IoT-Based Real-Time Health Care Monitoring System

Saravanan S[1], Dr. Hariharan I[2]
[1]21BEC1298 School of Electronics Engineering, VIT Chennai, [2]Guide

***Abstract*—The importance of modern healthcare to monitor the physiological signs continuously and in real time plays a great role, especially about elderly patients, especially for the people who don't have access to modern healthcare systems. Traditional health monitoring always involves physical visitation to a hospital or other established medical centres, which poses an inconvenience and is costly. The project herein focuses on the IoT-based real-time health monitoring system aiming at providing an efficient, portable, and accessible solution. Using an ESP32 microcontroller, it collects critical health parameters from connected sensors, such as heart rate, body temperature, and blood oxygen levels. The collected data is transmitted via Wi-Fi to the Thingspeak cloud platform, and authorized individuals can monitor health metrics in real-time. Also, using a custom python GUI application built in Python uses this data from the cloud and let us sends notifications and alerts on any abnormal reading. This project encourages the use of portable health care system where the patient doesn't have depend on hospitals to monitor their vitals. The idea proposed over here makes the user to have control over their health data and also minimised the visit to hospitals. The solution given here is to avail health care systems even too the people from no connect to modern infrastructure and health care**

I. INTRODUCTION

The most advanced progress in the IOT field is that it has transformed into one-to-many fields. For the betterment of patient life and health, care quality and managing the health data, it has posed many challenges in terms of integration with IOT devices. Along with the increasing demand for efficient, easily accessible healthcare to elderly people and for chronically ill patients. In the recent times it has become a necessary thing to develop a modern health care monitoring solution. It can be an IoT-enabled health monitoring system that can be built by using ESP32 micro controller to calculate the various health parameters like heart rate, temperature and blood oxygen. This data collected from sensors is processed and stored to cloud via thingspeak, and it also give access to analyse the person health data by monitoring in the app which fetches data every 10-15 seconds. This health care system reduced the frequent visits to hospitals and its very lower price makes the people to buy and make sure to spread health care system even to remote areas. This paper explains the design, making and implementation of the system for broader absorption of modern and advanced health care system.

II. SYSTEM ARCHITECTURE

The IoT-based real-time healthcare monitoring system consists of subcomponents connected by esp32 and where the data is collected from sensor like MAX30102, Ds18B20 and Oled powered by another ESP32 these connect to internet and one esp32 with sensors sends data to thingspeak and other esp32 with oled fetches that data and displays it on oled

The sensor modules can readily establish contact with the ESP32 microcontroller via I2C and GPIO interfaces for easy communication. Real-time data transmitted by ESP32 over WiFi to a cloud based platform(thingspeak),for processing the data sent from esp32 connected by sensors, which is kept and shown. It can be accessed from anywhere using a Pythonbased monitoring application which extracts and demonstrates health metrics to make it apt for remote health monitoring.

A. Block Diagram

This is the block diagram of the health care monitoring system, which shows the connections between the ESP32, sensors,oled, and python app.

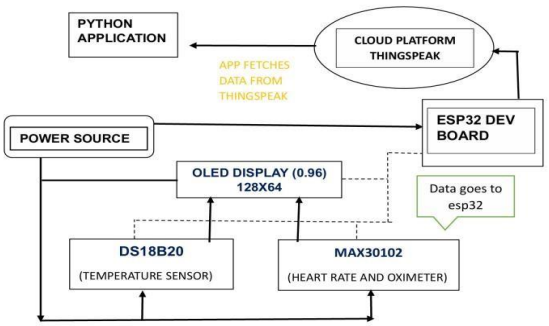

Fig. 1: Block Diagram of the IoT-Based Health Monitoring System

ESP32 with sensors collect the data from sensors and send it to thingspeak cloud platform and esp32 with oled fetches data from thingspeak and displas it on oled and the python custom GUI application fetches data from thingspeak and show us the results and let us decide to send mail or export as excel offered by the app options

III. HARDWARE COMPONENTS

The following hardware components were used in the system:

A. ESP32 Microcontroller

The esp32 acts as a processing unity for the systems, which have both wifi and bluetooth for fast IOT applications. As its low power device it is ideal for portable device.

B. Sensors

MAX30102: calculates SpO2 and heart rate with accuracy. DS18B20: A digital temperature sensor with accuracy up to 0.5-3°C.

C. OLED Display

This OLED display helps us to display our collected data either via TX RX pins or fetch data VIA esp32

D. Power Management

The system can be powered by external source via USB-B or some other power source and it must be implemented with efficient power consuming algorithms for the long and better battery life.

IV. SOFTWARE IMPLEMENTATION

A. Data Collection and Processing

Data collected by esp32 via sensors are processed and cleaned and stored in thingspeak before uploading it to thingspeak. This uses wifi to work and do all of the tasks.

B. Cloud Integration

Thingspeak is a very reliable platform to store and process and analyse the health and even helps the user to make it more understandable by making it visualizing the data in graphical way with in the application.

C. Python Application

The python application retrieves data from thingspeak and shows the data in graphical way to understandable and easy format to visualize it. The application also offers us the option to share the data via email, or export as excel in order to send the data to concerned officials or personals.

V. HOW IT WORKS

The MAX30102 and DS18B20 is used ESP32 to collect the data of the patient. The MAX30102 sensor calculate heart rate and SpO2 levels by passing the infrared light in the body; DS18B20 is used in measuring body temperature. The ESP32 process the data and sends the data to Cloud via thingspeak so that it can handle the process of managing the data

Fig. 2: Graphical interface offered in thingspeak

A. Communication

Communication protocol used by ESP32 to communicate with all the sensors is I2C.The esp32 has the power to connect to wifi and send the data to thingspeak and an python application can monitor the data and it also can be fetched by the oled powered by ESP32 in real time to monitor all the readings stored and processed in the cloud server via Thingspeak

VI. PYTHON MOBILE APP INTERFACE

In the Python app, the Thingspeak API request is sent using the real-time data delivered in order to show up on the user interface. The interface is simple on the app with heart rate and temperature and SpO2 levels, and also sends alerts to the user's end if readings surpass a predefined threshold.

Fig. 3: Python Application Interface for Real-Time Health Monitoring

The user can view current health data as well as receive instant alarms in case of any abnormal readings. In such a way, the user interface is engineered so that these users can readily access their health data.

A. App Output

The application is developed to monitor the real time data so that we can see the metrics of a person and act swiftly. Whenever an abnormal reading is detected the app gives us the option to share the output to the specific personals.

This python code uses fetch data by inbuilt library called requests which connects to thingspeak via API keys so that it can fetch the data whenever the user want to know.

VII. METHODOLOGY

In this project we used systematic approach to from sensors to micro controller through sequential selection and planning, Then we connected the sensors according to the proposed circuit block diagram above and integrated with the sensors.Once the sensors are ready to power, we power it up and make it ready to transmit the data from ESP32 to cloud and the custom made python application.

The system implemented process the data collected by sensors in esp32 and transfers it to thingspeak VIA HTTP protocols, and the custom-made python application will fetch the data from cloud using request library and in time the esp32 connected to OLED in other end acts as a display for view the data rather than showing it in graphical and statistical way.

VIII. THINGSPEAK CONNECTION

Thingspeak is a platform which acts a cloud for the devices to store data and collect and process it wireless mode and it also perform the analysis of data into raw and unique data for the users to see and make it statistical for the views The esp32 in the system first collect data from sensors then process it in the cloud, again the esp32 with oled fetches data to the esp32 and our python application can also fetch the dataThe Arduino code powered the ESP32 to send data for every 10-15 seconds for the cloud to pre-process and store the data.

IX. RESULTS AND DISCUSSION

A. Discussions:

1) Circuit implementation challenges and power issues: During the integration and setup of the healthcare monitoring system ,an issue has been noticed that when the person is using high power consuming temperature sensor, the data is accurate but the other things like heart rate and spo2 is not working or they stop giving new data, but when we are using the low power consuming sensor the data is quite fine and other readings are also quite good enough with minimal deviation compared to the actual data, here are the few images of the data which is sometimes experience variations and deviation.

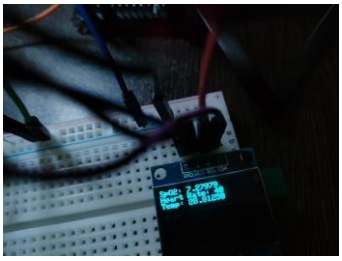

Fig. 4: Data when powered by High Power Temperature Sensor we can see that the oled is displaying the correct data, but the values for max30102 which calculate the heart rate and Blood Sp02 is not even changing the values

But when the data is powered by the low powered sensor, we can see that the values changes and the max30102 starts to work fine and display values quite accurate.

Fig. 5: Data when powered by low Power Temperature Sensor

2) Connectivity and Data uploading: Sometimes we even noticed that when the data gets uploaded it takes sometimes more than 10 seconds to the cloud, which could be affected by environment and surroundings and depends on signal strength of the network though. The sensors collect the data and process it through the esp32 and esp32 takes time to process the load of data and the analyses and send it.

Other than that remaining cloud data processing is good and best and reliable and can handle loads. And fetching of data is also good as we can see the data can be fetched fast even in the python application

3) User interface of the Application: The python Ui is quite reliable and fast enough to load the data and visualize in graphical way to make the users more understandable and the graphs are also plotted in the correct data matches the data fetched from thingspeak. The email, export data also works fine and immediate response from the application is given from the server and send mail or arranged in the excel format so that the user can access the data metrics later.

4) Sensor accuracy: Rather than the power management the sensors also start to have some deviations even after having many data, some data are off the chart like either too much high or too low so the accuracy of the sensor also has deviation like 2-5 deviation in the values. but it's quite good reliable enough to monitor.

B. Implementation and outputs of the system

The outputs of the system and the images are shown below: The figure-6 and 7 we can see that the fully implemented health care system is ready for the calculating the data and send to cloud and thingspeak.

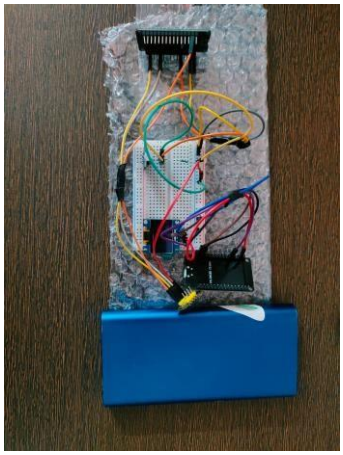

Fig. 6: Top view of the circuit

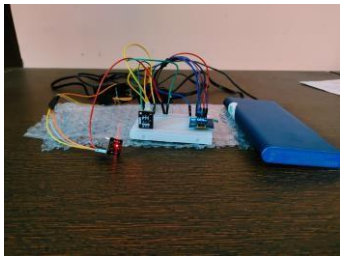

Fig. 7: Front view of the circuit

Application UI: The application is made full of python which can act like both front and backend for the project and collect data and visualise it. Graph visualisation of data: This helps

Fig. 8: Application UI

the user to understand the data very well with the good value plotting and indicating.

Fig. 9: Thingspeak Graph

The email notification also be sent to the specific recipient and the health personals to monitor and access the health problem soon and cure it.

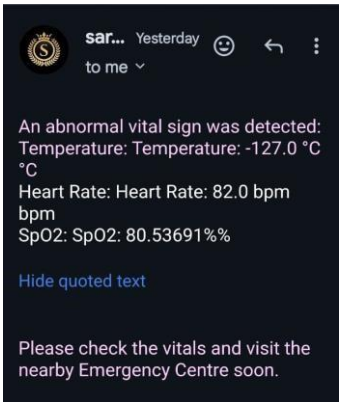

Fig. 10: Email message about health data

The app also offers the visualization graphs for the other user to understand the health data of the patients

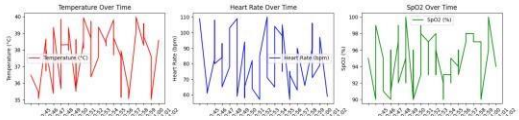

Fig. 11: Application graph

The application also has option to export the data as excel format in order to know the precise information at specific time the person can load the data and have it for future reference.

Time	Temperature (°F)	Heart Rate (bpm)	SpO2 (%)	Location
2024-11-17T02:03:47Z	104	0	78	Chennai
2024-11-17T02:03:47Z	105	30	81	Chennai
2024-11-17T02:03:47Z	99	60	81	Chennai
2024-11-17T02:04:02Z	83	70	69	Chennai
2024-11-17T02:04:02Z	91	70	79	Chennai
2024-11-17T02:04:02Z	91	82	88	Chennai
2024-11-17T02:04:02Z	102	61	90	Chennai
2024-11-17T02:04:02Z	100	79	98	Chennai
2024-11-17T02:04:18Z	100.4	84	91	Chennai
2024-11-17T02:05:17Z	82	78	97	Chennai
2024-11-17T02:05:32Z	95	73	96	Chennai
2024-11-17T02:05:32Z	98	73	96	Chennai

Fig. 12: Excel output

1) Comparison of data: The data has been collected by the medical-grade health care system with the comparison of our proposed system This data has been collected among ourself

TIME	HEARTRATE	SP02	TEMPERATURE
10:00	63	89	33
10:10	82	91	28
10:15	84	80	32
10:25	86	85	31
10:30	86	98	36
10•30	38	82	33

Fig. 13: Data from our proposed system and compared so that we can use the knowledge to overcome these deviations

TIME	HEARTRATE	SP02	TEMPERATURE
10:00	71	97	33
10:10	72	98	28
10:15	73	94	32
10:25	73	95	31
10:30	75	96	36
10•30	69	97	33

Fig. 14: Data from medical grade health care system

from these data we can see that the there is only slight deviation in the metrics even though we have faced many power management issues too. But the data from medical

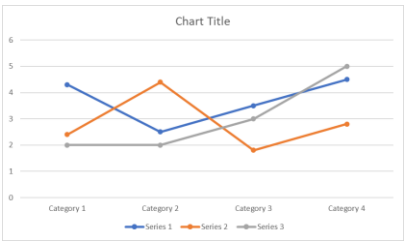

Fig. 15: Data from medical grade health care system

grade instrument is so accurate as it's using the high energy sensors which is powered by separate OS and other components which are worth a million and over

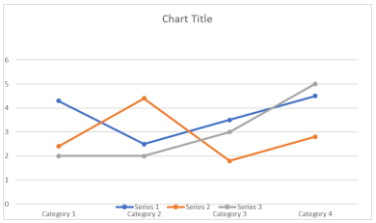

Fig. 16: Data from medical grade health care system

X. FUTURE WORK

In future we can add the system that it can operate it in separate it in own environment and merge with AI tools to process the data, and we can use more battery management strategies s that it can work for long time with just one charge at once or twice a week, we can even make the system to work without cloud by introducing the edge and fog computing so that the data processing and analysis can be done at sensor level which reduce the work load on system as well as cloud. we aim to improve the security of the system as the user's data is sensitive, we can add some blockchain techniques to process it and protect the data, this increases the trust among the users and helps to maintain the healthy trust between patient and health care monitoring device.

XI. CONCLUSION

In the end, the proposed device model can collect the data and send the data to thingspeak by calculating the Vitals like heart rate, Spo2 and temperature by collecting data by sensors like MAX30102 and DS18b20 powered by micro controller. These data gets processed in both sensor and cloud level and stored in it. The custom-made Python GUI app helps us to make the data available in detailed manner in both statistical ways and graphical ways to ensure the greater understandable data. The application also offers the user to export the data too. And for sure we faced many power related issues though while integrating multiple sensors but that machine still ultra reliable health care monitoring system which can be used for continuous health care metrics management. By this project we lay a solid foundation for the advancement of the system in this field and it also gives us the promise for future development where it can handle all the data autonomous instead of manually sharing and fetching the data.

ACKNOWLEDGMENTS

We would like to take this opportunity to thank our project guide, Dr.Hariharan I., for his able guidance

and valuable insight during the project. Further, we extend special thanks to Dr. Mohanaprasad K, Dr. Ravishankar A, Dr. Reena Monica, and Dr. John Sahaya Rani Alex for their support and encouragement. We would be thankful to VIT Chennai for providing resources and environment conducive to the proper completion of the project.

BIBLIOGRAPHY

[1] M. G. L. D. G. Stojanovic,"Wireless Health Monitoring System Using IoT," IEEE International Conference on Electronics, Communication and Aerospace Technology (ICECA), 2017.

[2] A. G. Rodrigues, A. K. N. de Lima, L. L. S. Souza, and J. M. Silva, "A Review of IoT Technologies in Health Monitoring Systems," IEEE Access, vol. 7, pp. 179870179885, 2019.

[3] S. D. Mishra and D. S. V. R. K. P. Kumar, "Realtime Health Monitoring using IoT with Cloud Integration," Proceedings of the International Conference on Smart Systems and Inventive Technology (ICSSIT), 2019.

[4] P. G. R. Krishnan, "Design and Implementation ofIoTbased Health Monitoring System," IEEE Transactions on Consumer Electronics, vol. 65, no. 4, pp. 509-518, 2019.

[5] M. S. Oliveira, "Implementation of a Remote Health Monitoring System with IoT Technology," IEEE International Conference on Networking, Systems and Security (NSysS), 2020.

[6] S. M. B. Nunes and L. M. de M. Ferreira, "A CloudBased IoT Architecture for Real-time Health Monitoring System," IEEE Access, vol. 8, pp. 159874-159883, 2020.

[7] B. J. Wei, X. F. Li, and H. Q. Yu, "Health Monitoring System Based on IoT and Cloud Computing," IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS), 2017.

[8] R. S. K. K. Sriram and P. K. R. Reddy, "Smart HealthMonitoring System Using IoT," IEEE International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2018.

[9] S. A. L. B. W. Anith, "IoT Based Health Monitoring System Using Wireless Sensors," IEEE International Conference on Circuit, Power and Computing Technologies (ICCPCT), 2017.

[10] A. P. Deshmukh, S. B. Patel, and V. B. Patil, "Designand Implementation of Wireless Health Monitoring System using IoT," IEEE Conference on Electrical, Electronics and Computer Science (EECS), 2018.

[11] A. K. Jain, A. Gupta, and M. B. Dinesh,"IoTBasedRemote Patient Health Monitoring System for Hospitals," IEEE Conference on Communication Systems and Network Technologies (CSNT), 2021.

[12] G. B. K. T. Suman,"Real-time Patient Health Monitoring System Using IoT with Cloud Integration," IEEE International Conference on Computational Intelligence and Communication Networks (CICN), 2019.

[13] T. Z. G. S. Khajuria,"Development of a RealTimeHealth Monitoring System Using IoT," IEEE International Conference on Emerging Trends in Computing, Communication and Nanotechnology (ICECCN), 2020.

[14] N. P. S. Rao and K. V. P. Reddy,"IoT-Based SmartHealthcare System for Remote Monitoring," IEEE International Conference on Smart Grid and Smart Cities (SGSC), 2020.

[15] H. K. Rao, A. L. K. Nair, and M. K. Patel, "HealthMonitoring Using Wireless Sensor Networks and IoT," IEEE Conference on Emerging Trends in Electronics, Communications and Networking (ETECN), 2019.

[16] R. K. Choudhury,"Low-Power IoT Based Health Monitoring System," IEEE International Conference on Advances in Computing, Communications, and Informatics (ICACCI), 2020.

[17] L. M. A. S. Chavan,"Wireless Health Monitoring System Based on IoT Technology," IEEE International Conference on Advances in Computing, Communications and Electronics (ICACCE), 2021.

[18] N. M. Prakash, R. S. K. Ravi, and S. P. K. Suresh,"Design of IoT-Enabled Smart Healthcare System for Remote Patient Monitoring," IEEE Transactions on Industrial Informatics, vol. 17, no. 3, pp. 1-11, 2021.

[19] A. K. Gupta, R. S. K. Srikant, and R. K. Ghosh, "HealthMonitoring System Using IoT for Chronic Disease Patients," IEEE International Conference on Electronics, Information, and Communication (ICEIC), 2018.

[20] S. G. R. Joshi,"IoT-Enabled Health Monitoring Systemwith Remote Access," IEEE International Symposium on Computing and Communication Systems (ISCCS), 2020.