Development of a Sensor-Based Heartbeat and Body Temperature Monitoring System for Remote Chronic Patients

Jimmy Obira Okello
School of Computational and Communication
Sciences and Engineering
Nelson Mandela African Institution of
Science and Technology
Arusha, Tanzania
jobira32@gmail.com

Ramadhani Sinde
School of Computational and Communication
Sciences and Engineering
Nelson Mandela African Institution of
Science and Technology
Arusha, Tanzania
ramadhani.sinde@nm-aist.ac.tz

Abstract-The growing number of chronic diseases have stretched the healthcare sector. Globally, more than 36 million deaths per year are attributed to chronic disease complications. This has increased the demand for telemedicine in managing chronic patients as they must be on continuous monitoring for a long time. The involvement of wireless sensor networks and cloud computing technology in the health sector is increasing due to the potential it possesses in remote sensing and monitoring applications. This paper presents a developed system prototype for monitoring the heartbeat rate and body temperature of chronic patients using sensors. The monitored data are sent to a cloud database in real-time via an internet connection using the ESP8266 wireless module. The approach involves connecting a heart pulse sensor, an MLX90614 contactless temperature sensor, and the ESP8266 module to the Arduino development board. The goal of this work is to create a system that interfaces chronic patients and medical personnel in an attempt to avert the effects of insufficient health facilities, especially in rural Africa. The patient's data in the cloud database can also be retrieved by medical personnel anytime in order to track the patient's conditions and to advise the patient accordingly. The sensed heartbeat and body temperature readings were processed, sent, and recorded in the cloud database effectively.

Keywords-Arduino; sensor networks; ESP8266 module; chronic diseases; pulse oximeter

I. Introduction

The management of chronic diseases has become one of the biggest challenges that health sector faces nowadays. The healthcare sector is struggling to meet the needs of people with chronic diseases as they need continuous monitoring for a long time, commonly outside the health facilities [1]. A chronic disease is basically any condition that persists for a year or more requiring continuous medical care which may curtail an individual's daily activities [2]. Eighty percent of the world's deaths from chronic diseases occur in low and middle-income countries [3] and although the biggest economic burden falls on the high-income countries, the burden on the low and middle-income countries is increasing along with population growth [4]. Globally, over 36 million deaths per year are attributed to chronic disease complications. Six out of ten adults in the USA have a chronic disease [3]. The challenges of chronic disease management have resulted to the increasing use of telemedicine. Telemedicine is the involvement of information technology in providing healthcare services to patients who may be located away from the health facilities [5]. The integration of wireless communication, cloud computing, and sensor networks in the field of medicine has facilitated and promoted telemedicine. In the last decade, there has been a rapid growth in the use of low cost wireless communication protocols and sensors in healthcare which has greatly promoted the use of telemedicine [6]. Sensor networks have also become a serious focus for research and deployment in the fields of oil and gas and agriculture [7, 8].

In this paper, some previous related works on remote patient management and monitoring were studied and analyzed to appreciate the trends and identify their gaps and a system that uses multiple sensors for collecting data, virtual instrument software for data processing, and wireless data transmission is proposed. A system that deploys wireless technology to detect any abnormalities in the patient bio signal and sends SMS alerts to doctors using Global System for Mobile Communication (GSM) has been proposed in [9]. A GSM based system that monitors the patient's health condition and passes messages to the doctor's mobile phones has been developed in [10]. However, GSM technology is becoming obsolete in most countries. A mobile application based on android for remote patient monitoring system has been proposed in [11] that focuses in monitoring the body temperature of patients and displaying the data on a mobile application. However, this does not cater for users of other mobile phones that use different operating systems. A system that involves electrical signals to monitor heart disease has

Corresponding author: Jimmy Obira Okello been developed in [12]. However, this system comes with a bulky strap that must be put around one's chest making it a bit uncomfortable. A Raspberry Pi based system that automatically monitors patient's heart rate, body temperature, respiration rate, and body movements has been proposed in [13]. However, the Raspberry pi is an expensive and complex board to program and use, making it a costly approach.

Medical professionals use body temperature, heart beat rate, blood pressure, and pulse rate for the tracking of a patient's health condition. Body temperature is one of the main indicators of normal or abnormal body functioning. Chronic patients require continuous monitoring of their body parameters, record keeping, and informing the doctors in case of any abnormality. This is sometimes a manual process and requires proper record keeping which is still a challenge as the doctors do not receive the data in real time for immediate response while sometimes the records are lost by patients. Taking into consideration the challenges of monitoring, record keeping, and real time transfer of data, the authors of this paper found it extremely paramount to develop a system that monitors and relays chronic patient's data in real time. Some of the research carried out in this area of patient's condition monitoring involves the use of expensive systems such as drones to capture data or GSM technology. However, little attention has been put on utilizing the low cost ESP8266 wireless protocol, MLX90614 contactless temperature sensor and heart pulse sensors in patient remote monitoring, and the use of free cloud services for proper record keeping and virtualization. The present work brings in the element of using low cost, low power heart pulse sensor, MLX90614 contactless temperature sensor, ESP8266 wireless module, and free cloud database. This developed system allows the extraction of a patient's data in a spreadsheet document which facilities an offline analysis. The system allows real time transmission of a patient's data to the database using a cheap wireless module thus allowing quick and easy response from the doctors. The major contribution of this paper is that the proposed system interfaces chronic patients and medical personnel that will help averting the effects of poor health facilities in areas such as the rural Africa.

II. MATERIALS AND METHODS

This paper proposes a monitoring system that detects the body temperature and the heartbeat rate of chronic patients with the aim of improving timely intervention to patient's conditions. The developed system prototype has two sides. On the patient's side the readings are displayed on a Liquid Crystal Display (LCD) screen whereas on the doctor's side the readings are stored in the ThingSpeak cloud database. The developed system uses the Arduino development board, a heartbeat sensor and a MLX90614 contactless temperature sensor, the ESP8266 wireless module, and a cloud database. The sensors collect health indicators from the patient and transmit them to the ThingSpeak database using the ESP8266 wireless module. The data from the cloud database can be retrieved any time by the doctors. The heartbeat sensor was connected to pin A0 of the Arduino Uno whereas the MLX90614 temperature sensor's SDA and SCL were connected to the Arduino's SDA and SCL pins respectively. The Arduino communicates with the ThingSpeak database remotely through an internet connection. The block diagram of the proposed system prototype is shown in Figure 1. The system comprises of hardware and software components and their details are explained below.

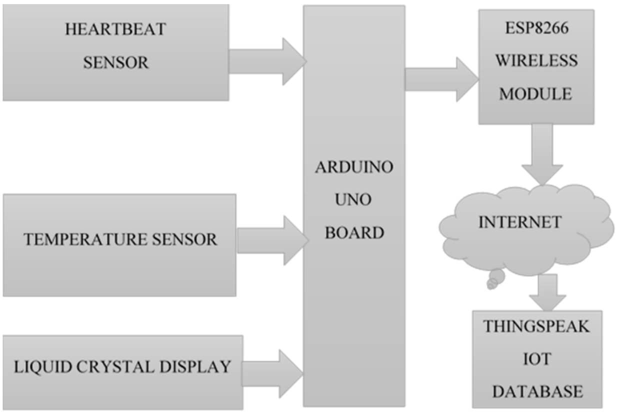

Fig. 1. Block diagram of the developed system.

A. Hardware Requirements

The hardware components of the system are summarized in Table I. The heartbeat sensor detects the heartbeat rate whereas the MLX90614 sensor detects the body temperature of the patient. These sensors are attached to an Arduino board where the collected readings are processed. The ESP8266 wireless module is attached to the Arduino board and is used to send the collected sensor data to the cloud database.

TABLE I. HARDWARE REQUIREMENTS SUMMARY

S/N	Required component	Quantity
1	Arduino Uno	1
2	Heartbeat sensor	1
3	MLX90614 sensor	1
4	ESP8266 module	1
5	Breadboard	2
6	Battery	2
7	Jumper wires	1 packet
8	LCD	1
9	Potentiometer	1

1) Arduino UNO

Arduino Uno is an embedded development board having different pins for analog, Pulse Width Modulation (PWM) and digital signal interfacing. It includes an Integrated Development Environment (IDE) for programing. The Arduino development board is used to interface sensors which are used to collect data from the environment. The Arduino board contains an ATMEGA32 microprocessor which was programmed to sense body temperature and heartbeat. The Arduino Uno was used in this project because of its cost, simplicity, and availability.

2) Heartbeat Sensor

Pulse Sensor is a heartbeat sensor designed for Arduino. It can be used to incorporate heartbeat monitoring projects in real time. The sensor side with the heart logo is usually placed onto a fingertip or earlobe to take measurements of the heartbeat. The sensor has three jumper cables which are plugged into Arduino pins. The sensor side contains the LED indicator and an ambient light sensor. The light from the LED strikes the fingertip, earlobe, or other body part and the reflected light is then used by the sensor in order to determine the heartbeat rate [14]. In this project, the sensor signal cable was connected to pin A0, the ground pin to GND pin of the Arduino, and the VCC pin was connected to the 3.3V pin of the Arduino.

3) MLX90614 Contactless Temperature Sensor

The MLX90614 contactless temperature sensor measures ambient and target temperatures. It utilizes infrared (IR) to be able to detect the ambient and target temperature without coming into physical contact. It utilizes the inter-integrated circuit (I2C) serial communication protocol when communicating with microcontrollers or other devices. It consists of four pins: SDA, VIN, GND, and SCL. The SDA pin is used for data transfer, the SCL pin is used for clock transfer during I2C serial communication, and the VIN pin is for power supply. This sensor can be used in a number of commercial, healthcare, and household applications. This sensor measures target temperature without making physical contact. All human beings emit IR energy and the MLX90614 sensor can calculate the temperature of a target from this emitted IR energy since the temperature is directly proportional to it. For this project, the SDA pin was connected to the SDA pin of Arduino, the SCL was connected to the SCL of the Arduino, while the VIN pin was connected to 5V, and the GND to the GND of the Arduino.

4) ESP8266 Wireless Module

The ESP8266 module can be used as standalone host receiving internet connection from a router or as a master providing internet connection to other devices. When the ESP8266 module is used to host application, it is booted directly from an external flash. It has on-chip memory to enhance the capacity of the developed applications. This module has a Radio Frequency (RF) antenna, filters, and power management mechanisms within a small board. ESP8266 also integrates a 32-bit processor, on-chip Static Random Access Memory (SRAM) with the ability to communicate with external devices and sensors using the General Purpose Input Output (GPIO) pins. For this project the ESP8266 wireless module was used and the pins were connected as follows: Tx pin to pin 2 of Arduino, Rx to pin 3 of Arduino, pins VCC and CH_EN connected to 3.3V and Reset and ground pins to the GND of Arduino.

5) Liquid Crystal Display

The LCD used for this project, is a 16×2 LCD which displays only 32 characters. This LCD is a 16 pin device which can display values when programmed with the help of Liquid Crystal library. The 16 pins include: VCC pin which is the power supply for the LCD and is usually connected to the Arduino's 5V, while the GND pin goes to Arduino's ground pin (0V). The Vo pin controls the contrast of the LCD and is connected to 5V for maximum brightness or to the signal pin of potentiometer for brightness adjustments. The Register Select (RS) pin is used to select between control command signals for LCD and data. The RS pin is connected to ground (LOW) to send command signals to the LCD and to 5V (HIGH) to send data. Read/Write (R/W) pin is used to select read or write mode. Connecting the pin to HIGH reads data from the LCD whereas setting it to LOW sends data to the LCD. Enable (EN) pin is used to enable or disable the LCD using HIGH and LOW signals respectively. Anode (A) and cathode (K) pins are used to power the LCD backlight by connecting to VCC and GND pins of Arduino respectively. The data buses (D0-D7) pins are used to carry data and commands between the LCD and Arduino. There are two modes used to send data, namely the 8-bit and the 4-bit modes. For the 8-bit mode, a byte is sent at once in pins D0 to D7 whereas in the slower 4-bit mode, 4 bits are sent in pins D4 to D7 twice. In this project, a 4-bit mode connection was used in order to reduce the number of cables used. Pin 1 was connected to the ground, VCC to 5V, Vo to signal pin of potentiometer, R/W to the ground, EN to pin 11, RS to pin 12, D7, D6, D5, D4 to Arduino's pins 4, 5, 6, 7 respectively, the anode to 5V and the cathode to ground.

6) Potentiometer

The potentiometer is a variable resistor, usually with 3 pins: GND, VCC, and output. In this project, the brightness of the LCD was controlled by a 10K potentiometer. The potentiometer ground pin was connected to Arduino's GND pin and the VCC to 5V of Arduino while the output pin was connected to the Vo of the LCD.

7) Arduino Integrated Development Environment

The Arduino IDE is a software which includes code editor, compiler, and uploader to upload programs to a board. The text editor is used for coding and the codes are uploaded to the Arduino hardware via serial communication.

8) ThingSpeak IOT Cloud Database

ThingSpeak is a web service that allows the collected sensor data to be stored in the cloud and aids the development of Internet of Things (IoT) applications [15]. It works with different development boards like Arduino, Raspberry Pi, and LPC1768. It uses REST API and HTTP that allows working with various programming languages. The sensor data from the Arduino are transferred to the cloud database for storage, processing, and visualization. ThingSpeak service also allows performing online analysis on the data. During this project, one channel called Contactless Sensor was set up in ThingSpeak having three fields. The first field is for the ambient temperature, the second field for body temperature, and the third field for heartbeat values. Each channel has a unique channel identifier (ID) and Application Programmable Interface (API) write and read keys and must be included in the coding. The write API key allows data to be sent to a channel while the read API allows data from a personal channel to be viewed by others. For this project the write API "Y2HQ4ZOYATCB8EE4" was used to allow sensor data get sent to the ThingSpeak database.

B. Validation

Validation is the procedure of checking key indicators aimed at verifying whether the specifications for the system match the intended purpose. It is meant to check whether the user specifications and requirements are realized [16]. For the developed system, different approaches were used for validation, including unit testing, integration testing, and system testing. The summary of the testing is shown in Table II.

TABLE II. SYSTEM TESTING SUMMARY

Requirement	Description	Score
Device powering up LED indicator	Different devices are connected to the power supply and the power indicator LED is observed.	Pass
Interfacing sensors with Arduino board	After programming, the output of the sensor measurements was checked on Arduino serial monitor.	Pass
LCD was added to the setup	The measured readings from the sensor were programmed to be shown on the LCD and this was observed.	Pass
Visualization of data	A hardware setup was interfaced with the IoT cloud database. The sensor data were visualized in graphical form.	Pass
Excel report	After visualization, the sensor data were extracted in Excel format for further local analysis.	Pass
Validation of the accuracy of the readings	The system readings were compared to the readings from other devices at the hospital.	Pass

C. *Unit Testing*

Unit testing is a technique used to verify the functional correctness of each module of the system [17]. For the developed system, the following modules were tested: ESP8266 module, heartbeat sensor, contactless temperature sensor, LCD, and potentiometer. Each was tested for proper powering and the interfacing with Arduino was checked for proper functionality.

D. *Integration Testing*

This is a testing technique that is used to verify whether the modules tested in the unit testing stage can be integrated to work smoothly and properly without problems [18]. Different functional modules were integrated and tested to confirm whether they were able to work together properly. For instance, the two sensors were all connected in addition to ESP8266, potentiometer and LCD in a bottom up approach and tested for any possible errors. The integration test was done by combining modules bit by bit to make sure they were working properly.

E. *System Testing*

System testing is a technique of putting the system as a whole and carrying out tests to proof whether it is working as planned and that it meets the intended end user's requirements. The combined outcome of all the modules was successfully tested during this testing stage. System testing does not look at the structural dimension of the program code but rather the visible functional correctness of the end product. The hardware was interfaced with ThingSpeak cloud database for data storage and virtualization. After the interfacing, the system was tested as a whole and was found to perform according to plan.

III. RESULTS AND DISCUSSION

The developed system prototype that monitors body temperature and heartbeat rate for chronic patients can be seen in Figure 2. A thingSpeak cloud database was utilized to store and virtualize the patients' data. The aggregated data can also be extracted in excel format for offline analysis, allowing the doctors to track chronic patient's data in real time and keeping track of the patient's health conditions. A 16×2 LCD was used to display the patient's body temperature and heartbeat readings. The patients can see their temperature and heartbeat readings while at the same time the readings are sent to the database. ThingSpeak was configured and 3 fields were created in order to receive ambient temperature, target temperature, and heartbeat readings from different remote sensors. The data were analyzed and organized in graphical form in terms of a line graph in different fields. The fields receive real time data from the sensor nodes corresponding to patient's body readings. Figures 3 and 4 show the body temperature and heartbeat data stored and visualized on thingSpeak cloud database.

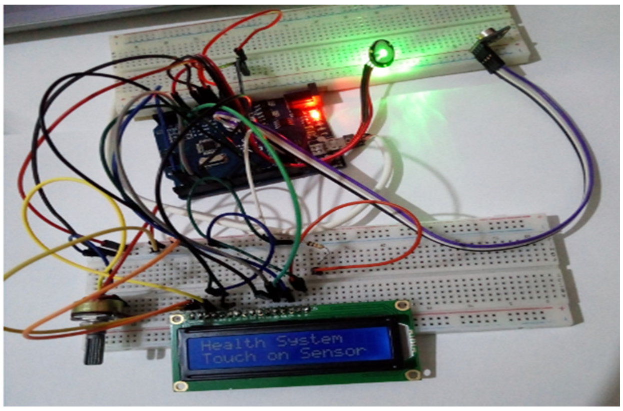

Fig. 2. Developed system prototype.

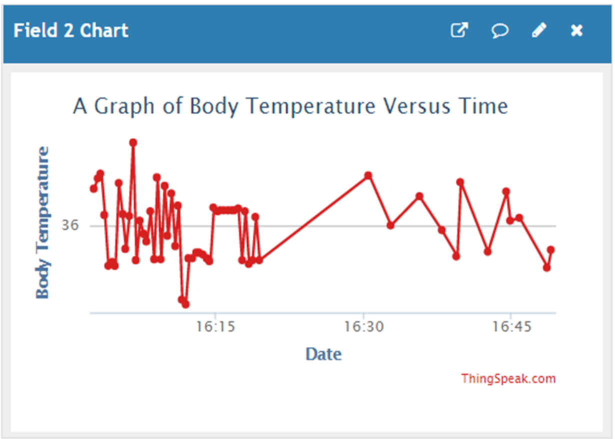

Fig. 3. ThingSpeak capture of body temperature readings.

Seventy people were examined by the system and their body temperature and heartbeat readings were displayed on the LCD and at the same time sent to the cloud database. Normal body temperature is 37°C and a person is said to have fever when the temperature is greater than 38.0°C [19] while body temperature between 35.5°C to 37°C is usually regarded as fine when assessing a patient's condition. Therefore, the temperatures taken by the developed system as shown in Figures 3 and 5 mostly fall in the normal body temperature category. Whenever the heart pumps blood, there is an increase in oxygenated blood whereas a decrease in oxygenated blood occurs when the heart relaxes. The heartbeat rate measured in Beats Per Minute (BPM) is basically determined by the amount of time between the increase and decrease in the oxygenated blood. The normal BPM for adults is about 72, for children it is about 90, and for babies about 120 [12]. The BPM values shown in Figures 4 and 6 were taken from adults and children only.

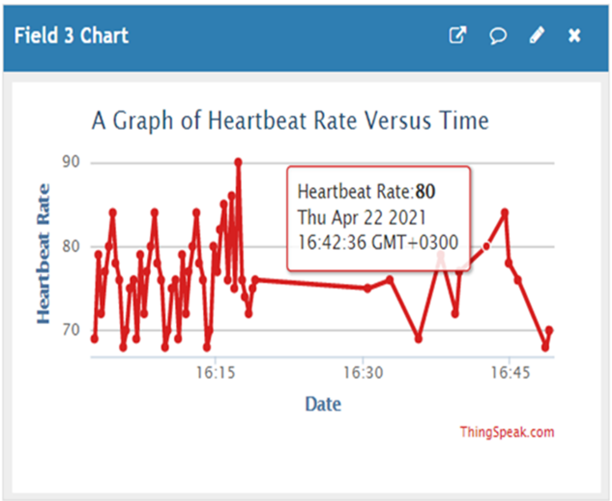

Fig. 4. ThingSpeak capture of heartbeat readings.

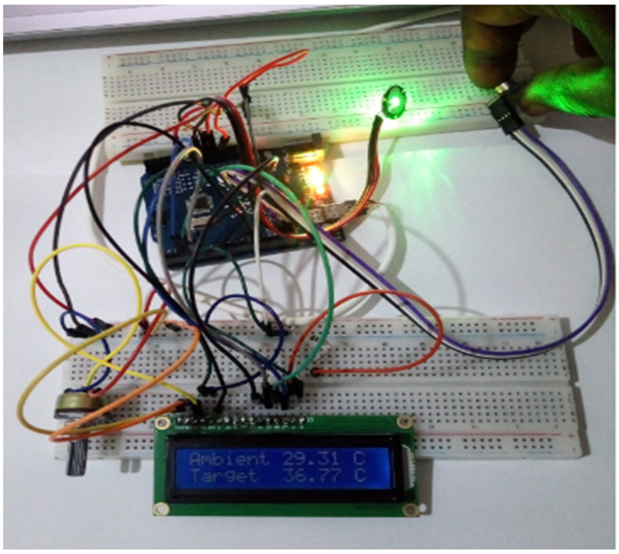

Fig. 5. Temperature taken by the system prototype.

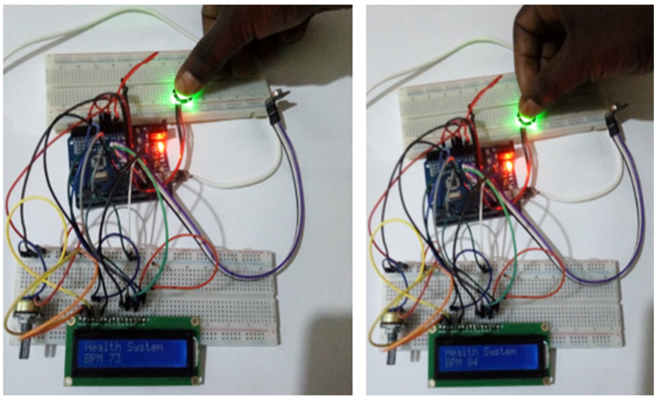

Fig. 6. Heartbeats taken by the system prototype.

IV. Conclusion

The scarcity of health facilities in rural areas, especially in Africa, is a serious hindrance to healthcare access. This poses serious risks and is even more paramount in chronic patients. The advancement of wireless sensor networks and cloud computing technology has created an opportunity for adopting telemedicine. In this paper, a system capable of monitoring body temperature and heartbeat rate of chronic patients has been developed. This system is capable of sending patients' data to a cloud database in real time. This system will bridge the gap in healthcare access caused by the insufficient healthcare facilities and will further improve the way of managing chronic patients as well as dealing with any sudden changes in the patient's health condition in order to avert health complications. The developed system consists of a heart pulse sensor, an MLX90614 contactless temperature sensor, an ESP8266 wireless module, an Arduino board, and a cloud database. The database virtualizes the received data in graphical and spreadsheet formats. The system has the potential of adding more health indicators for monitoring. Policy makers, governments, and different stakeholders such as the internet service providers should increase the internet coverage to enable such systems to be used in rural areas where health facilities are usually distant. Various stakeholders in the health sector should be able to sensitize the citizens on the use of telemedicine as an alternative to physical visits to health facilities. This will save time and enable quality services to reach even remote areas because telemedicine doctors will be able to serve patients from any location. In the future, more research can be done in the area of adopting sensors to improve remote patient monitoring. This system can be further improved by incorporating a dashboard pop up mechanism for immediate alerts.

Acknowledgment

The authors wish to thank the Center of Excellence for ICT in East Africa (CENIT@EA) and the Nelson Mandela African Institution of Science and Technology (NM-AIST) for their support. Finally, the authors wish to thank the Medical Concierge Group Uganda.

REFERENCES

[1] D. Beran, "Health systems and the management of chronic diseases: lessons from Type 1 diabetes," *Diabetes Management*, vol. 2, no. 4, pp. 323–335, 2012.

[2] "Chronic conditions prevention and management," *Northern Territory Government*, Apr. 28, 2021. https://health.nt.gov.au/professionals/chronic-conditions-health-professionals/chronic-conditions-prevention-and-management (accessed Jun. 25, 2021).

[3] "Chronic Diseases in America." CDC's National Center for Chronic Disease Prevention and Health Promotion, 2019.

[4] P. Jha, R. Nugent, S. Verguet, D. Bloom, and R. Hum, *Disease Control Priorities in Developing Countries*, 3rd ed. 2013.

[5] S. Sherin, R. J. Neethu, M. Yedu, V. R. Anand, and M. J. Jayashree, "Remote Patient Monitoring System," *International Journal of Distributed and Parallel Systems*, vol. 3, no. 5, pp. 99–110, Sep. 2012.

[6] P. Kakria, N. K. Tripathi, and P. Kitipawang, "A Real-Time Health Monitoring System for Remote Cardiac Patients Using Smartphone and Wearable Sensors," *International Journal of Telemedicine and Applications*, vol. 2015, Dec. 2015, Art. no. e373474**,** https://doi.org/10.1155/2015/373474.

[7] M. K. Villareal and A. F. Tongco, "Remote Sensing Techniques for Classification and Mapping of Sugarcane Growth," *Engineering, Technology & Applied Science Research*, vol. 10, no. 4, pp. 6041–6046, Aug. 2020, https://doi.org/10.48084/etasr.3694.

[8] B. F. Alshammari and M. T. Chughtai, "IoT Gas Leakage Detector and Warning Generator," *Engineering, Technology & Applied Science Research*, vol. 10, no. 4, pp. 6142–6146, Aug. 2020, https://doi.org/10.48084/etasr.3712.

[9] N. R. J. Madhu, "IoT based remote patient health monitoring system," M.S. thesis, Kansas State University, Manhattan, KA, USA, 2017.

[10] P. Srinivasan, A. A. Khan, T. Prabu, M. Manoj, M. Ranjan, and K. Karthik, "Heart beat sensor using fingertip through Arduino," *Journal of Critical Reviews*, vol. 7, no. 7, pp. 1058–1060.

[11] S. ul M. Abbasi, M. Daud, S. Ali, and A. Q. Ansari, "Design of Android-Based Remote Patient Monitoring System," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 6, 2018, https://doi.org/10.14569/IJACSA.2018.090627.

[12] B. Mallick and A. K. Patro, "Heart rate monitoring system using finger tip through Arduino and processing software," *International Journal of Science, Engineering and Technology Research*, vol. 5, no. 1, pp. 84–89, Jan. 2016.

[13] S. Rajkumar, M. Srikanth, and N. Ramasubramanian, "Health monitoring system using Raspberry PI," in *2017 International Conference on Big Data, IoT and Data Science (BID)*, Pune, India, Dec. 2017, pp. 116–119, https://doi.org/10.1109/BID.2017.8336583.

[14] "MAX30100 Pulse Oximeter and Heart-Rate Sensor IC for Wearable Health." Maxim Integrated.

[15] S. Pasha, "Thingspeak Based Sensing and Monitoring System for IoT with Matlab Analysis," *International Journal of New Technology and Research*, vol. 2, no. 6, pp. 19–23, Jun. 2016.

[16] M. Kamalrudin and S. Sidek, "A Review on Software Requirements Validation and Consistency Management," *International Journal of Software Engineering and Its Applications*, vol. 9, no. 10, pp. 39–58, Oct. 2015.

[17] T. Dybå and T. Dingsøyr, "Empirical studies of agile software development: A systematic review," *Information and Software Technology*, vol. 50, no. 9, pp. 833–859, Aug. 2008, https://doi.org/10.1016/j.infsof.2008.01.006.

[18] S. Nidhra and J. Dondeti, "Black Box and White Box Testing Techniques - A Literature Review," *International Journal of Embedded Systems and Applications*, vol. 2, no. 2, pp. 29–50, 2012, https://doi.org/10.5121/ijesa.2012.2204.

[19] M. Sund Levander, "Accuracy when assessing and evaluating body temperature in clinical practice. Time for a change?," in *EAT2012 Book of Proceedings*, Jan. 2012, vol. Appendix 1, pp. 25–32.