Pharos Engineering Science Journal (PES)

1

ISSN: 3009-6197 Vol. 1, No. 1, 2024 

# Enhanced nRF24-Based Wireless Communication Protocol for Patient's Temperature Monitoring

Mina M. Aziz[1, *], Hussein Seleem[1,2], Amira S. Ashour[1]

[1] Electronics and Electrical Communications Engineering Department, Faculty of Engineering, Tanta University, Egypt

[2] Electrical Engineering Department, Faculty of Engineering, Pharos University in Alexandria, Alexandria, Egypt

*Corresponding author's email: mina.medhat.ramzy@f-eng.tanta.edu.eg

***Abstract*—Temperature monitoring is considered one of the significant vital signs in clinical practice for detecting malignant hyperthermia, and overheating. For collecting and transferring body temperature measurements to a host computer for further monitoring over time, Internet-of-Things (IoT) system become essential. Several wireless communication protocols are now used to create wireless body area networks, which require different types of wireless transceiver modules. Nordic's nRF24L01 is one of the most remarkable wireless transceiver modules that can be employed for vital signs monitoring owing to its low power consumption, low price, and acceptable data rate. However, some important communication features are missing from the nRF24L01 module which are essential to increase the reliability and stability of the communication system for monitoring. In this paper, an extra communication layer is proposed to improve the communication scheme of nRF24L01. To test the proposed communication protocol, two devices were designed, namely the master device and the wireless sensor node with a temperature sensor. The experimental results' findings established that the connection status is probably tracked by both the slave node and the master device. Additionally, the error between the transmitted temperature reading and the received temperature readings is zero. The lowest recorded temperature reading was approximately 36.5 degrees Celsius while the highest recorded temperature reading was about 38.6 degrees Celsius for a volunteer patient.**

***Index Terms*— Wireless body area network, nRF24L01, internet of things.**

## I. INTRODUCTION

HEALTHCARE monitoring systems have become one of the cornerstones of modern medical practices, attracting significant attention from scientists and developers. Temperature assessment is considered a preliminary clinical examination that allows the physician to have an accurate insight into the patient's up-to-date condition. Accordingly, reliable and comfortable monitoring systems become a must by integrating sensors for the vital signs, such as the temperature, through IoT, and communication protocols [1].

Traditionally, monitoring systems in hospitals use wired sensors to monitor the patients' vital signs. Each sensor is connected directly to the monitor beside the patient's bed through multiple wires, which restrict the patients' mobility and causes discomfort [2]. Smart solutions for body temperature monitoring are raised with the progression in communication technology. For the transmission of information, RF data based on wave propagation via a wireless sensor system can be realized using the nRF24L01 module. This module has low power consumption, relatively long range, acceptable sensitivity, and acceptable data rate [3]. Previous studies have explored the use of the nRF24L01 wireless transceiver in healthcare monitoring systems and other wireless systems, for example, Wu *et al.* [4] designed a wireless alarm system using nRF24L01 without addressing the connectivity monitoring. Recently, Wedashwara *et al.* [5] implemented a system to track the effects of natural disasters, such as earthquakes using nRF24L01, but without designing a protocol to manage the communication between nodes or track connection status. Chen *et al.* [6] proposed a non-standard wireless communication protocol with a star topology based on the nRF24L01 chip to create a wireless body area network (WBAN) for gathering physiological signals. They added 6 control bytes in the data packet but without considering connectivity monitoring in their protocol.

In summary, while prior studies have utilized the nRF24L01 module for healthcare monitoring applications, they have not adequately addressed the need for connectivity monitoring as a management feature in these wireless systems. Subsequently, the key contribution of this paper is to develop an enhanced wireless communication protocol based on nRF24L01 that provides connectivity monitoring and other features that can be used in monitoring the patient's body temperature. To achieve this, an additional communication layer is added on top of the nRF23L01 architecture.

The proposed communication protocol's architecture will consist of three layers and an interface layer. The upper layer and the interface layer will be implemented on the microcontroller, while the two lower layers are already implemented on the nRF24L01. The new upper layer,

ISSN: 3009-6197 Vol. 1, No. 1, 2024 

integrated into the hosting microcontroller's code, introduces additional features. This new layer is responsible for incorporating essential management capabilities, including cryptography, connection orientation, connectivity monitoring, and more.

Thus, the main objective of this work can be summarized as follows:

- Develop the proposed wireless communication protocol that enhances the functionality of the nRF24L01.
- Design a hardware monitoring system including testing devices to implement the proposed wireless communication protocol.
- Test the connection between a master device and the wireless sensor node and discuss the results of these tests.

## II. Materials and Methodology

There are various wireless communication protocols that are used to create IoT systems, usually built on wireless transceiver modules operating in unlicensed industry, science, and medicine (ISM) radio bands [7-8]. Each wireless transceiver module has its own advantages and disadvantages, tailored to the specific requirements of the targeted system. Designing a WBAN for healthcare monitoring applications has main requirements as follows [9]:

1. Stability: the system must operate reliably and consistently.
2. Reliability: the communication links must be dependable and robust.
3. Low power consumption: to enable long-term, mobile use by patients.
4. Acceptable data rate: that meets the bio-signal specifications.
5. Short range: appropriate for a body area network environment.
6. Small size and light weight: for patient comfort and convenient deployment.

Guaranteeing these key criteria is crucial when selecting or developing a wireless protocol for healthcare monitoring WBANs. Nordic's nRF24L01 meets some of these key requirements which are appropriate for healthcare monitoring. It provides low power consumption as it consumes a current of 900nA in power down mode, and 26 µA in standby mode (idle mode). The nRf24L01 module also offers multiple power transmission levels and data rates up to 2MHz. It has four power transmission levels: 0, -6, -12, and -18 dBm and requires a voltage source ranging from 1.9 volts to 3.6 volts [3]. Nevertheless, the standard nRF24L01s' protocol lacks the required stability and reliability for healthcare applications.

Using the standard nRF24L01 protocol, the master will not notice the absence of the sensor node unless it attempts to send data and fails, as the standard protocol lacks connectivity monitoring. This leads to unreliable and unstable communication due to delays in detecting such events. Additionally, if another nRF24L01 device in the same area coincidentally uses the same receiver address, the master could mistakenly identify it as the intended receiver.

To address this, the proposed wireless communication protocol adds an additional communication layer on the top of the nRF24L01 architecture. This new layer provides enhanced connectivity monitoring feature that improves the overall stability and reliability of the wireless links [3].

The proposed WBAN system consists of a master device and wireless biomedical sensor node attached to the patient's body as shown in Figure 1. The sensor node is attached physically to the patient's body for measuring its temperature. It samples and processes the sensed temperature measurements, while the master is connected directly to the monitor beside the patients' bed. The sensor node contains the temperature sensor, microcontroller, battery, and nRF24L01 module. The master device also includes a microcontroller, battery, and nRF24L01. By building on the nRF24L01 hardware and enhancing the protocol with additional connectivity management capabilities, this proposed WBAN solution aims to meet the stringent requirements of healthcare monitoring applications.

### *A. Hardware circuitry*

As mentioned before the master and the wireless biomedical sensor node both contain a microcontroller and nRF24L01 wireless transceiver module. The specific microcontroller used is the Atmega32a from the AVR company. The master's microcontroller contains the software module of the master version of the proposed protocol, while the sensor node's microcontroller contains the software module of the slave version of the proposed protocol. In both circuits, the microcontroller is connected to the nRF24L01 module though Serial Peripheral Interface (SPI) connection. Additionally, the wireless sensor node includes a temperature sensor, which allows it to measure and transmit temperature data. By using the Atmega32a microcontroller and integrating the custom protocol software, the proposed WBAN system can leverage the capabilities of the nRF24L01 wireless transceiver to enable the desired connectivity monitoring and management features.

#### 1) *Atmega32a Microcontroller*

Atmega32a is a popular microcontroller providing 32Kbytes of flash memory and 2K bytes of RAM. These specifications are sufficient to host the proposed wireless communication protocol software and the temperature sensor control software. It also contains an SPI communication protocol to connect the nRF24L01 wireless transceiver. It

ISSN: 3009-6197 Vol. 1, No. 1, 2024 

operates on a 4.5 to 5.5 Volts power supply with 16MHz crystal oscillator. The controller also includes analog to digital converter (ADC) to read the output of the temperature sensor [10].

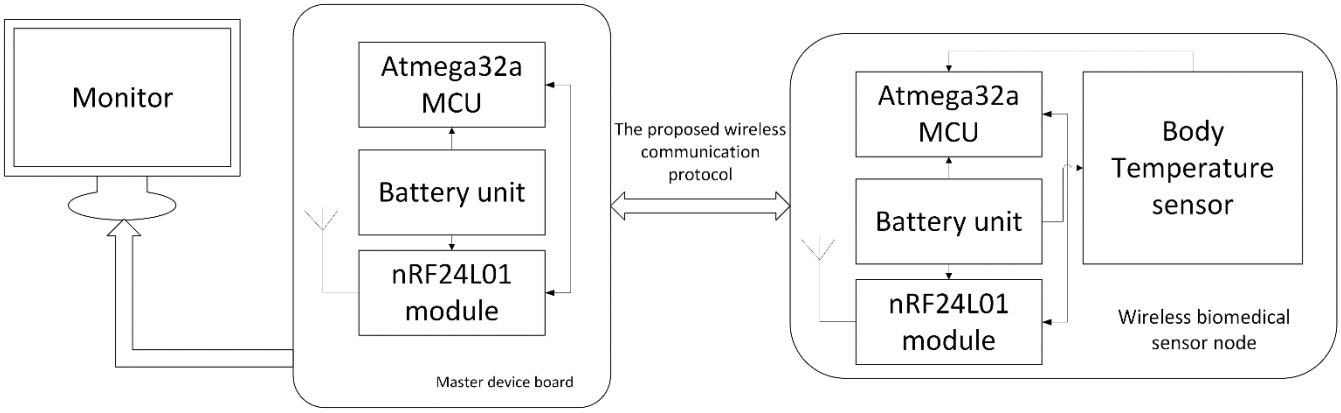


**Fig. 1.** Block diagram of wireless body area network using nRF24L01.

2) *nRF24L01Transceiver Module*

The nRF24L01 is a low-power 2.4 GHz wireless transceiver designed by Nordic Semiconductor. It integrates a complete 2.4 GHz RF transceiver, synthesizer, and baseband logic. It operates on a low voltage level of 1.9 to 3.6 Volts, suitable for battery-operated devices. The communication with the microcontroller will be via the onboard SPI interface, enabling fast easy integration [3].

3) *LM35 Temperature sensor*

LM35 is an integrated circuit used as a temperature sensor, which converts the temperature degree into an analog output voltage. The output voltage is linearly proportional to the sensed temperature in degrees Celsius. It provides typical accuracy of 0.24 degree Celsius at the room temperature. It has a scale of +10.0 milli-volt per 1°C which can be utilized in the software module of temperature sensing section in the biomedical sensor node. It operates on a power supply ranging from 4 to 20 volts [11].

*B. Software Requirements*

The microcontrollers of both the master device and the wireless biomedical sensor node are programed in C++ using the Atmel Studio IDE. The master device contains two main software modules. One module implements the master version of the proposed communication protocol. The other module is responsible for writing the data to the monitor. Similarly, the wireless sensor node also contains two main software modules to implement the slave version of the proposed communication protocol, and to read the temperature sensor, respectively. The second software module actives and uses the ADC peripheral to read the data from the temperature sensor. By connecting the AVCC pin to 5 volts, the step size (Δ) of the quantizer used in the ADC operation is calculated as:

$$\Delta = \frac{5}{2^{10}} volts \quad (1)$$

where Δ is the step size of the quantizer. The Lm35 temperature sensor generates an analog voltage proportional to the sensed temperature. This analog voltage is then measured by the ADC module of the microcontroller. The temperature reading ($T$) can then be calculated using the formula [11]:

$$T = \frac{Vin(mV)}{10.0(\text{mV}/\,°C)}\,°C \quad (2)$$

where Vin (mV) is the analog voltage presented to the ADC input. The temperature reading from the digital output of the ADC can be then calculated by:

$$T = \frac{\text{D} \times \Delta \times 1000(\text{mV})}{10.0(\text{mV}/\,°C)}°C \quad (3)$$

Where ($T$) is the scaled temperature reading in degree Celsius, ($D$) is the quantized output of the ADC, (Δ) is the step size of the quantizer in volt.

*C. Proposed Full-Stack Wireless Communication Protocol*

The nRF24L01 wireless module only provides the two lowest layers of the network architecture which are the physical layer and part of the data link layer. This limited implementation cannot ensure stability, reliability, or connectivity monitoring as mentioned previously. To address these limitations, an additional layer is added above the two layers provided by the nRF24L01. The proposed wireless communication protocol's

ISSN: 3009-6197 Vol. 1, No. 1, 2024 

network architecture consists of four layers as shown in Figure 2:

1. *Two low-level external layers:* These layers are embedded in nRF24L01 module, namely the physical layer, and the data link layer. They are external with respect to the hosting microcontroller.
2. *One high internal layer*: This layer is called "Link Control & data exchange" (LCDE). It is an additional layer embedded inside the microcontroller as a software module, so it is called internal. It completes the missing features from the nRF24L01's Data Link Layer.
3. *An application layer*: This layer is found above the entire network architecture.
4. *Abstraction interface*: This layer is in-between the internal and external layers to ensure abstraction of the internal layer from the external layers, enabling future upgrades.

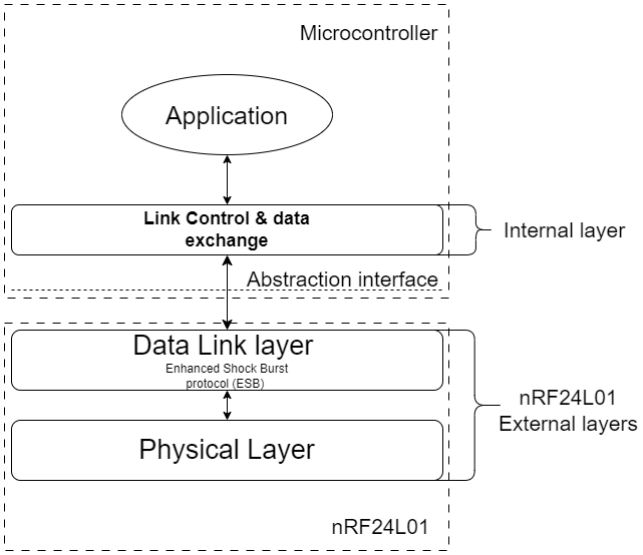


**Fig. 2.** The proposed wireless communication protocol's network architecture.

This hierarchical structure facilitates the abstraction of the application layer from the lower layers, which enables the upgrade and development of the lower-level components without affecting the upper layers.

1) *External layers*

*a) Physical layer*

The physical layer of the nRF24L01 module is the lowest layer of the network architecture. It operates in the 2.4 GHz ISM band. It provides information about the frequency band, the modulation technique, the supported data rate, the power of transmitted and received signals, and the antenna used for transmission and reception.

- *Frequency Band*

The nRF24L01 operates in the unlicensed 2.4 GHz frequency band. This band is divided into 126 RF channels, which can be selected using higher-level network layers that provide frequency division multiplexing and multiple access [3].

- *Modulation technique*

The nRF24L01 uses Gaussian Frequency Shift Keying (GFSK) as its modulation scheme, which is a widely used modulation technique in wireless communication systems. It employs Gaussian filtering to shape the transmitted signal, which results in improved spectral efficiency and reduced out-of-band emissions [12]. This modulation technique provides smooth transitions between the frequency states to enhance the system's protection from interference and fading.

- *Data rate*

The nRF24L01 module supports a range of data rates, offering flexibility for various applications. Specifically, it supports data rates 250 kbps, 1Mbps, and 2 Mbps, allowing for efficient transmission of data over short to medium distances.

- *Power of the transmitted and received signals.*

The nRF24L01 transceiver contains its own power amplifier, which has four different levels of transmission power. Recent studies have found that the exposure to high power RF signals at the 2.4 GHz ISM frequency band may lead to thermal effects and potential risks to human health [13]. Therefore, the recommended power level for transmission is -6 dBm.

*b) Data Link Layer*

Enhanced Shock Burst protocol (ESB) is the data link layer that is used in the nRF24L01 module, this is the second layer in the communication protocol. This layer enhances the efficiency of data transmission. It provides features like packet format, flow control (automatic retransmission, packet acknowledgment), physical addressing for destination only, error detection, and access control.

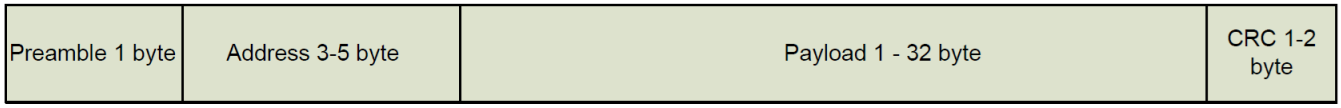


**Fig. 3.** The ESB packet with payload (0-32 bytes).

ISSN: 3009-6197 Vol. 1, No. 1, 2024 

- *Packet format*

The ESB packet format consists of five parts, as shown in Figure 3, which are as the following [3]:

- *Preamble*: A one-byte synchronization pattern (either 10101010 or 01010101) to help the receiver detect the start of the frame [14]. The preamble depends on the first bit of the address part, where if the address starts with 0, then the preamble will be 01010101, while if the address starts with 1, then it will be 10101010. This is done to be sure that there are as many transitions as possible in the start of the packet to help the receiver in detecting and synchronizing the demodulator [3].
- *Address:* 3, 4 or 5 bytes that specify the destination address. It is used by the receiver to accept or deny the whole packet. The length of the address is set using the upper layers of the protocol.
- *Packet Control Field:* It is the third part of the packet; it consists of 9 bits (payload length of 6 bits, Packet Identification (PID) of 2 bits and No Acknowledgment (NO_ACK) of 1 bit).
  - *Payload length*: specifies the number of bytes in the payload (from 0 to 32 bytes) [3]. The code 000000 = 0-bytes (used for Acknowledgment (ACK) packet only). The code 100000 = 32-bytes. More than 100000 in means do not care. It is used in dynamic payload length, Otherwise, the length is constant and do not care.
  - *PID*: the 2-bit PID along with the Cyclic Redundancy Check (CRC) are used by the receiver device to determine if a packet is retransmitted or new when more than one data packet is lost on the link [3]. For each packet transmitted, the transmitter assigns new (PID). If the receiver receives 2 consecutive packets with the same (PID), the receiver checks the (CRC). If the (CRC) matches, then the receiver discards the second packet and considers it as a replication of the first one.
  - *NO-ACK*: inform the receiver whether the transmitter waits for the acknowledgement or not [3].
- *Payload:* the main part of the packet, which contains the main data that is to be sent. It can be from 0 to 32 bytes.
- *CRC:* it is the main error detection mechanism in the packet. It is one or two bytes. It is calculated over the address, packet control field and payload. The polynomial of the 1 byte is [3]:

$$X^8 + X^8 + X + 1. \quad (4)$$

While the polynomial of the 2 byte is [3]:

$$X^{16} + X^{12} + X^5 + 1. \quad (5)$$

- *Access control*

The NRF24L01 module uses carrier sense multiple access with collision avoidance (CSMA/CA) to avoid data collisions. Before transmitting, the module checks that the channel is clear. However, if the channel is busy, it waits for a random backoff period before checking again. This helps to reduce the chances of collisions between devices trying to transmit simultaneously.

2) *Abstraction interface.*

The Abstraction interface ensures the abstraction between the internal layer of the nRF24L01 and the external layers. It is primarily based on the SPI communication protocol to interact with the nRF24L01. This layer handles functions like changing the channel, transmission power, sending packets, and initializing the nRF24L01 module.

3) *Internal layer (Link Control & Data Exchange)*

This layer is implemented as a software module inside the hosting microcontroller. It added features to enhance communication and create reliable IoT systems, such as tracking connection status, creating packets, and validating packets.

### *D. Connection status*

Tracking the states of the connection between devices is a very important issue. The Master-sensor link has its own states and procedures. The master has three different states which are "Searching", "waiting for first encryption key", and "connected" status as shown in Figure 4. While the slave 's three statuses are "waiting", "connected waiting for first encryption key", and "connected" status as shown in Figure 5. The Master has a stack for the connected slave device that stores some important information about the connected slave such as the frequency channel, Encryption key, status, etc. Also, the slave has its stack to track its communication status and parameters.

1) *Searching and waiting status:*

"Searching" status for Master and the "waiting" state for slave are both complementary states. Those are the idle state for the master and the slave at the start of the connection. The master keeps searching for the slave by using the "search" command. when it has this state for the slave in its stack. The "search" command is sent one time very a known period. At the same time, while the slave in this state it keeps waiting till it receives the "search" command, and it ignores any other commands. At this state, both the master and the slave are on the "Search connect channel". This channel is used only for establishing the connection.

2) *Connected waiting for first encryption key:*

This state is found in both the master and the slave devices. After the slave receives the "search" command and responds by the "connect" command. The slave then operates in the frequency channel that the master has sent to the slave in the search command. The slave waits till the master sends "New

ISSN: 3009-6197 Vol. 1, No. 1, 2024 

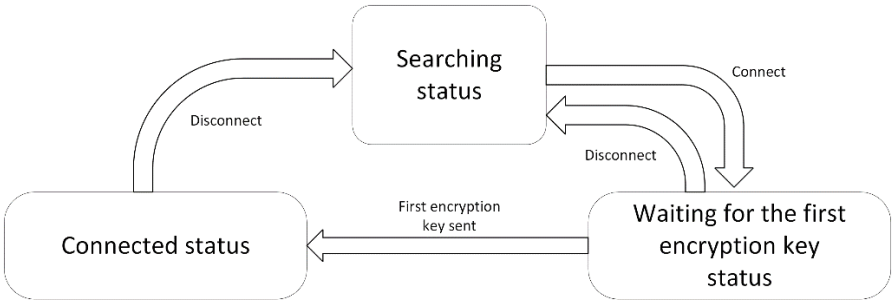


**Fig. 4.** Master's connection status diagram.

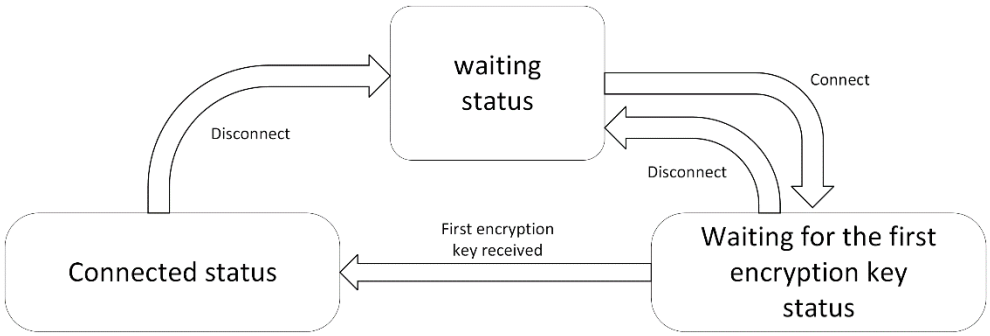


**Fig. 5.** Slave's connection status diagram.

session key" command to get the first encryption key. After that, the slave replies with "New session key accepted" command. Then, both the master and the slave enter the "connected" states

3) *Connected state:*
This is the idle state after the connection directly. In this state the slave waits for commands from the master. The slave does not start the communication, the master starts the communication. The master sends commands to the slave according to the timing table defined in the protocol.

*E. Creating and validating the packets*

Instead of using the full 32 bytes data section of the nRF24L01's ESB packet format in sending data only, the proposed protocol adds additional overhead information to enhance the communication reliability and stability. Both the master and the slave devices in the WBAN system use the same general packet format, despite the differences in the specific command types they may utilize. The general packet format consists of 6 main fields as shown in Figure 6. The fields are the version number, the number of bytes, the sender address, the packet type, the data, and the digest. This packet is to be sent as the payload of the datalink layer in the external layers of the nRF24L01 module. In the proposed system, the current version of the protocol uses the version number of 0x01, this field is important for future upgrades and backward compatibility with older versions.

In addition, the number of bytes represents the number of bytes sent in the payload, it helps in validating the packet at the receiver. Also, the sender address was specified from byte 2 to byte 5, which is a 4-bytes field that defines the address of the sending device, enabling connection-oriented communications between the master and slave devices. This field is necessary because the underlying nRF24L01's Enhanced Shock Burst protocol does not include the sender address. Moreover, the packet type which is a 1-byte field, defines the type of packet being transmitted. Each numeric value represents a specific command or message type. In this protocol, the length of the data payload is determined by the specific command or message type, where the data bytes, ranging from 7 to 7+n, where n is

| Version number | Number of bytes | Sender address Byte1 | Sender address Byte2 | Sender address Byte3 | Sender address Byte4 | Packet type | Byte0 | Byte n-1 | Hash |
|---|---|---|---|---|---|---|---|---|---|
| Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 7+n-1 | Byte 7+n |

**Fig. 6.** General format of the packets.

ISSN: 3009-6197 Vol. 1, No. 1, 2024 

the number of data bytes. Lastly, the hash (Digest), which is the last byte in the packet that is responsible for ensuring data integrity is calculated using the following formula:

$$digest = \left(\sum_{k=0}^{n-2} data_k\right) mod\ 0xff \quad (6)$$

Where n is the length of the packet. The digest is then placed at the end of the packet. The hash function can be upgraded or changed in future upgrades of the protocol. This structured packet format, with the defined fields, allows the proposed wireless communication protocol to enhance the reliability and stability of the data exchange between the master and wireless biomedical sensor nodes in the WBAN system.

The packet type depends on the "command type" field in the packet, there are some packets that can be sent from the master while the others from the slaves. There are two main types of packets in the proposed protocol, the control packets and the data exchange packets. The control packets are used in establishing new connections, beacon exchange, new encryption keys exchange and new channel exchange. While the data exchange packets are used in reading and writing data. The general format of the packet facilitates the validation process of the packets. The validation process starts by checking the "version number" field in the received packet, if this value exceeds the current version of the protocol on the device, the packet is discarded. The second process of validation is checking the "number of bytes" field, if it doesn't agree with the length of the packet, then the packet is discarded. After that the sender's address field is checked, if it doesn't match the sender address where the packet is supposed to come from, then the packet also is discarded. Lastly the "digest" field, if it doesn't match the packet's digest, the packet is discarded. So, using this procedure, the probability of accepting a wrong received packet is very hard.

### *F. Cryptography in the protocol*

To ensure the security of communication in the upper layer of the protocol, two encryption techniques are implemented: a fixed password and a session key. Each wireless sensor node is assigned a unique fixed password of 4 bytes, which is used before the session key is generated and exchanged. This fixed password primarily encrypts the index of the first frequency channel sent to the slave and the session keys when transmitted from the master to the slave. Session keys, with a length of 2 bytes, are randomly generated by the master. The first session key is sent to the slave while both devices are in the "Connected Waiting for First Encryption Key" status. Periodically, the master generates a new session key and sends it to the slave, encrypted with the fixed password. This process ensures the security of transmitted data and reduces the key lifetime of the session key, compensating for the shorter length of the session key.

### *H. Message Sequence Charts (MSCs) of the protocol.*

The connection sequence can be illustrated using the MSCs shown in Figure 6. It starts from the master as it sends a search packet to the slave while being in the searching status, then if the slave receives this packet it responds by a connect packet. Then the master and the slave enter the "waiting for the first encryption status". In this new status the master and the slave exchange the first encryption key. To make sure that the connection is stable a beacon packets exchange is performed between the master and the slave every fixed period in the connected status.

A sample of data exchange between a master and a slave can be illustrated by Message sequence charts as shown in Figure 8. While being in the Connected status the master can read or write data from or to the slave. If the beacon period is exceeded without any communication between the master and the slave, the master performs a beacon exchange to make sure that the connection is still working.

The connection orientation feature can be illustrated by a Message sequence chart as shown in Figure 9. In this MSC there are two scenarios that describe how the protocol deals with failed data exchange operations by retransmitting the same data again after a defined resend period when the "data received ack packet" is not received.

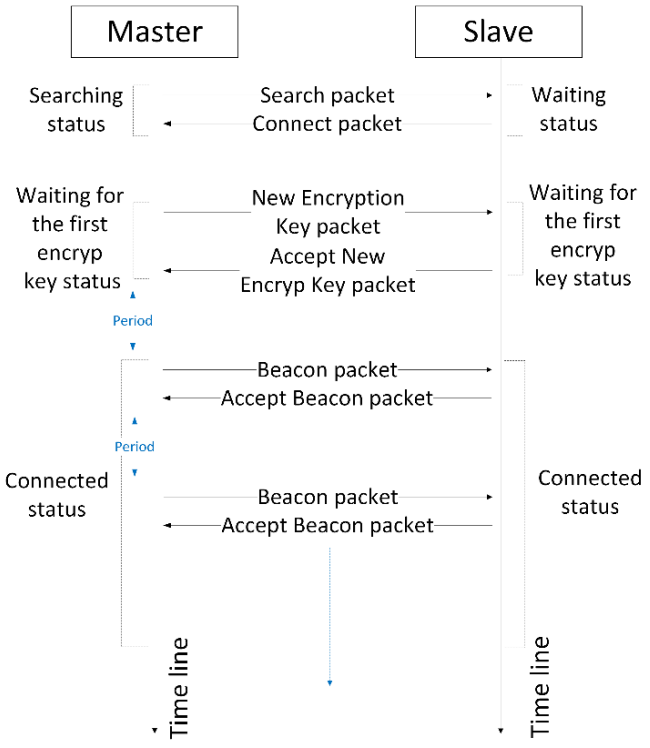


Fig. 7. Message sequence chart of connection sequence.

ISSN: 3009-6197 Vol. 1, No. 1, 2024 

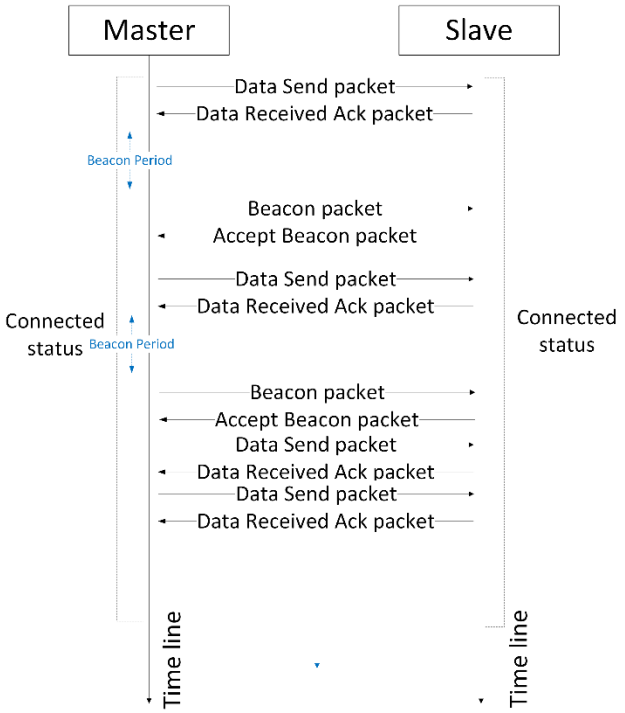


Fig. 8. Message sequence chart of data exchange.

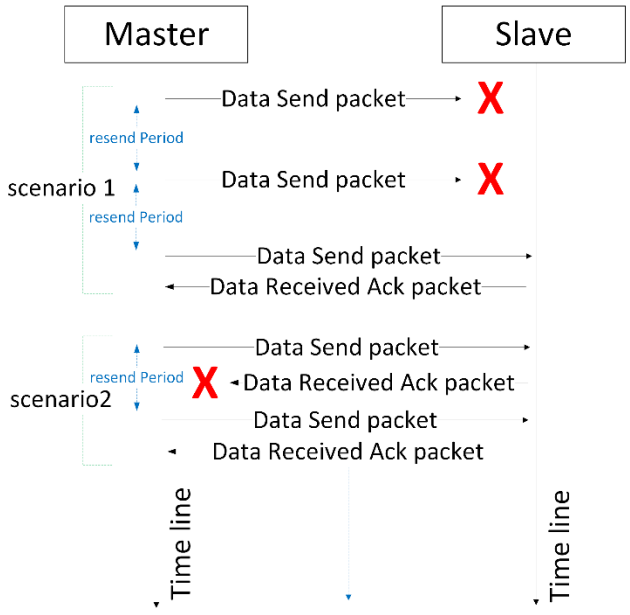


Fig. 9. Message sequence chart illustrating connection orientation feature.

## III. Experiment and Results

To evaluate the proposed wireless communication protocol, a WBAN was implemented with two devices. The first device is the gateway circuit, which serves as the master, as shown in Figure 10 its PCB board. The other one is the temperature sensor circuit which acts as the slave, as shown in Figure 11 its PCB board. Both The master and the slave circuits are based on the Atmega32a microcontroller. Each circuit also includes a USB TTL connection, which connects the microcontroller to a computer, and an nRF24L01 chip connected to the microcontroller through SPI. The temperature sensor circuit contains an LM35 temperature sensor, which converts the temperature into an analog signal.

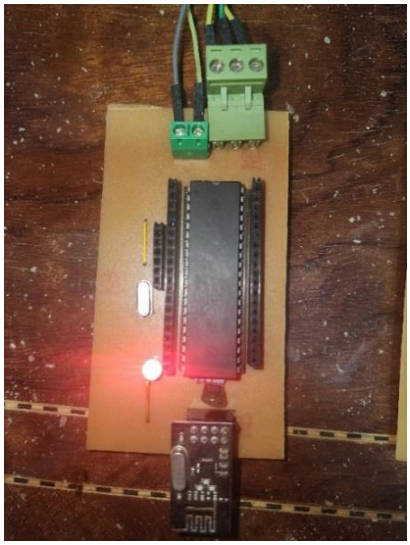

**Fig.10.** Wireless Master device PCB.

This analog signal is measured by the Atmega32a microcontroller ADC every second, and the microcontroller then converts the digital reading into a temperature reading as given in Eq. (3).

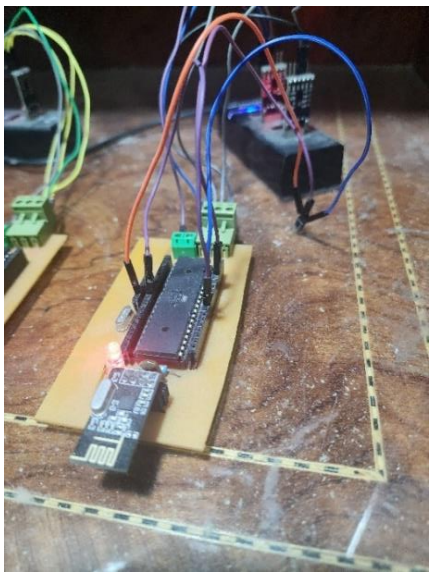

**Fig. 11.** Wireless biomedical temperature sensor node PCB.

The slave circuit also sends the same temperature value to the computer at the same time. The master circuit uses the proposed

ISSN: 3009-6197 Vol. 1, No. 1, 2024 

wireless communication protocol to read the temperature reading every 30 seconds and immediately sends the read value to the computer. A chart of the temperature readings which were sent directly by the slave's microcontroller and the temperature readings sent by the master's microcontroller using the protocol every 30 seconds is shown in Figure 12.

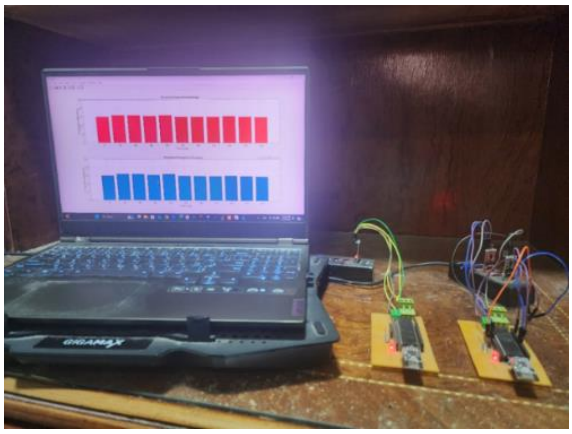

**Fig. 12.** The whole experiment system.

### *A. Monitoring system test*

The configuration of the proposed communication protocol setup for this test was as follows: 1Mhz data rate, and -6 dBm transmission power. Accordingly, several test cases were conducted for evaluating the proposed monitoring system to track the patient's temperature readings. This experiment was repeated 4 times. It was found that the least recorded temperature reading was about 36.5 degrees Celsius, while the highest recorded temperature reading was about 38.6 degrees Celsius. In each test, the received and transmitted temperature readings are recorded for 300 seconds in real time way, and the number of readings recorded during this period was 11 readings. As follows, four scenarios were conducted to evaluate the proposed monitoring system along with a connectivity monitoring test.

*1) Scenario 1: Gradual Increase in Patient's Temperature,* in this experiment, the temperature recording showed that the patient's temperature was gradually increasing over time as shown in Figure 13. The average temperature reading was 37.227 degrees Celsius. Furthermore, the standard deviation of the readings was 0.56937, and the variance was 0.324182. The relatively high standard deviation and variance indicate that the temperature readings were more dispersed and variable, rather than tightly clustered around the mean. This shows that the patient's temperature was gradually changing, either rising or falling, over the course of the experiment.

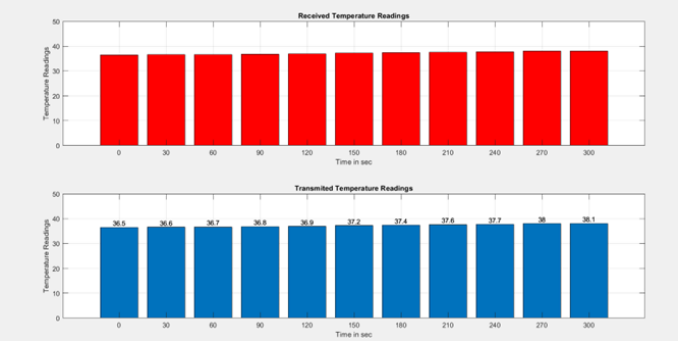


**Fig. 13.** Temperature recording of transmitted and Received readings in case 1.

*2) Scenario 2: Steady High level of Patient's Temperature,* in this experiment, the temperature recording showed that the patient's temperature was steady in a relatively high value as shown in Figure 14. The average temperature reading was 38.2545 degrees Celsius. Also, the standard deviation of the readings was 0.16949, and the variance was 0.028727. The relatively low standard deviation and variance indicate that the temperature readings were tightly clustered around the mean. This indicates that the patient's temperature was stable and not changing significantly during the experiment.

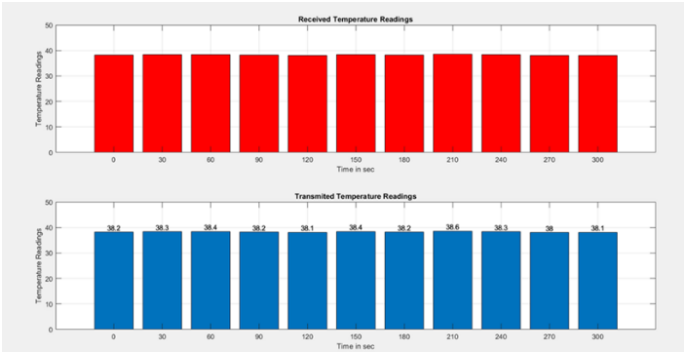


**Fig. 14.** Temperature recording of transmitted and Received readings in case 2.

*3) Scenario 3: Gradual Decrease of Patient's Temperature,* in this experiment, the temperature recording showed that the patient's temperature was gradually decreasing over time as shown in Figure 15. The average temperature reading was 37.554 degrees Celsius. Additionally, the standard deviation of the temperature readings was 0.494699, and the variance was 0.244727. The relatively high standard deviation and variance indicate that the temperature readings were more dispersed and variable, rather than tightly clustered around the mean. This shows that the patient's temperature was gradually changing, either rising or falling, over the course of the experiment.

ISSN: 3009-6197 Vol. 1, No. 1, 2024 

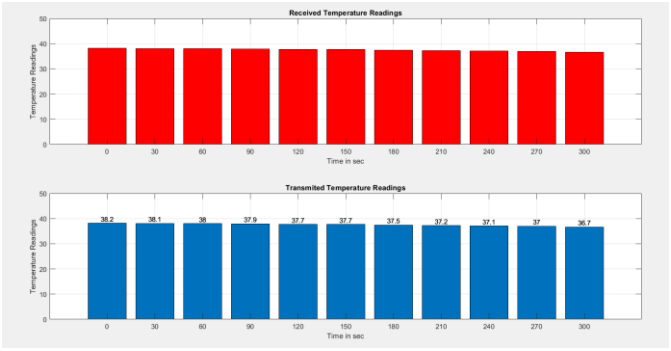


**Fig. 15.** Temperature recording of transmitted and Received readings in case 3.

*4) Scenario 4: Normal Patient's Temperature,* in this experiment, the temperature recordings showed that the patient's temperature readings were steady and within normal readings as depicted in Figure 16. The average temperature reading was 36.6181 degrees Celsius. Additionally, the standard deviation of the temperature readings was 0.087386, and the variance was 0.007636. The relatively low standard deviation and variance indicate that the temperature readings were tightly clustered around the mean, suggesting consistent and stable temperature profile for the patient.

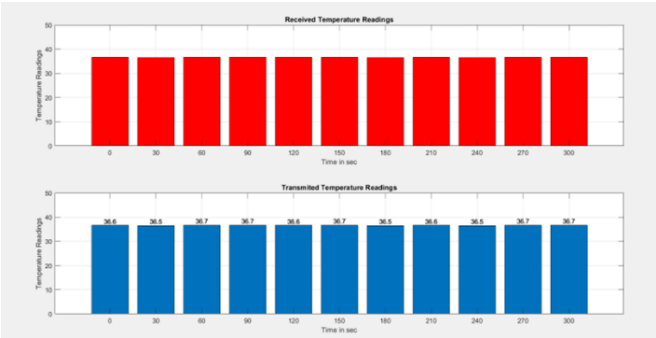


**Fig. 16.** Temperature recording of transmitted and Received readings in case 4.

The executed experiments showed that the system was stable for a period of 300 seconds without any errors or missed temperature readings, with a cross-correlation value between the transmitted and received temperature readings equals to one in the four executed experiments. This indicates that the transmitted and received temperature readings were identical, demonstrating the reliability and accuracy of the data transfer process. This was accomplished by making the use of the new added layer (Link Control & Data Exchange) in the protocol, which added the feature of connection orientation to the protocol, and avoiding some logical errors such as receiving wrong packets that is sent to other devices having the same receiving address by the validation process.

### ***B. Connectivity monitoring test***

The connection status is monitored by both the master and the slave devices. After successfully completing the connection procedure shown in Figure 17, both the master and the slave will update their connection status to "connected".

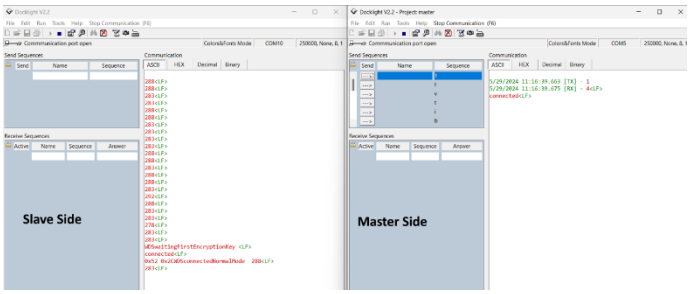


**Fig. 17.** Slave side and Master side when connected.

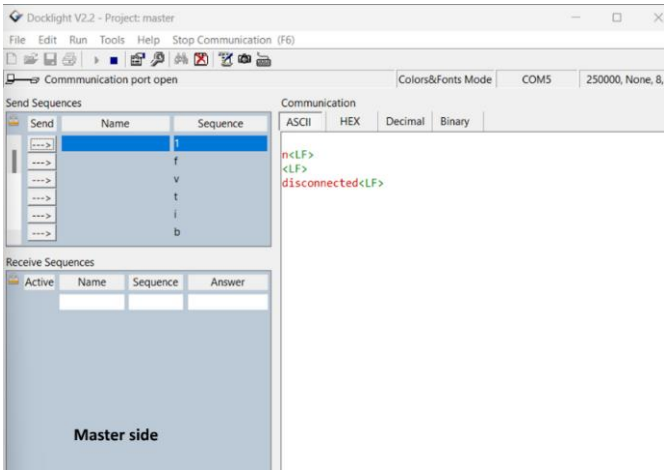


**Fig. 18.** Master side when slave disconnects.

If the slave device goes offline or becomes disconnected for any reason, the master will detect this and change its own connection status to "disconnected" as depicted in Figure 18. The master will then attempt to reconnect to the slave according to the connection sequence specified in the proposed protocol. Conversely, if the master device goes offline or becomes disconnected, the slave will detect this and update its own connection status to "disconnected" as shown in Figure 19.

Reliability in a communication system is typically measured using metrics such as packet delivery ratio, error rate, mean time between failures, and acknowledgments. Stability is assessed through metrics such as connection uptime. In our manuscript, the new layered structure is designed to enhance both reliability and stability through several improvements and we added the connection orientation feature which includes acknowledgments and improvement of the packet delivery ratio and error rate as shown in the tests, the cross correlation between the sent and received signals is one. Also, the mean

ISSN: 3009-6197 Vol. 1, No. 1, 2024 

time between failures is improved using beacon exchange which detects any problem in the connection in short time, then automatically start a new connection procedure from both sides the master and the slave as shown in the "Connectivity monitoring test" subsection. Also, without the connection tracking that is implemented in the proposed protocol, the connection uptime will be unspecified because there is no connection between devices is made. By testing the connection between the master and the slave it is found that the connection can stay stable for more than one day, and in case of any disconnection, the system starts a new connection procedure automatically which increases the stability of the communication.

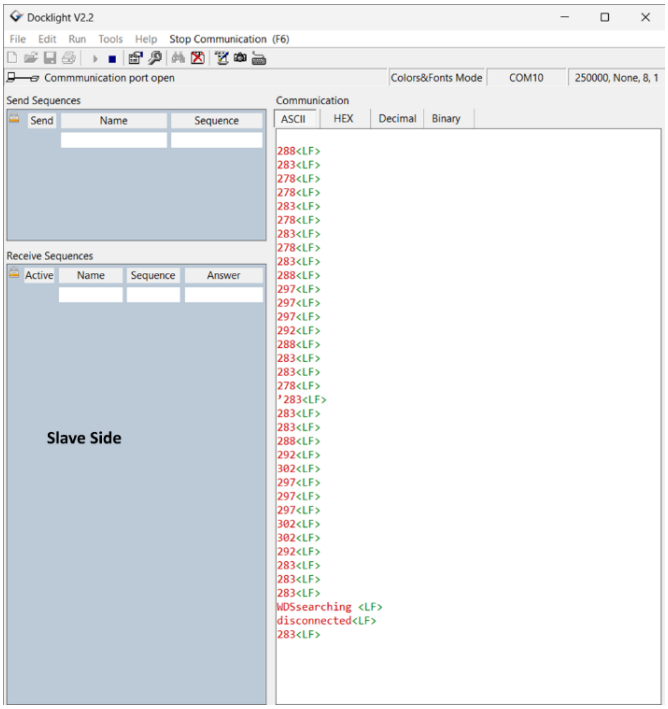


**Fig. 19.** Slave side when Master disconnects.

## IV. Conclusion and Future Work

In this paper, a new wireless communication protocol was developed based on modifying the conventional protocol of Nordic's nRF24L01 module, by adding a new extra layer that enhanced the whole communication protocol. The proposed protocol proved its reliability and stability to be used in healthcare monitoring systems. As, it is found that the sent and received temperature values are identical based on the cross-correlation value calculated in each scenario Also, both the master and slave detect the status of connection all the time. Due to the effectiveness of the proposed protocol for temperature monitoring, it is recommended to apply the proposed system to monitor other vital signs. However, the modified protocol includes connection tracking capabilities, it could benefit from further enhancements to improve reliability and stability. Potential improvements include implementing time division multiplexing, adding new connection modes, and testing the protocol with a larger number of wireless sensor nodes.

## References

[1] N. Verma, S. Singh, and D. Prasad, "A Review on existing IoT Architecture and Communication Protocols used in Healthcare Monitoring System," Journal of The Institution of Engineers (India): Series B, Jun. 2021, doi: https://doi.org/10.1007/s40031-021-00632-3.

[2] Booma Devi Sekar, J. Ma, and M. Dong, "Wired and Wireless Distributed e-Home Healthcare System," IGI Global eBooks, pp. 663–706, Jan. 2020, doi: https://doi.org/10.4018/978-1-7998-1204-3.ch037.

[3] Veerandi Kulasekara, Sachintha Balasooriya, J. Chandran, and Ilya Kavalchuk, "Novel Low-Power NRF24L01 Based Wireless Network Design for Autonomous Robots," Nov. 2019, doi: https://doi.org/10.1109/apcc47188.2019.9026452.

[4] G. Wu, J. Tao, and X. Xu, "Application and Design of Wireless Community Alarm System Based on nRF24L01 Module," Jun. 2019, doi: https://doi.org/10.1109/ccdc.2019.8832864.

[5] W. Wedashwara, M. S. Yadnya, I. W. Sudiarta, I. W. A. Arimbawa, and T. Mulyana, "Solar Powered Vibration Propagation Analysis System using nRF24l01 based WSN and FRBR," JOIV : International Journal on Informatics Visualization, vol. 7, no. 1, pp. 15–21, Mar. 2023, doi: https://doi.org/10.30630/joiv.7.1.1592.

[6] Z. Chen, C. Hu, J. Liao, and S. Liu, "Protocol architecture for Wireless Body Area Network based on nRF24L01," Sep. 2008, doi: https://doi.org/10.1109/ical.2008.4636702.

[7] S. Al-Sarawi, M. Anbar, K. Alieyan, and M. Alzubaidi, "Internet of Things (IoT) communication protocols: Review," IEEE Xplore, May 01, 2017. https://ieeexplore.ieee.org/abstract/document/8079928

[8] S. Elhadi, A. Marzak, N. Sael, and S. Merzouk, "Comparative Study of IoT Protocols," SSRN Electronic Journal, 2018, doi: https://doi.org/10.2139/ssrn.3186315.

[9] N. Verma, S. Singh, and D. Prasad, "A Review on existing IoT Architecture and Communication Protocols used in Healthcare Monitoring System," Journal of The Institution of Engineers (India): Series B, Jun. 2021, doi: https://doi.org/10.1007/s40031-021-00632-3.

[10] C. R. Carroll, "First Experiences with the AVR ATmega32 Microcontroller," Iowa Research Online (The University of Iowa), Oct. 2014, doi: https://doi.org/10.17077/aseenmw2014.1012.

[11] H. Mansor, M. H. A. Shukor, S. S. Meskam, N. Q. A. M. Rusli, and N. S. Zamery, "Body temperature measurement for remote health monitoring system," 2013 IEEE International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA), Nov. 2013, doi: https://doi.org/10.1109/icsima.2013.6717956.

[12] T. S. Rappaport, Wireless Communications. Cambridge University Press, 2024.

[13] M. Fernandez, H. G. Espinosa, D. Guerra, I. Peña, D. V. Thiel, and A. Arrinda, "RF Energy Absorption in Human Bodies Due to Wearable Antennas in the 2.4 GHz Frequency Band," Bioelectromagnetics, vol. 41, no. 1, pp. 73–79, Nov. 2019, doi: https://doi.org/10.1002/bem.22229.

[14] Thinh Hung Pham, V. A. Prasad, and A. S. Madhukumar, "A Hardware-Efficient Synchronization in L-DACS1 for Aeronautical Communications," IEEE transactions on very large scale integration (VLSI) systems, vol. 26, no. 5, pp. 924–932, May 2018, doi: https://doi.org/10.1109/tvlsi.2018.2789467.