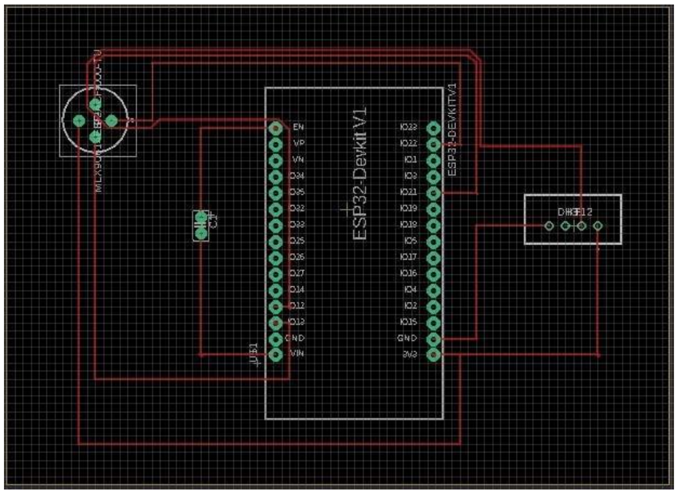

Figure 9: ESP32-1 PCB Design (MLX90614-DHT11-JP-D500G)

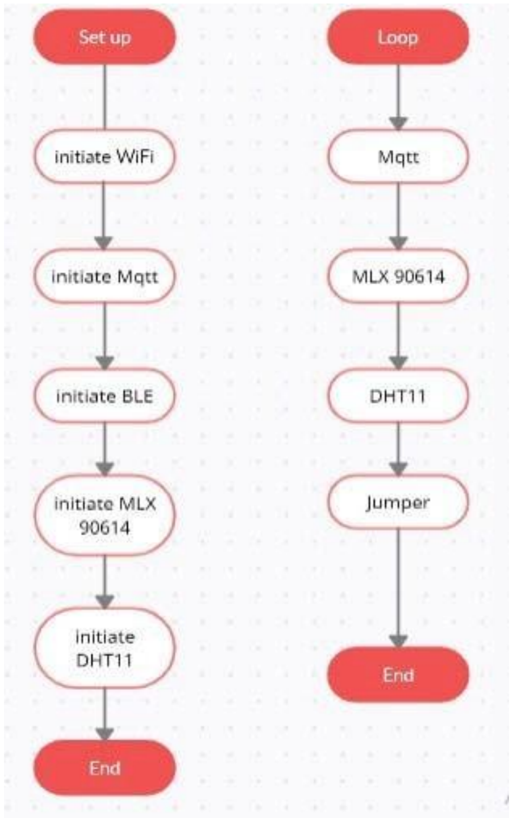

Figure 10: Flowchart ESP32-1

The Second System is for the ECG sensor AD8232 we chose to put this sensor alone on an ESP32 as it works in an infinite loop, This Sensor graphs heart beats preforming Electrocardiograph that can be enhanced with signal processing. using Arduino IDE for data acquisition. In the Data acquisition process as shown in figures [11], [12], [13].

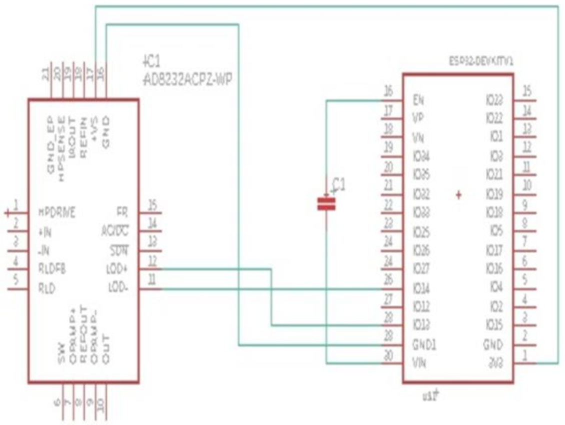

Figure 11: ESP32-2 Connected to AD8232

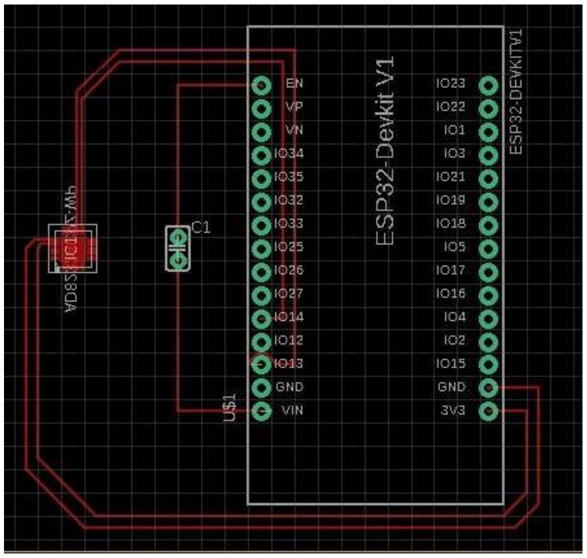

Figure 12: ESP32-2 PCB Design AD8232

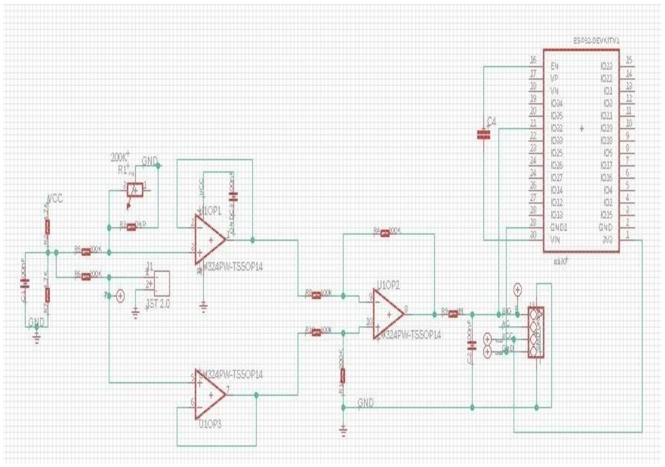
Figure 14: ESP32-3 Connected to GSR

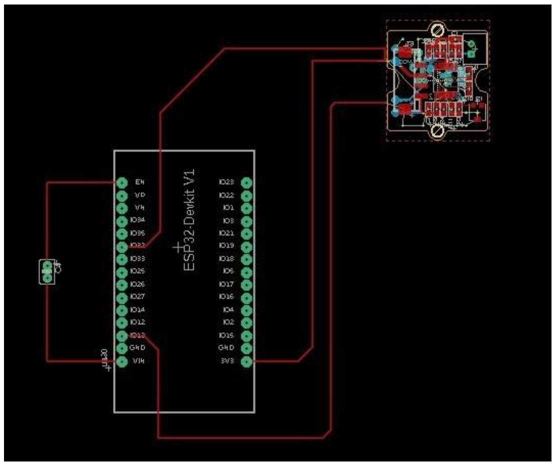

Figure 15: ESP32-3 PCB Design Grove's Seeed GSR

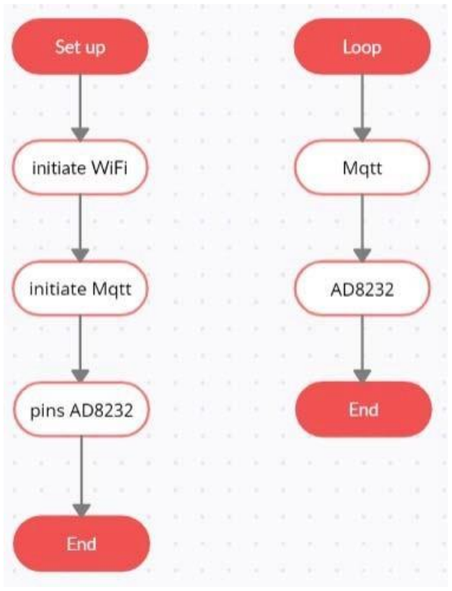

Figure 13: Flowchart ESP32-2

The Third System is for the GSR sensor we chose to put this sensor alone on an ESP32 as it works in an infinite loop, The GSR sensor as discussed before would identify mood swings by measuring skin resistivity. So, in our approach we would measure its analog output as shown in figures [14], [15], [16].

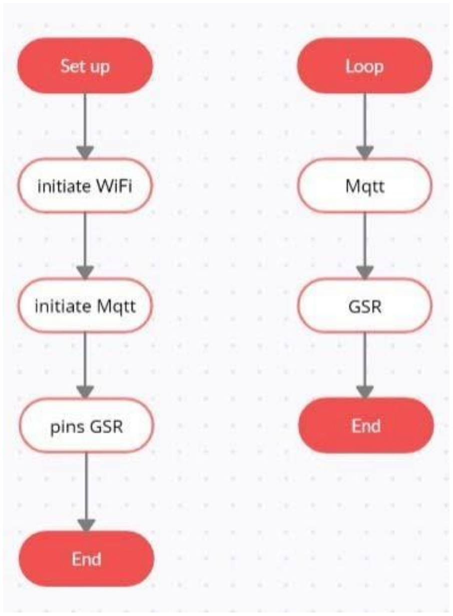

Figure 16: Flowchart ESP32-3

V. Implementation

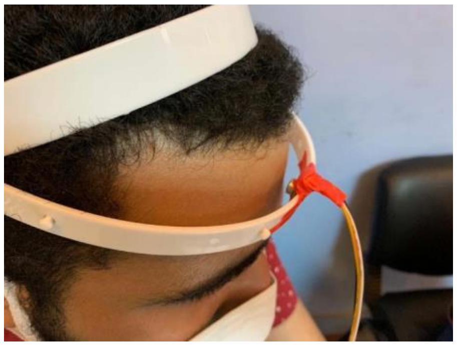
Figure 17: MLX90614 Headband

Figure [17] shows the mlx90614 headband prototype, Figures [18], [19] show the assembly process of the project.

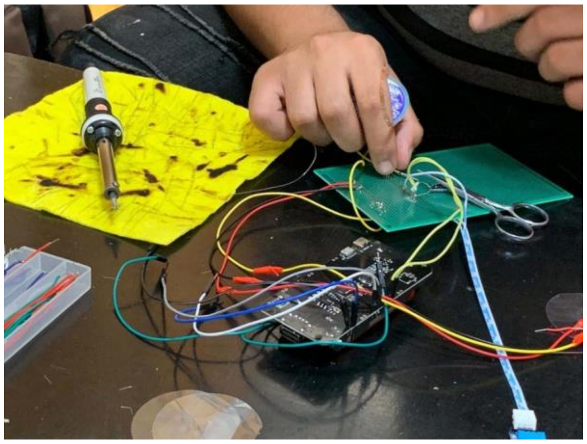
Figure 18: Connecting MLX90614 & DHT11 to a Single ESP32

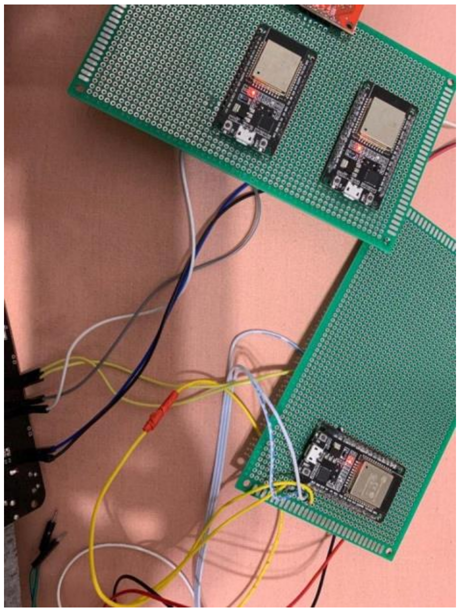
Figure 19: After Connecting All Three ESP32s to the Power Source & Their Sensors

Figure [20] shows the overall complete system with all its peripherals

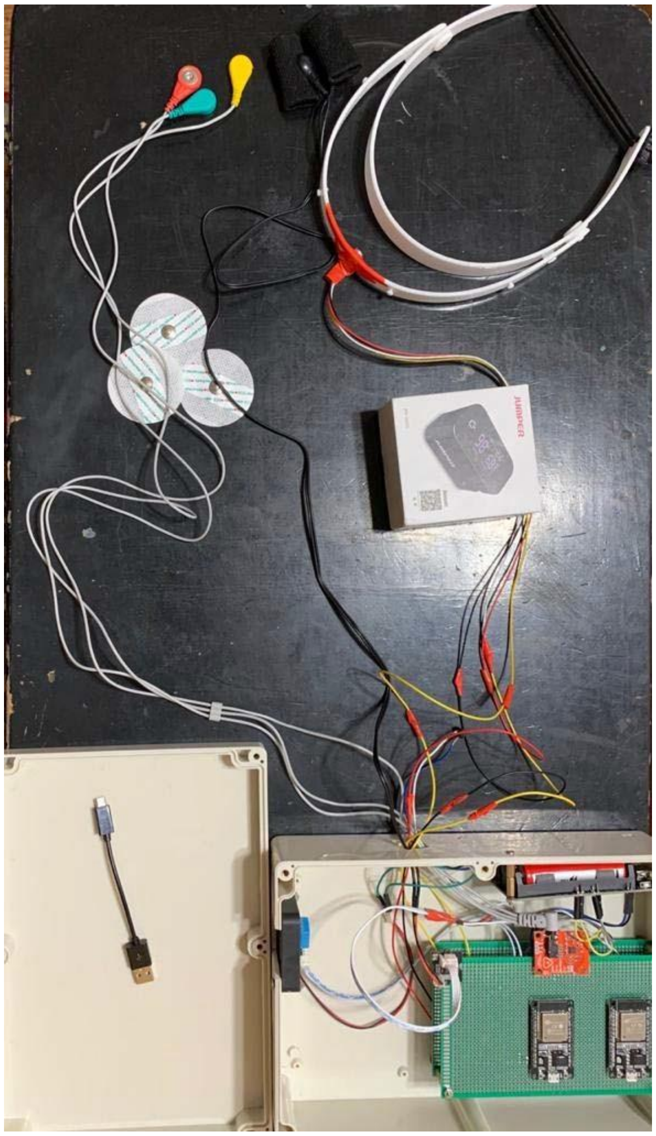
Figure 20: Overall Project

In figure [21] a test was made on the complete system by connecting a 24 year old male to the system and testing the complete integration of the system by sending the data to the Mobile APP which will be discussed later and the results can be seen in figure [25].

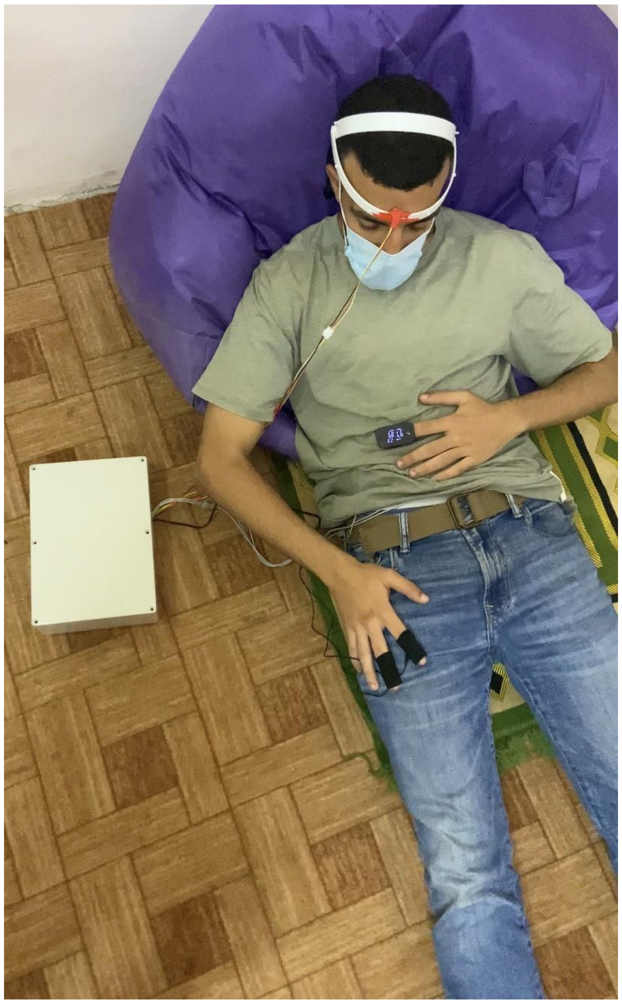

Figure 21: A Test of the overall system on one of our colleagues

VI. Mobile Application

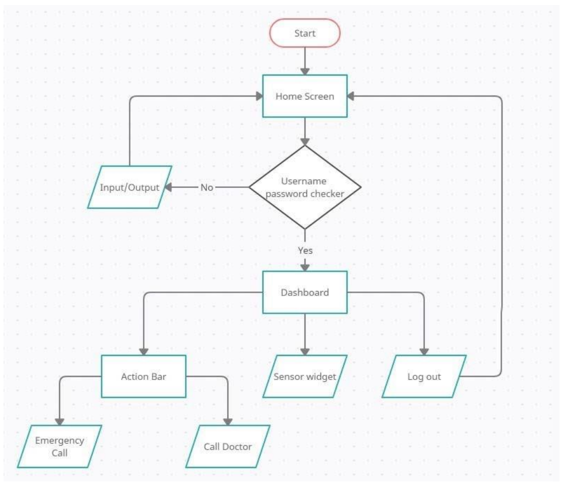

Figure 22: Mobile App Flowchart

We constructed the mobile app using Flutter so in this section we will understand the code of our mobile application. It is split up into multiple files to maintain readability and good structure. In our main file we have a declaration for our Home Screen: Which is declared with a Stateless Widget.

In our home screen, we have a CupertinoPageScaffold Widget with a navigation bar that holds our App title "Cloudcare", and a single child that holds the Login Screen widget.

Our login screen mainly contains two input fields, and a login button, all of them are custom made widgets to encapsulate each one with its properties and methods. UserInputField contains only style parameters, while the LoginButton widget contains the login button with its style parameters alongside some methods to handle the login process. we check the user's input and based on it, we either call pushDashboard method – which simply loads the Sensors screen – or we reject the login credentials and show them a dialog explaining that.

The dashboard screen is mainly built with a CupertinoPageScaffold widget like the home screen, with two main components (or widgets): SensorsWidget and ActionsBar.

The ActionsBar widget simply holds quick actions for the doctor to take, like calling the patient via phone or calling the emergency if the case is urgent.

SensorDataCard is a stateful widget since it changes its state and numbers with the values coming over MQTT connections with the hardware. We have a main configuration for the card and it consists of these parameters: topic, name, prefix, suffix, emoji. The topic variable handles the MQTT topic path since each card handles its own MQTT connection – to avoid full disconnectivity if one connection fails in its lifetime – and each card is independent of other cards. The name is the card name displayed on the screen. The prefix is a symbol displayed before the value, and the suffix is the opposite, a symbol displayed after the value. Lastly, the emoji, for each card we have a simple emoji describing the card's intention, like a heart for the BPM card, a Lungs emoji for the SpO2 card, and so on. We have a function called setupMQTTConnection that initializes our MQTT connection and adds onConnected and onDisconnected events listeners, which are predefined functions too.

And we have a function to handle received MQTT message, which is the sensor value, then we change the card's value to the new value.

And we connect to the MQTT server using a function called connect which completes all the logic after initiating a successful connection with the server.

VII. Results

The following figures present the user interface (login screen) & (dashboard) which are stateless widgets with a safe area in them, as for Figure [23], there are three children widgets which are the Username and password boxes as well as the login button, Figure [24] on the other hand has multiple stateful children widgets which show the data received from the MQTT broker, the detail of all widgets function has been previously discussed.

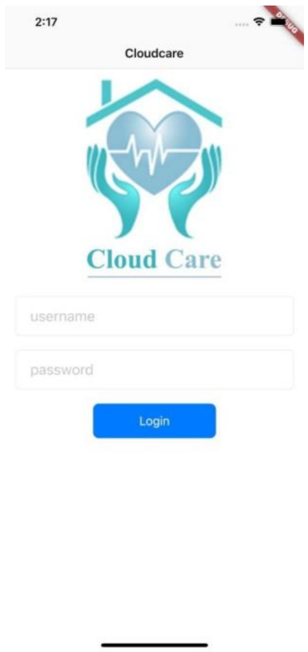

Figure 24: Login Screen Where the User Inputs their Credentials

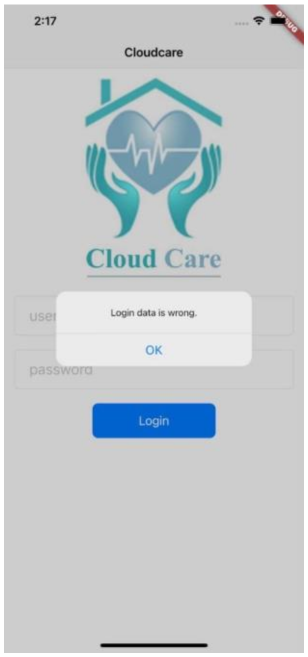

Figure 23: Wrong Credentials Interrupt

The figures [23],[24],[25], follow the User's Experience from opening the application to the login screen to the interrupt message if the user enters wrong credentials to the dash board for when he enters accepted credentials with a sample of one of the widgets.

The following figures [25],[26], the transmitted data of the complete system transmitted to the mobile app the patient is a 24-year-old male as seen in figure [21] under no fatigue at the time of testing hence the normal readings, the value of both ECG and GSR are not plotted but should it be it would show stable heart rate and equivalent to the average heart rate and emotional stability.

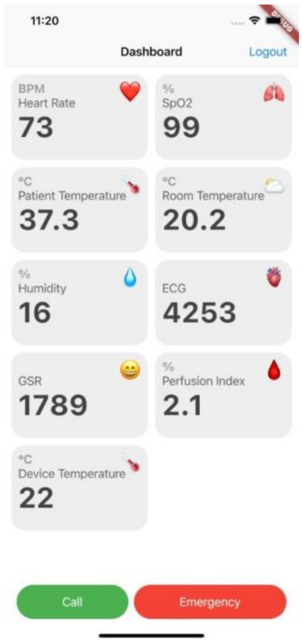

Figure 25: Dashboard/Accepted Credentials

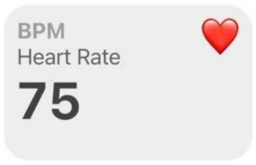

Figure 26: A Sample Widget

VIII. Conclusion

In this paper we discussed the recent problem of the new Corona-virus resulting in the COVID-19 disease and how using IoT and cloud server technologies we can solve that problem of patients' monitoring while avoiding contamination and infection, and how to reduce medical personnel to patient interaction all while maintaining the physician's ability to monitor all their patients using a multitude of sensors that acquire data about the patient's condition and process it and then send it directly to the patient's physician with adherence to data protection regulations like the HIPAA. As seen in the results, our sensors are capable of acceptable accuracy with room for future development. To sum it up using our proposed device, a physician can monitor its patient's oxygen in blood saturation level and their body temperature, and while maintaining constant monitor on their heart using ECG as well as their mood swings using GSR all while monitoring the patient's room temperature and humidity to provide an ergonomic space in which the patient can speedily recover from either COVID-19 or any other disease that is afflicting them, with acceptable accuracy and immediate warnings and the case of critical deterioration.

IX. Future Work

[1] Improving the accuracy of our sensors by further pre-processing the output of the sensors.

[2] Looking for alternative sensors with higher accuracy and lower tolerance.

[3] Testing the device under physician supervision.

[4] Studying additional ways for data acquisition.

Acknowledgment

We Would like to acknowledge and thank IoT&AI Egypt for their continuous support and mentorship across this project.

References

[1] Implementation paper published in IJSART International Journal for Science and Advanced Research in Technology, Approved in Volume 6, Issue 4, April 2020.(http://ijsart.com/Home/IssueDetail/36383)

[2] Pantelopoulos A., Bourbakis N.G. A survey on wearable sensor-based systems for health monitoring and prognosis. IEEE Trans. Syst. Manand Cybern. Part C (Appl. Rev.) 2010; 40:1–12. doi: 10.1109/TSMCC.2009.2032660. [CrossRef] [Google Scholar]

[3] B. E. Reddy, T. V. S. Kumar, and G. Ramu, "An efficient cloud framework for health care monitoring system," in 2012 International Symposium on Cloud and Services Computing, pp. 113–117, Mangalore, India, December 2012.

[4] Espressif Systems, "ESP32 Series data sheet," ESP32 Series Datasheet v3.5,Jan 2021.

[5] Hwang, H. C., Park, J., & Shon, J. G. (2016). Design and implementation of a reliable message transmission system based on MQTT protocol in IoT. Wireless Personal Communications, 91(4), 1765-1777.

[6] Melexis, "Datasheet Single and Dual Zone. Infra-Red Thermometer in TO-39,"MLX90614 family, Sep 2019 (rev. 013).

[7] ASAIR, "DHT11 Temperature & Humidity Sensor features a temperature &humidity sensor complex with a calibrated digital signal output.," DHT11 Humidity& Temperature Sensor v1.3, Mar 2017.

[8] Analog Devices, "Single-Lead, Heart Rate Monitor Front End," AD8232 Datasheet Rev. D, Aug 2012.

[9] Matthew D'Souza, Montserrat Ros, Adam Postula. 2006. Wireless Medical Information System Network for Patient ECG Monitoring. Digital System Design: Architectures, Methods and Tools, 2006, DSD 2006, 9th EUROMICRO Conference. pp. 617-624.

[10] S. Tyagi, A. Agarwal, and P. Maheshwari. A conceptual framework for iotbased healthcare system using cloud computing. In 2016 6th International Conference - Cloud System and Big Data Engineering (Confluence), pages 503–507, Jan 2016.

[11] B. Xu, L. D. Xu, H. Cai, C. Xie, J. Hu, and F. Bu. Ubiq- uitous data accessing method in iot-based information system for emergency medical services. IEEE Transac- tions on Industrial Informatics, 10(2):1578–1586, May 2014. ISSN 1551-3203.