Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)
Vol. 7, No. 1, April 2021, pp. 30-42
ISSN: 2338-3070, DOI: 10.26555/jiteki.v7i1.20209

Design and Implementation of a Smart, Interactive, and Portable System for Monitoring of Human Vital Signs

Haider Ismael Shahadi [1], Maha Khalid Kadhim [2], Nawal Mousa Almeyali [3], Ali Thamir Hadi [4]
[1]Department of Electrical and Electronic Engineering, University of Kerbala, Karbala, 56001, Iraq
[2]Department of Biomedical Engineering, University of Kerbala, 56001, Karbala, Iraq
[3]College of Medicine, University of Kerbala, 56001, Karbala, Iraq
[4]Department of Dialysis, Marjan Hospital, Hillah, 51001, Babil, Iraq

ARTICLE INFO

Article history:

Received March 12, 2021
Revised March 24, 2021
Accepted March 28, 2021

Keywords:

Patient monitor;
Smart system;
human vital signs;
Internet of things;
ThingSpeak;
ESP32-microcontroller;
Electrocardiography (ECG);
Heartbeat rate sensor;
Body sensor

ABSTRACT

Smart systems are characterized by their efficiency, high accuracy, and cost reduction. One of the important fields in which the smart system is used is health care, especially monitoring of human vital signs. In general, the conventional patient monitor is expensive, cannot be used for remote monitoring, and non-interactive. In many situations, it requires remote and portable monitoring for patients, such as in case of the area is outside the medical services, infected diseases (e.g., COVID-19, 20), and difficulties of a patient transferred. This paper proposes a smart, interactive, and portable monitor for vital human signs based on the Internet of Things (IoT). The proposed monitor is cheap and easy to use either directly by doctors and nurses or remotely by any person. The proposed system is designed using ESP32-microcontroller and vital-sign sensors. It measures three important vital signs, including heart rate, body temperature, and Electrocardiography (ECG), as well as the environment temperature of the patient. The measured signs can be monitored from anywhere in the world through a smartphone application in real-time. Furthermore, the doctor can send instructions and descriptions to the patients in real-time using the same phone application that is designed in this work.

Haider Ismael Shahadi,
Department of Electrical and Electronic Engineering, University of Kerbala, Al-Hillah Street, 56001, Karbala, Iraq.
Email: haider_almayaly@uokerbala.edu.iq

1. INTRODUCTION

Smart system is a new technology term that seems in many applications in our daily life such as in energy management, medical applications and healthcare management, industrial automation, and automotive. Smart systems have abilities to resolve complex problems, including taking over human cognitive tasks [1, 2]. In medical and healthcare applications, smart devices are mainly devoted to improving treatments and healthcare. The key components of the smart system in healthcare products are biomedical sensors. These sensors are miniaturized based on Micro-Electro-Mechanical System (MEMS) technology [3] in order to minimize the physical effect on the biologic system.

The vital signs are proof of the body's current physical functioning. They refer to a group of the most important medical signs that indicate the status of the body's vital (life-sustaining) functions. These measurements are taken to help assess the general physical health of a person, give clues to possible diseases, and show progress toward recovery. The normal ranges for a person's vital signs vary with age, weight, gender, and overall health.

Employing the smart system in vital signs measurement and monitor becomes an important research area in order to resolve some important problems on a conventional monitoring system. These problems include high prices of the conventional devices, difficult to be used by ordinary people, and none mobile. Therefore, several works published recently to solve some of the above problems.

In [4], a patient monitor based raspberry-pi and Lab-View have been designed. The designed patient monitor measures pulse rate, humidity, and oxygen saturation level. DHT11 sensor is used to measure temperature and humidity, and pulse and oximeter sensors are used to measure pulse rate and oxygen saturation level, respectively. The graphical representation of the patient's data is displayed in the doctor's console using Lab-View, and the critical readings of the patient are sent to the doctor's contact.

The researchers in [5] have proposed a patient monitor based on an ATMEGA8 microcontroller and GSM modem. The main idea of this system is to continuously monitor patients over the Internet. The temperature sensor (LM35) and pulse oximeter sensor (TCRT1000) are connected to the microcontroller, and the output is displayed on a screen. The collected data from the sensors are sent to the Web page through the modem.

Authors in [6] have added closed-circuit television (CCTV) cameras and smartphones application as well as some vital signs sensors in order to monitor the patients that have Infectious diseases. Authors in [7] have presented an Arduino-based IoT project used for monitoring the vital signs of humans. Also, in [8], a patient monitor is based on an Android platform application that provides the end-user with visualization ECG. Table 1 shows a summary of the related work and illustrates the specifications and drawbacks of each work.

In this paper, we attempt to overcome the drawback of the related work by combining a smart system and cloud computing to design and implement an efficient, portable, and cheap patient monitor system. The designed system employs a cheap microcontroller (ESP32) that has the ability to connect to the Internet through its built-in Wi-Fi communications. A spatial Android application is designed and implemented to be an interface and communication tool among the designed device, patient, and doctor—the data collected from body temperature, environment temperature, heart pulse, and ECG sensors. The microcontroller sends the collected data to the cloud in a continuous way in case of measuring. In the cloud, the new data, as well as the history of the measurements, are saved with its reading time. The Android application is updated automatically with the new reading of the cloud.

Table 1. Summary of the related work.

Ref.	Sensors	Method	Specification	Drawbacks
[4]	Temperature and humidity (DHT11), pulse, and oximeter.	Raspberry-pi collects data from sensors and display on Lab-View.	Graphical patient's data is displayed for the doctor.	No web server, no remote access to the patent's data.
[5]	Temperature (LM35), Pulse oximeter (TCRT1000), Liquid Crystal Display (16x2), GSM modem.	ATMEGA8 microcontroller collects the data from the sensors and sends the data to a Web page.	The sent data are encrypted, and it can be accessed anytime by the doctors using a unique URL link.	The system does not have a smartphone application. The cost still not cheap
[6]	SpO2, ECG, heart rate, blood pressure, and camera.	A technology for remote monitoring utilizing CCTV cameras and smartphones. Also, the data of the sensors are collected and send to the phone.	Patient vital signs can be visualized by smartphone. So, it alerts and manages patient care without being physically present bedside.	This system does not connect to the webserver to monitor and analyses the history of the patient vital signs
[7]	Ultrasonic sonar, pressure pad Pulse, temperature.	Arduino-Uno collects data from all sensors, then sends them to the cloud to save data. The doctor can display the data using the cloud site. .	The project provides fast data collection. The cost is fair.	Sonar sensor has a slow sensing rate, and it is very sensitive to temperature changes. Also, no smartphone application
[8]	Temperature, heartbeat rate, blood pressure.	Arduino-mega collects the data patient from the sensors and sends them to the webserver using Wi-Fi.	Managers/doctor can monitor the data through smartphone.	The system can save only part of the data in the webserver.

2. THE PROPOSED SYSTEM

The proposed system is designed to be simple, portable, and efficient based on IoT and smart systems. The system presents a solution to the patient monitoring by the doctor remotely and in real-time. The system includes three important aspects. The first is the hardware components of the (Patient Monitor Device), through which data is collected, then data uploaded to the Internet. The second represents the cloud for saving and analyzing the data. ThingSpeak is used as a cloud platform that contains the database (patient serial number and medical history). The third aspect is the web interface and smartphone application that displays the vital signs reports to the doctor and other users and allows by conducting between the doctor and patient. The overall system block diagram of the proposed patient monitor is shown in Fig. 1.

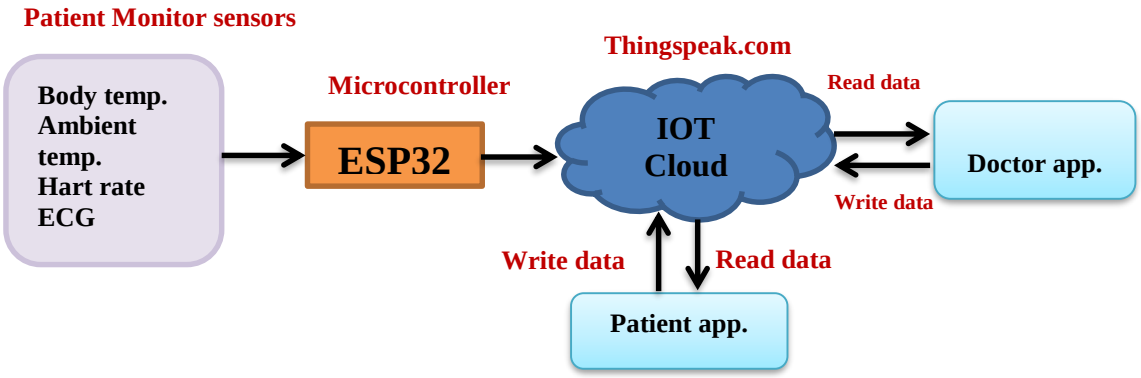


Fig.1. The proposed system of the patient monitor.

2.1. Hardware Description

The main parts in the hardware of the designed patient monitor are including microcontroller (ESP32), heartbeat rate sensor (APDS-9008), ECG sensor (AD8232), and temperature sensor (MLX90614). The microcontroller collects data from the sensors and sends it to the cloud platform using Wi-Fi wireless connection. The details of the hardware and connection method of each sensor are shown in the following subsections:

2.1.1 ESP32 Microcontroller

ESP32 is a powerful system on chip (SoC) microcontroller with integrated Wi-Fi (IEEE 802.11), dual-mode Bluetooth version 4.2, and a variety of peripherals. It is an advanced successor of the 8266 chip, primarily in the implementation of two cores clocked in different versions up to 240 MHz. Compared to its predecessor, except for these features, it also extends the number of GPIO pins from 17 to 36, and it is equipped with 4MB of flash memory [9]. ESP32 includes two cores (Xtensa LX6 processor made with 40 nm technology). CPU cores can be individually controlled. There is 520 KB of on-chip SRAM for data and instructions available. Some SoC modules, such as ESP32-Wrover, feature 4 MB of external SPI flash and an additional 8 MB of SPI PSRAM (Pseudo static RAM) [9].

The most important features of ESP32 are including small size, low power consumption, wireless connectivity (built-in chips, Bluetooth and Wi-Fi), and low price. Therefore, we employ this efficient microcontroller to get a portable device with minimum cost.

2.1.2 Heartbeat Sensor

Heartbeat rate is a vital sign, and its measurement is a standard clinical procedure. It is typically measured as the average number of pulses that occur per minute. This is the Pulse Rate Average (PRA) and is expressed in units of /min – pulses per minute. When a heartbeat occurs, blood is pumped through the human body and gets squeezed into the capillary tissues. The volume of these capillary tissues increases as a result of the heartbeat, but in between the Heartbeats (the time between two consecutive heartbeats), the volume inside capillary tissues decreases. The pulse sensor module has a light that helps in measuring the pulse rate. The change in volume between the heartbeats affects the amount of light that will transmit through these tissues [10]. In this work, the APDS-9008 heart rate sensor is used. The sensor consumes very low power at a voltage of 3-5 Volt. Therefore, it is powered directly from the ESP32 using Vcc (5 or 3.3 Volt) and GND pins. Once powered connect the signal pin of the sensor is connected to the analog to digital converter (ADC) pin of the

ESP32 (GPIO 27) to get the reading of the sensor. Fig. 2 shows the method of connection for the heartbeat sensor to the ESP32 microcontroller.

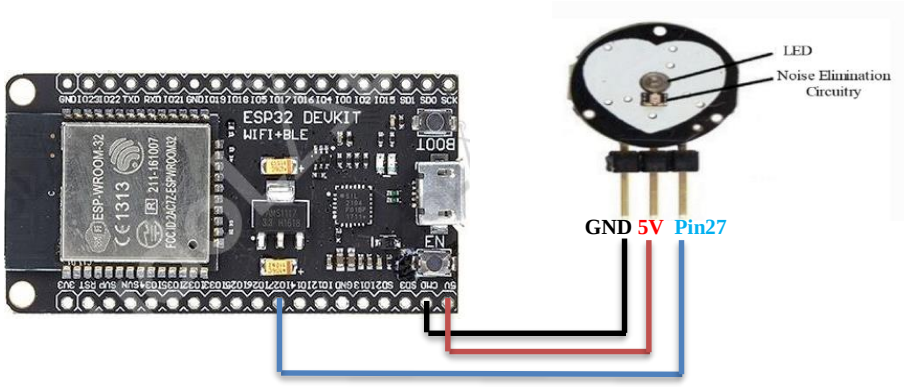


Fig.2. Connection heartbeat sensor to ESP32 microcontroller.

2.1.3 Electrocardiograph ECG Sensor

The AD8232 sensor is an integrated signal conditioning block for ECG that has been used in the design of the proposed system. This sensor is designed to extract, amplify, and filter small bio-potential signals in the presence of noisy conditions, such as those created by motion or remote electrode placement. The AD8232 module breaks out nine connections from the IC that you can solder pins, wires, or other connectors to. SDN, LO+, LO-, OUTPUT, 3.3V, GND provide essential pins for operating this monitor with an Arduino or other development board. Also provided on this board are RA (Right Arm), LA (Left Arm), and RL (Right Leg) pins to attach and use your own custom sensors. Additionally, there is an LED indicator light that will pulsate to the rhythm of a heartbeat [11].

In this work, the connection of AD8232-ECG Sensor with ESP32 microcontroller is achieved, as shown in Fig. 3. The sensor is powered by 3.3V from ESP32 and connects the GND-pin of the sensor to one of the GND-pins of ESP32. The output pin of AD8232 is an analog signal that is connected to the SVP-pin of ESP32 that known as A0. Also, LO+ and L0- of AD8232 are connected to D2 and D3 of ESP32, respectively.

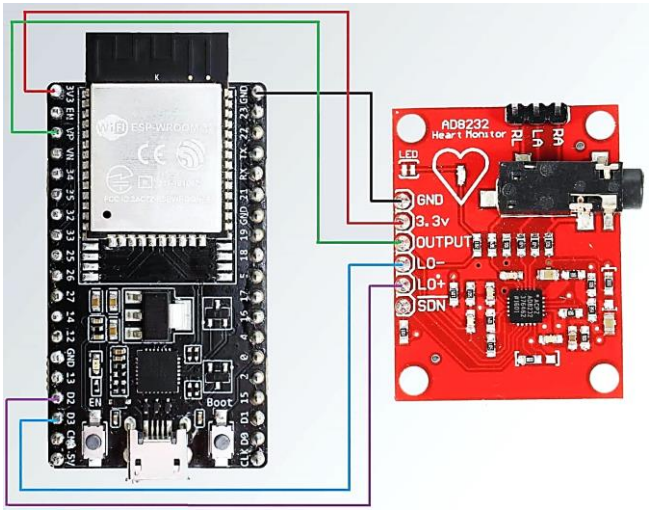


Fig. 3. Connection of ECG sensor to ESP32 microcontroller.

2.1.4 Body Temperature Sensor (MLX90614)

MLX90614 is an infrared thermometer for non-contact temperature measurements. Both the IR-sensitive thermopile detector chip and the signal conditioning ASIC are integrated into the same TO-39 CAN. MLX90614 has a low noise amplifier, 17-bit ADC, and powerful DSP unit, thus achieving high accuracy and resolution of the thermometer. The thermometer comes factory calibrated with a digital bus output giving full access to the measured temperature in the complete temperature range(s) with a resolution of 0.02°C [12]. The user can configure the digital output to be pulse width modulation (PWM). As a standard, the 10-bit PWM is

configured to continuously transmit the measured temperature in the range of -20 to 120°C, with an output resolution of 0.14°C [12].

This sensor needs a 4-wire connection. The VIN-pin on the MLX90614 sensor is connected to 3.3 Volt-pin from the ESP32 board and GND-pin from the sensor to GND-pin from the ESP32 board. Then, we connect the SDA (Serial Data) pin from the temperature sensor to the SDA-pin from the ESP32 board (pin 21), and the SCL (Serial Clock) pin from the sensor to the SCL-pin from the ESP32 board (pin 22). Fig. 4 shows the connection of the body temperature sensor to the ESP32 microcontroller.

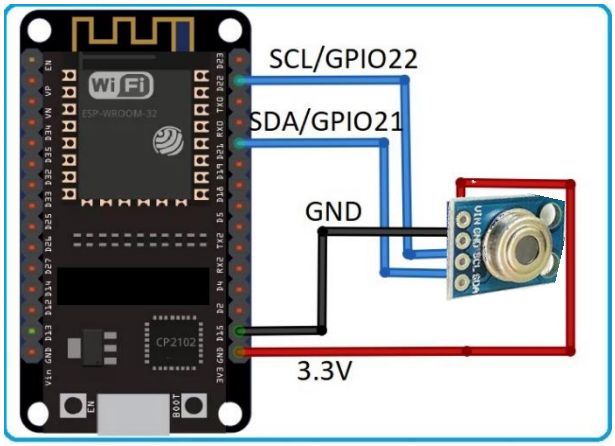


Fig. 4. Connection of the body temperature sensor with ESP32 microcontroller.

2.2. Software Description

Three software platforms are used in this work in order to program and interfacing the overall system in a remote manner. Firstly, Arduino IDE software is used to programming the hardware of the patient monitor system. Secondly, the ThingSpeak platform is used to received data from the microcontroller and create a database on the cloud. Also, it is used in data analysis and notification in emergency cases. Thirdly, the MIT-APP platform is used to design and implement a special Android application that allows the doctor to monitor remotely the patient vital signs in real-time from anywhere from the world. Furthermore, the doctor can see the history of the patient's vital signs and send instructions and descriptions to the patient through the application. More information about the used platforms is shown in the next subsections.

2.2.1 Arduino IDE

The Arduino IDE is an integrated development software present for Arduino devices and helps to code the Arduino microcontrollers to interface the sensors and other types of components and perform the operation on both local and global domains with the help of library functions. Arduino IDE is open-source software that is mainly used for writing and compiling the code into the Arduino Module. It is official Arduino software, making code compilation too easy that even a common person with no prior technical knowledge can get their feet wet with the learning process. It is easily available for operating systems like MAC, Windows, and Linux and runs on the Java Platform that comes with inbuilt functions and commands that play a vital role in debugging, editing, and compiling the code in the environment. The range of Arduino modules available is including Arduino Uno, Arduino Mega, Arduino Leonardo, Arduino Micro, and many more. Each of them contains a microcontroller on the board that is actually programmed and accepts the information in the form of code. The main code, also known as a sketch, created on the IDE platform will ultimately generate a Hex File which is then transferred and uploaded in the controller on the board. The IDE environment mainly contains two basic parts: Editor and Compiler, where the former is used for writing the required code and later is used for compiling and uploading the code into the given Arduino Module. This environment supports both C and C++ languages [13].

Fig. 5 and Fig. 6 illustrate the algorithm steps to program each part of the proposed system. Fig. 5 shows the flowchart of the connection and collecting data from the ECG sensor to the microcontroller. In Fig. 6(a), the flowchart of the process of connection and collecting data from the Body and Ambient Temperature sensor by the microcontroller. Finally, Fig. 6(c) demonstrates the process of collecting data from the heartbeat rate sensor by the microcontroller.

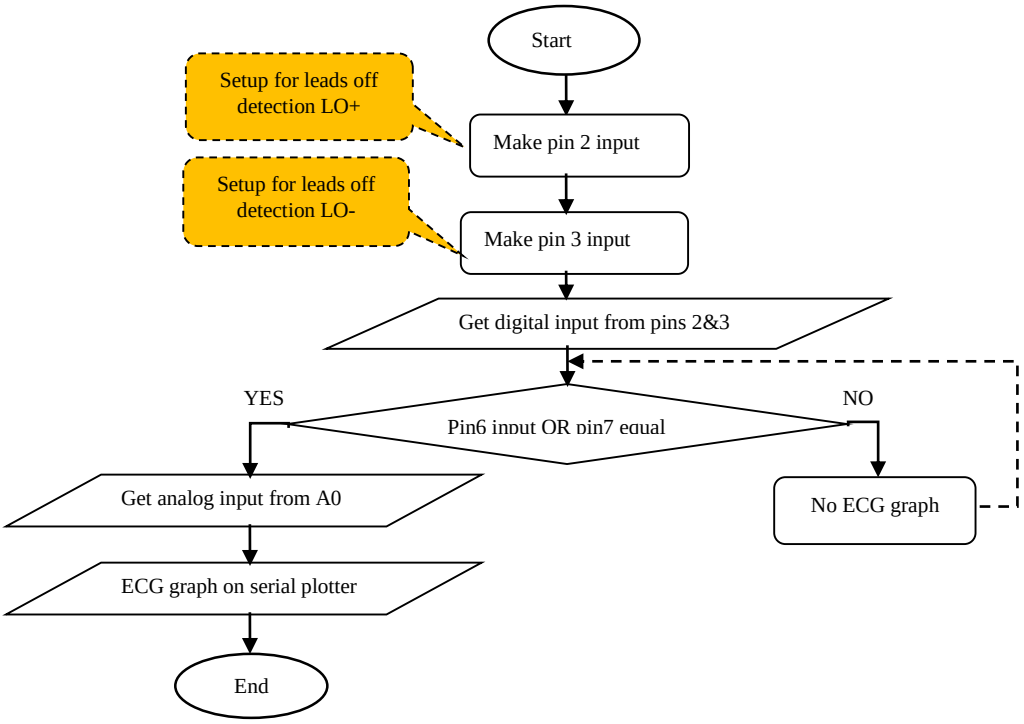
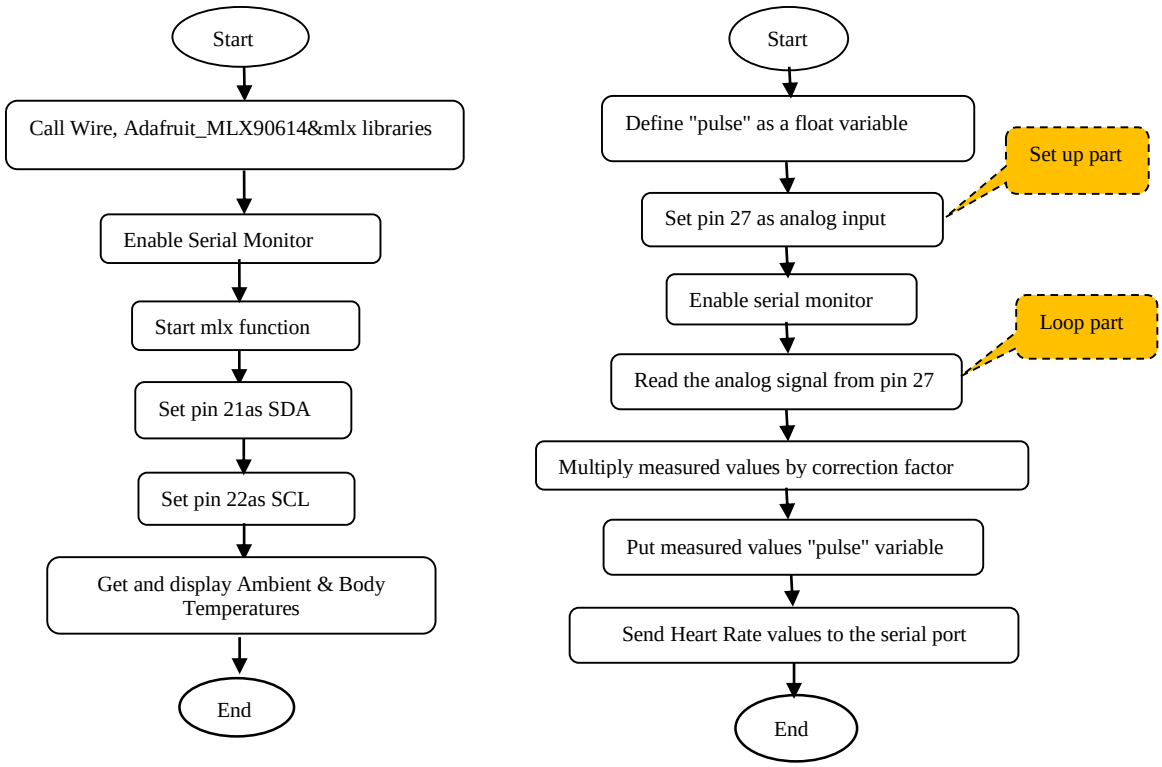


Fig. 5. Flowchart of ECG sensor data collection



(a) Data collection process of ambient and body temperature sensor.

(b) Data collection process of the heartbeat rate sensors.

Fig. 6. Flowchart of the processing of both body and heartbeat rate sensors

2.2.2 Thing Speak Account

Application Programming Interface (API) and web service for the "Internet of Things" (IoT), while the interpretation as to what should be understood under the term is changing over time, here we refer to enabling objects or simple devices to be identified and communicated with via the Internet. The Thing Speak API is an open-source interface that listens to incoming data, timestamps it, and outputs it for both human users (through visual graphs) and machines (through easily parse-able code). We look into practical examples using the ESP32 micro-controller as well as communication with graphical interface operating systems through a Python script. Our report concludes that Thing Speak is especially useful for smaller hardware projects where connectivity over the Internet is required but tends to require payment for some of their functionality and is consequently not open source.

The primary feature of Thing Speak functionality is the term 'Channel' that has a field for data, a field for location, field for status for varied sensed data. Once channels are created in the 'Thing Speak', the data can be implemented, and alternately one can process and visualize the information using the MATLAB and respond to the data with tweets and other forms of alerts. Thing Speak also provides a feature to create a public-based channel to analyze and estimate it through the public [14]. Fig. 7 shows how to make channel ID and its field in the ThingSpeak platform for our system.

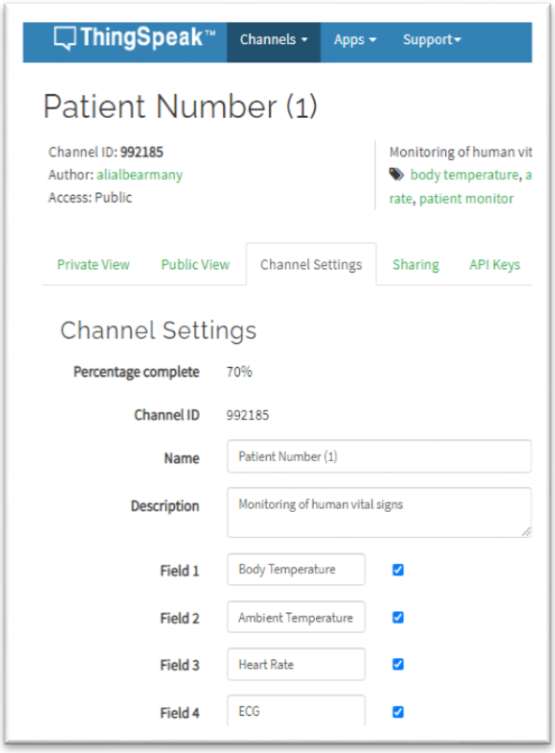


Fig. 7. Channels and fields setting in ThinkSpeak platform.

2.2.3 MIT-APP Inventor

MIT App Inventor is an online platform designed to implement and develop simple smartphone applications. The MIT App Inventor user interface includes two main editors: the design editor and the blocks editor. To aid in development and testing, App Inventor provides a mobile app called the App Inventor Companion (or just "the Companion") that developers can use to test and adjust the behavior of their apps in real-time. In this way, anyone can quickly build a mobile app and immediately begin to iterate and test [15]. Fig. 8 illustrates how we use MIT-APP to create an Android application for the proposed patient monitor. We have named this application a Sumer patient monitor, where the word "Sumer" refers to one of the ancient Iraqi civilizations.

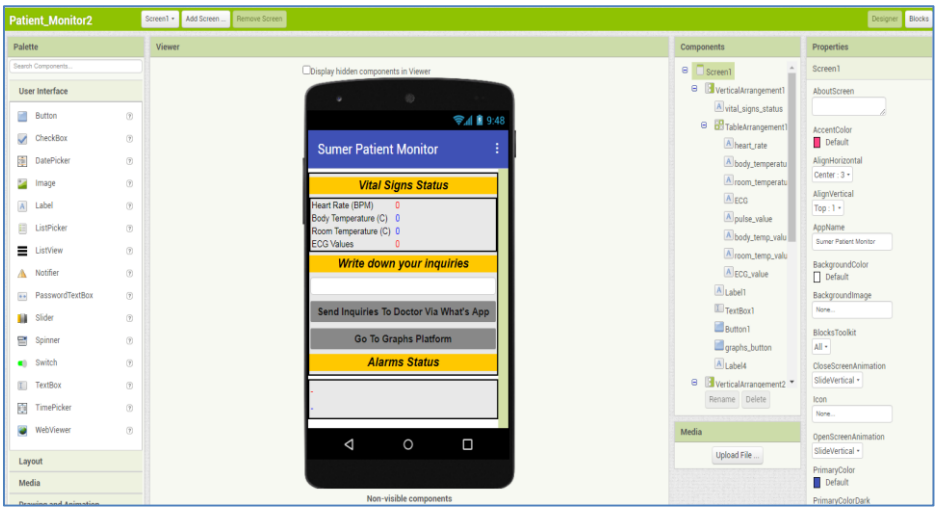


Fig. 8. The MIT-APP Platform that used in the designed Android application of the patient monitor system

3. IMPLEMENTATION OF THE OVERALL SYSTEM

The implemented patient monitor hardware components are shown in Fig. 9. All the sensors that include ECG, heartbeat rate, body, and environment temperature sensors are connected to the ESP32 board. Also, LEDs and buzzers are added to the hardware in order to alert the patient or whoever does the measurement for abnormal conditions when the ECG, temperature, heartbeat, environment conditions are outside the normal range. ESP32 is connected through Wi-Fi connectivity to access the Internet. The board interfaces with the patient and doctor by a smartphone application that is connected to the cloud through the ThingSpeak platform.

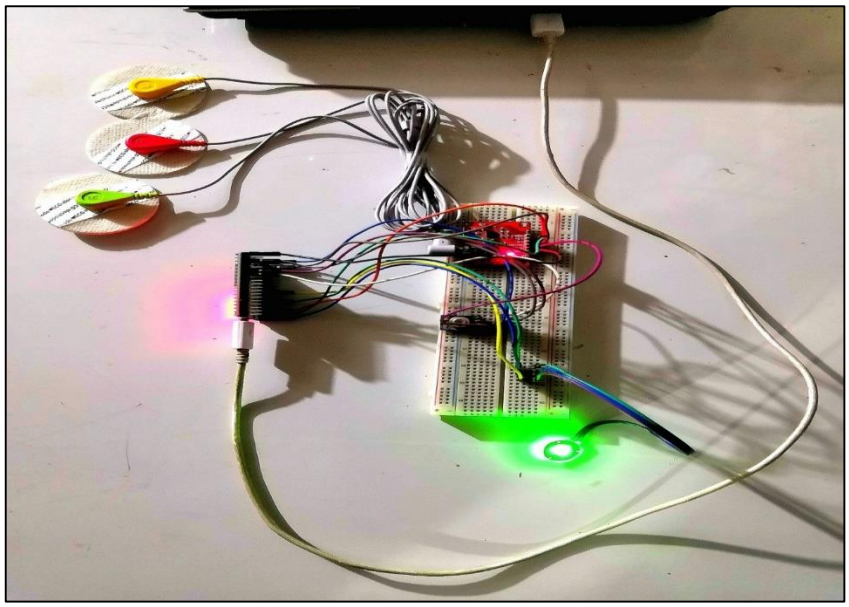


Fig. 9. The implemented hardware of the designed patient monitor system

Fig. 10 shows a flowchart of the overall processes of the designed patient monitor. The wireless network name (through Wi-Fi), password, and the Thing Speak channel ID are entered as input to the setup part. Then in set up, the program sets the pins of 21 and 22 as an SDA and SCL respectively for taking the analog signals of the body and environment temperatures automatically because these pins are the I2C default pins and sets pin 27 as an analog input where the pulse sensor is connected to get heartbeats of the patient. After that, set the pin of the ECG analog signal that will be pin A0. Another execution steps are the enabling of serial communication for monitoring measurements at baud rate 9600, enabling of Wi-Fi communication for allowing ESP32 to access the internet network, and the last step of setting up part is calling of ThingSpeak

function to start communication between the patient monitor device and Thing Speak platform. In the loop part of the programming code, the reading values of the body and ambient temperatures, heart rate, and ECG are continuously collected from the sensors that are connected to the patient's body. The collected data are sent to the channel of the ThingSpeak platform. The designed Android applications, patient and doctor applications, are connected and updated automatically to the cloud database via the ThingSpeak platform.

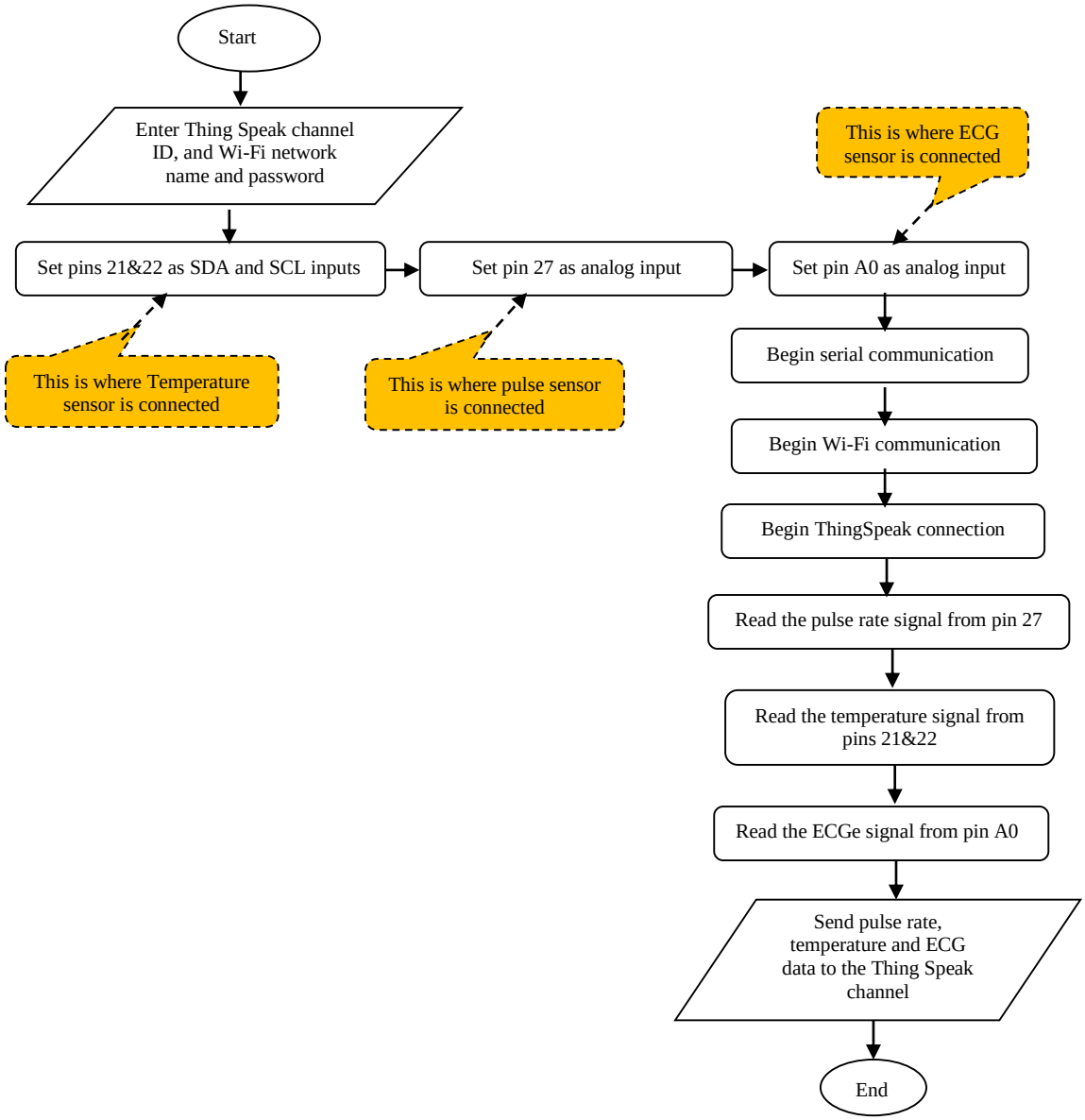


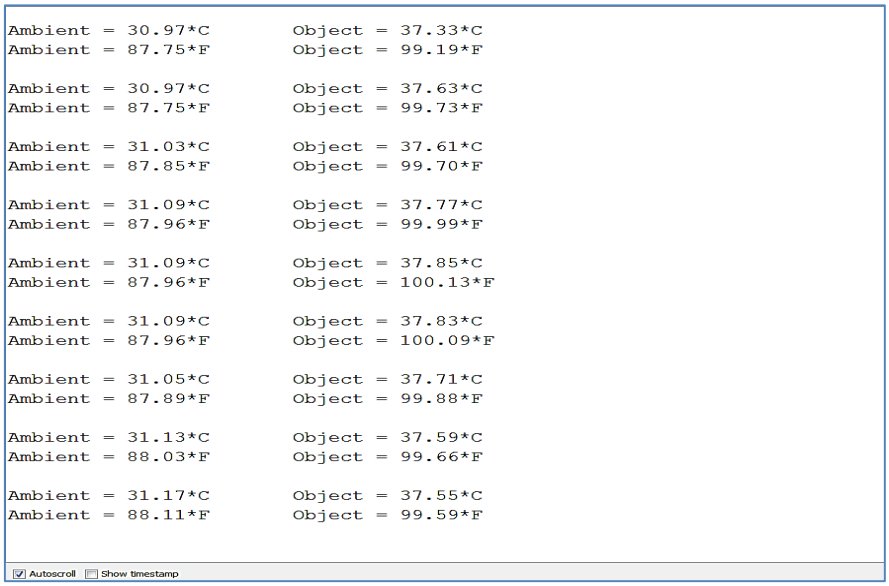
Fig. 10. Flowchart of the programming for the overall patient monitor hardware system

4. RESULTS AND DISCUSSION

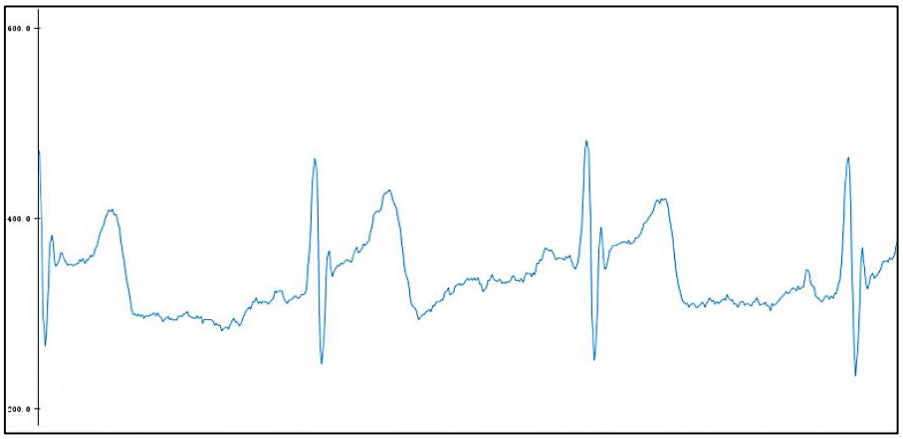
Samples of the results obtained from the proposed patient monitor system are shown in Fig 11. These results show serial data received by the computer from the hardware system. The ambient and body temperature are shown in Fig. 11(a), while the heartbeat rate shown in Fig. 11(b) and ECG plot is shown in Fig 11(c). Fig. 12 shows two ECG readings we achieved in the hospital for the same person with a time difference of about 10 minutes between the two readings, the first recorded ECG readings by our device and the second one is by ECG device of the hospital, which is an expensive device that named "Marquette 12SL ECG Analysis" from GE Healthcare (MAC 2000).

Fig 13. shows the results from the designed ThinkSpeak channel with five fields; field 1 for the body temperature, field 2 for the ambient temperature, field 3 for heart rate, field 4 for ECG, and field 5 for displaying ECG as a two-dimensional graph. The values of readings in this figure are received from the ESP32 microcontroller via the Internet and displayed on the platform in the way that is shown in Fig. 13. The data of the sensors are sent every 15 seconds to the doctor, and patient applications from the Thing Speak channel. The applications are updated automatically in real-time. Therefore, the doctor can monitor the patient in real-time when the sensors are connected to the patient body.

The doctor application that we designed and implemented enables the doctor to monitor several patients by selecting the required serial number. When the doctor presses a specific number, the results of the device associated with that number will be immediately seen, and if there is no device are operating previously, an error message will be displayed indicating that there is no device associated with the serial number that was pressed. Fig. 14 shows how the sensor's reading is displayed on the doctor and patient applications. The application of the doctor allows him/her to communicate with the patient and send instructions and descriptions to the WhatsUp .application of the patient. Furthermore, the doctor and patient, easy access to the database and history of the patient by press the "Go to Graph Platform" push button on his smartphone applications, as shown in Fig. 14. Also, the patient can send inquiries to the doctor. In case of critical values of sensor reading, for example, if the body temperature greater than 40°, or heartbeat rate > 125, the key of the alarm status changes the color to red, and an alarm sound plays for 10 seconds in both applications.



(a) Ambient and body temperature readings.



(b) ECG graph that is red by ECG sensor.

Fig. 11. Sample of the results for the sensor readings from the proposed system

(a)

(b)

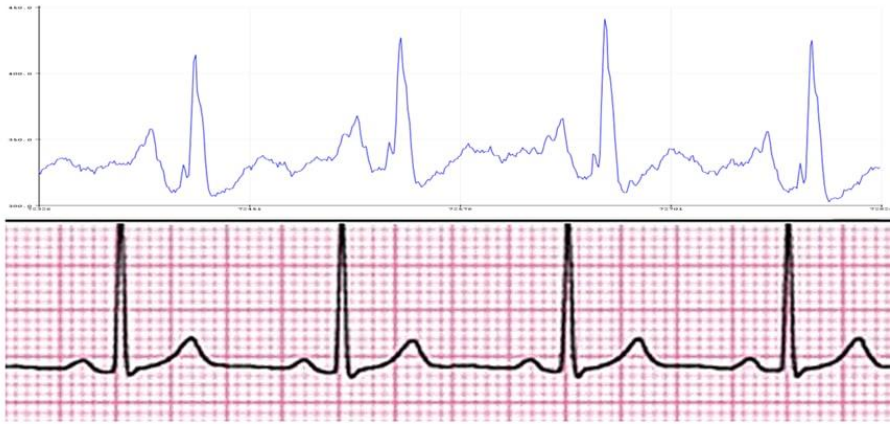


Fig. 12. ECG reading for the same person by: a) Our device, b) Device in hospital.

Fig. 13. Samples of the results on the ThingSpeak channel

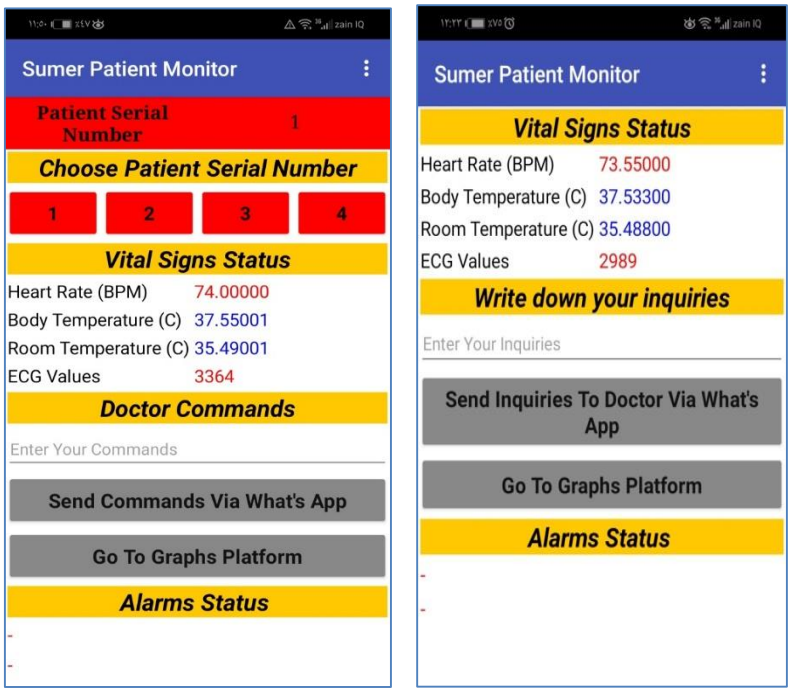


Fig. 14. The Interfaced window of the doctor and patient smartphone applications.

5. CONCLUSIONS

A smart and portable patient monitor system has been designed and implemented in this paper. The implemented system allows the doctor to monitor his/her patients from any place in the world in real-time. In the implemented system, no need for a special skill to be used. Also, it is very cheap and requires low power consumption. The obtained results from the implemented system are similar to the standard results that are obtained from the famous devices that are used in the field of vital signs measurements. The system combines the smart modules that control by the ESP32 microcontroller with the cloud database. The wireless connection between ESP32 and Internet is achieved via Wi-Fi connectivity. The cloud has access to the microcontroller to upload the sensors reading every 15 seconds. The cloud platform (ThingSpeak) creates a special ID with a specific password for each channel (patient monitor device). Each channel has five fields that represent the sensor readings. In addition to the cloud, smartphone applications for patients and doctors have been designed in this work. These applications are updated automatically by the cloud database. Furthermore, they enable the doctor and patient to chat and comment in real-time. Moreover, both patient and doctor can access the history of the patient's vital signs. The doctor can monitor several patients in a very short time.

REFERENCES

[1] M. Glesner and F. Philipp, "Embedded systems design for smart system integration," *IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, 2013, pp. 32–33. https://doi.org/10.1109/ISVLSI.2013.6654611

[2] R. Jadhav, R. Kulkarni, S. D. Perur, G. L. Kulkarni, and P. Kunchur, "Prominence of Internet of things with Cloud: A Survey," *International Journal of Emerging Research in Management &Technology*, vol.6, Issue 2, pp. 40–43, 2017.

[3] M. Talal et al., "Smart Home-based IoT for Real-time and Secure Remote Health Monitoring of Triage and Priority System using Body Sensors: Multi-driven Systematic Review," *Journal of Medical Systems*, vol. 43, no. 3, pp. 2-43, 2019. https://doi.org/10.1007/s10916-019-1158-z

[4] K. Mohanraj, N. Balaji, and R. Chithrakkannan, "IoT based patient monitoring system using raspberry pi 3 and Lab view," *Pakistan Journal of Biotechnology*, vol. 14, pp. 337–343, 2017. https://doi.org/10.1109/ICOMET.2019.8673393

[5] B.K Bhoomika., K. Muralidhara, "Secured Smart Healthcare Monitoring System Based on IOT," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 3, issue 7 pp. 4958–4961, 2017. https://doi.org/10.2139/ssrn.2941100

[6] A. Singh, B. N. Naik, S. L. Soni, and G. D. Puri, "Real-Time Remote Surveillance of Doffing during COVID-19 Pandemic: Enhancing Safety of Health Care Workers," *Anesth. Analg*, pp. E112–E113, 2020. https://doi.org/10.1213/ANE.0000000000004940

[7] P. Valsalan1, T. Baomar, A. Baabood, "IoT based Health Monitoring System for Elderly People," *Journal of critical reviews*, Vol 7, Issue 4, pp. 739-740, 2020. https://doi.org/10.31838/jcr.07.04.137

[8] M. G. Ayoub, M. N. Farhan, and M. S. Jarjees, "Streaming in-patient BPM data to the cloud with a real-time monitoring system," *Telkomnika Journal*, vol. 17, no. 6, pp. 3120–3125, 2019. https://doi.org/10.12928/telkomnika.v17i6.13263
[9] A. Maier, A. Sharp, and Y. Vagapov, "Comparative analysis and practical implementation of the ESP32 microcontroller module for the internet of things," *IEEE Conference on Internet Technologies and Applications (ITA)*, 2017, pp. 143–148. https://doi.org/10.1109/ITECHA.2017.8101926
[10] B. Mallick and A. K. Patro, "Heart Rate Monitoring System Using Finger Tip through Arduino and Processing Software," *International Journal of Engineering Research and Technology*, vol.6, issue 13 , pp.1-4, 2018.
[11] L. Dioren Rumpa, S. Suluh, I. Hendrika Ramopoly, and W. Jefriyanto, "Development of ECG sensor using arduino uno and e-health sensor platform: Mood detection from heartbeat," *Journal of Physics: Conference Series – IOPscience*, vol. 1528, no. 1, 2020. https://doi.org/10.1088/1742-6596/1528/1/012043
[12] G. Jin, X. Zhang, W. Fan, Y. Liu, and P. He, "Design of non-contact infrared thermometer based on the sensor of MLX90614," *Open Automation and Control Systems Journal*, vol. 7, no. 1, pp. 8–20, 2015. https://doi.org/10.2174/1874444301507010008
[13] M. Fezari and A. Al Dahoud, "Integrated Development Environment "IDE" For Arduino," retrieved 2018, available online at: https://www.researchgate.net/publication/328615543_Integrated_Development_Environment_IDE_For_Arduino.
[14] 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, 2016.
[15] E. W. Patton, M. Tissenbaum, and F. Harunani, *MIT app inventor: Objectives, design, and development*, in Computational thinking education, Springer, Singapore, 2019. https://doi.org/10.1007/978-981-13-6528-7_3

BIOGRAPHY OF AUTHORS

Haider Ismael Shahadi received his BSc degree in information engineering from the University of Baghdad-Iraq in 2001, his master's degree in Electronics and Communication Engineering from the University of Baghdad-Iraq in 2004, and his Ph.D. in Electronics and Communication Engineering from the Tenaga National University-Malaysia in 2014. Currently, he is an assistant professor at the department of electrical and electronic Engineering-University of Kerbala-Iraq. His research interests include digital signal and multimedia processing, data security, FPGA design and implementation, and embedded systems, IoT systems, and smart systems.

Maha Khalid Kadhim received her BSc degree in electrical engineering from the University of Babylon in 2009, MSc in 2017 in Communications and Electronic Engineering from the University of Babylon. Currently, she is an assistant lecturer Biomedical engineering-university of Kerbala-Iraq. Her research interest in communication and IoT systems.

Nawal Mousa Almeyali received her BSc degree in electrical engineering from the University of Al-Mustansiriya, her higher diploma in computers from the University of Technology, and her MSc in Industrial Management from the University of Tehran. Currently, she is an assistant lecturer in the College of Medicine-University of Kerbala. Her research interests in industrial system management.

Ali Thamir Hadi received his BSc degree in Biomedical Engineering from the University of Kerbala in 2020. Currently, he is a biomedical engineer in the Department of Dialysis- Marjan Hospital-Iraq. His research interest. Electronic systems for Medical applications.