A CYBER-PHYSICAL SYSTEM FOR SMART HEALTHCARE

by

Tlhakatswane Jerry Malapane

216088717

Submitted in partial fulfilment of the requirements for the degree of

Magister Technologiae

In the

Department of Electrical and Electronic Engineering Technology

Faculty of Engineering and the Building Environment

University of Johannesburg

Supervisor: Prof BS Paul

Co-Supervisor: Prof W Doorsamy

SUMMARY

The increasing number of patients in hospitals is becoming a serious concern in most countries owing to the significantly associated implications for resources such as staff and budget shortages. This problem has prompted researchers to investigate low-cost alternative systems that may assist medical staff with monitoring and caring for patients. In view of the recent widespread availability of cost-effective internet of things (IoT) technologies such as ZigBee, WiFi and sensors integrated into cyber-physical systems, there is the potential for deployment as different topologies in applications such as patient diagnoses and remote patient monitoring.

In this dissertation, a cyber-physical system to be implemented in smart healthcare is designed, developed, and tested. The purpose of the system is to enable medical staff to monitor and diagnose patients remotely by collecting, processing and analysing real-time vital signs data. The integration of sensors for measurement of patients' vital signs, patient fall detection and patient room conditions is presented. The system is based on the use of technologies that are compatible with Arduino microcontrollers, sensors and data exchange mechanisms between sensor nodes. MATLAB software is used for developing a graphical user interface and a fuzzy logic inference system. Wireless sensor network topology integrates the data into one common system to provide an easy means of real-time monitoring of the patient's health status.

This work deals with the design, implementation, and prototype testing of the subsystems of an overall cyber-physical system based on smart healthcare. The cyber-physical system involves the use of wireless sensor network devices and a wearable sensor device, which collects patients' vital sign data and patient room environment data. A fuzzy logic algorithm is used to analyse the patient's condition continuously in real-time. A set of fuzzy rules is implemented to diagnose the condition of patients based on incoming vital signs and environmental conditions via the wireless sensor network. The fuzzy algorithm, together with Thingspeak, serves as an IoT platform that outputs early warnings to doctors and nurses in the event of detection of abnormal patient status. The presented concept for a cyber-physical system for smart healthcare is intended to serve as assistive technology in healthcare and support an efficient, reliable and cost-effective healthcare service.

ACKNOWLEDGEMENTS

First of all, I would like to take this opportunity to thank the lion king of Judah (God of Abraham, Isaac, and Jacob) for the strength and shower of steadfast love that He is always giving me. I would like to acknowledge a few people who have been part of this research.

I would like to thank my wife, Mpho, for love, care, unlimited inspiration, and support during the course of this research.

I am sincerely and eternally grateful to both of my supervisors at the University of Johannesburg, Prof B.S. Paul, Institute for Intelligent Systems, and my co-supervisor, Prof W. Doorsamy, Department of Electrical and Electronic Engineering Technology. Their guidance, motivation, and unlimited support helped me to progress during the course of this research project.

Moreover, I would like to thank my fellow research mates for group discussions and helping one another on our projects. This has been a pleasant journey.

Finally, I would like to thank all my family and friends who supported me.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

AES	Advanced Encryption Standard
CPS	Cyber-Physical System
CMS	Central Monitoring System
DSR	Design Science Research
ECG	Electrocardiogram
4IR	Fourth Industrial Revolution
FLC	Fuzzy Logic Controller
GSR	Galvanic Skin Response
GUI	Graphic User Interface
IEEE	Institute of Electrical and Electronic Engineers
IoT	Internet of Things
I/O	Input and Output
MATLAB	Matrix Laboratory
NSF	National Science Foundation
PSD	Patient's Side Device
UART	Universal Asynchronous Receiver Transmitter
USB	Universal Serial Bus
WSN	Wireless Sensor Networks
WLAN	Wireless Local Area Network
WiFi	Wireless Fidelity

LIST OF PUBLICATIONS

- W. Doorsamy, B.S. Paul and J. Malapane. "The Internet of Things in Health Care: Transforming the Industry with Technology". In: Z. Mahmood (ed), *The Internet of Things in Industrial Sectors, Computer Communication and Networks, https://doi.org/10.1007/978-3-030-24892-5_11 , 2019. Pages 261 – 278.*

- T. Malapane, B.S. Paul and W. Doorsamy. "Wearable Sensors for Remote Monitoring in Smart Public Hospital", In: *Proceedings of the International Conference on Industrial Engineering and Operations Management Pretoria/Johannesburg, South Africa, October 29 – November 1, 2018.*

CHAPTER 1: INTRODUCTION

1.1 BACKGROUND

Cyber-physical systems (CPS), which integrate communication and computation capabilities with real-world physical activities, are currently regarded as developing technology. In 2008, the US National Science Foundation (NSF) acknowledged CPS as a highly important research area and in the meeting of the Council of Advisors on Science and Technology, United States, the President listed CPS as a first research priority [1]. Applications of CPS are very popular in the improvement of the systems, forming building blocks in the fourth industrial revolution (4IR), such as smart healthcare, smart factories, smart homes and smart cities [2]. CPS, together with the internet of things (IoT), plays a significant role in communication and information technologies because of its affordability and design simplicity. With an increase in the technological revolution of CPS in the world, this leads to the development of many innovations in healthcare. Technologies such as wireless sensor networks (WSN), wearable sensor devices, and ambient systems become increasingly pervasive, which makes CPS a reality by interconnecting with the IoT to be capable of accessing the internet, which helps to exchange and collect data [3]. A CPS in smart healthcare is a modern technology encompassing highly complex and sophisticated embedded systems designed with network communication proficiencies that are accountable for monitoring and controlling the physical dynamics of patient's bodies [4].

1.2 PROBLEM STATEMENT

Currently, public healthcare facilities are overpopulated because of the low fees for admission and consultation. This causes more work pressure and a shortage of health professionals to maintain and monitor patients. Admitted patients suffer most, including patients who are critically ill and have been diagnosed with diseases such as hypertension, diabetes, and cardiac problems. These patients may suffer unnecessarily and even die, as they need to be monitored continuously, which is done physically by doctors and nurses who must manually monitor each patient at specific time intervals. Wired monitoring systems need to be attached to the body of the patient and limit the patient's movements. When patients need to use the toilet, nurses must remove the equipment attached to a patient to allow the patient to move freely. Most healthcare facilities still using old, traditional ways of capturing records of patients by using paper files and storing these in shelves that occupy space in healthcare facilities. Information about patients recorded in files may be misplaced or lost.

Most ongoing research in healthcare still advertises the use of wired monitoring systems provided by sensors and a computing system, whereas the CPS and IoT systems have not been properly introduced. The motive of the researcher is to design and develop systems reflecting the latest futures, including the IoT, ambient sensors, and a WSN to observe the patient continuously in real-time, supplied by CPS technology.

1.3 RESEARCH AIM AND OBJECTIVES

The main aim of this research is to design a CPS that is integrated with the IoT in smart healthcare facilities. The system will analyse and monitor the conditions of patients in real-time, including observing the environmental conditions of patients' rooms. The CPS consists of a WSN, which collects all sensing data from different microcontrollers equipped with a set of sensor devices, and a wearable sensor device attached to the arm of a patient to measure vital signs parameters. An IoT based on Thingspeak together with the MATLAB application is accountable for the analysis and collection of information data and visualisation of the results.

The specific objectives are as follows:

i. To analyse the current challenges faced in healthcare facilities.
ii. To conduct an extensive literature review on an existing CPS based on healthcare.
iii. To design and develop a low-cost system to be used in the healthcare environment.
iv. To develop a WSN.
v. To design a fuzzy logic inference engine to analyse data and detect the status of the patient's condition.

1.4 RESEARCH QUESTIONS

The aim is to design a system that is unique and enables smart healthcare, using a CPS. In doing so, the following questions were formulated to address the research problem:

i. What literature is available on the existing CPS for smart healthcare?
ii. What are improved methods of collecting, analysing, and monitoring data of patients in real-time?
iii. How could the system generate alerting messages to medical staff and doctors?

iv. How does the fuzzy logic algorithm in healthcare improve the proposed system compared to existing systems?

Through this research, each of these questions will be analysed and explored in detail by designing and experimentally testing a CPS for smart healthcare.

1.5 RESEARCH METHODOLOGY

In this research, the methodology used is built on the *Design Science Research* (DSR) process model, as shown in Figure 1.1. DSR delivers a set of analytical methods that provide new knowledge, by giving researchers understanding and enlightenment on existing research and publications [5]. The progression is intended to create and provide innovative design, including analysing, testing and providing conclusive results. Over the years, DSR has been engaged in the fields of computer science, software engineering and information systems [6].

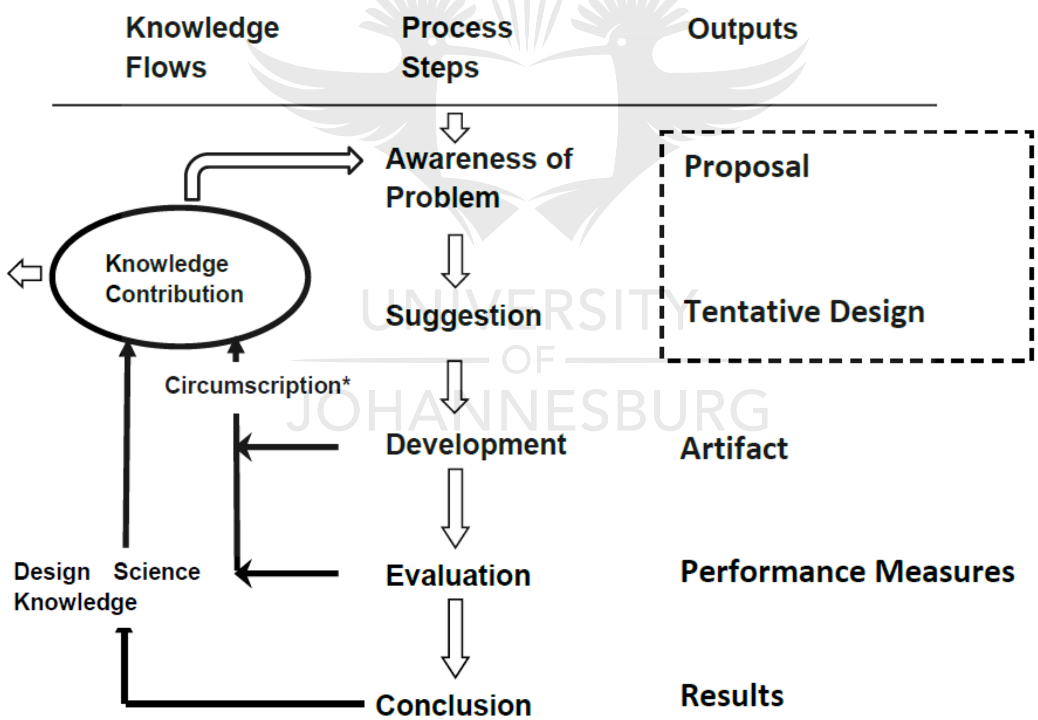


Figure 1.1: DSR process model [5].

The approach is summarised in Figure 1.1, starting with awareness of the problem, followed by the research suggestion, including a literature review, experimental design, system testing and analysis of results, and finally the conclusion. This research methodology employs the DSR process model for the development of the intended CPS for smart healthcare.

1.6 OUTLINE OF THE DISSERTATION

Chapter 2 – Literature Review

This chapter focuses on the literature review of different technologies and provides research that has been done on achieving design on CPS for smart healthcare. A methodical literature review has been conducted to gain an understanding of related work done on the CPS and the prediction of patient diagnosis. The gaps in current research will be highlighted to propose the research question.

Chapter 3 – Methodology

An overview of the proposed CPS for smart healthcare is presented in this chapter. This includes a discussion of prototype operation and theoretical strategy. Various types of sensing components will be listed and described in this chapter.

Chapter 4 – System Design

This chapter presents the steps in implementing, circuit designs, and simulation of the system. It includes the validation and evaluation of components used in the system. This includes data collection, data storage and analysis of data.

Chapter 5 – Analysis and Evaluation

In this chapter, the researcher discusses the analysis and implementation of the algorithm using MATLAB. Fundamental concepts of fuzzy set theory and crisp methods for diagnosis of patients are presented, followed by a discussion on the effect of various design parameters and the explanatory ability of the diagnostic structure.

Chapter 6 – Results and Discussion

This chapter presents the discussion and analyses of the results achieved during the simulation of the design.

Chapter 7 – Conclusion

This chapter provides summaries of the main aspects observed in the dissertation and makes suggestions for future work to place these findings in a broader environment.

CHAPTER 2: LITERATURE REVIEW

The evolution of modern technology in healthcare is extremely fast-growing, such as CPSs that participate in both physical procedures and computation. The development of CPS is valuable in healthcare; however, such systems raise many questions, such as software compatibility, hardware design, and many others. Currently, only a few hospitals in the world have implemented the use of remote operations with the assistance of the automated hand and high-resolution cameras [6]. More research and development are required to enable complete, full self-controlled surgery when the CPS will be able to undertake surgery by itself without human management. Research on the CPS [8], [9] demonstrates monitoring systems on the IoT and has tried to identify and highlight critical issues in the health sector. Some research is based on a user interface to display information and help to collect data wirelessly, but the system cannot relay information remotely to be monitored[10]. Most ongoing research in healthcare still advocates the use of wired monitoring systems and the principle of CPS has not been introduced.

This chapter presents an overview of the CPS for smart healthcare and reviews existing patient monitoring systems. It also covers published works and relevant studies sourced from research knowledge bases such as IEEE Explore, Research Gate, and Science Direct. The chapter starts with definitions and an outline of the CPS related to healthcare. It investigates the evolution of smart healthcare through various development processes. A brief overview is given of wearable sensors, WSN and IoT as some of the main aspects of a CPS. The last part of this review focuses on the fuzzy logic algorithm, which is used to analyse and predict the condition of the patient and the healthcare environment.

2.1 Cyber-Physical Systems in Healthcare

CPSs are designed for the replacement of embedded systems [17]. They assimilate modern computing technology with communication and information systems. A CPS integrates the features of the microcontroller processing system and communication networks with the physics of physical systems functioning in the real world [18]. CPSs are poised to play an important role in the healthcare industry. Research on CPSs in the healthcare sector focuses on a real-time smart sensor system that monitors and provides warning of the condition of a patient in healthcare. This includes telemedicine systems that qualify for remote healthcare service facilities and self-controlled remotely functioning service robotics that can help patients with physical activities. CPSs in the healthcare sector are used as responsive to users, life-critical, and networked system medical devices that are cooperatively involved in treating a patient in a detailed medical situation [10]. Humayed *et al*. [11] explain the concept of CPS in healthcare by identifying three important components, namely communication, computation, and monitoring. These devices contain a physical component, such as sensing or an actuator and a networking component for communication. This system is used as a collection of sensor data or as a wireless body area network (placed around a patient's body) that communicates with the controller.

In [12] the authors describe a CPS designed for the healthcare monitoring system as involving three layers. The first layer is the information collection layer, which collects data from different nodes, such as temperature sensors, heartbeat sensors, and other sensors. Captured data are then sent to the converter, where the noise is filtered and converted to a comprehensible setup using an advanced encryption standard (AES) algorithm. The second layer, the data management layer, consists of a distributed file storage module and distributed parallel computing (DPC) module. This layer sends encrypted data to the cloud via the WiFi module; in this layer, DPC also provides the corresponding processing and analysis methods. The application layer, which is the final layer, provides the user with decrypted data that can be visualised and analysed. Doctors can even prescribe medication to patients using the methods in the application layer.

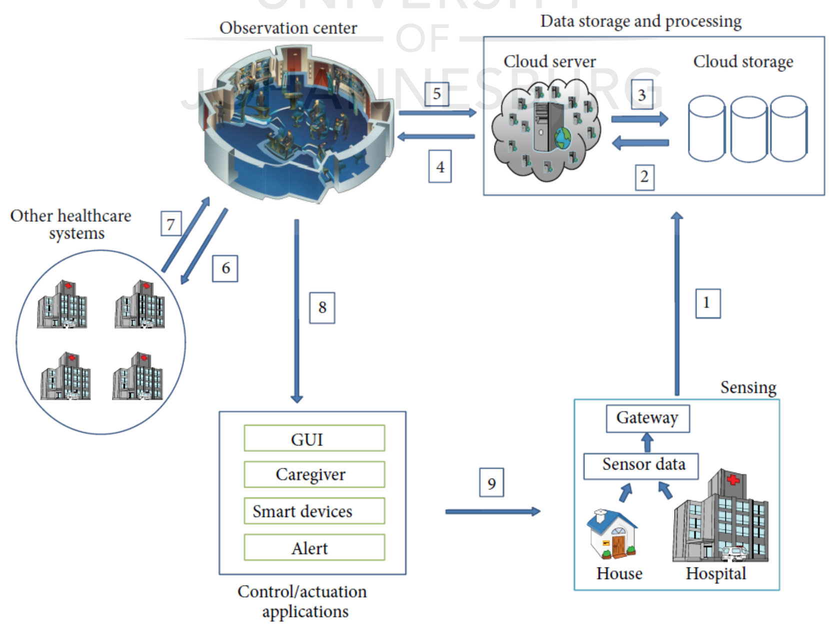


Figure 2.1: Diagram of CPS for healthcare [14, 15, 16]

In Figure 2.1, the researchers in [14, 15, 16] demonstrate the functionality of the CPS for healthcare. Arrow (1) represents sensor information captured from patients using different sensors and sent to cloud storage through the gateway; Arrow (2) is managed data collected from sensors sent to the cloud server and processing queries are performed in real-time; (3) stored sensor data are managed in cases of request-response; (4) system analysis processing is finished; an announcement alarm is generated and sent for monitoring; (5) medical staff access patient information that is stored from the cloud; (6) and (7) medical staff request and receive consultation from other healthcare systems; (8) medical staff and doctors deliver the final decision to the actuation components and finally (9) essential measures are finalised on the patients.

Sonntag *et al*. [19] define the combination of passive and active operator input methods in medical environments for data discovery and data achievement to lend real-time decision support. The objective involvements are practical components for combining digitised patient data, sensory data, and manual data acquisition, including an interchange computer server. Secondly, the addition of data acquisition systems to a hospital testbed environment creates semantic methods for medical evidence. This ingenious aspect is a general assessment of individual patients based on expressions, ontologies and textual patient records whereby different passive and active real-time patient data are used to facilitate medical decision support [20]. Tercero *et al*. [21] cover some practical methods of analysis based on a CPS for endovascular surgery replication and claim that the implementation of cyber-physical emulators for other areas of medicine depends on research into photo-elastic supplies for human tissue modeling that would enable measurable assessment of acknowledgment using surgical tools and a genuine illustration of human tissue.

2.2 Wireless Sensor Network Technologies for Healthcare

There is an extensive range of wireless telecommunication technologies that can be used to achieve the implementation of a healthcare remote monitoring system. The existence and operation of personal area systems such as WSNs, ZigBee Bluetooth, and radio frequency identification, combined with the integration of wireless networks such as WiFi, 3G, and WiMAX, could serve to supply a full, remote monitoring system. However, there are challenges and several issues that involve the common standards and protocols that have to be addressed for the successful integration of these technologies [22]. A WSN is the main aspect that enables the technology of the CPS and the IoT. A WSN connects several sensor and actuator nodes into a network through wireless communication. The sensor and actuator nodes are usually smaller, with reasonable prices, easy to design and maintain, but their functionality and capability are inadequate because of their resources [23].

In [24] the researchers studied operational wireless methods for short-range data transmission, also counting the current connected wireless applications in healthcare facilities. The main interest of the researchers involves hypothetically studying the distinct requirements of exploiting wireless technologies in a healthcare environment, especially during anaesthesia conditions and in the intensive care unit. This was done by interviewing health clinicians and practitioners and measuring the qualities of the current market situation. Crucial parameters were explained by the clinical doctors, including electrocardiogram (ECG) readings, temperature, heart rate (HR), blood pressure (BP), and respiration rate (RR). At present, wireless systems in the healthcare environment use the Web Map Tile Service standard on which telemetry applications are built and the WLAN standard, which is appropriate for information transmission in local area networks (LANs) in limited environments. There are no practical resolutions for a short distance in data communication from patient sensor nodes to a patient monitoring system, but theoretically, methods that could work in the near future are built on Wireless Personal Area Network standards. These wireless network methods consist of ZigBee, Bluetooth, and UWB. Other appropriate methods might depend on inductive or capacitive coupling. The implementation of wireless methods depends on guaranteeing the dependability of data transmission, abolishing interruption by other wireless devices and assuring the security of information of a patient and the safety of the patient, in addition to low and economical power consumption.

2.2.1 ZigBee

ZigBee is an international industrial license-free standard for wireless radio networks that provides low power, low cost, and low complexity. The ZigBee protocol is designed on IEEE 802.15.4 functionality by adding proficiencies for more flexible network topologies, smart communication routing, and upgraded security [25]. The operation of ZigBee 250 kbps involves a low data rate with a frequency of 2.4 GHz. The transmission range is generally between 10 to 100 meters but may increase up to 3 Km with maximum transmit power. There are three types of devices in the ZigBee network, namely routers, coordinators and end devices. These devices can be configured into several ZigBee network topologies, such as a star, a cluster tree, mesh topology and peer-to-peer [26].

In [27] the researchers propose a wireless remote monitoring system in the case of cardiac patients by using ZigBee, based on an adjustable responsibility rotation being extracted to sensors. In an intra-hospital telemedicine situation, ECG information of patients is assimilated through ECG sensor nodes having transmission capability, and this ECG information is received by a personal digital assistant (PDA) kept at the remote desk of the nursing station and communicating via the ZigBee network. ECG data are finally transmitted to the doctor's PDA. If the duty cycle is diverse in terms of the load or number of active sensors, the total energy expended in idle mode can be evaded and the total energy expended by sensors is reduced while increasing the total network lifetime. Comparative analyses of the energy proficiency of ZigBee sensors indicate a changed proportion of the duty cycle based on the energy consumption parameter under adjustable load circumstances. The methods used in presentation assessment consume more energy in the transmit mode, in the received approach and in the shiftless method.

In [28], the authors present the development of a Zigbee network for rehabilitation patients using a multiple-channel HR monitoring system. The system monitors patients' HR continuously to provide early warning alarms in real-time to a physical therapist. The whole system is designed based on the patient's side device (PSD) and the central monitoring system (CMS); the system is connected wirelessly via the Zigbee network, as shown in Figure 2.2. The PSD has been employed to be wearable so that it may use less electrical power and be able to monitor numerous patients with their own identification numbers so that the CMS can monitor multiple PSDs simultaneously through the wireless network.

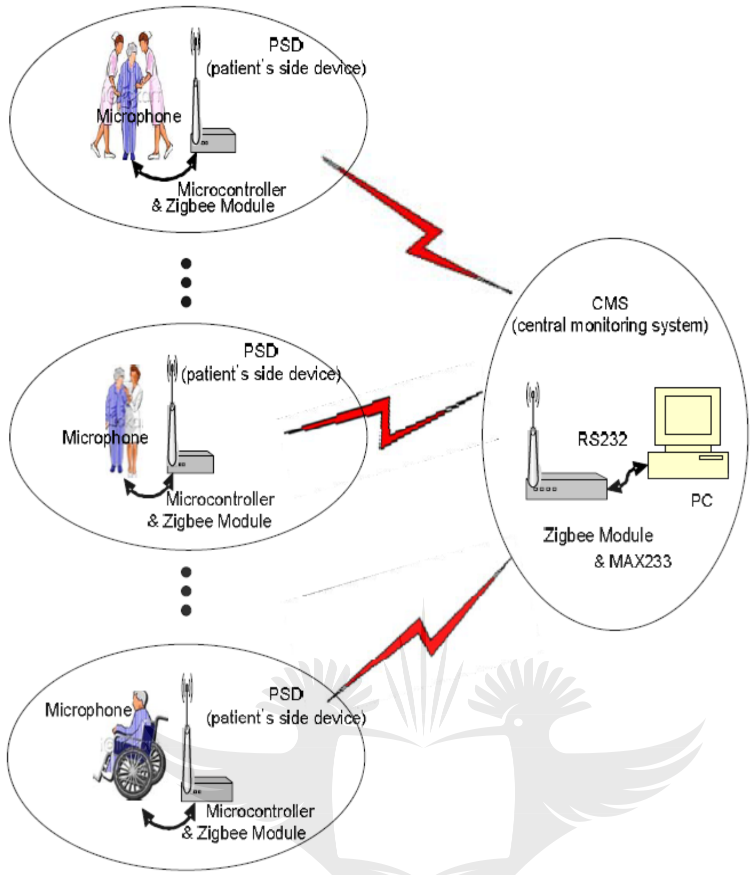


Figure 1.2: Zigbee network multiple channels HR monitoring system [25]

2.2.2 Comparison of ZigBee, WLAN, and Bluetooth

ZigBee is an IEEE 802.15.4 standard set of protocols for data communication that has been established as a low data rate solution with a very long battery life and very low complexity. It is designed to function in a licensed free worldwide short-range radio frequency band. The standard ZigBee module uses 16 channels at 2.4 GHz, ten channels ranging from 902- 928 MHz, and one channel at 868-870MHz. The maximum data rates for each band are 250, 40 and 20 kbps individually [29].

One of the best-known forms of wireless technology is the Bluetooth network system that enables all electrical devices to communicate with one another. Bluetooth systems communicate in a short-range in the 2.5 GHz ISM frequency band. The systems are integrated into the group of IEEE 802.15.1 standards and considered for small and low-cost devices. The Bluetooth Special Interest Group has established the Health Device Profile (HDP) capable of delivering practice models for fitness and healthcare devices. The HDP is also able to connect devices wirelessly, such as pulse weight scales, oximeters, glucose meters, BP and thermometer monitors to link devices such as cell phones, PDAs, laptops and personal computers [30].

Wireless LAN (WLAN) is part of protocols for data communication specified by the IEEE 802.11 standard. WLAN can be described as successful ethernet without wires. It permits users to access information communication data, such as the internet, at a high throughput speed of up to 11 megabits per second (Mbps). WLAN connectivity is typically needed to link and simplify analytical data exchange between several telemedicine systems in a medical facility through wireless communication. The link can provide very high-speed communication data that exchanges information capability between two devices [31]. Table 2.1 below summarises a comparison of ZigBee, WLAN, and Bluetooth.

Characteristics	**ZigBee**	**WLAN**	**Bluetooth**
IEEE Standards	802.15.5	802.11	802.15.1
Nominal Range	10 – 100 meters	100 meters	10 meters
Data Rate	20 – 250 Kbps	54 Mbps	1 Mbps
Power Consumption	Very low	High	Medium
Operating Frequency	868/915 MHz, 2.4 GHz	2.4 GHz	2.4 GHz
Network Topology	Star, mesh, peer to peer	BSS	Piconet
Modulation type	BPSK, O-QPSK	BPSK, QPSK, M-QAM	GFSK
Spreading	DSSS	CCK, OFDM	FHSS
Security	AES block chipper	AES block chipper	AES block chipper
Error checking and correction code	16-bit CRC	32-bit CRC	16-bit CRC

Table 2.1: ZigBee, WLAN Bluetooth Comparison [31]

2.3 Wearable Sensor Device for Healthcare

Wearable devices in the healthcare sector are used to monitor and store real-time data on patients' physiological conditions and activities. Wearable sensors used in health monitoring systems contain devices that have one or many sensor nodes. Each sensor node is generally designed with a radio transceiver and receiver, a low-speed processing unit, and limited memory. The sensors can measure vital and physical signs such as electrodermal activity (EDA), electromyogram, BP, ECG, arterial oxygen saturation, HR, blood temperature and RR [32].

Zhou et al. [33] developed a wearable sensor device that detects and analyses patients' activities. The designed sensor device is mostly used in healthcare facilities for elderly people by monitoring their daily activities and sending notification alert messages to their caretakers and doctors that inform them about their health status. This device assists patients to be independent, secure, and able to work around health facilities freely. The system is still facing some difficulties in resolving the problems of identifying the activity of the user, such as running, walking, eating and so on. It is also difficult to discover a portion of helpful information from monitoring user activities and transmitting these data to the approved party. Users' activities are stored in a data server and the system uses an algorithm to select these activities to obtain useful data and share these with supporters.

The authors of [34] have designed and deployed a wearable device that measures ECG parameters, with an application system using an Android OS platform. This wearable system can monitor and diagnose the heart condition of patients in real-time. The patient must wear a sport-shirt on which the ECG sensors are built. Healthcare professionals use their smartphones to access information data of patients wirelessly in real-time. As shown in Figure 2.3, the proposed system will not only help patients in healthcare facilities but can be applied in remote hospital analysis systems to help elderly patients in terms of diagnostics and self-testing.

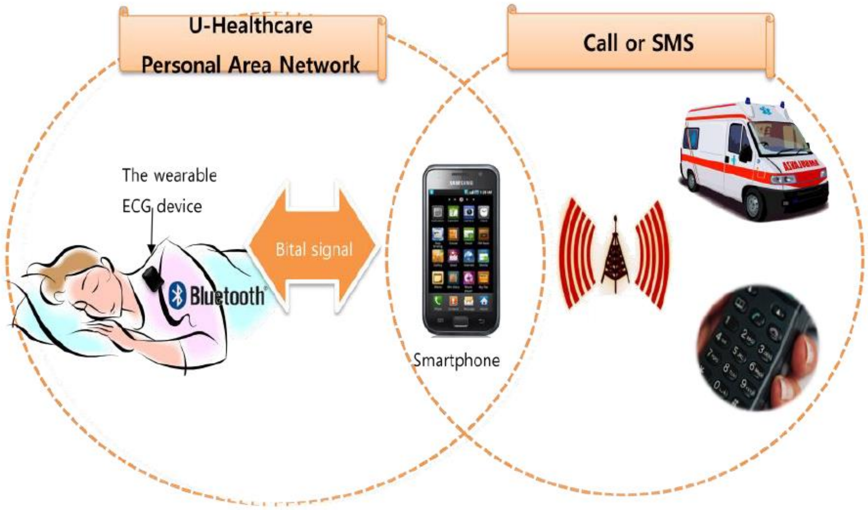


Figure 2.3: Overall healthcare environment system using wearable ECG device [34]

Wearable Galvanic Skin Response Monitoring System

The galvanic skin response (GSR) monitor is a wearable system of sensor devices used for measuring patients' vital parameters in healthcare. The GSR is also known as EDA, defined as the constant and universal alteration of electrical properties of human skin. GSR is measured from a part of the skin that has many sweat glands, such as the fingers, palms, or soles of the feet. In an energetic measurement, a DC voltage is applied across two on-body electrodes and the skin conductance is obtained through Ohm's law by taking the readings of the current. Figure 2.4 [35] below demonstrates a schematic diagram of a wearable GSR monitoring system.

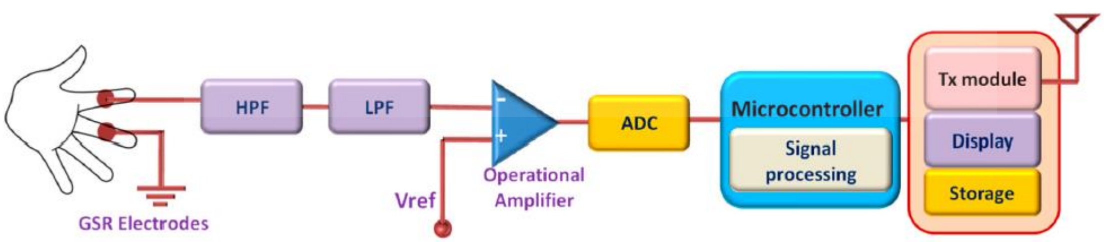


Figure 2.4: Schematic diagram of a GSR monitoring system [35].

Garbarino et al. [36] designed a multiple-sensor wristband called Empatica E3 that measures GSR, photo plethysmogram (PPG), motion, and body temperature sensors. The GRS sensor provides a high active processing speed range measurement between 0.01 microseconds and 100 microseconds at 900 picosecond resolution. The device that has been designed to capture data from a patient has dimensions of 4 cm× 4 cm and has been developed to be embedded in a wristband. It is designed to measure and log data from all four sensors continuously for 38 hours. The system can stream real-time data using the Bluetooth LE wireless medium. Low-power multi-parameter monitoring, wireless connectivity capability, and long battery life are key features of the implemented system.

2.4 Smart Healthcare Systems

SMART Healthcare is an open standard-based information technology platform that connects healthcare and computing, enabling the establishment of intelligent and data-driven medical services. It creates operational efficiencies that support the throughput of patients and doctors with better management of medical resources and improved profitability. The key concept of SMART healthcare includes confidentiality, reliability, and an electronic healthcare record management system. The functionalities of the smart healthcare system are deployed by the availability of the IoT in hospital facilities. Akpojaro *et al*. [37] explained SMART healthcare systems such as the IoT system deployed in healthcare; they define SMART as self-directed, motivated, adaptive, resource-enriched technologies embedded in healthcare. The IoT is a system in which computing capability and network connectivity are extended to sensors, digital machines and objects, allowing these devices to exchange information and use data with minimal human intervention.

In [38] the researchers proposed the design of a smart healthcare system and its methodological architecture. In this design medical information is captured by sensors and actuators, whereafter the gathered data is processed via mobile and intelligent networks. Collected data go to cloud computing for data analysis, using complex algorithms. Medical professionals can subsequently observe these analysed data to facilitate analyses and treatment recommendations in the smart hospital system. Implementation of a smart healthcare system is divided into five features, namely an intelligent network, cloud computing, data analysis, diagnosis, and treatment. The proposal of the healthcare system will ensure improvement in patients' monitoring and analysis by moving toward anticipation and early discovery of disease and will benefit those who are in intensive care observation for critical health conditions. In [38] it is stated that the introduction of an IoT in smart healthcare can improve productivity, which will lead to more effective use of the time of the medical caretaker. The medical staff can have more time to improve patient care, including the improvement of the hospital environment. This would be of prodigious advantage to the patient and lead to better quality treatment.

Kumar et al. [40] established a patient-monitoring system in smart healthcare that is controlled by a raspberry pi to monitor the vital signs of patients, such as the body temperature, respiration level, heartbeat rate, and body movement. Data about the patient are received by using sensors and are presented on a screen, using puTTY software. However, the system is not designed for emergency alerts and alarm notification for assisting family members or doctors to give information to the patient in case of an unexpected condition. In [41] the author demonstrated monitoring a patients' health condition by using the smart hospital system. The health condition of patients can be monitored by using the spark kit. It collects data of a patient about the temperature and heartbeat rate and transmits an alert warning if any of the parameters obtained crosses a predefined threshold value.

Multi-access Edge Computing for Smart Health

The IoT system is intended to encourage a substantial flow in demand for computing resources, data, as well as networking structures to aggregate the anticipated multitude of integrated devices. All these exciting demands will require modification of existing system infrastructure, as well as cloud computing technologies. The European Telecommunications Standards Institute introduced the Industry Specification Group and mobile edge computing to increase the intelligence to the edge of the network, along with higher storage and processing capabilities [42]. Edge computing defines the empowering technologies that allow computation to be generated at the edge of computing; cloud services serve as downstream data and IoT services use upstream data. Edge computing permits efficient data processing in large amounts to eliminate costs by ensuring data-steam acceleration, including real-time data processing without latency [43]. Mao *et al*. [44] propose that edge computing can assist in smart healthcare by enabling better insight into the heterogeneous healthcare environment to provide affordable and quality patient care.

In [45] the authors offer a short description of the proposed multi-access edge computing-based architecture for smart health applications. The proposed system architecture, shown in Figure 2.5, ranges from the information foundations located on or around patients to the service providers. It contains the following major components:

Hybrid sensing sources: Multiple sensing devices surround patients and are attached to them to source data. These include wearable non-medical and medical sensors.

Mobile or infrastructure edge node: This can be a mobile device that works as a WBAN and contains several sensor nodes that measure different vital signs or an infrastructure edge node that is deployed near the patient to recover data collected by the body area sensor network and transmit it to the cloud.

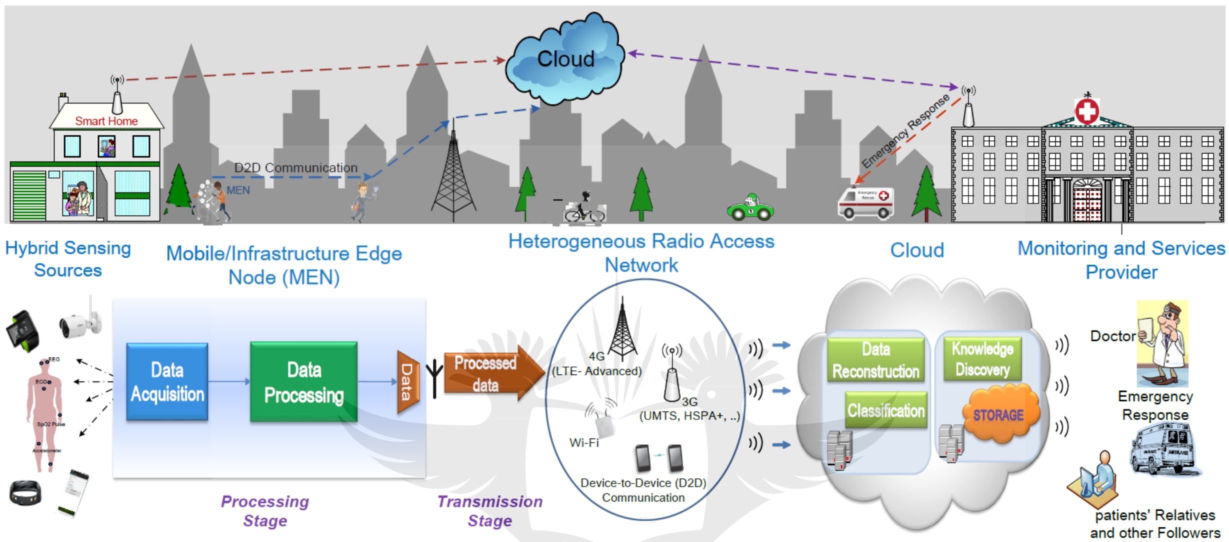


Figure 2.5: Proposed smart health system architecture [45].

Edge cloud: It is a local edge cloud where complex data analysis methods for pattern recognition, data storage, trend discovery, and population health management can be enabled. An example of the edge cloud can be medical healthcare, which monitors and records patients' state while providing the required help.

Monitoring and Services Provider: A health service provider can be a doctor, an artificial intelligence ambulance, or even a patient's relative, who provides preventive, curative, emergency, or rehabilitative healthcare services to the patient.

2.5 Fuzzy Logic System

The concept of fuzzy logic was first developed by Dr. Lotfi Zadeh of the University of California at Berkeley in the 1960s [46]. A fuzzy logic system is a computer-based algorithm system whose reaction is described by a set of linguistic rules. The algorithm characterises expert information in the form of IF-THEN, makes rules that relate vague input statements by vague output decisions or actions. The IF part of the rule is called the antecedent and defines imprecisely the system states, while the THEN part of the rule is called the consequent and signifies actions that could be taken to remedy the state condition [47]. A fuzzy system consists of necessary components to implement a fuzzy algorithm, as shown in Figure 2.6. In the fuzzification component, the crisp input data are collected by sensors and computed based on membership functions that are explained. The interface component communicates with the fuzzy reasoning method by applying an appropriate fuzzy operator to obtain the fuzzy set to accumulate in the output variable. The defuzzifier converts the set to a crisp output by applying a precise defuzzification method [48].

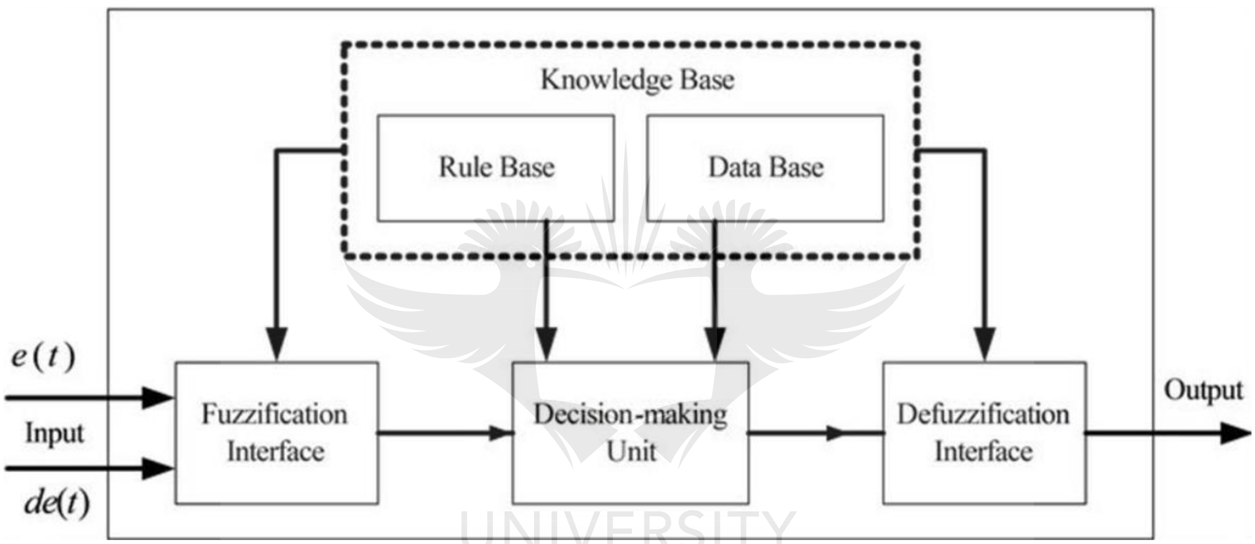


Figure 2.6: Components of the fuzzy logic system [48].

The fuzzy logic system is one of the decision support systems proven to be an effective method to provide clarity in healthcare facilities. The aim of fuzzy logic in healthcare is to provide soft decision strategies approximating human decision-making to build a flexible information-processing system. It minimises human decision-making by working out a simple way to determine a definite solution from imprecise information.

In [49] the researchers propose an intelligent decision-making system for estimated reasoning that can almost handle the indistinctness of critical risk for human health using a fuzzy logic control system. This contains a diagnosis of healthcare risk related to pulse rate, BP, and kidney function. The glomerular filtration rate is used to analyse the level of BP and kidney function. Under this concept, the fuzzy logic control system is presented to represent the parameters that may result in a risk to human health and examination by using a rule-based factor implemented using the MATLAB tool program.

In [50], the authors reported the discoveries of an insulin pump and the implementation of a fuzzy logic system to perform as an artificial algorithm in diabetes patients. The paper demonstrates a closed-loop control system for regulating blood glucose levels in type-1 diabetic patients. The purpose of the controller is to avoid hypoglycaemia, which is a blood glucose concentration above 3 mmol/l, and a reduction in the blood glucose level; the blood glucose average should be less than 7 mmol/l in diabetic patients. The control algorithm uses the knowledge of physicians to treat diabetics by using a Mamdani-type fuzzy logic controller (FLC). Using the analysis and the capabilities of the fuzzy controller assessment, the results of FLC are compared with a proportional integral derivative controller for each diabetic patient. The results have shown that FLC yields more constant results in the presence of uncertainty in system parameters, which may vary from patient to patient [51].

In [52] the researcher chooses a fuzzy logic algorithm system as the model to detect abnormal vital signs in patients with chronic heart failure and trigger a notification to alert the doctor and relatives. Fuzzy logic is a multivalued logic where truth values fall between any real numbers 0 and 1. Fuzzy logic handles the partial truth concept, in the truth value range between false and completely true [53]. The researcher determines three membership functions that are chosen for each input in a patient-monitoring system, with three input sensors, namely HR (ECG), temperature and heartbeat sensors. In adult patients with chronic heart failure, the resting HR ranges from 60 to 70 beats per minute. This is proportionally lower with age. The mean HR is slightly lower in males than in females of the same age. In patients with heart failure, a resting HR of more than 80 BPM could cause myocardial dysfunction, which further deteriorates their condition. Using the HR sensor, the ambient HR data from the HR sensor is to be grouped into three fuzzy sets (low, medium, and high), as shown in Figure 2.7 below.

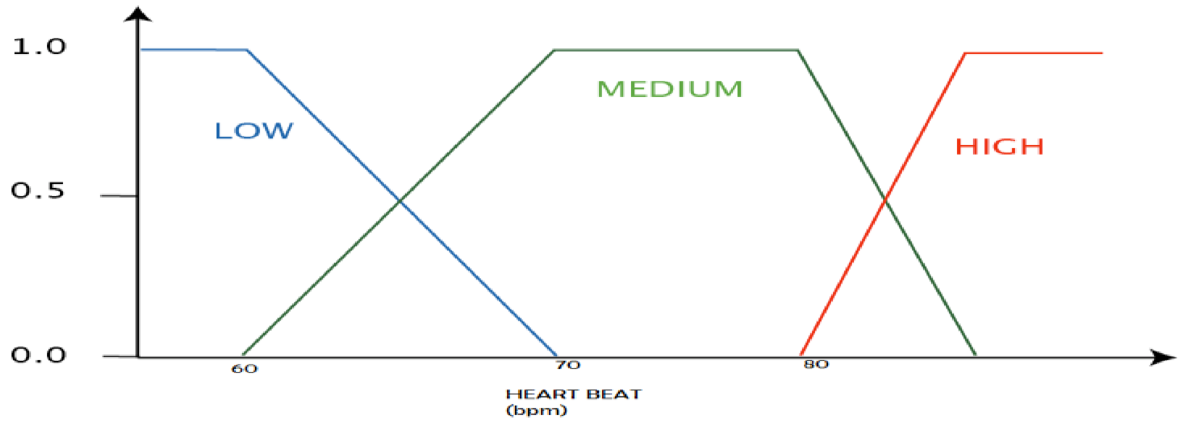


Figure 2.7: Fuzzy Membership Sets for HR sensor [53].

2.6 Conclusion

In this chapter, various articles and appropriate designs for this study are reviewed, as well as similar work that led to the development of the proposed system. The chapter examined different research projects, approaches, and techniques that were used for the CPS. The challenges facing the healthcare system were also discussed and different proposed and employed approaches were reviewed. Most research in the healthcare system that was reviewed still indicated unresolved problems, which included the use of proper quality sensors with low cost, among others the development of a real-time monitoring system. This research will consequently attempt to fill some gaps in the literature by adjusting and adapting the CPS for smart healthcare.

CHAPTER 3: METHODOLOGY

The aim of this research study is to develop a CPS based on smart healthcare. In the previous chapter, the background concepts related to the CPS for smart healthcare were reviewed. This chapter will explain the possible approaches that were used to design the system, including the hardware and software involved. To achieve this design, different sources of information are used, including books, journals, conference proceedings, and websites.

3.1 Research Methodology

In this chapter, the researcher investigates the type of sensors used in building the system and the interconnection of wireless communication technologies. A qualitative investigation was used to carry out a reasonable analysis of various sensor technologies and different types of wireless sensor network topologies that would enable the implementation of the CPS. The design of the CPS involves the following:

- Identifying the various components to be used in the design (Arduino boards, sensors, WiFi modules, XBee modules, etc).
- Programming the Arduino boards, using C language code.
- Creating and programming wireless sensor network topology.
- Using the Thingspeak platform to receive data from the device.
- Using the MATLAB GUI for monitoring the dashboard and fuzzy logic algorithm.

3.2 Description of Hardware Components

The system hardware requirements establish the features and services that the CPS should provide and the limitations under which it must operate. Defining the system requirements helps the designers to make a better choice of components and optimises resources. This section highlights the functional and non-functional component requirements for the proposed system.

3.2.1 Arduino Uno/Micro

Arduino Uno is a microcontroller board built on the ATmega328P Integrated Circuit. It has 14 digital input/output pins, of which six can be used as pulse-width modulation (PWM) outputs and six as analogue inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button [54]. "Uno" indicates one in Italian and it was meant to describe the release of

Arduino Software (IDE) 1.0. The Uno board and version 1.0 of Arduino Software IDE were the reference versions of Arduino.

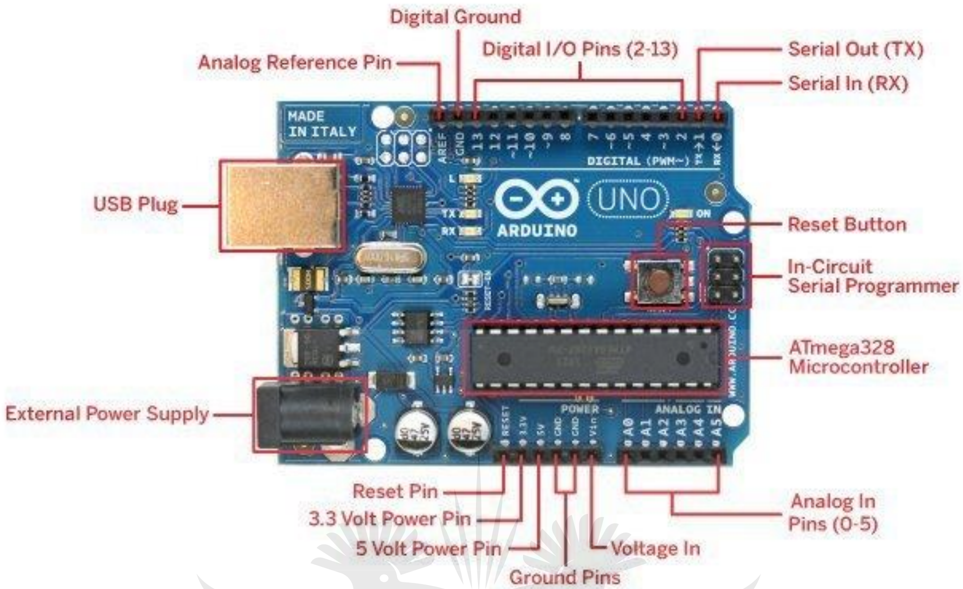


Figure 3.2: Arduino Uno board layout.

Technical data	
Microcontroller	Atmega328P
Operating voltage	5V
Digital I/O pins	14 (6 provide PWM channels)
Analog input pins	6
DC current per I/O pin	20 mA
DC current for 3.3V pin	50 mA
Flash memory	32 KB (0.5 KB used by bootloader)
SRAM	2 KB
EEPROM	1 KB
Clock speed	16MHz

Table 3.1: Technical specification of Arduino UNO

The Uno board is the first in the development of USB Arduino sheets and the reference demonstration for the Arduino stage [54]. Arduino micro has similar functionality as Arduino Uno, but it is smaller in size, uses the ATmega32u4 Integrated Circuit, and has 20 digital input/output pins, of which seven can be used as PWM outputs and 12 as analogue inputs [54]. Table 3.1 below gives the specifications for Arduino Uno and Table 3.2 shows the specifications for Arduino micro.

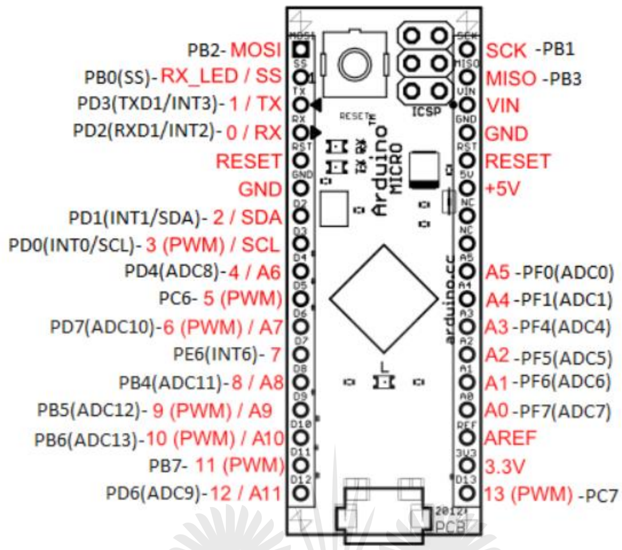


Figure 3.3: Arduino micro-board layout.

Technical data	
Microcontroller	Atmega32u4
Operating voltage	5 V
Digital I/O pins	20 (7 provide PWM channels)
Analog input pins	12
DC current per I/O pin	40 mA
DC current for 3.3V pin	50 mA
Flash memory	32 KB (4 KB used by bootloader)
SRAM	2.5 KB
EEPROM	1 KB
Clock speed	16 MHz

Table 3.2: Technical specification of Arduino micro

3.2.2 WiFi Module

The ESP8266 module is a WiFi empowered system on the chip module established by the Espressif system. ESP8266 is a WiFi microchip at a low price, with microcontroller capability and full TCP/IP stack. It is mostly used for designing IoT embedded applications. The ESP8266 is capable of accommodating applications and the elimination of all WiFi interacting functions from another application processing unit. There are few applications to the pre-program ESP8266 module, but the better way of using it is with AT command set firmware by simply connecting to the Arduino device. The module represents the integration of sensor devices and the internet cloud [53].

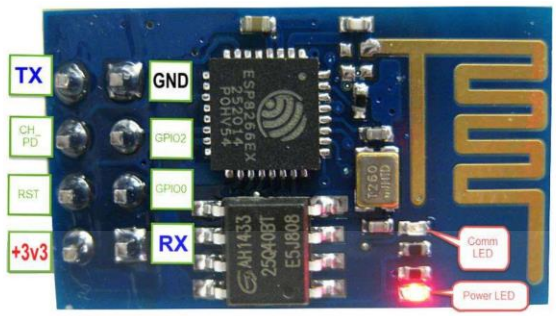


Figure 3.3: ESP8266 WIFI module

ESP8266 comes with the following capabilities and features:

- 2.4 GHz WiFi (802.11 b/g/n, supporting WPA/WPA2),
- General-purpose input/output (16 GPIO),
- Inter-Integrated Circuit (I^2C) serial communication protocol,
- Analogue-to-digital conversion (10-bit ADC)
- Serial Peripheral Interface (SPI) serial communication protocol,
- I^2S (Inter-IC Sound) interfaces with direct memory access (sharing pins with GPIO),
- Universal Asynchronous Receiver Transmitter (UART) (on dedicated pins, plus a transmit-only UART can be enabled on GPIO2), and
- Pulse-width modulation.

ESP8266 consists of a 32-bit RISC CPU design by the Tensilica Xtensa L106 running at 80 MHz (or overclocked to 160 MHz). It has a 64 KB boot ROM, 64 KB instruction RAM, and 96 KB data RAM. External flash memory can be accessed through SPI [54].

3.2.3 XBee Radio Module

The XBee radio module is a wireless transceiver that uses a fully implemented IEEE 802.15.4 protocol for data communication that provides features needed for robust network communication in WSN. XBee support network topologies such as point-to-point, point-to-multipoint, and peer-to-peer.

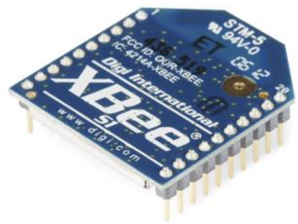


Figure 3.4: XBee radio module

The XBee module can transmit and receive data in a radius of approximately 10 to 100 meters, has low power consumption and a low data rate and provides network security [13]. Any microcontroller that has a UART interface or TTL port can communicate with the XBee module, as shown in the diagram below.

Figure 3.5: System data flow diagram in a UART-interfaced environment [55]

3.2.4 Pulse Sensor

The Arduino pulse sensor is a plug-and-play heart-rate sensor mostly used in the Arduino microcontroller. In essence, it syndicates a modest optical HR sensor with strengthening and noise filtering circuitry, making it fast and easy to get dependable pulse readings. When the heartbeat sensor is functional, the beat LED flashes in correspondence to each heartbeat. This digital output can be integrated directly to the microcontroller to measure the beats per minute (BPM) rate. It works on the principle of light modulation by the circulation of blood through fingers at each pulse.

Figure 3.6: Pulse sensor

The pulse sensor that the researcher uses is fundamentally a photo plethysmograph. The heart pulse signal that comes out of a photo plethysmograph is similar to the oscillation of voltage, and it has an expectable wave shape, as shown in Figure 3.7. The representation of the pulse wave is called a PPG. The pulse sensor amped responds to the comparative changes in light intensity. If the quantity of light on the sensor remains constant, the signal value will remain at (or close to) 512 (midpoint of ADC range). When there is more light, the signal amplitude will increase and when there is less light the signal amplitude will reduce. Light from the green LED that is reflected in the sensor changes during each pulse.

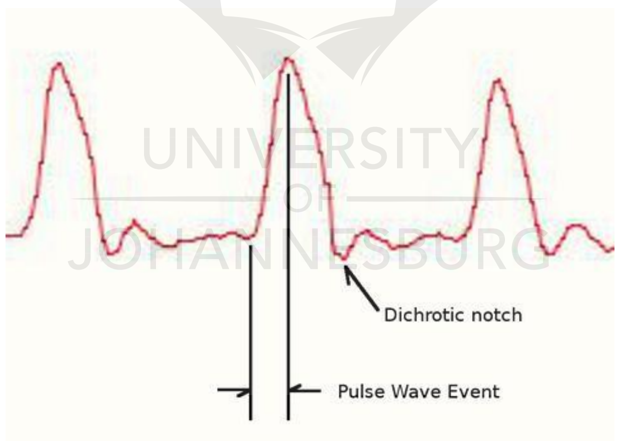


Figure 3.7: Pulse rate wave

A normal resting HR for an adult ranges from 60 to 100 BPM. A low HR demonstrates better heart function and cardiovascular fitness; an athlete may have a normal resting HR close to BP< 40. It is better to consult a medical practitioner if the HR is lower than 60 BPM if one is not an athlete.

3.2.5 DHT11 Temperature and Humidity Sensor

The DHT11 is a low-cost digital humidity and temperature sensor. It uses a thermistor and a capacitive humidity sensor to measure the environmental air and emits a digital signal on the data pin. The Arduino microcontroller provides timing for the DH11 to take data. In this study, the DH11 sensor is used to measure the temperature and humidity of a patient's room. Figure 3.8 shows the DHT11 sensor and DHT11 specifications in Table 3.3.

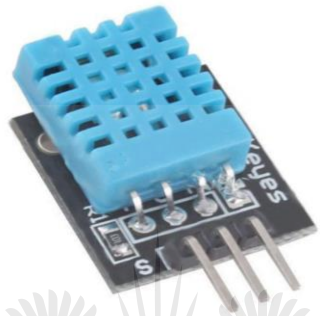


Figure 3.8: DHT11 sensor

Item	Measurement Range	Humidity Accuracy	Temperature Accuracy	Resolution	Package
DHT11	20-90 %RH 0 – 50 ^{o}C	±5%RH	±2^{o}C	16-bit	Four-pin Single Row

Table 3.3: Specifications of DHT11

3.2.6 LM35 Temperature Sensor

The LM35 is an accurate integrated-circuit temperature sensor with an output relative to the temperature. It is a three-terminal linear temperature sensor from National Semiconductors characterised by low self-heating and does not cause more than 0.1°C temperature rise in still air. The LM35 measures temperature from -55°C to +150°C with output voltage increases in 10 mV per degree Celsius. The pinout of the LM35 is shown in the figure below.

Figure 3.8: LM35 temperature sensor [56]

3.2.7 X-Band Motion Detector Module

The X-band motion sensor operates in the 10.525GHz X-band frequency and represents detected body activities through oscillations in its high and low outputs. The X-band motion sensor is a well-known component in automatic door openers and security systems and can sense body movements in a room, yard, or even through a wall. Its sensitivity can be adjusted manually with a potentiometer, contributing through-line of sight detection up to 9 m [57].

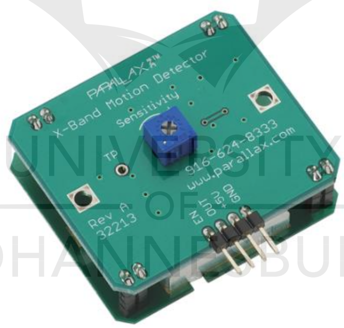


Figure 3.9: X-band motion detector [57]

The working principle of the X-band radar system is represented using the block diagram shown in Figure 3.10. The built-in transmit antenna sends constant frequency signals to the human body and waits for the rebound signals from an object or movement of a person. The reflected signal is received by the Rx antenna and is then enlarged by a +LAN. The in-phase - I and quadrature – Q phase indicators are used to decrease the null-point problem, as shown in the diagram below.

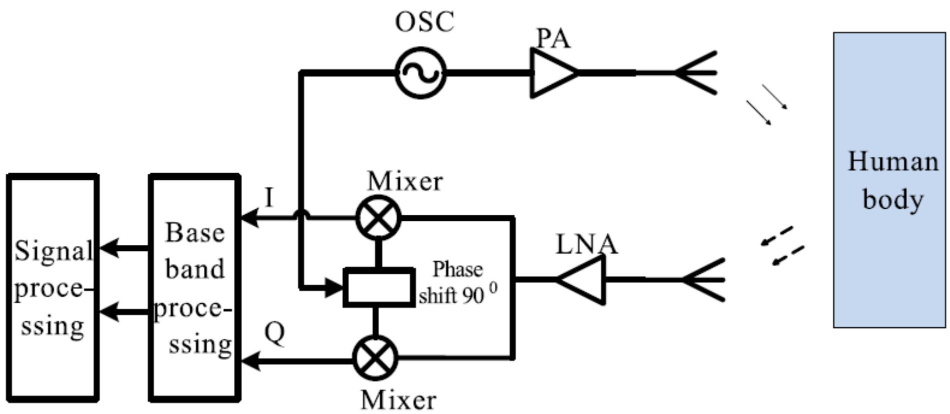


Figure 3.10: X-band radar system block diagram

The transmitted signal from the X-band radar sensor device can be defined by:

$$Y_T = \sqrt{2P_T}\cos\left(2\pi f t + \emptyset(t)\right) \qquad \textbf{(3.1)}$$

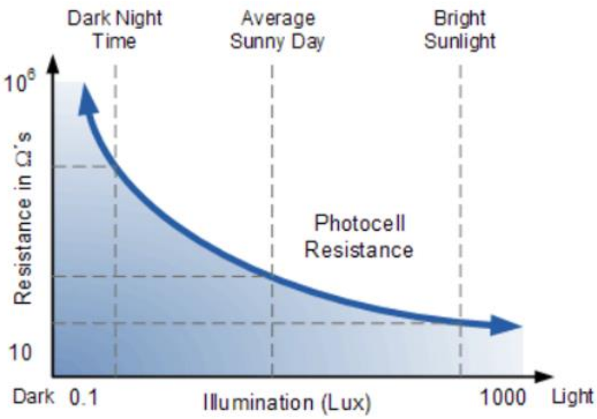
where f is the carrier frequency, P_T is the power of the transmitter, $\emptyset(t)$ is the phase noise and t is the transpiring time. The original stage of the transmitting signal is assumed to be zero. The signal Y_T is then reproduced by the human body. The movement of the human body can be measured as the periodic signal x (t). The distance from the human's location to the antennae of the X-band radar sensor system is d (m). The received signal Y_R can be expressed as follows:

$$Y_R = A_R\cos\left(2\pi f t - \frac{4\pi d}{\lambda} - \frac{4\pi x(t)}{\lambda} + \emptyset\left(t - \frac{2d}{c}\right)\right) \qquad \textbf{(3.2)}$$

where λ is the wavelength of the signal, c is the speed of light and A_R is the return signal amplitude that depends on the transmitted power and the antenna gain.

3.2.8 Photoresistor

A photoresistor, also known as a light-dependent resistor (LDR), is a light-sensitive device most commonly used to designate the presence and absence of light or to measure the light intensity. As shown in the graph below, in the dark its resistance is very high; it can be up to 1MΩ, but when the LDR sensor is visible to light, the resistance drops quickly. This may imply that significantly fewer ohms can be measured, depending on the light intensity.



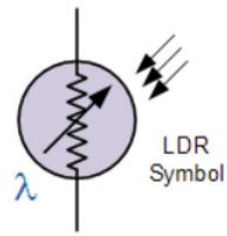


Figure 3.11: LDR light graph and LDR symbol

3.2.9 Passive Infrared Sensor

A passive infrared sensor (PIR sensor) is an automated sensor that detects levels of infrared radiation. A PIR is mostly used to detect motion but does not provide details of the type of movement and the distance from an object. Its sensing range is up to 7 m in a 100-degree cone. The figure below shows the back and front of a PIR sensor.

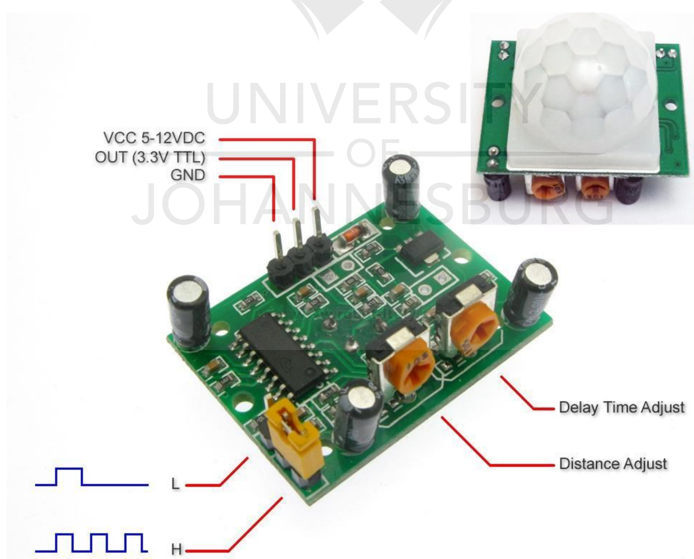


Figure 3.12: Back and front of PIR sensor [58]

3.2.10 Ultrasonic Sensor

An ultrasonic sensor detects an object and measures the distance between the sensor and object by sending an ultrasonic wave from its emitter and detecting a reflection of the wave from an object with its receiver. The ultrasonic sensor measures the distance to an object by measuring the time between the emission and reception. The range distance for the ultrasonic sensor is from 2 cm to 400 cm. An ultrasonic sensor with pin descriptions is shown below in Figure 3.13.

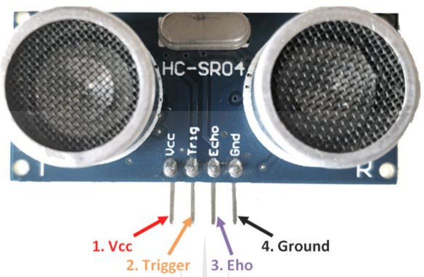


Figure 3.13: Ultrasonic sensor

3.3 Design and Configuration Software

A few computer software applications are used to complete the system, from sensors to monitoring the display. The software is used to program the system to give better results. Most of the programming software is sensitive and must be understood first before programming the device to avoid damage. This section describes and shows computer software and applications that are used in designing the system.

3.3.1 X-CTU Software

X-CTU software is a free user-friendly application for configuring and managing Digital radio frequency modules (XBee devices) through a simple graphic interface. The software is provided by DIGI and can be downloaded free online. The X-CTU can be used to set different parameters, such as the XBee destination MAC address and network ID, as well as communication types such as broadcast, multicast, or unicast. In this study, three of the XBee radios were configured, two of them as routers and one as a coordinator. Figure 3.14 and Figure 3.15 show a block diagram of the XBee configuration as a router and coordinator.

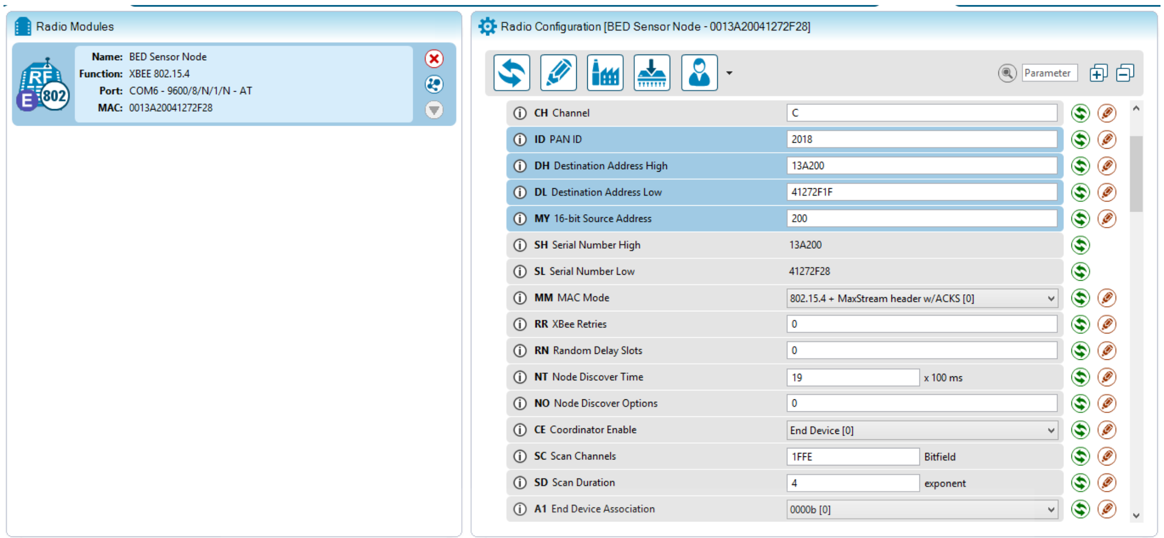


Figure 3.14: XBee configurations for router nodes

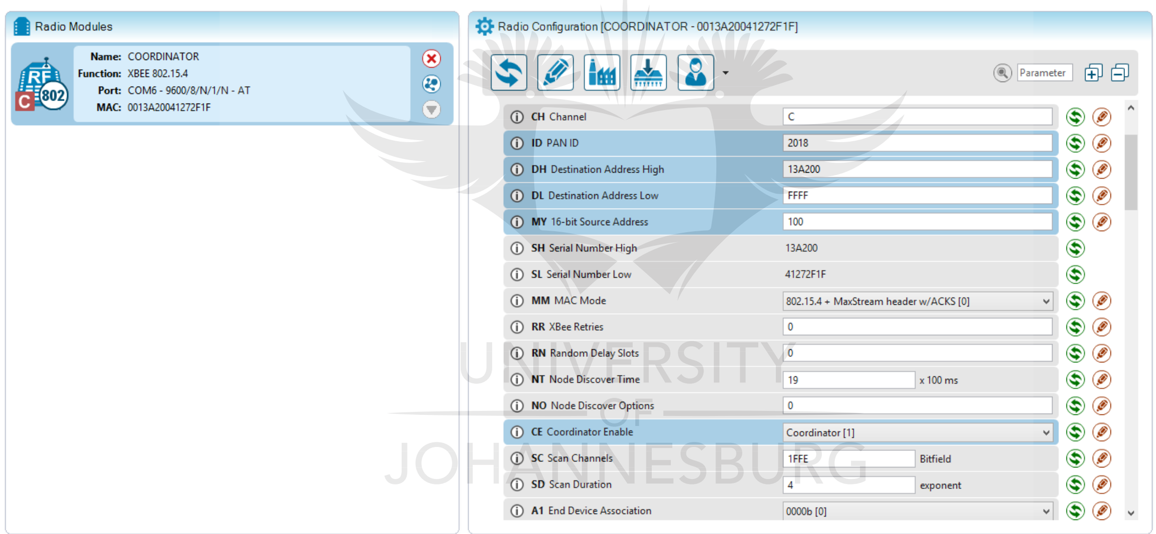


Figure 3.15: XBee configurations for coordinator nodes

3.3.2 Arduino Integrated Development Environment

The Arduino integrated development environment (IDE) tool allows the user to compile and write code for the Arduino micro-controller. Arduino language is a simplified language made up of C and C++ languages using special rules of code configuring. The code is uploaded to the Arduino board via a serial port, usually a USB port. User-written code needs two elementary functions, for opening the sketch (code) and the main program loop that is assembled and linked with a program stud main into an executable returning executed program with a GNU toolchain, which is encompassed in the IDE distribution. The Arduino IDE is developed with the program AVR

Downloader Uploader to translate the executable program into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board firmware. Figure 3.6 shows the Arduino IDE.

Figure 3.16: Arduino IDE

3.3.3 Thingspeak

ThingSpeak is an IoT platform for building an IoT prototype. It is an open-source application originally launched in 2010 by ioBridge as a provision in support of IoT systems, which allows the user to collect, visualise and analyse live data. Users can design IoT systems without the need for setting up extra servers. Data collection is done using REST API or MQTT and data analytics and visualisation are done using MATLAB analytics. The ThingSpeak service is operated by MathWorks. To use the ThingSpeak platform application, one must create a new MathWorks Account or log into one's existing MathWorks Account.

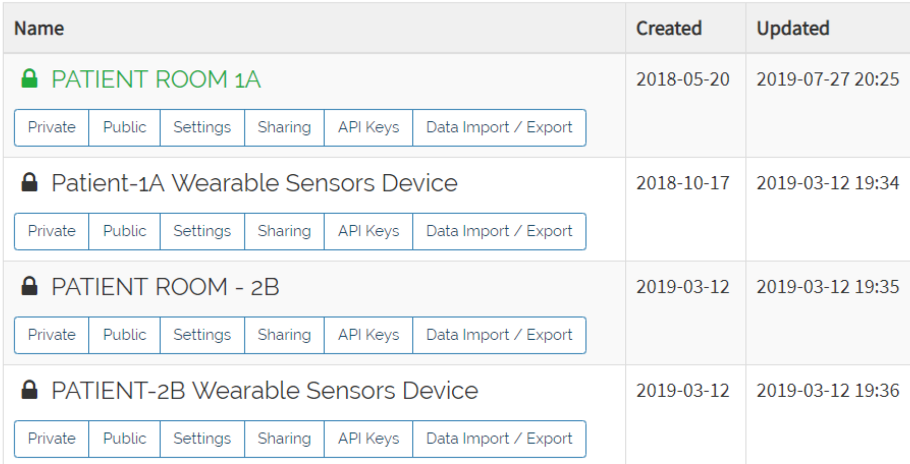


Figure 3.17: ThingSpeak platform application.

In the Thingspeak server, it takes 15-second delays for each data entry. Firstly, the sensor data from Arduino arrive in the storage and are then graphically shown in the display. The information entered in the storage is a channel- and field-specific. That means it will go to the exact field of the channel specified by the user. As shown in Figure 3.17, there are four free channels and in each channel are eight fields that can be used.

3.3.4 Fritzing

Fritzing is open-source software for generating and designing breadboard diagrams and PCB layout design. It produces an imaginative ecosystem for users. It has a large community helpdesk and facilities associated with Arduino. Users can simply share their designs, ideas, and prototypes on the Fritzing platform [59].

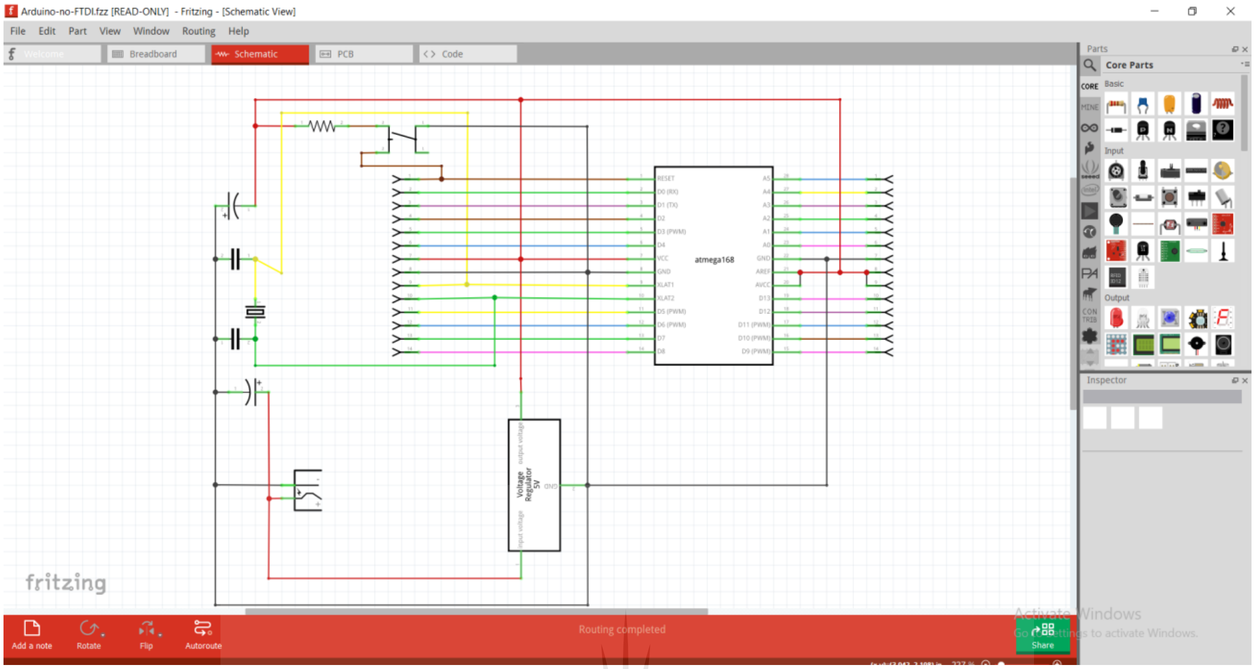


Figure 3.18: Fritzing schematic window view

3.4 Fuzzy Logic System

A fuzzy logic system (FLS) is a nonlinear classification whose behaviour is defined by a set of linguistic rules in a rule base. The concept of a fuzzy logic system has been effectively applied to many industrial systems as a system that reproduces human thinking. The fuzzy logic system offers an actual tool for unfolding the characteristics of a system that is too difficult to acknowledge through accurate scientific analysis.

If-Then Fuzzy Rule

Assuming there are two inputs, *x*, and *y*, to the system and the output is *z*, the fuzzy if-then rule is as follows:

$$\text{IF } x \text{ is } \boldsymbol{A_i} \text{ and y is } \boldsymbol{B_i} \text{ then z is } \boldsymbol{C_i},$$

where *x* and *y* are linguistic variables. ***A, B,*** and ***C*** are fuzzy sets considered by membership functions.

The fuzzy membership function for input parameters is defined from a crisp set to present the degree of truth. By analysing these inputs and their variety one can suggest the membership functions of input variables using equations. Fuzzy membership expression will be as follows :

$$\mu_{Low}(x) = \begin{cases} 1 & x < a \\ \frac{b-x}{b-a} & a \leq x \leq b \end{cases} \tag{3.3}$$

$$\mu_{Mid}(x) = \begin{cases} \frac{x-a}{b-a} & a \leq x \leq b \\ 1 & b \leq x \leq c \\ \frac{d-x}{d-c} & c \leq x \leq d \end{cases} \tag{3.4}$$

$$\mu_{High}(x) = \begin{cases} \frac{x-c}{d-c} & c \leq x < d \\ 1 & x \geq b \end{cases} \tag{3.4}$$

where a, b, c, and d are given membership functions of fuzzy numbers that are graphically considered in the figure diagram below.

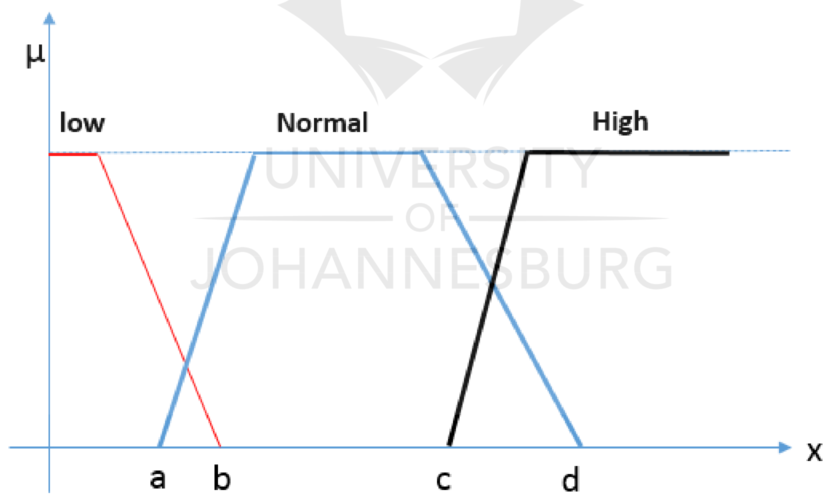


Figure 2: Inputs variables example

In this project, the MATLAB fuzzy toolbox is used for implementation and simulation of patient data analysis. In order to implement a fuzzy expert system, the representative steps are followed to determine the input and output variables, the collection of appropriate membership functions, and the design of the fuzzy rules database. The FIS editor is used for defining the output, input variables, and membership function editor. Full details of a fuzzy logic system are provided in chapter 5.

3.5 System Analytics

Thingspeak is an IoT analytics platform service that allows a user to combine, analyse, and visualise live data streams in the cloud, as demonstrated in Figure 3.19. All sensing data from CPS and wearable sensors are captured and saved using the Thingspeak IoT platform. This provides real-time visualisation of data posted by devices. The Thingspeak web platform communicates with the WiFi module that is programmed with the same application programming interface key, to provide security and immunity of the system against cyber attack.

The MATLAB code is available inside the Thingspeak–IoT platform for the analysis of data and provides the output results. Furthermore, data are exported in .csv file format to be viewed using the design of the MATLAB graphic user interface (GUI) and analysed using the fuzzy logic system. After the system had been built, experiments were carried out to validate the established background for complete robustness and reliability. The assessment of the system was performed in an indoor home environment.

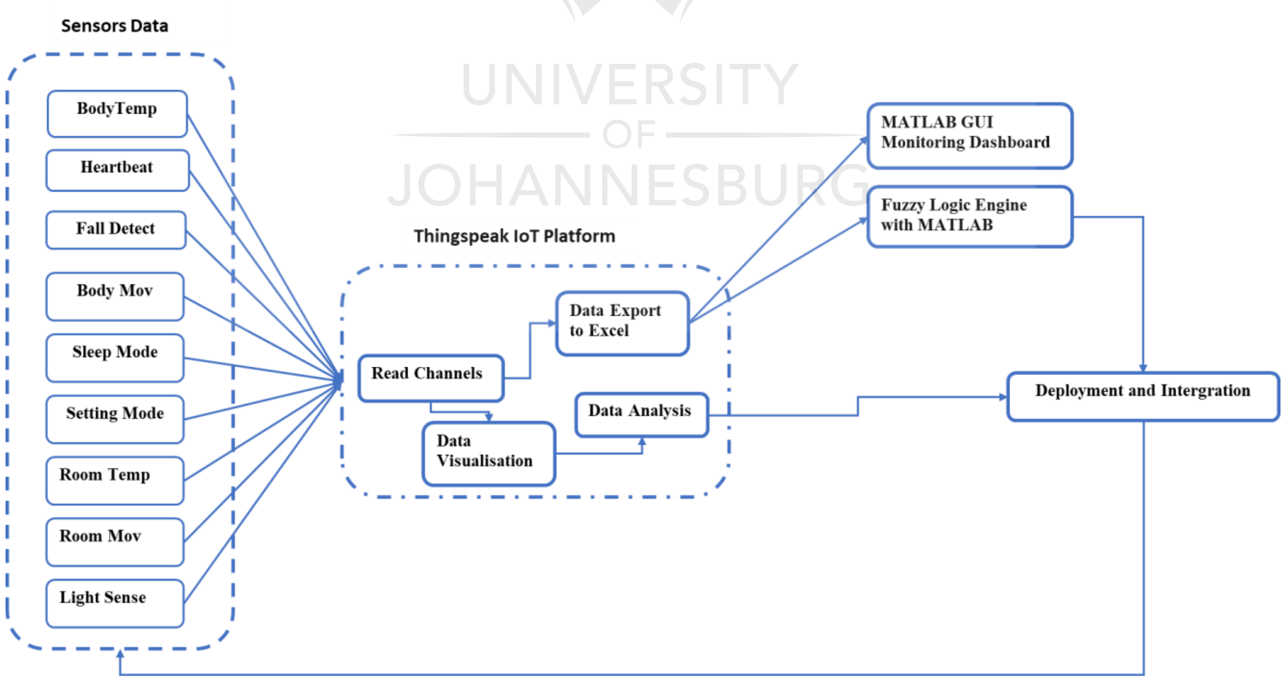


Figure 3.19: Overall system software diagram

Figure 3.20 is a fuzzy logic control flowchart diagram that how how data flow from patient monitoring system and final decisions are made. Below flow chart demomstrate the collection of data from sensors and adaption of the rule-based approach represented by fuzzy logic system.

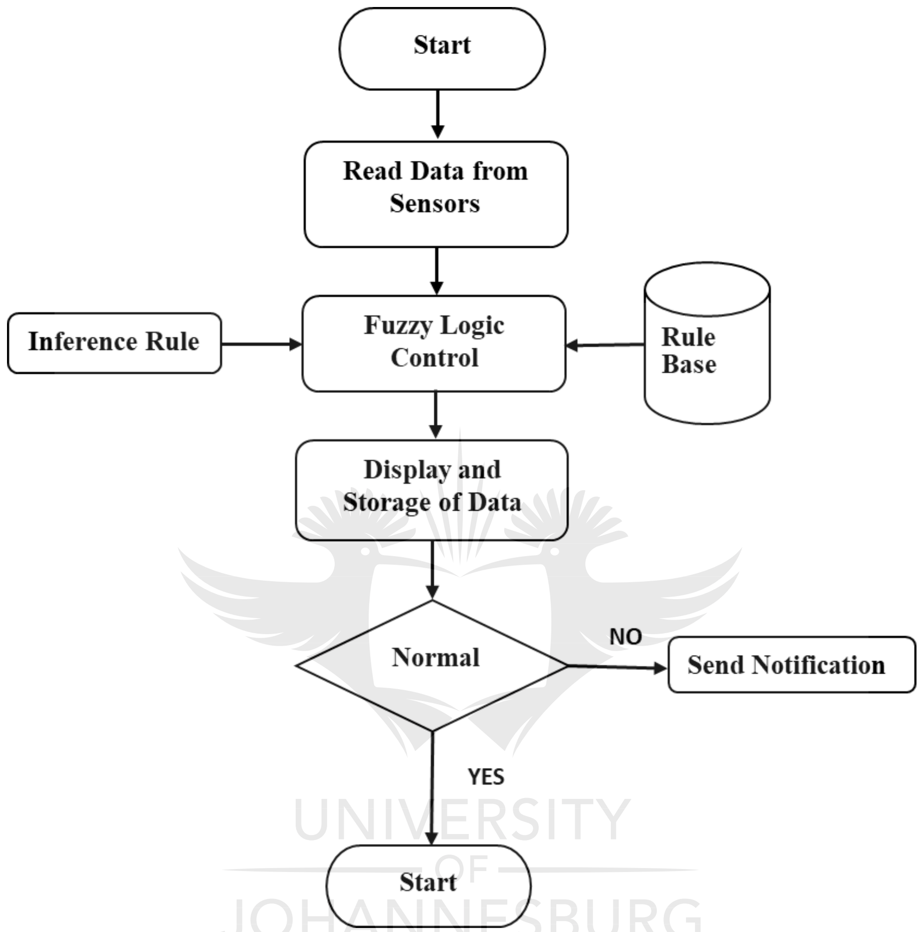


Figure 3.20: Fuzzy logic control data-flow diagram

3.6 Conclusion

This chapter introduced the overall methodology used in the proposed system, the description of building components, and an explanation of the software involved in designing the system. The design of the hardware aspect of the prototype through a careful collection of components aligns with the proposed system. The design of the graphic interface is emphasised in order to address the accessibility issue of patients' vital data remotely in real-time. The next chapter explains the system design that uses electronic components and software obtained from the research study.

CHAPTER 4: SYSTEM DESIGN AND ARCHITECTURE

The CPS is intended to enhance the efficiency of smart healthcare in current society by supplying a better, consistent, and appropriate healthcare system to patients and medical professionals. The successful implementation of this proposed system could result in a significant drop in healthcare costs by reducing clinical monitoring and emergency procedures. The main objective of the CPS is to monitor physiological parameters collected from patients and generating data records that will be stored in a cloud database using the Thingspeak platform. These data will be available to doctors at any time and be analysed using a fuzzy logic system for future prediction. The design of CPS, as shown in Figure 4.1, is divided into a few categories in other to be manageable. The patient's room has three sensor nodes that detect the condition of a patient sleeping on a bed or sitting in a chair and another sensor node checks the patient's room environment, such as room temperature, light and motion.

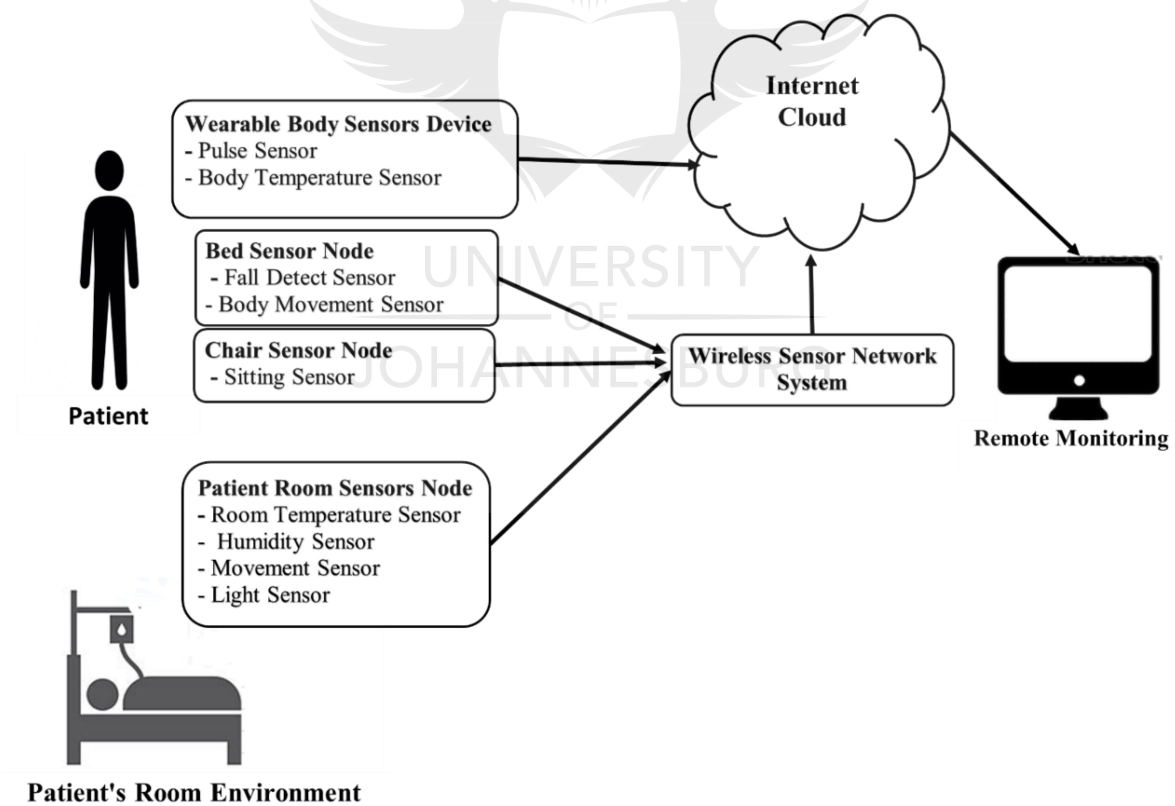


Figure 4.1: The design diagram of CPS

All three sensor nodes are interconnected by the WSN system. A wearable sensor device is attached to a patient's arm for the detection of the body temperature and HR of the patient. The device is portable and allows patients to move freely while data are captured and continually monitored by health staff through a dashboard and the internet.

4.1 Wearable Sensors

A circuit diagram of wearable sensors is shown in Figure 4.2. The Arduino code for this device is included in Appendix A. In the device designed in this study, the temperature sensor (LM35) and pulse sensor are connected to Analog input A0 and A1 of the Arduino micro-board. Two AAA (1.5 V each) batteries are used to provide 3 V, which is converted to 5 V by the step-up voltage regulator. The output of the voltage regulator is connected to the input voltage pin of the microcontroller, as shown in Figure 16. The ESP8266 WiFi module is used to collect body temperature and pulse rate sensing data and transfers these to the Thingspeak platform, which is an open-source cloud service.

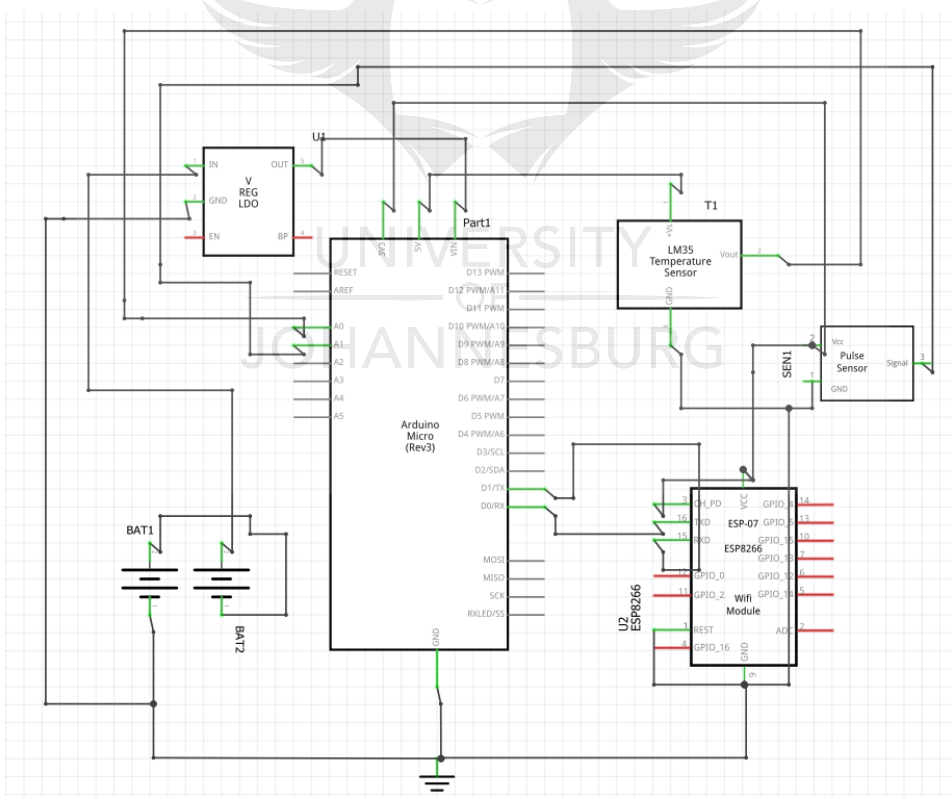
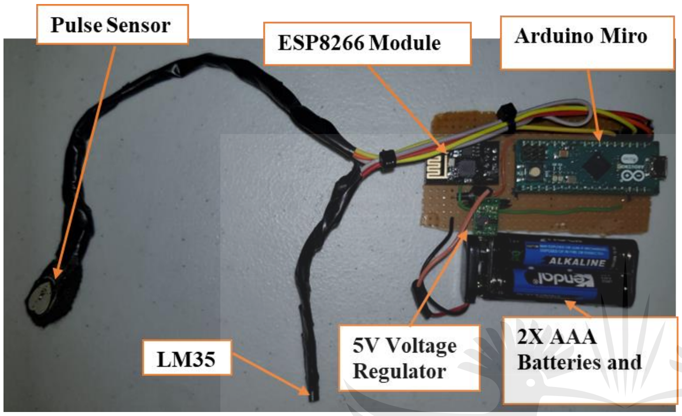


Figure 4.2: Circuit diagram of wearable sensors device

Figure 4.3 shows the circuit setup after soldering all the hardware components on the PCB board, to make the device wearable on the wrist of the user using Velcro strips, as shown in Figure 4.4. The wearable sensor system is implemented in such a way that the mini USB adapter can be used only when one needs to configure code on the Arduino. Finally, once the device is powered up and is connected to the internet on Thingspeak, it displays all the patient's vital parameters continually in a period of one minute.



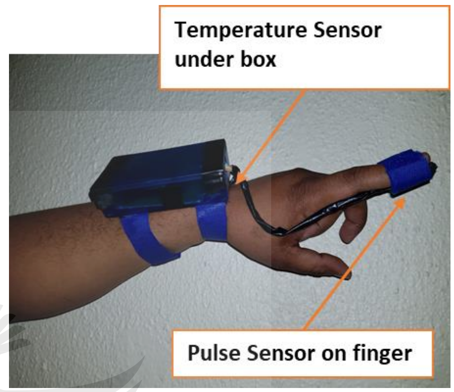


Figure 4.3: Wearable sensors circuit setup. **Figure 4.4:** Completed wearable device.

4.2 Bed Sensor Node and Chair Sensor Node

The bed sensor node and chair sensors node are similar in design, but with different sensors connected. The chair sensor node only uses one pressure mat sensor and two ultrasonic sensors to detect the patient sitting in the chair and transmit data through WSN using the XBee module. The bed sensor node consists of three pressure mat sensors, three ultrasonic sensors, and one X-band motion sensor, as demonstrated in the circuit diagram below (Figure 4.6). The bed sensor node is designed to sense the presence of a patient when sleeping on a bed, the condition of the patient when sleeping by detecting if a patient is sleeping comfortably or moving too much on the bed and detecting the patient falling from the bed.

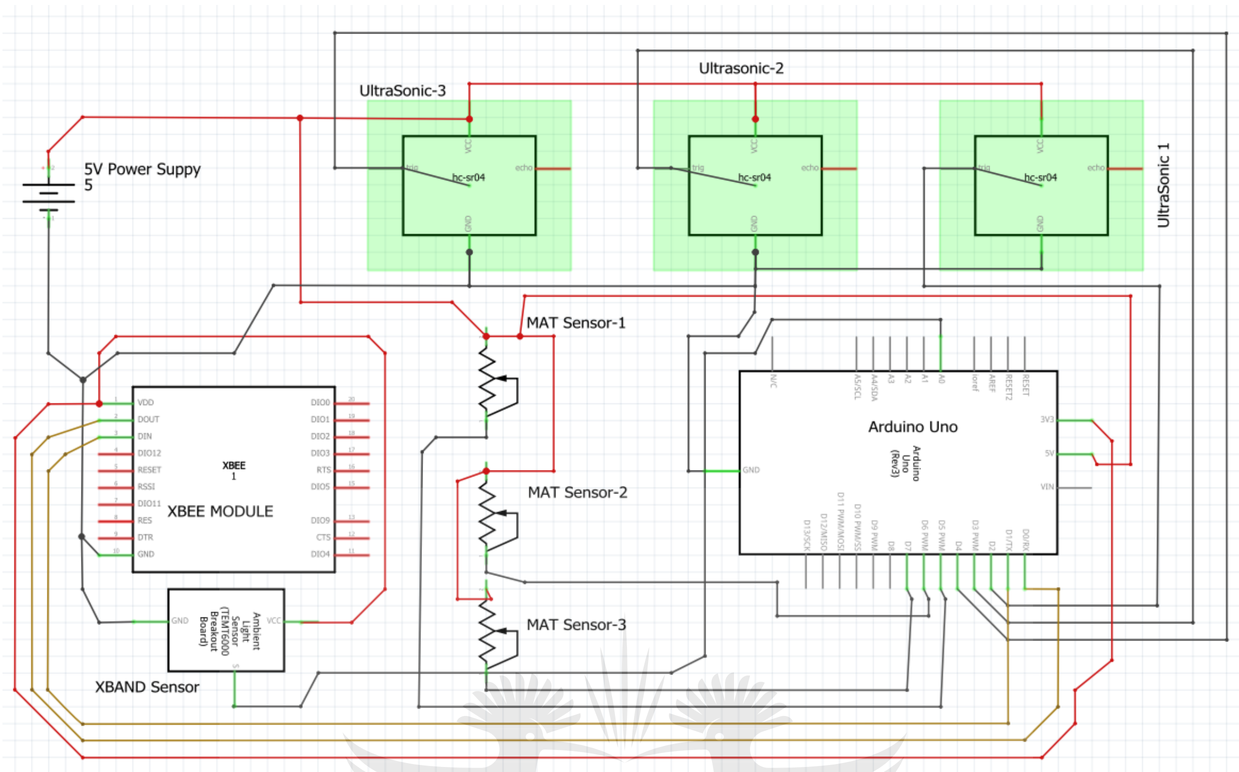


Figure 4.6: Bed sensor node circuit diagram

Wireless network topology is used to connect both the chair sensor node and bed sensor node to the coordinator sensor node. The diagram in Figure 4.7 shows the completed design for the bed sensor node, but the pressure mat sensors and ultrasonic sensors are not shown, as these are connected to the bed and floor.

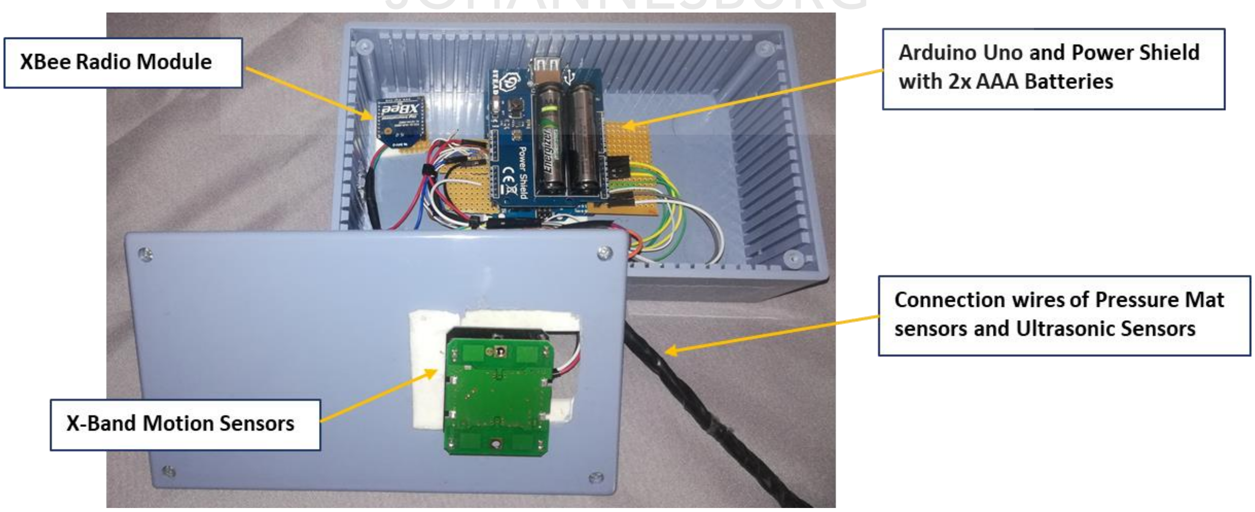


Figure 4.7: Completed bed sensors node

4.3 Patient Room Sensor Node

The patient room sensor node consists of a humidity and temperature sensor, motion detector sensor, light density sensor, XBee module, and an ESP8266 WiFi module. In this design, the Arduino power shield module is used to provide 5 V voltage to the device. The patient room sensor node is designed as a gateway device that collects sensing information from the bed sensor node and chair sensor node and passes that information to the internet cloud. The XBee module that is installed on the patient room sensor node acts as coordinate in WSN, which only receives data but does not transmit data.

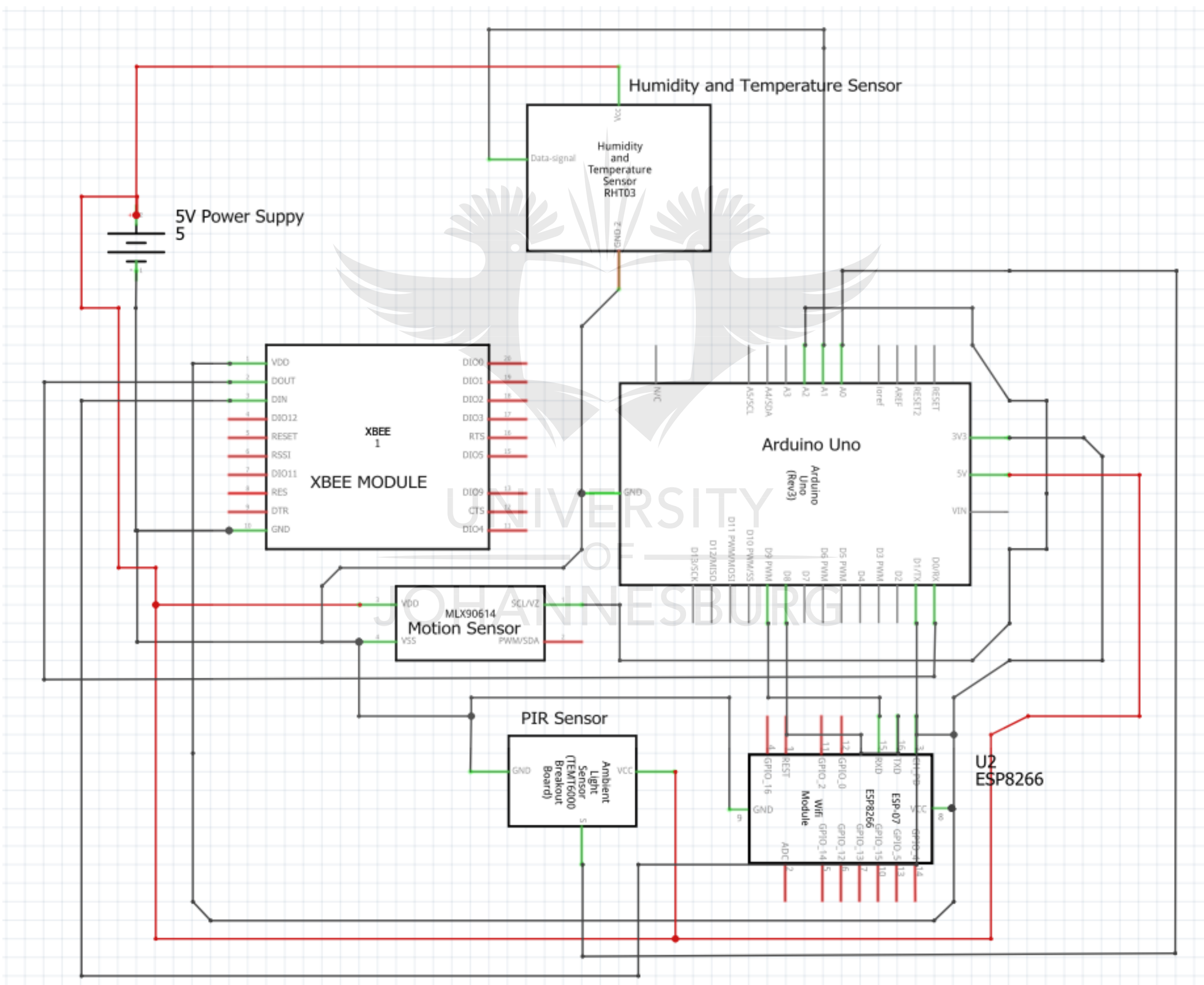


Figure 4.8: Patient room sensor node circuit diagram

The design topology of WSN on this system is star topology or point-to-multipoint topology, with only one coordinator node and multiple end-user nodes. Room environment sensors are connected to the microcontroller to detect the room temperature, humidity, light, and motion around the patient's room, as shown below in Figure 4.9.

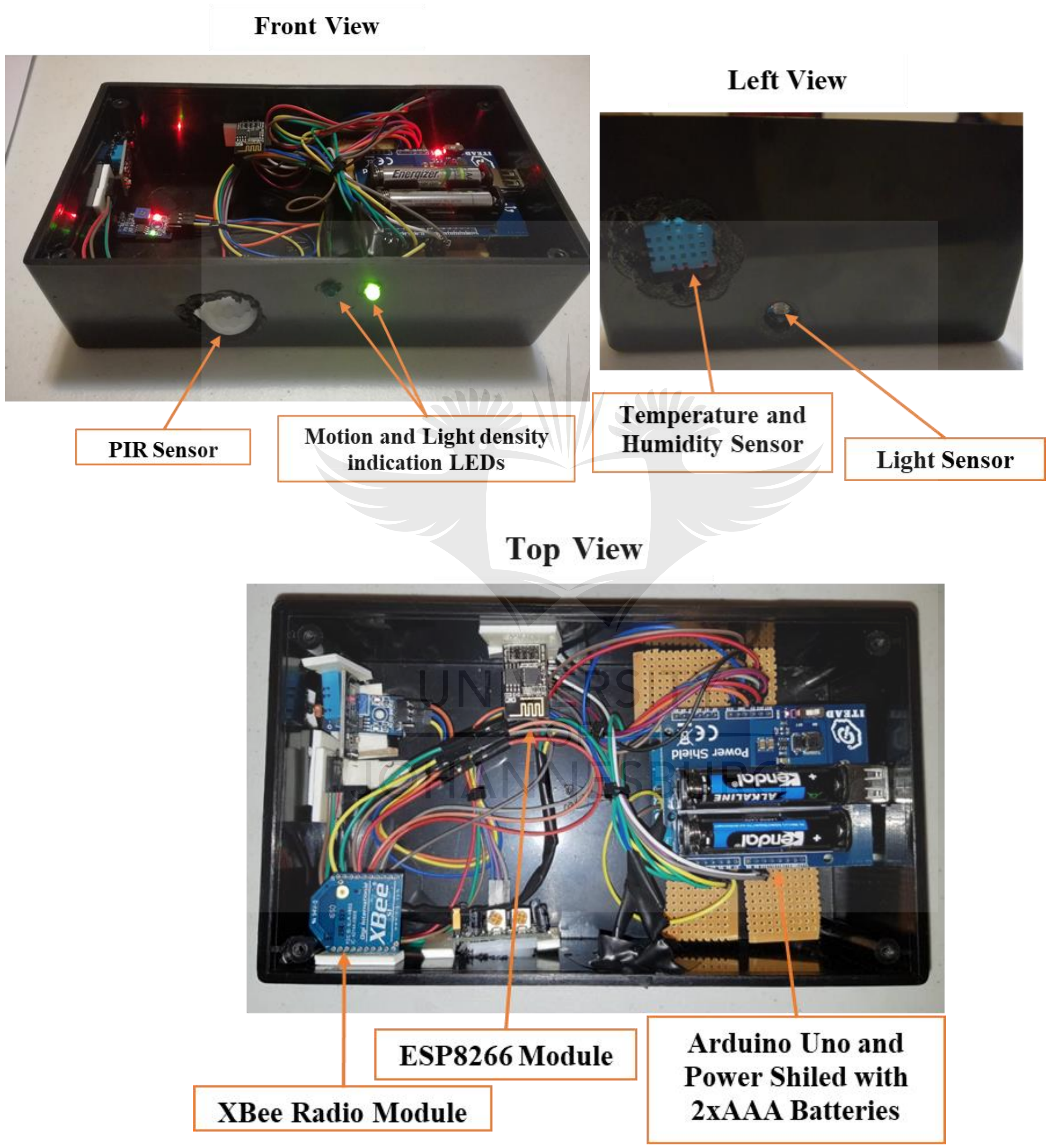
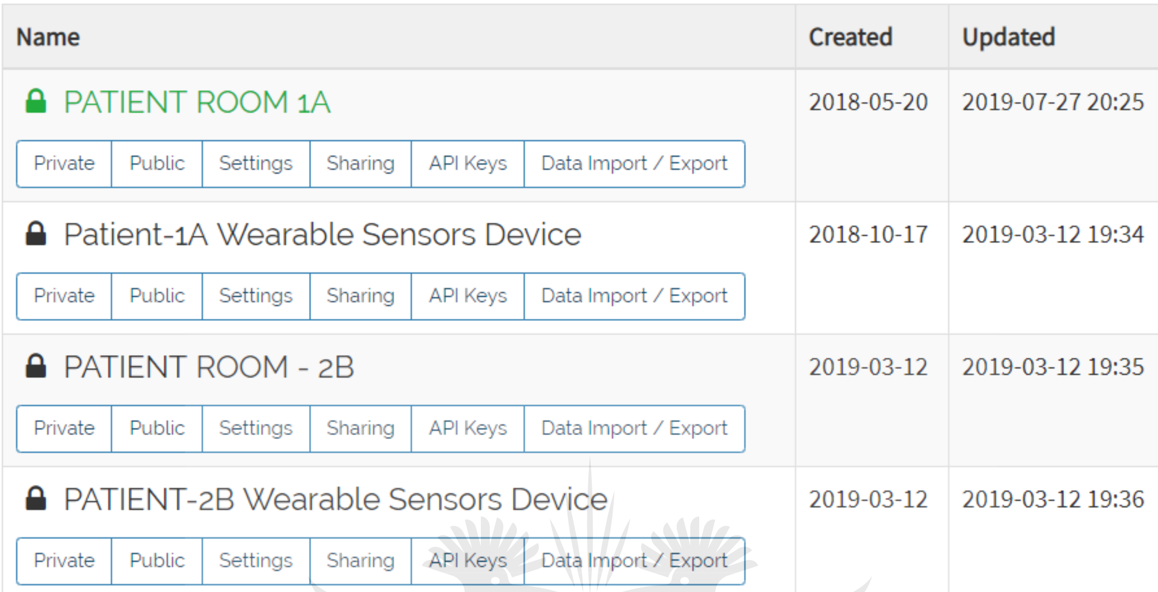


Figure 4.9: Completed patient room sensors node.

A few sensor devices are available in the room of the patient. Sensors are used to monitor patient falls, sleeping conditions, motion inside the room of the patient, and environmental conditions in the room, such as room temperature, humidity and light. Figure 4.10 shows four channels that have been created; inside each channel up to eight fields can be configured.



Name	Created	Updated
PATIENT ROOM 1A Private | Public | Settings | Sharing | API Keys | Data Import / Export	2018-05-20	2019-07-27 20:25
Patient-1A Wearable Sensors Device Private | Public | Settings | Sharing | API Keys | Data Import / Export	2018-10-17	2019-03-12 19:34
PATIENT ROOM - 2B Private | Public | Settings | Sharing | API Keys | Data Import / Export	2019-03-12	2019-03-12 19:35
PATIENT-2B Wearable Sensors Device Private | Public | Settings | Sharing | API Keys | Data Import / Export	2019-03-12	2019-03-12 19:36

Figure 4.10: Thingspeak channels

4.4 Conclusion

The CPS consists of wireless sensor network devices and a wearable sensor device. Both devices were fully implemented and designed successfully. The hardware circuits were constructed, and the embedded software programs were loaded to produce a functional CPS. Evaluation results showed that all electronic components and sensors attached to the Arduino microcontroller boards were working. The star network topology used in the wireless sensor network was tested and showed data had been transferred from the sensor nodes to the coordinator node. The software components of the full design of the CPS were properly employed and evaluated, using MATLAB R2016a, X-CTU for WSN, and C++ Arduino IDE.

CHAPTER 5: ANALYSIS AND EVALUATION

In this chapter, we analyse data captured from the CPS by using a fuzzy logic system algorithm. The data will be divided into three categories: firstly wearable sensors data (pulse rate, body temperature), secondly data from the patient's room (fall detection, body movement, sleep and sitting mode) and finally the patient's room environment data (room temperature and humidity, light intensity and motion detection). The fuzzy logic system method is chosen because it can represent the problem in the classification of mathematical probabilistic with a full range of operatives to associate indeterminate information in a better way and can make real-time decisions.

The GUI is designed for monitoring patients' vital signs and room environment. After the system had been constructed, experiments were done to authenticate the established outline of general robustness, consistency, etc. The assessment of the system was undertaken in an indoor home environment. The details of each analysis components are described in the next subsections. Figure 5.1 shows the structure of a fuzzy logic system with multi-input sensors

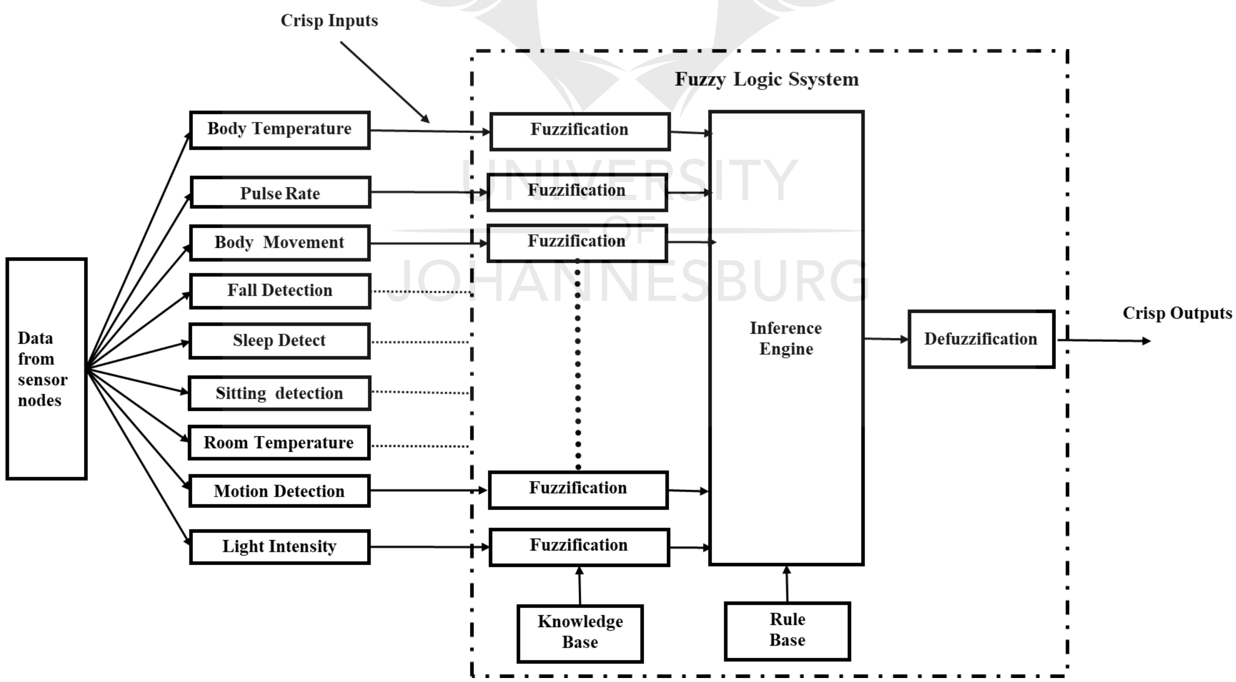


Figure 5.1: Structure of a fuzzy logic system with multi-input sensors

5.1 Analysis of Patient Parameters

The typical stages followed in implementing fuzzy logic systems include the purpose of the input and output variables, the selection of appropriate membership functions, and the construction of the fuzzy rules database. These components are described in the following sections. Figure 5.2 shows the whole patient parameter structure of the fuzzy logic system, which includes wearable sensors device parameters (pulse rate and body temperature) and patient room parameters (fall detection, body movement, sleep mode and sitting mode). The system consists of six inputs, reasoning rules, and output.

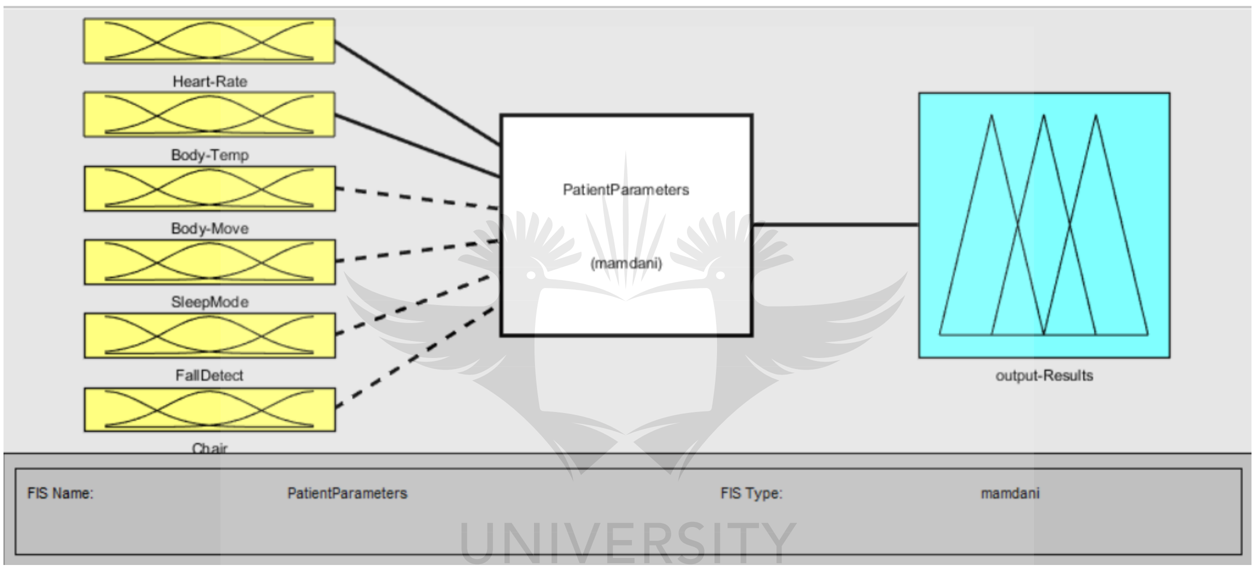


Figure 5.2: Patient fuzzy system structure

5.1.1 Input Variables

Body Temperature: For body temperature, there are three fuzzy sets (low, normal, and high), as defined in Table 5.1. Membership functions of body temperature fuzzy sets are trapezoidal and described in Figure 5.3.

Table 5.1: Body temperature ranges.

Input Field	Range	Fuzzy Sets
Body Temperature	< 36.5	Low
	36 - 38.5	Normal
	>38	High

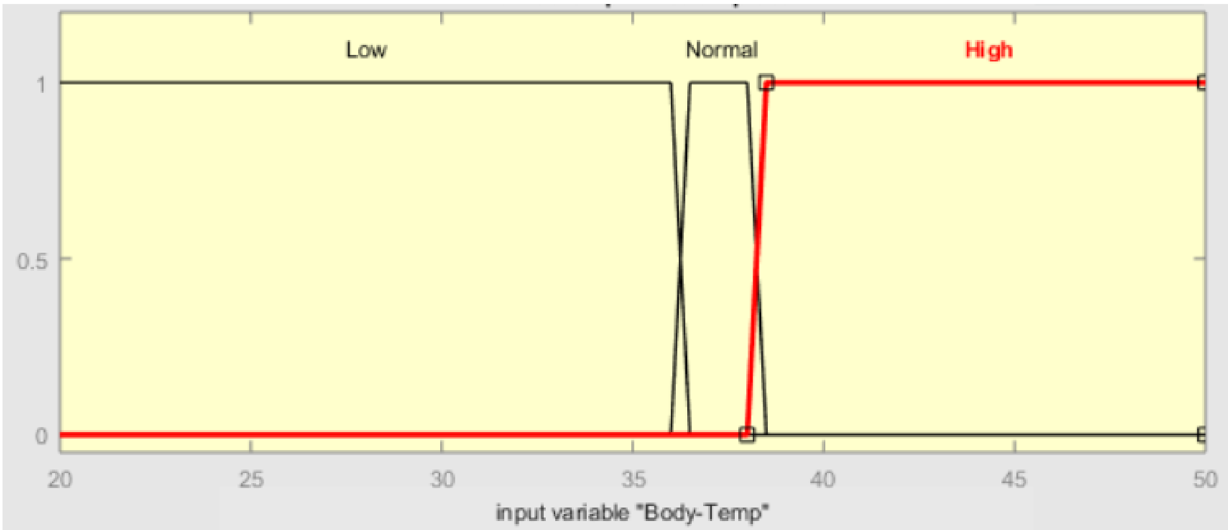


Figure 5.3: Membership functions of the body temperature parameter.

Heart rate: For HR, the researcher used three fuzzy sets (low, normal, and high) as defined in Table 5.2. The membership functions of HR fuzzy sets are trapezoidal and described in Figure 5.4.

Table 5.2: Heart rate ranges that correspond to each fuzzy set

Input Field	Range	Fuzzy Sets
Heart Rate	< 60	Low
	60 - 110	Normal
	>110	High

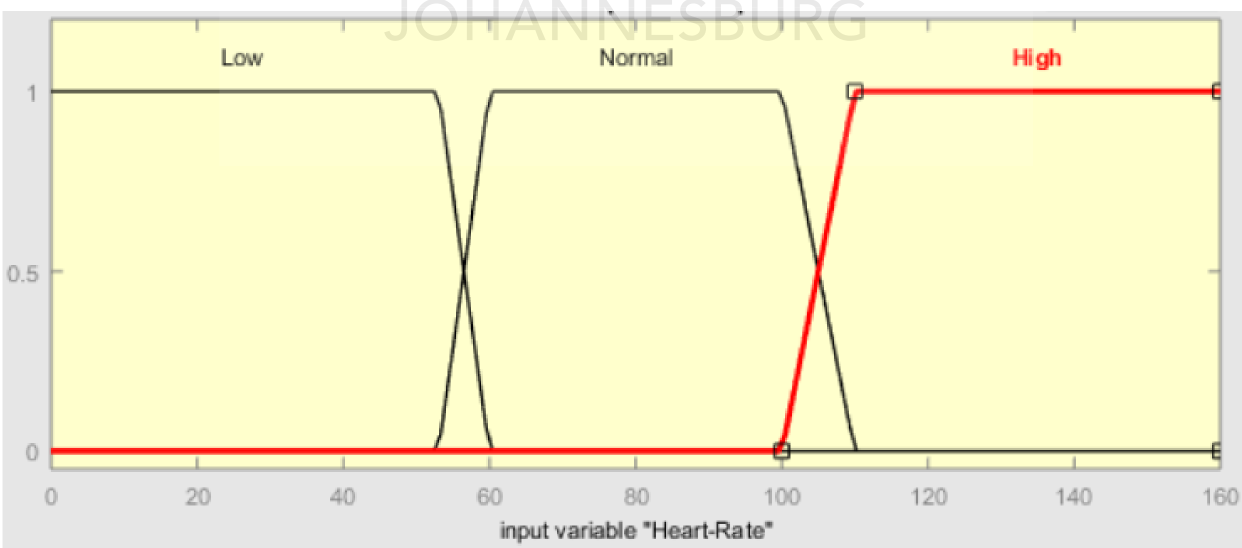


Figure 5.4: Membership functions of the HR parameter

Body Movement: There are four fuzzy sets (low, normal, high, and very high), as defined in Table 5.3. **Low** indicates when a patient is not available in bed, **Normal** defines that the patient is resting, **High** indicates that the patient is restless and **Very High** that the patient is in a critical condition. The membership functions of body temperature fuzzy sets are trapezoidal and described in Figure 5.5.

Table 5.3: Body movement range

Input Field	Range	Fuzzy Sets
Body Movement	0	Low
	1	Normal
	2	High
	3	Very High

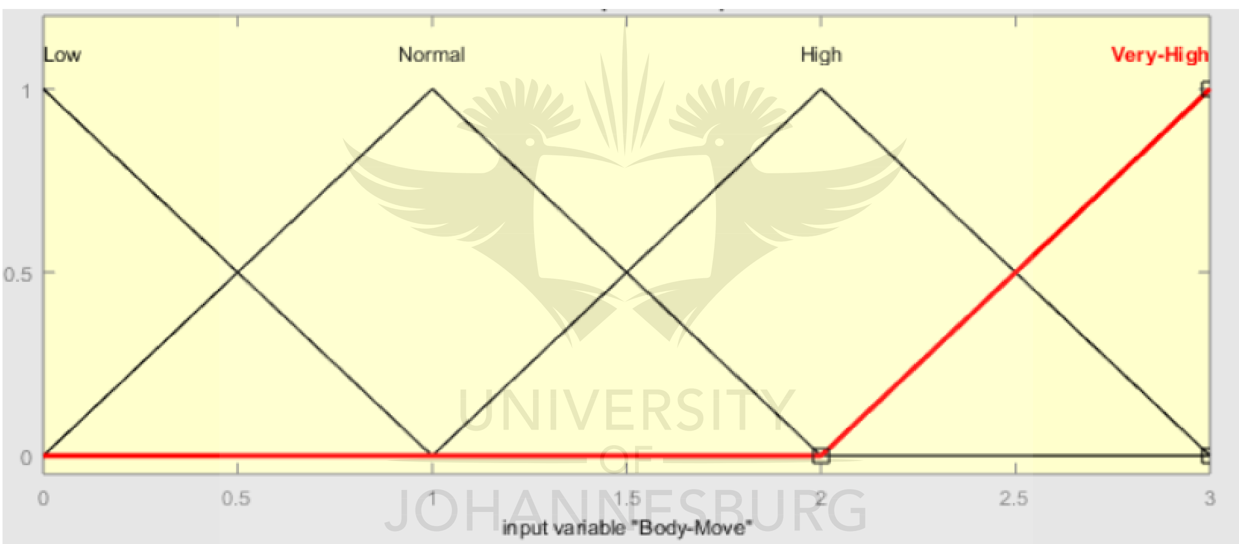


Figure 5.5: Membership functions of body movement

Sleep Mode, Fall Detection, and Sitting Mode: These three inputs have the same fuzzy sets, namely low and high. The sleep mode defines whether the patient is sleeping on the bed, while fall detection determines the critical condition of the patient fallen from the bed. The last is sitting mode, which indicates whether the patient is sitting on a chair. Figure 5.6 shows the membership functions of sleeping mode fuzzy sets.

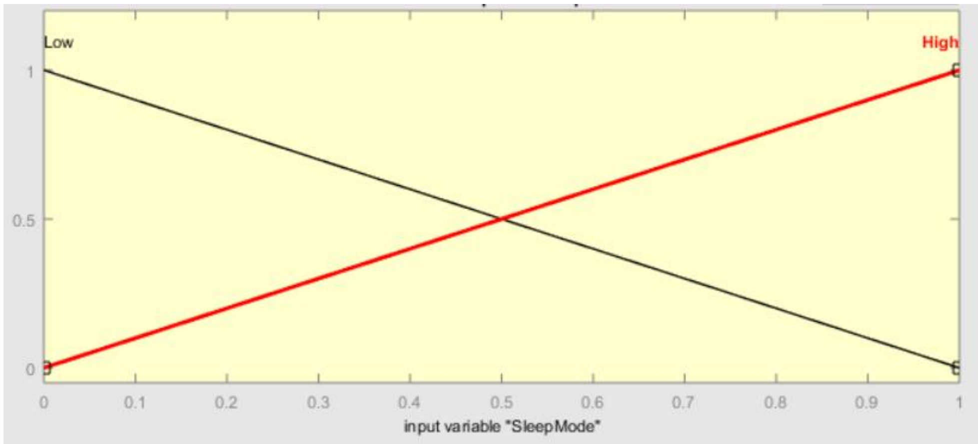


Figure 5.6: Membership functions of the sleep mode

5.1.2 Output Variables

The output of the patient fuzzy logic engine has three output variables ranging from 0 to 2. The three fuzzy sets for the output variable are Normal = 0, Risk = 1, and Critical = 2. These variables state the condition of the patient in three stages. The membership functions for these sets are triangular. The membership functions are shown in Table 5.4 and Figure 24**.**

Table 5.4: The output variable ranges that correspond to each fuzzy set.

Output Field	Range	Fuzzy Sets
Risk Group	0 - 0.5	Normal
	0.5 - 1.5	Risk
	1.5 - 2	Critical

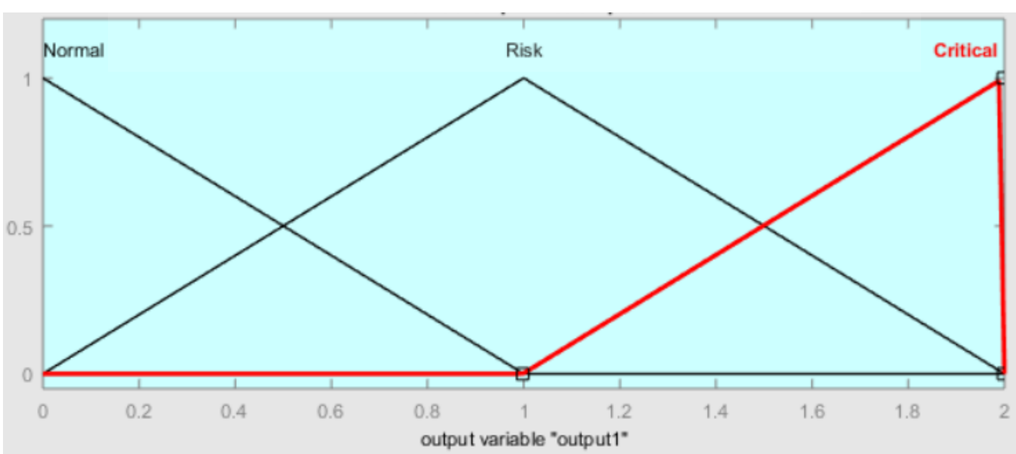


Figure 5.7: Membership functions of the output variable (wearable sensors system).

5.2 Fuzzy Rule Base Inference

When the input variables and membership functions are defined, the next step is to establish the rules, which are the important part of the fuzzy inference system; the excellence of results depends on the fuzzy rules. The rules are characterised by IF (antecedents) and THEN (conclusions) statements and the number of rules relies on the number of membership functions for the input variables. The number of rules is found using the formula in equation 1 below:

$$N = p1 \times p2 \times \ldots\ldots\ldots \times pn \qquad \text{Eq-1}$$

where N is the total number of probable rules for a fuzzy system and pn is the number of linguistic terms for the input linguistic variable N.

The fuzzy rules for patient parameters were designed based on the modified early warning score system by using patient physiological parameters. Since there are six inputs, HR represents three membership functions, body temperature three membership functions, body movement four membership functions, and other inputs have two membership functions. The maximum number of rules is 144 (3x3x4x2x2); although it is not necessary to complete all the rules to get results, the researcher defines all possible 144 rules exactly. Figure 5.6 below shows the designed fuzzy rules of patient parameters. The results and experiments of patient parameters, including defuzzification of the system, will be discussed in the next chapter.

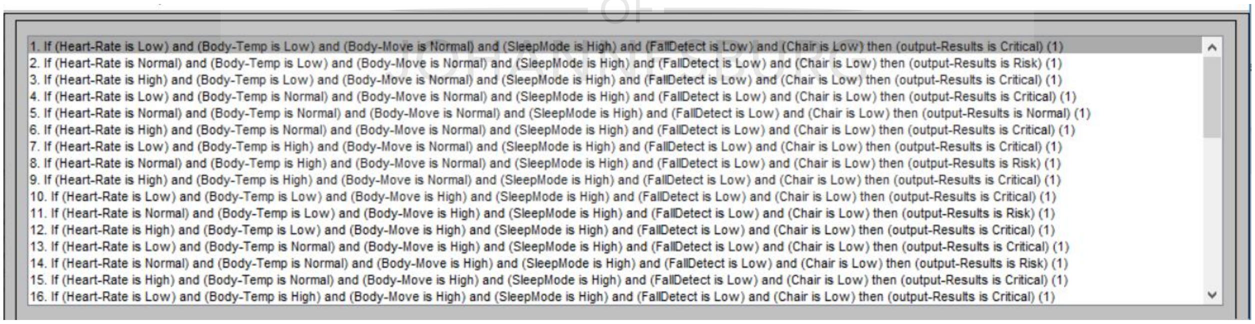


Figure 5.6: Patient fuzzy logic system rules.

5.3 Reaction Display

The Thingspeak platform has the privilege of using the react and ThinkHTTP option. The reactions are sent via e-mail and the social networking site Twitter. React works with Twitter to perform actions when channel or field data meet a certain condition. Tweet sending is relatively easy with the reaction option of Thingspeak, but in order to send an e-mail, a third-party website push inbox can be used. Figure 5.7 shows how to configure reaction messaging easily on Thingspeak that will send an alert to the Twitter application [61].

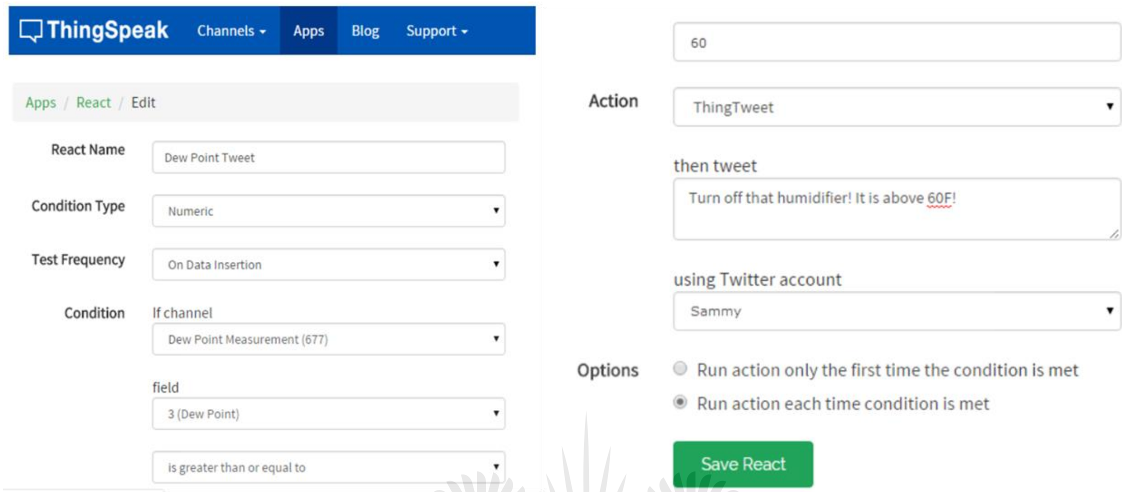


Figure 5.7: Twitter message configuration [61].

5.4 Conclusion

To conclude this chapter, the fuzzy logic system was established and tested successfully. The membership functions, the rules, and the structure were checked several times and the necessary changes were made to finalise the design. The logical creation of each fuzzy rule was inferred to provide a result at an assembled magnitude for each output membership function. The Thingspeak platform was configured correctly to provide sensing results from Arduino boards and to provide reaction messages to Twitter in case of abnormal patient conditions.

CHAPTER 6: OVERALL SYSTEM TESTING

This chapter presents the testing of the CPS prototype that was implemented and developed. The design of software and hardware prototypes was achieved in previous chapters, together with the functionality of microcontrollers and electronic sensors that were used to build the circuit prototype. The tests performed were used to determine the validation of the system and measure the design accuracy. The functionality of tests was intended to ensure all bugs were fixed and the system performed perfectly. Initially, the patient wore a wearable device equipped with sensors. The data from patient vital signs, body temperature, and pulse rate were sent to the internet cloud and stored in a computer using MATLAB GUI programming software. In the second test, to determine the environment of the patient's room and physical activities of the patient, sensors were installed in the bed of the patient to monitor the patient's sleep and detect falls.

6.1 Testing of Wearable Sensor Device

A wearable sensor device consists of a pulse sensor and temperature sensor, both integrated with the Arduino microcontroller unit. Both sensors are powered by 5 V from the Arduino board and ESP8266 WiFi module, also powered by 3.3 V Arduino output voltage. The Arduino micro-development board contains the ATmega328 microcontroller connected to a laptop through a USB cable and powered up by 5 V voltage that is supplied by the laptop.

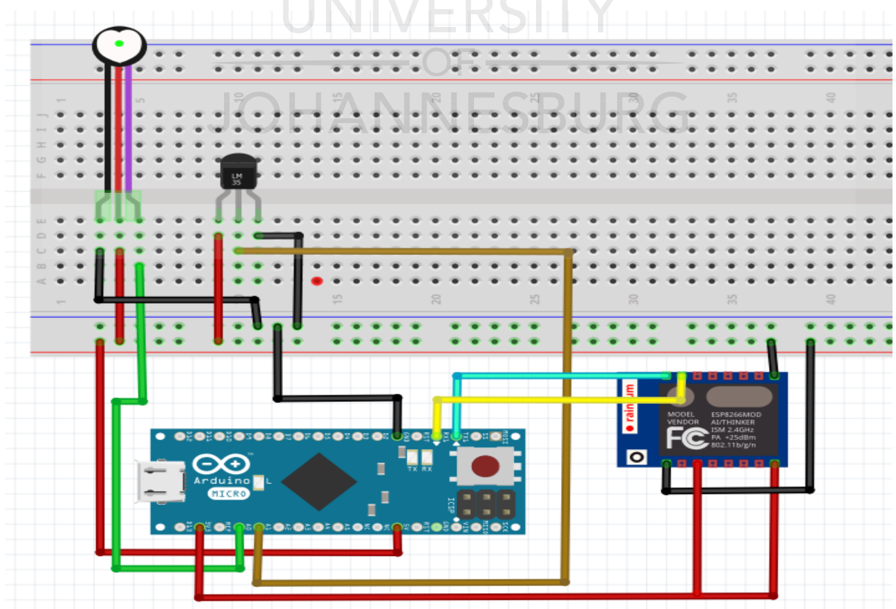


Figure 6.1 : Wearable sensors circuit diagram

After the device has been powered up, Arduino Genuino software is used to program the device. Real-time data of the sensors are captured from the Arduino board and transmitted to the Thingspeak cloud and MATLAB GUI through the WiFi module. For test purposes, before data are transmitted from the microcontroller serial monitor, the real-time data are used to view the data of sensors, as shown in Figure 6.2 below. For the temperature sensor, readings are mostly accurate, as they should be between 35 and 37 degrees Celsius for normal body temperature.

Figure 6.2: Real-time serial monitoring readings

Figure 6.3 shows the tests of patient body temperature and HR captured from the Thingspeak platform displayed in real-time. The pulse rate sensor measures the value based on the pressure of blood flow. The normal range of HR is from 60 BPM to 110 BPM. As shown below, the testing measurement is approximately 92 BPM.

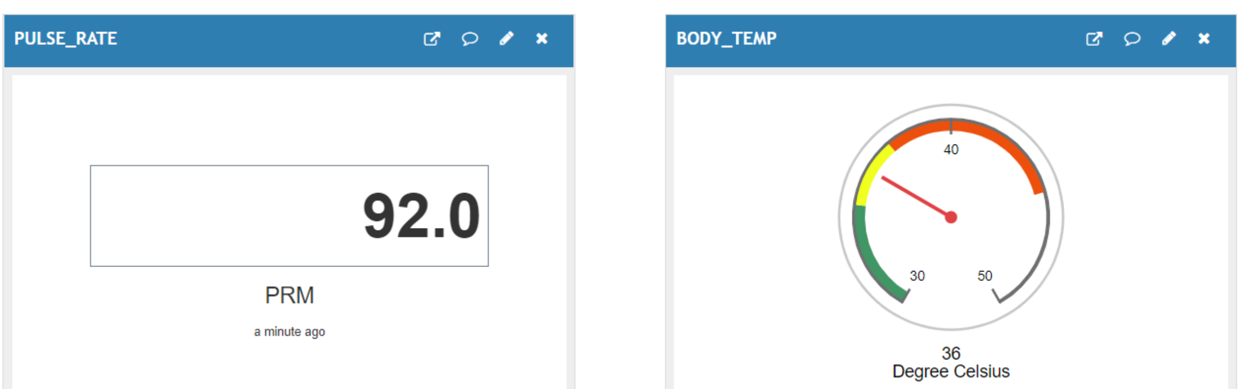


Figure 6.3: Pulse rate and body temperature

6.2 Testing of Sensor Nodes system

In the image below, the setup test for the bed sensors node is shown. A micro-controller is connected with sensors to detect body movement, sleep mode, and falls the patient might have. The overall completed system consists of a room sensor node, wearable sensors, a bed sensor node, and a chair sensor node, as demonstrated in chapter 4. Arduino Uno microcontrollers are used in the three sensor nodes and an Arduino-micro microcontroller is used in the wearable sensor device. The Arduino development board uses C-code programming software, as all sensor nodes and the wearable sensor device are programmed differently, as shown in Appendix A.

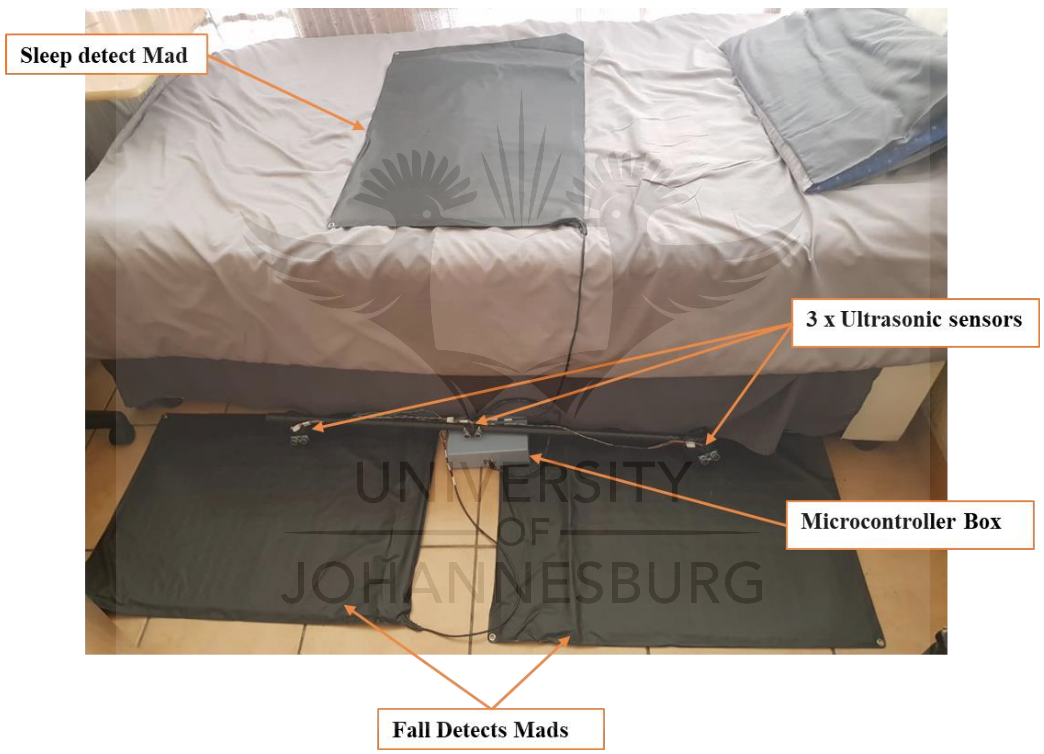


Figure 6.4: Bed sensors node test setup

After all three sensors nodes have been configured using the Arduino program and XBee X-CTU software, the nodes are powered up using a 5 V Arduino power shield installed in each sensor's node, which is supplied by two AAA batteries in each shield module. The bed sensor node and chair sensor node send data to the room sensor node, which acts as coordinator, with the WSN configured as a star topology. All data are received from the coordinator node and sent to the

Thingspeak internet cloud using the ESP8266 WiFi module. Figure 6.5 demonstrates the testing output achieved using the bed sensor node and room sensor node.

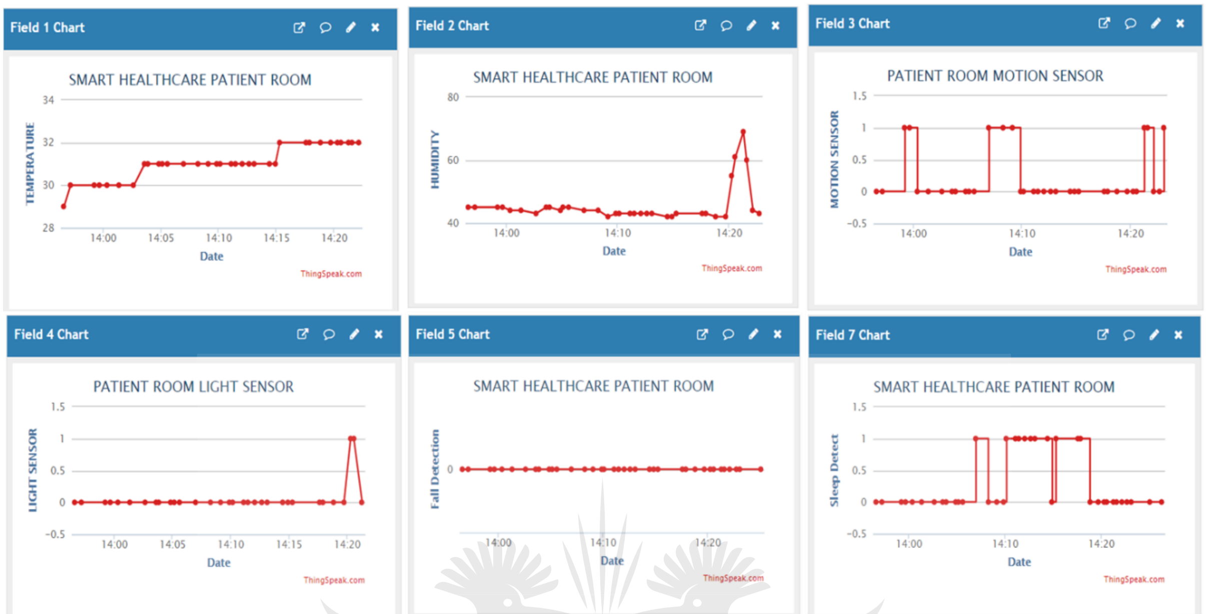


Figure 6.5: Thingspeak sensors real-time output test results

6.3 MATLAB Graphic Interface

After all, data had been captured from the sensors using microcontrollers and transmitted to Thingspeak cloud via XBee and ESP8266 WiFi modules, the MATLAB graphic interface unit was used for data storage, visualisation, and analysis. In this final stage, a monitoring dashboard was created to monitor patients continuously in real-time. At the same time, patients' vital signs and environmental data were stored in a computer server. Finally, a fuzzy logic algorithm was designed to analyse the data of the patients, as shown in the next chapter.

CHAPTER 7: RESULTS AND DISCUSSION

In this chapter, the experimental results are presented and discussed. The results include the analysis of a patient's parameter data by using a fuzzy logic system. Real-time assessments should be conducted to collect enough information from different types of patients, from different age groups, and different room conditions. The proposed CPS is designed to assist doctors and medical staff to record, monitor, and diagnose the problems that may occur during hospitalisation of a patient by analysing and being capable of predicting multilayer physical signs, including falls detection risk.

The researcher will discuss a few scenarios to provide experimental results. First, the patient's sensor information will be captured and sent to the ThingSpeak cloud, when the sensor parameters of the patient are not normal. The ThingSpeak platform will send alert messages via tweet immediately to inform doctors and medical nurses. At the same time, sensor data are stored to be analysed, using the fuzzy logic algorithm, and monitored by MATLAB GUI.

7.1 Case Study

This section will provide experimental results by using the fuzzy logic system for two different patients' conditions. The first is a case study for Patient-1A, as shown in the monitoring dashboard that involves patients with high body temperature, and the second will be fuzzy control in a feedback system involving Patient-2B with an abnormally high HR. The fuzzy logic system consists of six inputs, as explained in chapter 5, but only two experimental scenarios with be explained in this section.

7.1.1 Case 1: (One Abnormal Input – Body Temperature, Five Normal Inputs, and One Output Result)

In this case, the parameters below are captured, generated from the Thingspeak server. Only six patient parameters will be used, namely HR, body temperature, body movement, fall detection, sleep mode, and chair mode. For experimental analysis on the data in Figure 6.1 below, the researcher uses patients' parameters as five normal inputs and one abnormal input. To obtain proper results by using a fuzzy logic system, a few steps will be used to get results.

Step 1: Inputs Fuzzification

The researcher's main goal is to predict findings and diagnose the patient by using six different inputs and to make an appropriate decision, depending on the output. Firstly the researcher receives the input data produced by sensors and adjusts the group to which these belong to each of the suitable fuzzy sets through membership functions. The system developed in this study is built on 144 rules, as explained in the previous chapter. Before the rules can be assessed, the inputs must be fuzzified according to linguistic sets.

For inputs, the researcher considers HR, defined as three levels in the range from 40 BPM to 150 BPM for adults as LOW, NORMAL, and HIGH. Body temperature has three levels ranging from 20°C to 50°C, defined as LOW, NORMAL, and HIGH. Other inputs range from 0 to 1, respectively LOW and HIGH, such as sleep mode; when the patient is sleeping this will be High and when the patient is not on the bed it will be Low. Membership functions have been explained in chapter 5.

1	created_at	Room Temp	Room Humidity	Motion Detect	Light detect	Heart Rate	Body Temp	Body movement	Sleep Mode	Chair Mode	Fall Detect
2	2019-01-15 1	25	45	1	1	65	37	2	1	0	0
3	2019-01-15 2	25	45	1	1	65	37	2	1	0	0
4	2019-01-15 3	25	45	1	1	65	37	2	1	0	0
5	2019-01-15 4	25	45	1	1	65	37	2	1	0	0
6	2019-01-15 5	25	45	1	1	65	37	2	1	0	0
7	2019-01-15 6	25	45	1	1	65	37	2	1	0	0
8	2019-01-15 7	25	50	0	1	65	37	2	1	0	0
9	2019-01-15 8	25	50	0	1	65	39	2	1	0	0
10	2019-01-15 9	25	50	0	1	65	39	2	1	0	0
11	2019-01-15 10	27	50	0	1	70	39	2	1	0	0
12	2019-01-15 11	27	50	0	1	70	39	2	1	0	0
13	2019-01-15 12	27	50	0	1	70	39	2	1	0	0
14	2019-01-15 13	27	50	0	1	70	39	2	1	0	0
15	2019-01-15 14	27	50	0	1	70	39	2	1	0	0
16	2019-01-15 15	27	55	0	1	70	38	2	1	0	0
17	2019-01-15 16	27	55	0	1	70	38	2	1	0	0
18	2019-01-15 17	27	55	0	1	70	38	5	1	0	0
19	2019-01-15 18	27	55	0	1	70	38	5	1	0	0
20	2019-01-15 19	27	55	0	1	70	38	5	1	0	0
21	2019-01-15 20	26	55	0	1	68	38	5	1	0	0
22	2019-01-15 21	26	55	0	1	68	38	5	1	0	0
23	2019-01-15 22	26	55	0	1	68	39	5	1	0	0
24	2019-01-15 23	26	55	0	1	68	39	5	1	0	0
25	2019-01-15 24	26	55	0	1	68	39	8	1	0	0
26	2019-01-15 25	26	55	0	1	68	39	8	1	0	0
27	2019-01-15 26	26	55	0	1	68	39	8	1	0	0
28	2019-01-15 27	26	52	0	1	68	39	8	1	0	0
29	2019-01-15 28	26	52	0	1	68	39	8	1	0	0

Figure 6.1: Patient 1A vital and environmental parameters data

Step 2: Fuzzy Rules

After the inputs have been fuzzified, the rules are defined in each situation. If the antecedent of the agreed rule has more than one part of the rules, the fuzzy operator is applied to obtain only one number of rules, but this applies only to the output function. The input of the fuzzy operator is six membership principles from fuzzified input variables, which must result in one output as a single truth value.

In this system, the maximum number of rules is 144. Although it is not compulsory to satisfy all the rules to get the results, all possible 144 rules are precisely defined. The following first nine rules are based on derived fuzzified inputs. In the rules below, the researcher only focusses on HR inputs and body temperature input by assuming a patient is sleeping.

RULE 1: IF *(HR is Low)* ***AND*** *(Body Temperature is Low)* ***AND*** *(Body Movement is Normal)* ***AND*** *(Sleep Mode is High)* ***THEN*** *(Output Result is* ***Critical****)*
RULE 2: IF *(HR is Normal)* ***AND*** *(Body Temperature is Low)* ***AND*** *(Body Movement is Normal)* ***AND*** *(Sleep Mode is High)* ***THEN*** *(Output Result is* ***Risk****)*
RULE 3: IF *(HR is High)* ***AND*** *(Body Temperature is Low)* ***AND*** *(Body Movement is Normal)* ***AND*** *(Sleep Mode is High)* ***THEN*** *(Output Result is* ***Critical****)*
RULE 4: IF *(HR is Low)* ***AND*** *(Body Temperature is Normal)* ***AND*** *(Body Movement is Normal)* ***AND*** *(Sleep Mode is High)* ***THEN*** *(Output Result is* ***Critical****)*
RULE 5: IF *(HR is Normal)* ***AND*** *(Body Temperature is Normal)* ***AND*** *(Body Movement is Normal)* ***AND*** *(Sleep Mode is High)* ***THEN*** *(Output Result is* ***Normal****)*
RULE 6: IF *(HR is High)* ***AND*** *(Body Temperature is Normal)* ***AND*** *(Body Movement is Normal)* ***AND*** *(Sleep Mode is High)* ***THEN*** *(Output Result is* ***Critical****)*
RULE 7: IF *(HR is Low)* ***AND*** *(Body Temperature is High)* ***AND*** *(Body Movement is Normal)* ***AND*** *(Sleep Mode is High)* ***THEN*** *(Output Result is* ***Critical****)*
RULE 8: IF *(HR is Normal)* ***AND*** *(Body Temperature is High)* ***AND*** *(Body Movement is Normal)* ***AND*** *(Sleep Mode is High)* ***THEN*** *(Output Result is* ***Risk****)*
RULE 9: IF *(HR is High)* ***AND*** *(Body Temperature is High)* ***AND*** *(Body Movement is Normal)* ***AND*** *(Sleep Mode is High)* ***THEN*** *(Output Result is* ***Critical****)*

Step 3: Defuzzification

The decisions are based on testing all the rules in the fuzzy inference system. These rules must be integrated with others to make a decision. This process is called aggregation, whereby fuzzy sets that demonstrate the outputs of each rule are combined into one single fuzzy set, before being passed on to the defuzzification process for crisp output generation. Several inference methods exist. In our system, the root sum square (RSS) inference engine is developed using the formula presented below in Equation 2:

$$\text{RSS} = \sqrt{\sum R^2} = \sqrt{R_1^2 + R_2^2 + R_3^2 +, \ldots, + R_n^2} \qquad [61] \ldots \text{Eq-2}$$

where $R_1^2, R_2^2, R_3^2, \ldots, R_n^2$ are quality values of types of rules that share the same decision.

Finally, the defuzzification of the information into a crisp output is achieved using the fuzzy centroid algorithm. Given the data in Figure 6.1, the output result is generated using input data. According to the data the HR is normal, but the body temperature is high. This can be checked by fuzzy rules as well, as **Rule 8 has** been invoked.

RULE 8: IF *(HR is Normal)* ***AND*** *(Body Temperature is High)* ***AND*** *(Body Movement is Normal)* ***AND*** *(Sleep Mode is High)* ***THEN*** *(Output Result is* ***Risk****)*

Figure 6.2 shows the output results from the input data provided. The results can be identified in a few steps of fuzzy rules. It can be seen that the output of a few steps resulted in Rule 8 (HR = 80, Body Temp = 39 and output result = 1). These outcomes satisfied design rule 8.

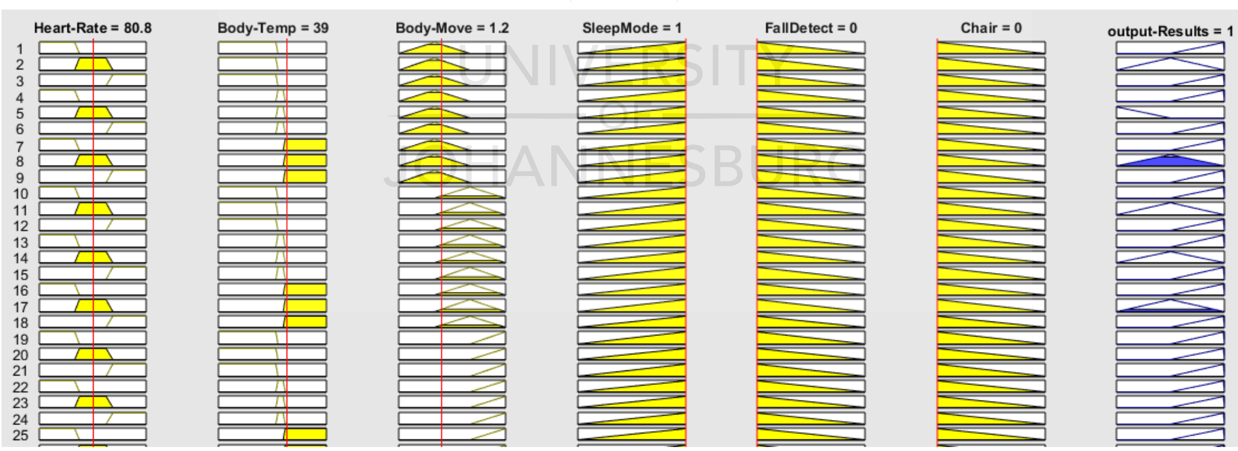


Figure 6.2: Output results (Risk) when rule 8 was invoked with abnormal body temperature

The relationship between inputs and output can be found in Figure 6.3, which shows that from 0°C to 36°C there is an output of 1, which defines the condition of the patient at risk and from 38°C to 50°C, also warning of the patient being in a risky condition. The types of diagnoses of patients are provided in monitoring dashboard information.

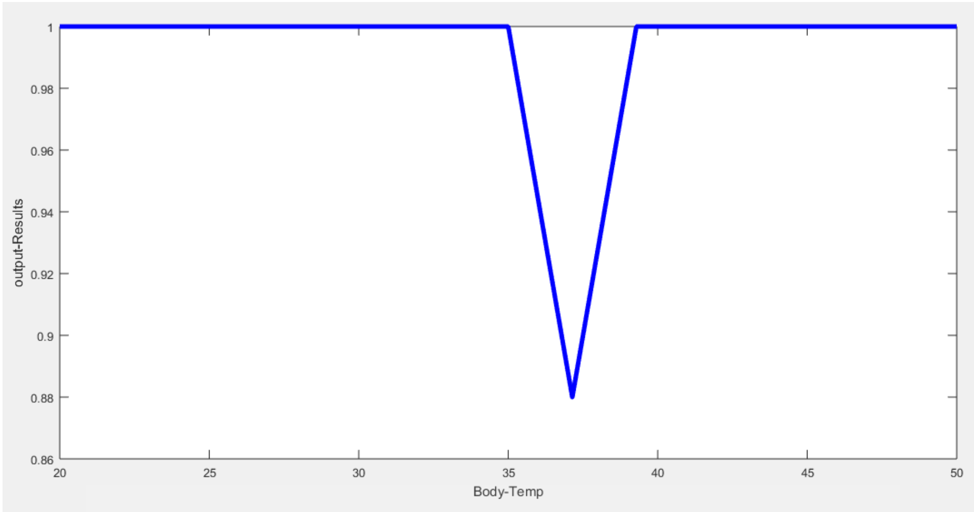


Figure 6.3: Relation between body temperature and output

7.1.2 Case 2: (One Abnormal Input – Heart Rate, Five Normal inputs, and One Output Result)

In this subsection, there are also six fuzzy inputs, as in case1, but instead of abnormal body temperature of the patient, the HR of an adult patient is too high, as shown in the patient data in Figure 6.4. The first step is to consider the inputs and regulate the grade of membership functions to which these belong in each fuzzy set.

created_at	Room Temp	Room Humidity	Motion Detect	Light detect	Heart Rate	Body Temp	Body movement	Sleep Mode	Chair Mode	Fall Detect
2019-01-18 1	23	59	0	1	116	36	5	1	0	0
2019-01-18 2	23	59	0	1	116	36	5	1	0	0
2019-01-18 3	23	59	0	1	116	36	5	1	0	0
2019-01-18 4	23	59	0	1	116	36	5	1	0	0
2019-01-18 5	23	59	0	1	116	36	5	1	0	0
2019-01-18 6	23	59	0	1	116	36	5	1	0	0
2019-01-18 7	23	59	0	1	116	36	5	1	0	0
2019-01-18 8	23	50	0	1	116	36	5	1	0	0
2019-01-18 9	23	50	0	1	116	36	0	0	0	1
2019-01-18 10	23	50	0	1	116	36	0	0	0	1
2019-01-18 11	23	50	0	1	116	36	0	0	0	1
2019-01-18 12	23	50	0	1	116	36	0	0	0	1
2019-01-18 13	23	50	0	1	116	36	0	0	0	1
2019-01-18 14	23	50	0	1	116	36	0	0	0	1
2019-01-18 15	23	55	1	1	116	36	0	0	0	1
2019-01-18 16	23	55	1	1	116	36	0	0	0	1
2019-01-18 17	23	55	1	1	116	36	0	0	0	1
2019-01-18 18	23	55	1	1	116	36	0	0	0	1
2019-01-18 19	23	55	1	1	116	36	0	0	0	1
2019-01-18 20	23	55	1	1	116	36	0	0	0	1
2019-01-18 21	23	55	1	1	116	36	0	0	0	1
2019-01-18 22	23	55	1	1	116	36	0	0	0	1
2019-01-18 23	23	55	1	1	116	36	0	0	0	1
2019-01-18 24	24	55	1	1	116	36	0	0	0	1
2019-01-18 25	24	55	1	1	116	36	0	0	0	1
2019-01-18 26	24	55	1	1	116	36	0	0	0	1
2019-01-18 27	24	52	1	1	116	36	0	0	0	1
2019-01-18 28	24	52	1	1	116	36	0	0	0	1

Figure 6.4: Patient 2B vital and environmental parameters data

Before the rules can be assessed, the inputs must be fuzzified according to fuzzy linguistic sets. As mentioned in the previous case, the system is built on 144 rules and each rule depends on determining the inputs into a few different types of fuzzy linguistic sets.

By checking the rules, in this case, rule 6 is being invoked according to input fuzzification and applying fuzzy rules, as explained in the previous subsection.

RULE 6: IF *(HR is High)* ***AND*** *(Body Temperature is Normal)* ***AND*** *(Body Movement is Normal)* ***AND*** *(Sleep Mode is High)* ***THEN*** *(Output Result is* ***Critical****)*

After applying a few steps, the same in case 1, and defuzzification, the outcomes result satisfied fuzzy rule 6, where HR = 116, body temperature = 36 and output result = 2 or greater than 1.5).

Figure 6.5: Output results (critical) when rule 6 was invoked with abnormal HR

The relationship between HR inputs and output in Figure 6.6 shows that an HR less than 60 BPM and more than 110 BPM is defined as the patient is in a critical condition, which requires urgent attention from doctors and nurses.

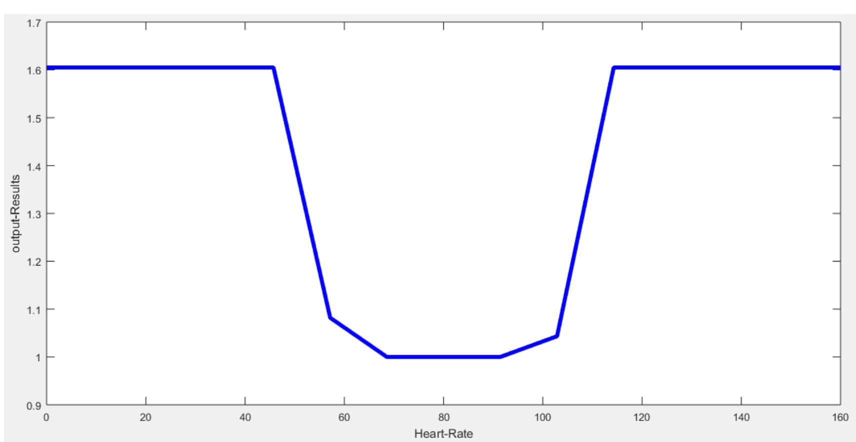


Figure 6.6: Relation between HR and output

The output result had to be any of the output membership functions from rule 1 to rule 144. Figure 6.7 shows the surface view of the Z-output of HR input in the y-axis and body temperature in the x-axis. The Z-output can be in any of six inputs as defined by patient data parameters.

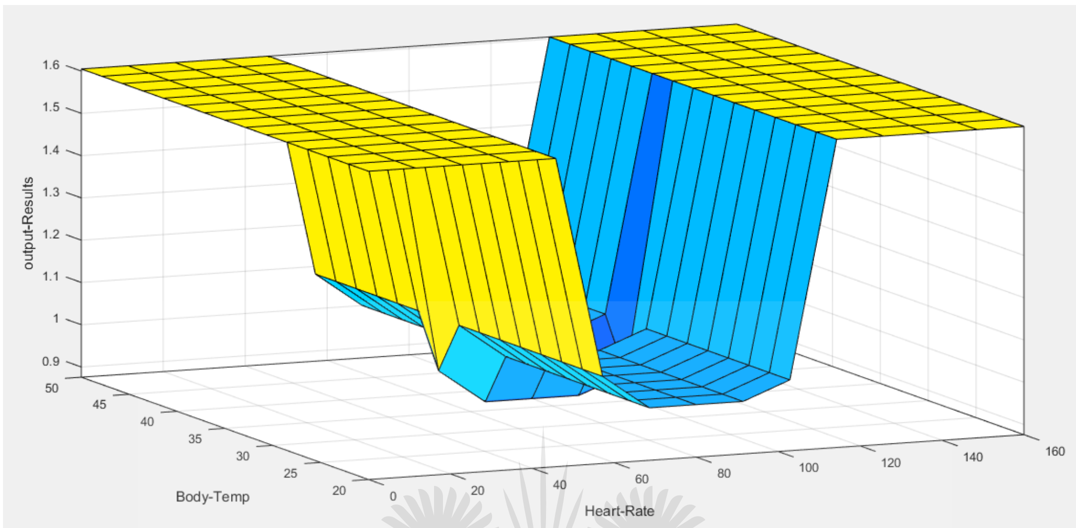


Figure 6.7: HR vs body temperature vs output

The graph in Figure 6.7 shows that the output is normal on body temperature, around 37°C, and HR from 60 BPM to 110 BPM.

7.2 Smart Healthcare Monitoring Dashboard

Thingspeak collects the data values from different sensors used in the system. These data are transferred to the MATLAB system. In the system described in this document, two applications in MATLAB are used. The first is a fuzzy logic system that analyses and predicts patients' vital sign parameters and GUI used for monitoring and diagnosis of patients. The monitoring system in real-time is established for several patient parameters such as HR, body temperature, and fall detection; these include patient room environmental parameters such as humidity, room temperature, and light.

Figure 6.8 below is a smart healthcare monitoring dashboard designed using MATLAB GUI. The monitoring dashboard shows the output results of five real-time vital sign parameters for patients and output results of five environmental conditions of rooms of patients.

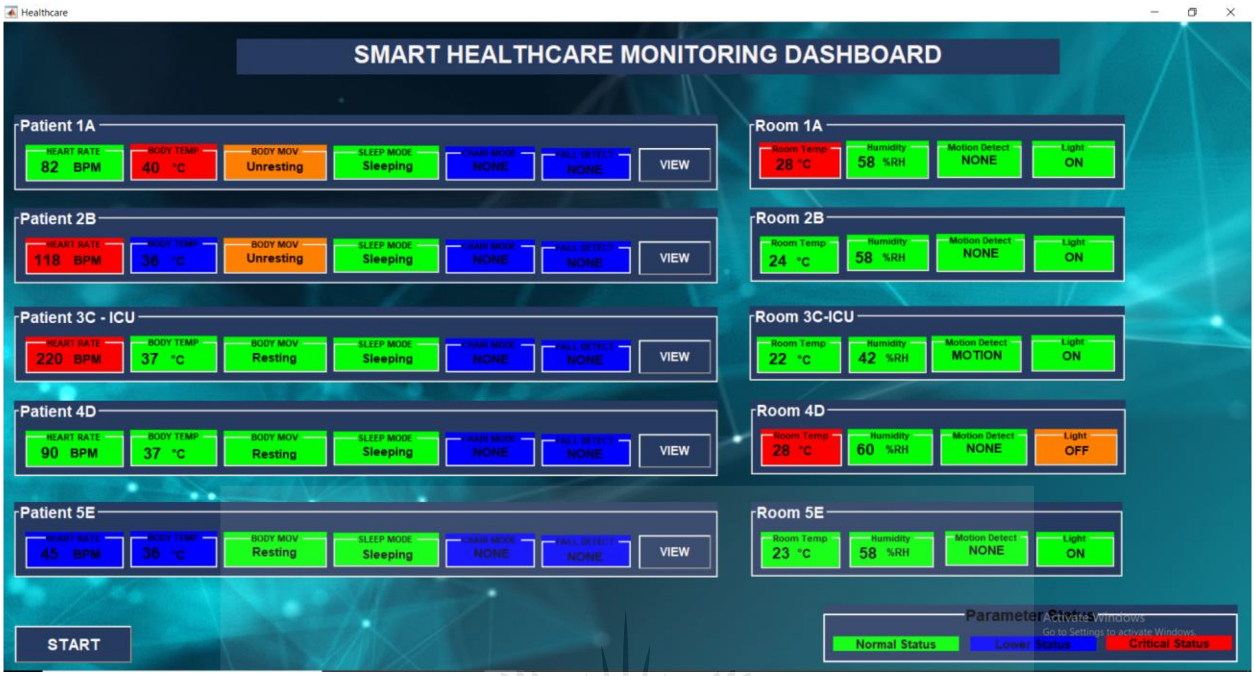


Figure 6.8: Smart healthcare monitoring dashboard

As shown in Figure 6.8, medical staff can easily monitor the conditions of each patient. By viewing the information on Patient 1A, medical staff and doctors will be able to see the real-time condition of the patient and notice that the patient is restless and the body temperature is high. On the other hand, medical staff can inform the maintenance team to check the air conditioning as the monitoring system indicates that the temperature in Room 1A is high.

The GUI includes other displays that allow operators to view patient information and diagnosis in the system. By clicking on the view icon for each patient, doctors and nurses can see patient information and information on vital signs, including the patient's diagnosis, as shown in Figure 6.9.

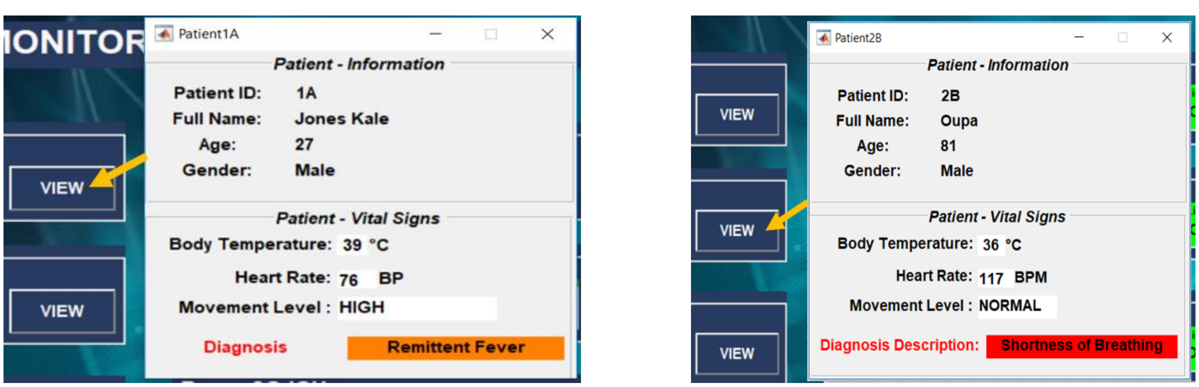


Figure 6.9: Patients details and diagnosis

7.3 Alarming Messages

When any sensor data reach the threshold, the Thingspeak application will send a report through a twitter message to inform doctors about patients' conditions. These alert messages will be the same as those displayed on the monitoring system. Tweet messages enable doctors or nurses who are not around the hospital or close to the monitoring system to react quickly to help the patients. Figure 6.10 demonstrates tweet messages sent to inform the doctor about the warning and patients' critical condition.

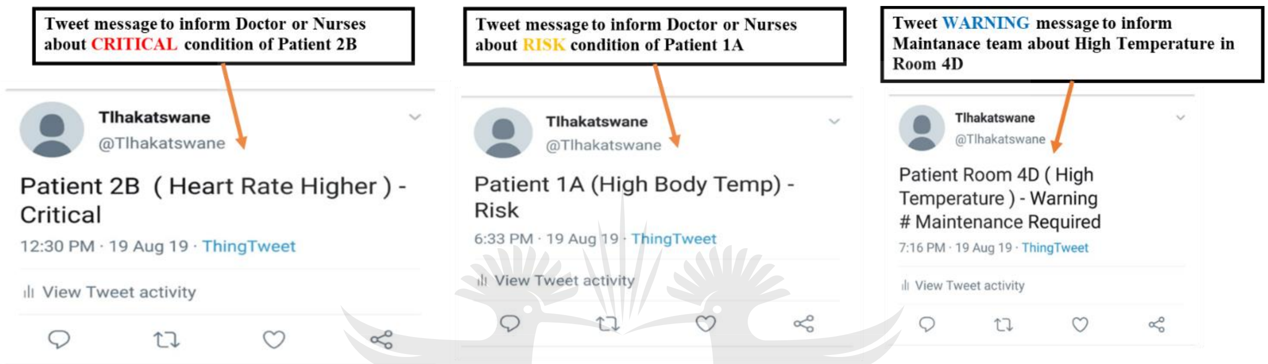


Figure 6.10: Tweet messages to inform medical staff and doctors

7.4 Conclusion

In this chapter, the researcher briefly explained the results of the experiments conducted. The resulting graphs and figures of experiments were analysed by using accompanying keys and an explanation of each graph was given. The fuzzy logic rules were designed for the fuzzy logic system and the MATLAB program was used to design a healthcare monitoring system and for the design rules as well as member functions of the inputs and outputs. Analysis of the invoked fuzzy logic rules was done in MATLAB and the results correspond to the design requirements. Alert messages were generated using the Thingspeak platform to inform medical staff and doctors about patients' conditions.

CHAPTER 8: CONCLUSION

In this dissertation, the researcher presented a CPS that is used in smart healthcare to provide a monitoring system for patients, such as patient body temperature, fall detection, and HR. The room environment, room temperature, humidity, and light density in the hospital are included in the monitoring. The system was not designed to replace medical staff, but to reduce effort and time overhead, as well as human-associated errors. Prototype devices were developed using a wearable sensor device and room inversive device entailing a WSN. This CPS monitors the patient's physical signs as well as relevant environmental conditions continuously in real-time. It furthermore provides data analysis by using a fuzzy logic algorithm that helps to classify abnormal and normal sensor data.

Fuzzy logic was chosen as an appropriate computational method owing to its simplicity and ease of tuning to monitor changing vital signs values. Then the technique was employed in software, providing a multistage process for classifying the condition of patients, using fuzzy functions for each of several experimental vital signs, and then integrating these, using rules to predict the overall patient health status. The ThingSpeak reaction platform sends alert messages to doctors and medical staff if an abnormal vital sign is detected.

Against this background, the research exposes the fact that owing to high numbers of patients in healthcare with deteriorating conditions, constant monitoring of patients in real-time is important. Experimental testing was carried out to validate that the system is acceptable and easy to use. It is estimated that if fully adopted and supported, the system will help deliver smart healthcare. The researcher has contributed a possible key to growth in fields of computing, namely the IoT and real-time data capture and analysis through fuzzy logic by integrating these into the field of a CPS based on healthcare. The prototype developed provides an inclusive resolution that is affordable, efficient, and provides real-time diagnosis and monitoring of patients in healthcare.

8.1 Original Contributions

The contribution of this dissertation is a few characteristics in terms of computing architecture, applications, and an algorithm, in the situation of various CPSs. More importantly, the original contributions of this dissertation are summarised as follows:

- In association with the Department of Health, this system could be implemented in both public and private hospitals for monitoring patients' and patients’ rooms.
- The fuzzy logic algorithm has been shown to yield a realistic technique for use in dealing with the general problem of categorizing the health status of patients when irregularities occur in vital signs and other conditions. It is estimated that a more extensive system could be built by including further parameters and more complex rules, using the same essential algorithm.
- The system should provide a large amount of sensor data that can be explored for further improvement in this field. The data can be used in research in academia, with recommended collaboration with universities for further development and research.

8.2 Limitations of the Study

There are few challenges hindering the progress of the full development of CPS in Smart Healthcare, this involves software reliability and computational intelligence and privacy. The limitations in this research include the capturing complete set of actual data from patients in hospitals was difficult as such information considered to be confidential and should not be allowed from unauthorized persons. Secondly, the system requires an uninterrupted internet connection and power supply to function and capturing of patient data. IoT is a new generation of the internet but still depends mostly on the internet, where they are still challenges of speed and latency to capture data from IoT devices.

8.3 Future Work

In future work, new guidelines might be included to improve CPS in smart healthcare. This research is the first step in smart healthcare. It can be predicted that smart healthcare information will be accessible on the cloud, where different types of monitoring from various universal devices would be integrated through CPSs. Data analytics and machine learning techniques should be mostly considered to be used as part of CPSs to collect data for dedicated care, truthful results, and individualised treatment. CPSs in healthcare entries can provide more wide-ranging consultation and support in healthcare matters than pure information environments. The introduction of 5G network, in future will make IoT and CPS devices to be more fast on speed and capturing of real-time data. CPS in the future should provide better and more extensive care in healthcare emergencies.

REFERENCES

[1] Wang J., Abid, H., Lee S., Shu, L. and Xia, F. "A secured health care application architecture for cyber-physical systems," *Control Engineering and Applied Informatics*, vol. 13, no. 3, pp.101–108, 2011.

[2] Paredes, A.C., Malfaz, M. and Salichs, M.A. "Signage system for the navigation of autonomous robots in indoor environments." *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 680-688, 2014.

[3] Lee, E.A. "Cyber-physical systems: Design challenges." Technical report, EECS Department, University of California, Berkeley, January (CPS Steering Group), 2008.

[4] Zanella A., Bui N., Castellani A., Vangelista L., and Zorzi M. "Internet of things for smart cities." *IEEE Internet Things J*, 1 (1):22-32, 2014.

[5] Vaishnavi, V., Kuechler, W., and Petter, S. (Eds.). "*Design Science Research in Information Systems*" December 20, 2017. URL: http://www.desrist.org/design-research-in-information-systems/, 2004/17.

[6] Tsichritzis, D. "The dynamics of innovation." In P.J. Denning and R.M. Metcalfe, (Eds.), *Beyond Calculation: The Next Fifty Years of Computing* (pp. 259-265). New York: Copernicus Books,1998.

[7] Networking and Information Technology Research and Development Program (2009), High-Confidence Medical Devices: Cyber-Physical Systems for 21st Century Health Care.

[8] Alani A. and Rao K.P., "IoT based smart healthcare system," vol. 05, no. 01, pp. 1214–1221, 2018.

[9] J. Illegems, "The internet of things in health," Fac. Econ. en Bedrufskd., p. 71, 2017.

[10] Jha, S.K. "Medical cyber physical system," vol. 4, no. 5, pp. 819–823, 2014.

[11] Lee, E.A. The past, present and future of cyber-physical systems: "A focus on models." *Sensors (Switzerland)*, vol.15, no. 3, pp. 4837–4869. doi: 10.3390/s150304837, 2015.

[12] Abdulmalik Humayed, Jingqiang Lin, Fengjun Li, and Bo Luo. "Cyber-physical systems security - A survey." *IEEE Internet of Things Journal*, 4(6):1802-1831, 2017.

[13] Yin Zhang, Meikang Qiu, Chun-Wei Tsai, Mohammad Mehedi Hassan, and Atif Alamri, "Health-CPS: Healthcare cyber-physical system assisted by cloud and big data." *IEEE Systems Journal*, vol., no. 1, March 2017.

[14] Wu, F.-J., Kao, Y.-F. and Tseng, Y.-C. “From wireless sensor networks towards cyber physical systems,” *Pervasive and Mobile Computing*, vol. 7, no. 4, pp. 397–413, 2011.

[15] Lee, I., Sokolsky, O. and Chen S. et al., “Challenges and research directions in medical cyber- physical systems,” *Proceedings of the IEEE,* vol. 100, no. 1, pp. 75–90, 2012.

[16] Wang, J., Abid, H., Lee, S., Shu, L. and Xia, F. “A secured health care application architecture for cyber-physical systems,” *Control Engineering and Applied Informatics,* vol. 13, no. 3, pp.101–108, 2011.

[17] Frazzon, E.M., Hartmann, J., Makuschewitz, T. and Scholz-Reiter, B. “Towards socio-cyber- physical systems in production networks.” *Procedia CIRP*, vol. 7, pp. 49–54, 2013.

[18] Rajkumar, R. “A cyber-physical future.” Proceedings of the IEEE, 100:1309-1312, 2012.

[19] Sonntag, D., Zillner, S., Chakraborty, S., Lorincz, A., Strommer E. and Sera¯ni L. “The medical cyber-physical systems activity at EIT: A look under the hood.” In 2014 IEEE 27th Int. Symp. Computer-Based Medical Systems (CBMS), pp. 351–356, 2014.

[20] Chen, J.-J. and Chakraborty, S. “Resource augmentation for uniprocessor and multiprocessor partitioned scheduling of sporadic real-time tasks.” *Real-Time Systems*, vol. 49, no. 4, pp. 475–516, 2013.

[21] Tercero, C., Kodama, H., Shi, C., Ooe, K., Ikeda, S., Fukuda, T., Arai, F., Negoro, M., Kwon G. and Najdovski, Z. “Technical skills measurement based on a cyber-physical system for endovascular surgery simulation.” *International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 9, no. 3, pp. e25–e33. doi: 10.1002/rcs.1467, 2013.

[22] Vouyioukas, D. and Karagiannis, A., 2011. “Pervasive homecare monitoring.” In: P.G. Graschew (ed.). *Telemedicine Techniques and Applications.* Shanghai: InTech, pp. 45-70.

[23] Li, S., Xu, L., and Wang, X. (2013). “Compressed sensing signal and data acquisition in wireless sensor networks and internet of things.*” IEEE Transactions on Industrial Informatics*, in press DOI: 10.1109/TII.2012.2189222.

[24] Lo Bello L. and Toscano, E. “Coexistence issues of multiple co-located IEEE 802.15.4 ZigBee networks running on adjacent radio channels in industrial environments.” *IEEE Transactions on Industrial Informatics*, vol. 5, no. 2, pp. 157 -167, May 2009.

[25] Elankavi, R. Kalaiprasath R. and Udayakumar, R. "Wireless ZigBee network cluster-capacity calculation and secure data conveyance using Indegree," *International Journal on Smart Sensing and Intelligent Systems Special Issue*, vol. 1, pp. 175-185, 2017.

[26] Bidai, Z. and Maimour, M. "Interference-aware multipath routing protocol for video transmission over ZigBee wireless sensor networks," in *International Conference on Multimedia Computing and Systems (ICMCS)*, 2014.

[27] Pathak, S., Kumar, M., Mohan, A., and Kumar, B. “Energy optimization of ZigBee based WBAN for patient monitoring energy optimization of ZigBee based WBAN for patient monitoring,” *Procedia - Procedia Comput. Sci.*, vol. 70, no. January, pp. 414–420, 2016.

[28] Paksuniemi M., Sorvoja H., Alasaarela E. and Myllyla R. “Wireless sensor and data transmission needs and technologies for patient monitoring in operating room and intensive care unit.” In Proceedings of the 27th IEEE Annual International Conference of Engineering in Medicine and Biology Society, pp. 5182-5185, Shanghai, China, 2005.

[29] ZigBee Alliance [Online]. Available: http://www. zigbee.org/

[30] Bluetooth, 2015. Available online: http://www.bluetooth.com/Pages/Bluetooth-Home.aspx (accessed on 1 May 2016).

[31] Nemati, E., Deen, M., and Mondal, T. “A wireless wearable ECG sensor for long-term applications.” *IEEE Commun. Mag*., vol 50, pp. 36–43, 2012.

[32] Garbarino, M., Lai, M., Tognetti, S., Picard, R., and Bender, D. “Empatica E3—A wearable wireless multi-sensor device for real-time computerized biofeedback and data acquisition.” In Proceedings of the 4th International Conference on Wireless Mobile Communication and Healthcare—Transforming Healthcare through Innovations in Mobile and Wireless Technologies, Athens, Greece, 3–5 November 2014; pp. 39–42.

[33] Jeon, B., Lee, J. and Choi, J. “Design and implementation of a wearable ECG system,” vol. 7, no. 2, pp. 61–70, 2013.

[34] Guo, R., Li, S., He, L., Gao, W., Qi, H., and Owens, G. “Pervasive and unobtrusive emotion sensing for human mental health. In Proceedings of the 7th International Conference on Pervasive Computing Technologies for Healthcare, Venice, Italy, 5–8 May 2013; pp. 436– 439.

[35] Akpojaro, J. Bello, R. and State, B. “Smart healthcare system : A primer,” pp. 143–149, 2019.

[36] Jeong, J.S., Han, O. and You, Y.Y. “A design characteristics of smart healthcare system as the IoT application,” *Indian J. Sci. Technol.*, vol. 9, no. 37, 2016.

[37] Hanada, E. “Managing the environment for IoT systems - Toward a ‘smart hospital,’” *Int. J. Comput. Softw. Eng.*, vol. 3, no. 2, 2019.

[38] Kumar, R. and Pallikonda Rajasekaran, M. “An IoT based patient monitoring system using raspberry Pi.” In Proceedings of the 2016 International Conference on Computing Technologies and Intelligent Data Engineering (ICCTIDE'16)

[39] Siva1, S., Suresh, P., Seeba Merlin S. and Punidha, R. “A smart heart rate sensing system in the internet of things.” *IJCTA*, vol. 9, no.9, pp. 3659-3663, 2016.

[40] Hu, Y.-C., Patel, M., Sabella, D., Sprecher, N. and Young, V. “Mobile edge computing a key technology towards 5G,” *ETSI White Paper*, vol. 11, no. 11, pp. 1–16, 2015.

[41] Satyanarayanan, M., Bahl, P., Caceres, R. and Davies, N. "The case for VM-based cloudlets in mobile computing," *IEEE Pervasive Comput*., vol. 8, no. 4, pp. 14–23, Oct./Dec. 2009.

[42] Mao, Y., You, C., Zhang, J., Huang, K., and Letaief, K.B. "A survey on mobile edge computing: The communication perspective." *IEEE Commun. Surv. Tutor*. **2017**, 19, 2322–2358.

[43] Zadeh LA., "Fuzzy Sets*." Information and Control*, vol. 8, no. 3, pp. 338-353, 1965.

[44] Biswas A. and Modak, N. "A fuzzy goal programming approach for fuzzy multiobjective stochastic programming through expectation model, pp.124 – 135, Springer Verlag, (ISSN: 1865-0929, eISSN: 1865-0937), 2012.

[45] Sugeno, M.T. Takagi, "Fuzzy identification of systems and its application to modelling and control," *IEEE Trans Syst Man Cybern*, vol. 15: pp. 116–132.

[46] Mayilvaganan M. and Rajeswari, K. "Risk factor analysis to patient based on fuzzy logic control system," *Blood Press*., vol. 60, no. 4, p. 40, 2014.

[47] Maleki A. and Geramipour, A. "Continuous control of blood glucose in TIDM using fuzzy logic controller in insulin pump: A simulation study," In Proceedings of the 2011 2nd Int. Conf. Control. Instrum. Autom. ICCIA 2011, pp. 122–127, 2012.

[48] Al Masud, & Rashid, S. M. (2012). "A novel approach to introduce cloud services in healthcare sectors for the medically underserved populations in South Asia." *International Journal of Engineering Research and Applications*, 2(3), pp. 1337-1346.

[49] https://www.google.com/search?q=arduino+uno&client=firefoxbab&source=lnms&tbm=isch&sa=X&ved=0ahUKEwip2qiR5aHTAhUJQo8KHX9AJAQ_AUICCgB#imgrc=YWmB1xWbeYqNrM:

[50] "Arduino UNO". Arduino.org.N.p., 2017. Web. 24 Mar. 2017.

[51] https://www.google.co.za/search?q=arduino+micro&source=lnms&tbm=isch&sa=X&ved=0 ahUKEwjp69-Xw7PfAhV2ThUIHXAQCWEQ_AUIDigB&biw=1536&bih=754:

[52] Benchoff, B. (August 26, 2014). "New chip alert: The ESP8266 WiFi module ". Hackaday Retrieved 2015-06-24.

[53] https://www.electroschematics.com/wp-content/uploads/2015/02/esp8266-datasheet.pdf

[54] http://www.instructables.com/id/LM35-Temperature-Sensor/

[55] https://www.parallax.com/sites/default/files/downloads/32213-X-BandMotionDetector-v1.1_0.pdf

[56] https://www.lelong.com.my/hc-sr501-motion-sensor-arduino-ir-bodypassive-infrared-sensor-module-deanwong-I5322208-2007-01-Sale-I.htm

[57] Hajek, P. Fuzzy logic. In *Stanford Encyclopedia of Philosophy*. Retrieved from http://plato.stanford.edu/entries/logic-fuzzy/, 2010.

[58] Oliva, F., Sormani, P., Contri, R., Campana, C., Carubelli, V., Cirò, A., and Ammirati, E. "Heart rate as a prognostic marker and therapeutic target in acute and chronic heart failure." *International Journal of Cardiology*. https://doi.org/10.1016/j.ijcard.2017.09.191, 2017.

[59] Fritzing Software, http://fritzing.org/home/

[60] Sert, I. A. and Saritas U. "A fuzzy expert system design for diagnosis of prostrate cancer," In Proceedings of the International Conference on Computer Systems and Technologies, Kanya, Turkey, 2003.

[61] https://www.mathworks.com/help/thingspeak/act-on-your-data.html

[62] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," IEEE Internet of Things Journal, vol. 3, no. 5, pp. 637–646, Oct 2016.

APPENDIX A

Coordinator Node Code:

```
//----------Cordinator Sensor Node--------------------------------------//


#include "DHT.h"
#include <SoftwareSerial.h>
#include <stdlib.h>
#define Rx 10
#define Tx 11
String Api = "******";      // Wifi ID
String Passwrd = "*********"; // Wifi Password
String Api = "2GHZO9JY8X5YMR8W";
String Host = "api.thingspeak.com";
String Port = "80";
int countTrueCom;
int countTimeCom;
boolean found = false;
int valSens = 1;
int ledPin = 13;
int PirLed = 4;
int PirSens = 3;
int PirState = LOW;
int PirVal = 0;
int LIGHT_SENSOR_PIN = 5 ;
int LIGHT_LEDPIN  =  6 ;
int Light_switch_state = 0 ;
SoftwareSerial esp8266(RX,TX);
int lm35Pin = A0;
#define DHTPIN 2
#define DHTTYPE DHT11
```

```
DHT dht(DHTPIN, DHTTYPE);
bool started= false;
bool started2  = false;
char incomingByte ;
char msg[5];
char msg2[5];
byte index;
int Fall_Detect ;
int XBAND_OUT ;
int ChairSen1 ;
int ValSen ;
int FallSensor ;
int SleepDetect ;
void setup() {
  pinMode(PIRled, OUTPUT);
  pinMode(PIRsensor, INPUT);
  pinMode(LIGHT_LEDPIN, OUTPUT);
  Serial.begin(9600);
  dht.begin();
  Serial.begin(9600);
  ESP.begin(115200);
 Serial.available() ;
  sendCom("AT",5,"OK");
  sendCom("AT+CWMODE=1",5,"OK");
  sendCom("AT+CWJAP=\""+ AP +"\",\""+ PASS +"\"",20,"OK");
}

void loop() {
  float H = dht.readHumidity();
  float T = dht.readTemperature();
 //---------------------Bed Sensors --------------------------------------//
```

```
while (Serial.available()>0){
    incomingByte = Serial.read();
    XBAND_OUT = atoi(msg2);
   if(incomingByte == 'A')
   {
     Fall_Detect = Serial.read();
   }
   if(incomingByte == 'B')
   {
     Fall_Detect = Serial.read();
   }
    if(incomingByte == 'C')
   {
     Fall_Detect = Serial.read();
   }
    if(incomingByte == 'D')
   {
     Fall_Detect = Serial.read();
   }
   if(incomingByte == 'E')
   {
     XBAND_OUT = Serial.read();
   }
   if(incomingByte == 'F')
   {
     SleepDetect = Serial.read();
   }
 }
ValSen = Fall_Detect ;
int SleepSen = SleepDetect ;
```

```
int SleepSensor ;
int Body_Movement;
Body_Movement = XBAND_OUT - 48 ;
if(ValSen == 48 ){
 FallSensor = 0 ;
}
if(ValSen == 49){
 FallSensor = 1;
 }
if(SleepSen == 48 ){
 SleepSensor = 0 ;
}
if(SleepSen == 49){
 SleepSensor = 1;
 }
 PirVal = digitalRead(PIRsensor);
 if (PirVal == HIGH) {
  digitalWrite(PirLed, HIGH);
  delay(100);

  if (PirState == LOW) {
   Serial.println("Motion detected!");
   PirState = HIGH;
  }
 }
 else {
   digitalWrite(PIRled, LOW);
   delay(200);

   if (PirState == HIGH){
    Serial.println("Motion stopped!");
```

```
      PirState = LOW;
    }
  }
Light_switch_state = digitalRead(LIGHT_SENSOR_PIN);
  if (Light_switch_state == LOW)
  {
    digitalWrite(LIGHT_LEDPIN, LOW);
    Serial.println("LIGHT OFF");
  }
  else
  {
    digitalWrite(LIGHT_LEDPIN, HIGH);
  }
  if (isnan(t) || isnan(h)) {
    Serial.println("The sensor can not be read");
  } else {
    Serial.print("FALL Detected: ");
    Serial.println(FallSensor);
    Serial.print("Humidity: ");
    Serial.print(h);
    Serial.print(" %\t");
    Serial.print("Temperature: ");
    Serial.print(t);
    Serial.println(" *C");
    Serial.println(XBAND_OUT);
    Serial.println(Body_Movement);

 String getData = "GET /update?api_key="+ Api +"&"+ field +"="+String(t);
 getData +="&field2=";
 getData += String(h);
 getData +="&field3=";
```

```
 getData += String(PIRval);
  getData +="&field4=";
 getData += String(Light_switch_state);
 getData +="&field5=";
 getData += String(FallSensor);
 getData +="&field6=";
 getData += String(XBAND_OUT);
 getData +="&field7=";
 getData += String(SleepSensor);
 sendCommand("AT+CIPMUX=1",5,"OK");
 sendCommand("AT+CIPSTART=0,\"TCP\",\""+ Host +"\","+ PORT,15,"OK");
 sendCommand("AT+CIPSEND=0," +String(getData.length()+4),4,">");
 esp8266.println(getData);delay(500);countTrueCommand++;
 sendCommand("AT+CIPCLOSE=0",5,"OK");
  }
}

void sendCommand(String command, int maxTime, char readReplay[]) {
  Serial.print(countTrueCommand);
  Serial.print(". at command => ");
  Serial.print(command);
  Serial.print(" ");
  while(countTimeCommand < (maxTime*1))
  {
    esp8266.println(command);
    if(esp8266.find(readReplay))
    {
      found = true;
      break;
    }
    countTimeCommand++;
```

```
  }
  if(found == true)
  {
    Serial.println("OYI");
    countTrueCom++;
    countTimeCom = 0;
  }
  if(found == false)
  {
    Serial.println("Fail");
    countTrueCom = 0;
    countTimeCom = 0;
  }
  found = false;
 }
```

Sensor Node code:

```
#include <NewPing.h>
#define SonarA 2
#define SonarB 3
#define SonarC 4
#define Max_Dist 200
#define Num_Sonar 3
#define led 12
//---------------XBAND Variables -------------------------------//
const int XBAND_ledPin = 13;
const int XBAND_maxCounter = 2000;
const int XBAND_maxMIDI = 80;  //127
const int xBandPin = A1;
  NewPing sonar[NUM_SONAR] = {
  NewPing(SonarA, SonarA, Max_Dist),
  NewPing(SonarB, SonarB, Max_Dist),
```

```
  NewPing(SonarC, SonarC, Max_Dist)
};
int distance[NUM_SONAR];
int Floor_MATpin_1 = 5;
int Floor_MATpin_2 = 6;
int Bed_MATpin_3 = 7;

int light_1 = 13;
void setup(){
  Serial.begin(9600);
  pinMode (Floor_MATpin_1,INPUT);
  pinMode (Floor_MATpin_2,INPUT);
  pinMode (Bed_MATpin_3,INPUT);
  pinMode (light_1, OUTPUT);
  pinMode(led, OUTPUT);
  pinMode(xBandPin, INPUT);
  pinMode(XBAND_ledPin, OUTPUT);
}
void loop (){
updateSonar();
  int MAT_1 = digitalRead(Floor_MATpin_1);
  int MAT_2 = digitalRead(Floor_MATpin_2);
  int MAT_3 = digitalRead(Bed_MATpin_3);
  int Fall_Detect ;
  int Sleep_Detect ;
//---------------------UltraSonic Sensors and Floor Mats ----------------------------------//
   if (distance[0] <= 60 && distance[1] <= 60 && distance[2] <= 60 && MAT_1 == HIGH &&
MAT_2 == HIGH) {
     Fall_Detect = 1 ;
     Serial.print('A');
     Serial.print(Fall_Detect);
```

```
    delay(50);
}
else if (distance[0] <= 60 && distance[1] <= 60 && MAT_1 == HIGH ) {
    Fall_Detect = 1 ;
    Serial.print('B');
    Serial.print(Fall_Detect);
    delay(50);
}
else if (distance[1] <= 60 && distance[2] <= 60&& MAT_2 == HIGH) {
    Fall_Detect = 1 ;
    Serial.print('C');
    Serial.print(Fall_Detect);
    delay(50);
}
  else {
    Fall_Detect = 0 ;
    Serial.print('D');
    Serial.print(Fall_Detect);
    delay(500);
  }
  if (MAT_3 == HIGH){
   Sleep_Detect = 1 ;
  }
  if (MAT_3 == LOW){
   Sleep_Detect = 0 ;
  }
   Serial.print('F');  //Starting symbol
   Serial.println(Sleep_Detect);
   delay(50) ;
//-----------------------------------XBAND Sensor-----------------------------------------------//
```

```
int XBAND_counter;
  XBAND_counter = 0;
  for(int idx = 0; idx < XBAND_maxCounter; ++idx)
  {
      int XBAND_sensorValue = analogRead(xBandPin);
      if(XBAND_sensorValue)
      {
        ++XBAND_counter;
      }
            digitalWrite(XBAND_ledPin, XBAND_sensorValue ? HIGH : LOW);
      delayMicroseconds(50);
  }
  if(XBAND_counter)
  {
    int XBAND_OUT = map(XBAND_counter, 10, XBAND_maxCounter, 10,
XBAND_maxMIDI)
    Serial.print('E');
    Serial.println(XBAND_OUT);
    delay(50) ;
  }
}
void updateSonar() {
  for (int i = 0; i < NUM_SONAR; i++) {
    distance[i] = sonar[i].ping_cm();
    if (distance[i] == 0)
      distance[i] = 0 ;    //MAX_DISTANCE;
  }
}
```