Ingénierie des Systèmes d'Information
Vol. 27, No. 4, August, 2022, pp. 521-529
Journal homepage: http://iieta.org/journals/isi

Design and Implementation of Temperature and Humidity Monitoring System Using LPWAN Technology

Muhammad Sani Danladi*, Muhammet Baykara

Department of Software Engineering, Faculty of Technology, Fırat University, Elazığ 23119, Turkey

Corresponding Author Email: 191137130@firat.edu.tr

https://doi.org/10.18280/isi.270401

Received: 6 May 2022
Accepted: 20 July 2022

Keywords:
data transmission, internet of things, long-range, LoRaWAN, LPWAN, temperature and humidity monitoring

ABSTRACT

Temperature and humidity monitoring is an integral aspect of human lives and has several applications ranging from greenhouses, laboratories, food industries, server rooms, data centers, and so on. However, the primary technologies that drive these systems suffer from numerous drawbacks such as deployment cost, coverage, and power consumption. This paper aims to employ new and affordable technology, namely low power wide area networks, to implement a temperature and humidity monitoring system (THMS) due to their low cost, low power consumption, and long range in data transmission. A LoRa-based THMS using a Raspberry Pi 3 B+ gateway with a single channel packet forwarder and an Arduino UNO end device with a DHT11 sensor was designed and implemented. After registering the gateway and the end device on The Things Network (TTN), temperature and humidity values of 22.0°C and 33.0% were recorded by the Arduino serial monitor and the TTN application server. The above implementation clearly shows that sensor values can be effectively transmitted using LoRa and LoRaWAN over long distances with minimal power consumption.

1. INTRODUCTION

The IoT sector has witnessed several technological advancements and growth in the past decades. The above has paved the way for many companies to embrace new ideas in manufacturing wireless devices, particularly for consistent environmental conditions monitoring such as temperature and humidity. IoT devices built with wireless technologies like Bluetooth, wireless sensor networks (WSN), ZigBee, Wi-Fi, and Radio-frequency identification (RFID) [1-4] are widely employed to transmit sensor data from the end nodes to gateways and later sent to the application server for further use by authorized users. Several temperature and humidity monitoring systems have been developed using these wireless technologies and found applications in greenhouses [5], server rooms [6], data centers [7], and so on. In greenhouses, for instance, any wrong combination of temperature and humidity will result in an abnormal spread of diseases to the crops, thereby reducing the yield and profit of the farmers [8]. Server rooms, data centers, laboratories, and food industries also need a proper environmental conditions monitoring and control system. However, the wireless technologies used to monitor these environmental conditions are severely confronted with high power consumption, short-range, and increased deployment costs. For instance, despite being affordable and widely employed, Bluetooth can only transmit data to approximately 10 meters [9]. Also, despite its ability to transfer large amounts of data at high speeds, Wi-Fi can only send data within 50 to 100 meters, depending on the frequency band used by the device manufacturer [10]. Additionally, a constant electric supply is required to power the routers resulting in high energy demand. In certain instances, solar energy and an uninterruptible power supply (UPS) must be used to keep these devices up and running, resulting in extra overhead costs.

Low power wide area network technologies (LPWANs) are the new set of affordable wireless network solutions that enable IoT devices' sensor data to be transmitted over long-range using just battery-operated devices [11]. Examples of LPWANs include LoRa and LoRaWAN, NB-IoT, SigFox, Weightless, and so on. These technologies are made to operate in either the licensed or unlicensed frequency bands or both for remote and secured data transmission. Essential features that make LPWAN technologies to be suitable for IoT devices include:

(1) A typical range of 10 to 40 km in rural areas (with a lesser, 2 to 5 km in urban areas resulting from the presence of obstacles);

(2) Limited energy consumption with more than ten years of battery life and,

(3) A radio chipset costs less than 20 USD.

This paper aims to design and implement a temperature and humidity monitoring system (THMS) using LoRa and LoRaWAN. A Raspberry Pi 3 B+ Lora gateway with a single channel packet forwarder and an Arduino UNO end node with a DHT11 sensor was designed and implemented. Two Adafruit RFM95W LoRa Radio Transceiver Breakouts, each with an operating frequency of 868 MHz, another license-free industrial, scientific, and medical (ISM) frequency band allowed by LoRa Alliance regional parameters for Europe [12] were employed for both transmission and reception. Furthermore, the gateway and the end device are registered on The Things Network (TTN), a free and open-source infrastructure for IoT devices, to demonstrate just one of the numerous applications of the available LPWAN technologies accordingly.

The remainder of this study is arranged as follows: Chapter 2 provides a review of previous and related works. A concise description of LoRa and LoRaWAN, the technology employed in this paper, is provided in Chapter 3. Chapter 4 details the materials and methods used to design and implement the THMS and the steps in registering and connecting the gateway and the end device to TTN. All the obtained results and proper discussions were given in Chapter 5. Chapter 6 concludes the study. Finally, precise and valuable recommendations concerning future work are also offered.

2. RELATED WORKS

Several works were conducted using standard wireless network technologies for temperature and humidity (TH) measurement and monitoring. For instance, a wireless sensor network (WSN) is an essential wireless communication technology that delivers more reasonable edges than wired communication in cost, scalability, and robustness. They are broadly used with various microcontrollers (MCU) like Arduino, Raspberry Pi, etc., for prototyping and real-life deployment of IoT systems where energy, memory, reliability, etc., are crucial. WSNs employed a multi-hub path (a connection of numerous sensor nodes connected in several topologies such as star and mesh configuration) to relate sensor data to the nearest sink nodes (gateway or base station) [13]. The gateways send these sensor data to the internet via wired Ethernet, Wi-Fi, or cellular networks. Gao et al. [14] proposed a WSN-based remote THMS. The study used ATmega 128L as the MCU unit that provides the suggested system's network controlling requirement. Also, DS18B20 and SHT75 sensors were used for TH measurement, where nRF905 wireless transceiver modules transmitted the sensor data to the cloud. The system proved scalable and automated in wirelessly sending the sensor data over a considerable long range. Similarly, Guan et al. [15] designed and implemented a short-range wireless communication-based real-time THMS in the agricultural and industrial sectors. The research work introduced an algorithm that intelligently assigned frequency, reduced energy consumption, and solved resource allocation issues. They used an SHT11 sensor to send the sensor data to the STC89C52 MCU unit. The data was then transmitted using the nRF905 wireless transceiver module and MAX232 chip for serial interfacing between these devices. Radio-frequency integrated circuits (RFICs) are analog circuits operating from 3 kHz to 2.4 GHz [16]. Chang and Hung [17] exploited RFIC's capabilities incorporated with RH/T sensors to directly initiate a continuous measurement of the internal TH of concrete in real-time. Later on, they obtained a remarkable performance compared to the temperature measurements with a Pt100 resistance thermometer and eventually analyzed and deduced that the quantity of reinforcing steel and the strength of the RF signal in the concrete were negatively correlated. Wang and Chi [18] also used an AVR MCU unit-based approach to design and implement a system that sensed specific environmental TH conditions with a DHT11 sensor using a serial peripheral interface (SPI) bus communication. The nRF24L01 wireless transceiver module was used to transmit sensor data, and a Dot matrix LCD12864 display module displayed the results of the measurements to enable wireless communication and real-time monitoring. In addition, a potentiometer was used to set low and high sensor data levels for proper regulation.

The positive contributions of Machine Learning (ML) to various aspects of human lives cannot be overemphasized. When appropriately trained, ML models are famously known for their high accuracy and performance in prediction in multiple applications such as image and speech recognition, natural language processing, self-driving cars, etc. In light of this, several research works have been conducted to predict environmental TH values with ML models. For instance, Bellido-Jiménez et al. [19] proposed a novel ML-based approach that used several ML classifiers on the intra-daily temperature dataset to predict solar values in different geo-climatic conditions such as elevation, sea distance, and aridity. They compared their outcome to empirical methods and found that root-mean-square error (RMSE) values were 7.56% in arid, and 45.65% in humid sites, Nash-Sutcliffe efficiency coefficient (NSE), and R2 value was up to 60% in summer. Then they evaluated the results of the ML classifiers and concluded that Support Vector Machine (SVM) and Random Forest (RF) classifiers were the most preferred in terms of overall performance. Although very challenging due to its nonlinearity, relative humidity prediction is another essential research dimension. Qadeer et al. [20] proposed an RF classifier algorithm-based ML model to estimate an air-based energy system's relative humidity (RH). Their study linked the Aspen HYSYS v10 process simulator with MATLAB 2019a to provide an environment necessary for data mining processes. They employed the dry and wet bulb thermometer to compare the results of their model, where an absolute deviation was observed. They further analogized the RF performance to an SVM and discovered a better overall performance of about 74.4%. Also, some researchers opted to exploit the high prediction accuracy of convolutional neural network (CNN) models. For instance, Jung et al. [21] used a CNN to monitor and regulate different inhabitants' indoor temperature and energy consumption. Their work used a 1D CNN to recognize inhabitants' activities and reinforce learning (RL) models for indoor temperature control. Their findings showed that the system automatically controlled the indoor temperature in real-time, with a 10.9% decrease in thermal discomfort and preserved energy consumption. Kreuzer et al. [22], however, used long short-term memory (LSTM) and convolutional LSTM (convLSTM) networks to enhance temperature forecasting compared to the traditional naive forecast approach. They used five different weather stations to put up a case study in Germany and observed that their models worked best in longer horizons prediction. After using numerous datasets to check the systems' performances, they deduced that convLSTM provided the best results.

In attempting to improve performance accuracy and efficiency in IoT devices, scores of researchers embedded LPWAN technologies with either machine learning (ML) or deep learning (DL) techniques. Parvez et al. [23] suggested a development architecture for the LoRa gateway communication technology by employing Artificial intelligence (AI) techniques. The study used the MySignals development shield, Arduino medical devices, and eHealth development platform to remotely monitor and maintain the sensor. Reduced complexity, minimum decision-making delay, and less deployment cost were the crucial findings of such integration. LPWAN technologies are specifically faced with payload size constraints compared to other wireless technologies. To minimize these issues, Bernard et al. [24] embedded a deep LSTM model in the LoRa sensor and network infrastructure to improve data transmission efficiency by using the occupancy dataset obtained from cellular base stations to train the model. They suggested a lossy compression technique, significantly reducing the gateway's sensor data traffic. They further analyzed the system's lossy compression error rate, energy consumption, and model performance, and the outcome was remarkable. Another noteworthy investigation was conducted by Ouameur et al. [25] that employed ML and DL models to measure and localize an indoor location by leveraging the LoRaWAN channel state information in a dataset depicting four different spots. The framework further provided entry to the sensor data and physical layer where information like the spreading factor, the receiver's signal strength, frequency hopping signature, etc., can also be accessed. They used a multilayer neural network to predict these locations with 98% accuracy. Wearable devices such as smartphones and glasses are becoming more and more integrated into our lives. Despite the immense benefits of monitoring peoples' fitness, health conditions, and so on, they are greatly limited by relying on smartphones and tablets for decision-making and data transmission to the internet. Sanchez-Iborra [26], in his work, reviewed and examined the aspects of embedding LPWANs (LoRaWAN) with tiny ML (TinyML). His research work further conducted an experimental study and investigated coverage and intelligence of the integration. The investigation eventually inferred that these paradigms fit the next generation of wearable devices. Loukatos et al. [27] used a combination of LoRa and Wi-Fi networks and remote cloud service-based voice recognition (VR) engines to enable smartphones to operate specific farming chores for the elderly and crippled remotely. They further used the embedded boards to facilitate offline VR in case of a shortage of internet which warranted remote connection to the VR servers. Also, the end nodes were able to reply by voicing the status of the current farming activity by playing a prerecorded message or via speech synthesis technique.

3. LORA AND LORAWAN

LoRa, short for long-range, is a Semtech-developed radio frequency modulation technology that allows exceptionally long-range data transmission for low-power and wide-area network (LPWAN) technologies. LoRa is a physical (PHY) layer for transceiver modules that use the Chirp Spread Spectrum (CSS) modulation to give great sensitivity and robustness against noise and interference. Figure 1 depicts a CSS modulation where a chirp, a sinusoidal signal whose frequency increases or decreases with time, is shown to increase with time, known as up-chirp. Conversely, if the chirp's frequency decreases with time, it is called a down-chirp [28]. Uplink channels have a set bandwidth of 125 or 500 kHz, whereas downlink channels have a fixed bandwidth of 500 kHz, resulting in a trade-off between sensitivity and data throughput. Lora's unique orthogonal spreading factors allow the network to provide adaptive power and data rate optimizations for each end device, extending the battery life of linked end devices [29].

As shown in Figure 2, LoRa provides the physical layer for the radio transceivers to transmit the modulated signals. However, LoRaWAN provides the medium access control (MAC) layer protocol, depicting how the end devices effectively communicate with the gateways (base stations).

They are created on top of LoRa and are managed by the LoRa Alliance company.

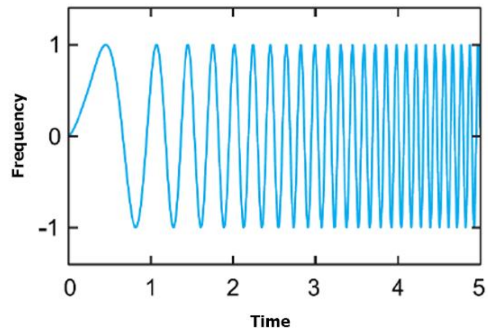

Figure 1. LoRa's chirp spread spectrum (CSS) [29]

Changing the spreading factor requires changing the modulation technique, which allows for a more extended transmission range but consumes more power. Because of the preceding, a standard, wholly supported eight-channel LoRa gate can accommodate a payload ranging from 11 to 242 bytes in size from up to 60,000 end devices when relaying sensor data over around 5 km in an urban setting and at least 15 km in rural settings.

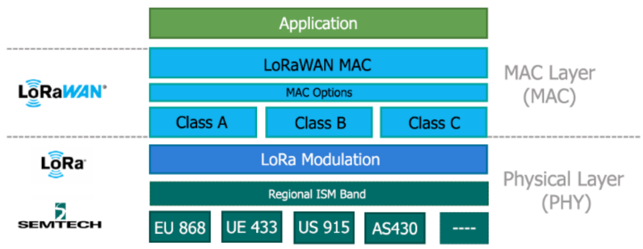

Figure 2. LoRaWAN protocol stack [29]

The MAC layer in LoRaWAN enables the communication between LoRa end nodes and gateways, sending sensor data to the cloud. LoRaWAN employs a star topology, meaning that end devices can only communicate with gateways and not with one another [30]. Simultaneously, multiple gateways can be connected to network servers, where application servers can be created for users to monitor and control the status of the payloads. As shown in Figure 2, there are three types of LoRaWAN operation, which are explained further below:

Class A: Because of their constant sleeping, devices in this category are the most energy-efficient. They only become active when sensors detect a new value and transmit an uplink message. During this brief window, the server can only send a downlink message.

Class B: Devices in this class use less energy than those in Class A. However, they can be scheduled to wake up, which generates a downlink message window and receives new sensor data.

Class C: Devices in this category use a lot of energy because they continually listen for downlink messages. Only while sending an uplink message that they listen to the uplink.

4. MATERIALS AND METHODS

This section provides information on the materials and methods used to develop the designed system. The system comprises two parts: the end device responsible for sensing changes in temperature and humidity and the gateway that receives the transmitted data from the end device. The gateway then encrypts and forwards the final packet to TTN. Both parts require software and hardware components to accomplish the mentioned task. The hardware provides the avenue for interacting with the physical environment and houses the MCUs and other essential system elements. The software contains the programs and algorithms that control the hardware through designated libraries and modules like the SPI modules for MCU to I/O units communication, radio head module for RF transceiver breakout configuration, and DHT library for controlling the DTH11 sensor, and many more.

4.1 Software requirement

Arduino IDE: The Arduino Integrated Development Environment (IDE) is open-source software that enables coding, debugging, and code upload to the Arduino physical board. The Arduino IDE is available for Windows, Linux, macOS, and even web versions for working online and is frequently updated for bugs and necessary changes [31]. This study utilized Windows version 1.8.19 of the Arduino IDE.

Raspberry Pi Imager: The Raspberry pi OS, formerly known as Raspbian, is a free Debian-based operating system improved for the Raspberry Pi system-on-chip hardware. It has more than 35,000 precompiled software bundles for use by the hardware [32]. However, the Raspberry Pi Imager is software that provides a fast and straightforward approach to installing the different versions of the Raspberry Pi OS on the microSD card, which is later used by the hardware [33].

PuTTY: PuTTY is also an open-source secure shell (SSH) and Telnet client software for reliable network services [34]. The software enables us to connect to the Raspberry Pi desktop environment via the headless connection method. PuTTY is also available for Windows and Linus OS.

Single Channel Packet Forwarder: A packet forwarder is a program that facilitates the transmission of uplink and downlink packet data between a concentrator (LoRa transceiver module) and the network server via IP/UDP communication protocol [35]. There are two packet forwarders types: single-channel packet forwarders and multi-channel packet forwarders. The implemented system used the single packet forwarder to develop a Raspberry Pi LoRa gateway. However, only the multi-channel packet forwarder is fully LoRaWAN compatible.

RadioHead Library: The RadioHead library is a radio packet library enabling the transceiver module to send and receive a message in a packet format through the embedded MCU [36]. It allows the RFM95W transceiver module on the end device side to transmit sensor data with the help of an embedded microprocessor, in this case, Arduino UNO.

Adafruit DHT11 Library: The DHT11 library is a simple Arduino library designed to control the low-cost TH sensor provided by Adafruit Industries (an open-source hardware company that designs and manufactures numerous electronic components and various products).

Arduino LMIC Library: Arduino LMIC library is a modified version of the IBM LMIC (LoraMAC-in-C) designed to operate in the Arduino environment. This library permits SX1272/76 transceivers and several RFM9x compatible transceiver modules for data transmission [37].

4.2 Hardware requirement

Raspberry Pi 3 B+: The raspberry Pi, 3 Model B+, is used in implementing the single-channel LoRa gateway. The Pi board is the final version of the third generation single-board computers manufactured by Raspberry Pi Foundation, a partner of Broadcom based in the United Kingdom. Raspberry Pi 3 B+ board is a powerful chip-based computer with several impressive capabilities and specifications. The board is built with a 1.4GHz 64-bit quad-core processor, faster Ethernet, dual-band wireless LAN, and Power-over-Ethernet (PoE) support. It also has 1GB RAM, a 4.2/BLE Bluetooth, 4 USB 2.0 ports, a 40 GPIO pin, full-size HDMI, a microSD port for OS, and data support, to name a few [38]. It is a fantastic board that is very affordable and useful in developing many stunning applications.

Arduino Uno: The second outstanding MCU unit employed is the Arduino Uno board. Arduino Uno board is a popular board that has been widely used to make many applications due to its simple design and functionality. It is widely considered the starting point for developing IoT applications. Arduino Uno is a 63.6 x 53.4 mm board designed with an Atmega328P microcontroller. It has 14 digital I/O pins (out of which six (6) offer Pulse-width modulation PWM) and six (6) analog input pins. Uno board has other features like 16 MHz clock speed, 2 KBstatic random-access memory (SRAM), and 1 KB electrically erasable programmable read-only memory (EEPROM), etc. Uno is derived from Italian, which means one, and it marks the first Arduino IDE version 1.0. [39].

Adafruit RFM95W LoRa Radio Transceiver Breakout 868 MHz: RFM95W LoRa radio module is the key component in this implementation. For long-distance wireless data transfer, two endpoints, typically a transmitter and receiver (or transceivers), are necessary to send and receive data. Two Lora radio modules are required to develop the system in the same light. These radio modules enable packetized data transmission between the end node and gateway, provided their radio modules are designed with the same frequency and use the same encryption key. They are also easily compatible with Arduino Uno and Raspberry Pi boards. Both use the same 868 MHz European license-free ISM bands following the LoRaWAN regional parameters with a simple wire antenna. The transceivers are based on the SX1276 LoRa module with the SPI interface capable of transmitting sensor data over about 2 Km depending on several factors such as antenna type, frequency, obstacle, and power out [40].

Whip or Wire Antenna: Antennas are essential components in wireless communication. They act as an interface between the transmitted radio waves and the electric current moving in the metallic conductors of the system. The LoRa radio modules facilitate using different antennas connection like the regular wire, u.FL or RP-SMA connection. This system used the wire antenna for cost-effectiveness and range up to 2 km [41]. The length of the antenna was computed using the quarter-wave whip antenna approach and was obtained by dividing the radio transceivers' wavelength by a value of 4, as given below.

$$\lambda = \frac{v}{f} \tag{1}$$

where, λ=Wavelength, v=Velocity, f=Frequency.

Here, the velocity of a radio signal in a vacuum is 299,792,458 m/s. And the frequency of the radio module is 868 MHz. Therefore, we can convert it to Hz by adding six zeros resulting in 868,000,000 Hz. Therefore,

$$\lambda = \frac{299{,}792{,}458}{868{,}000{,}000}, \lambda \approx 0.345\ m, \lambda \approx 34.5\ cm \quad (2)$$

After dividing the wavelength by 4, we obtained the desired length of the wire antenna as follows:

$$l = \frac{\lambda}{4}, l = \frac{34.5}{4}, l \approx 8.6\ cm \quad (3)$$

DHT11 TH Sensor: The DHT11 sensor is a low-cost, basic device widely employed in several long-term IoT prototyping for measuring TH in its surrounding environment. The component uses a capacitative humidity sensor and a thermistor to make these measurements. It contains an 8-bit chip for analog and digital signal conversion. The chip also emits the TH digital signal values, which are easily detected by the Arduino board. It can measure temperature between 0 to 50°C with a 2°C error margin. Also, with the DHT11, values ranging from 20 to 90% for the humidity measurement are detected with a 5% error margin. The DHT11 used in this design is a three-pin type, eliminating the need for a resistor. Other features of the DHT11 sensor include 3 to 5 Vpower with an I/O pin, 2.5 mA maximum current in requesting data, less than 1 Hz sampling rate, etc. [42].

Other Components: Other essential but straightforward components used in accomplishing the design include two half-size breadboards for testing, a 9 V battery power supply for the Arduino board, a 5 V regulated adapter, and a 16 GB microSD memory card for the Rasberry Pi board, soldering iron and lead, and 22 AWG solid-core hook up wires.

4.3 Simulation and breadboarding

As can be seen in Figure 3. the complete setup of the end device and gateway developed with fritzing design automation software was appropriately shown.

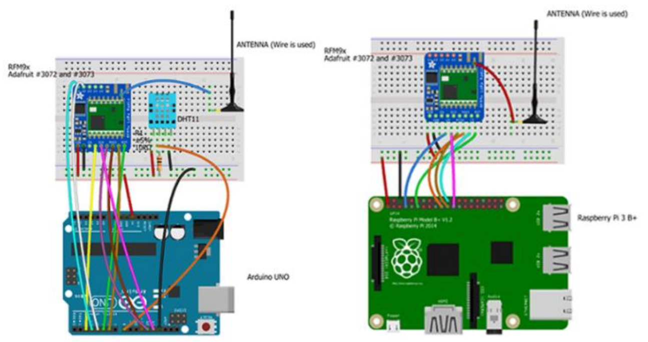

Figure 3. Breadboard designs of the end device and raspberry Pi 3 B+ LoRa gateway

The software provides a simple GUI for the full system realization. For the end device, all the required components, namely Arduino Uno board, DHT11 sensor, Adafruit RFM95W Lora radio module, 10 kΩ resistors, and antenna, were arranged and connected accordingly. The LoRa radio module and DHT11 libraries had to be imported into the software as they were initially unavailable. According to the design, all connections were ensured, and the system was set and ready for testing with the breadboard. The same approach was extended to the gateway setup, comprising only three components: Raspberry Pi 3 B+, Adafruit RFM95W Lora radio module, and antenna. The fritzing breadboard setups for the end device and gateway are ready for implementation.

4.4 System implementation and TTN registration

The Adafruit RFM95W Lora radio modules come with a set of 0.1-inch male header strips that need to be soldered to facilitate their usage with the breadboard or a universal printed circuit board (PCB) for a more concrete system realization. However, the universal PCB was not used in our cases as the complete soldering of the system is not required.

Table 1. Arduino Uno, RFM95W Radio Module, and DHT11 Wiring Connection

Arduino Uno	RFM95W	DHT11
2	G0	
3	G1	
4	G2	
5	RST	
6	CS	
8		DATA
11	MOSI	
12	MISO	
13	SCK	
5V		VCC
GND		GND

Table 2. Raspberry Pi and RFM95W Radio Module Wiring Connections

Raspberry Pi Board	RFM95W
1	VIN
6	GND
7	G0
11	RST
19	MOSI
21	MISO
22	CS
23	SCK

Only the male header strips were soldered to the radio module for proper deployment on the breadboard—Tables 1 and 2 outline the wiring connections for the end node and gateway and their respective components. The end device and gateway units were later configured after soldering the male headers to the radio modules and properly making the connections shown below. To configure the gateway for remote connection, the Raspberry Pi OS was written on the microSD card with the help of the imager. Afterward, the memory card is inserted into the Raspberry board to enable the desktop environment. There are many ways of accessing the Raspberry OS ranging from the headless connection via the SSH protocol to using software that allows virtualization. The board can also be directly connected to an external monitor through the HDMI port and can be controlled with a keyboard and monitor just like a regular computer. The approach used in this implementation is the headless connection approach to minimize cost. The PuTTY software facilitated the SSH headless connection. Then the Raspberry OS was updated through the command prompt, and the SPI function was allowed for communication between the board and its pins. Other requirements for the board are installing the wiringPi library and cloning the single-channel packet forwarder repository from Github, which can be found here [43]. The final actions for the gateway realization are setting the frequency of the radio module (868 MHz) and TTN network server IP (63.34.215.128) in the C++ file ("main.cpp") from the clone folder. These actions were essential for the remote connection to TTN.

After setting up the gateway and making the end device ready, they were registered on TTN for online access. An account had to be created to enable the gateway, application registration, and further configuration and can be found here [44]. Upon completion of the above process, the network session key (NtwkSKey), application session key (AppSKey), and device address (DevAddr) were taken for activation. There are two activation methods on TTN: over-the-air activation (OTAA), which automatically activates the devices, and activation by personalization (ABP). The required details are manually typed in the end device program for successful packet transmission from the gateway to TTN. This design employed the ABP activation method, and the three necessary parameters NTWKSK, APPSK, and DEVADDR were used in the code. Using the Arduino IDE, the Arduino-LMIC code [45] was updated with the above parameters and other necessary details such as the Arduino activation code for the DHT11 sensor, the pin mapping for the RFM95W LoRa radio module, and the end node frequency. The updated program was compiled and uploaded to the Arduino board, which sent the LoRaWAN packet with TH sensor values as payload to TTN with a matching frequency and encryption. After finally completing the system realization, the Raspberry pi gateway successfully sent the encrypted uplink message. However, to see the actual sensor value, TTN provides several payload formatters like JavaScript, Cayenne low power payload (LPP), etc., to decrypt the payload into human-readable formats just like in the Arduino IDE serial monitor. As simple as it seems, this task was challenging due to so many changes and recent updates in TTN. However, despite such challenges, the payload was finally decrypted using the JavaScript payload formatters option.

5. RESULTS AND DISCUSSION

This section captures all the experimental results and necessary discussions about the complete design and implementation of the LoRa-based THMS with a single channel Raspberry Pi 3 B+ gateway and Arduino Uno board end device. Figure 4 shows the respective TH values of 22.00°C and 33.00%, obtained from the Arduino IDE serial monitor. These are the values detected by the DHT11 sensor from its current surrounding environment. The values are displayed in 115200 baud of the serial monitor after 30 seconds, set earlier in the Arduino-LMIC program source code. The serial monitor started with a "Starting" welcome message. However, this message can be changed to any wording of choice. It does not affect the program execution. After every uplink message, the packet is queued and waiting for the network server to initiate a connection before transmission.

After the network successfully receives the uplink message, the device listens for any possible downlink message before it goes back to sleep, which is a technique that improves the battery life span. The above process is called the receiving window and is denoted by Rx and usually takes between 1 to 2 seconds. The downlink is simply an acknowledged message in case of confirmed delivery. The obtained result in the figure clearly shows the functionality and performance of the Arduino Uno-based end device in sensing TH values.

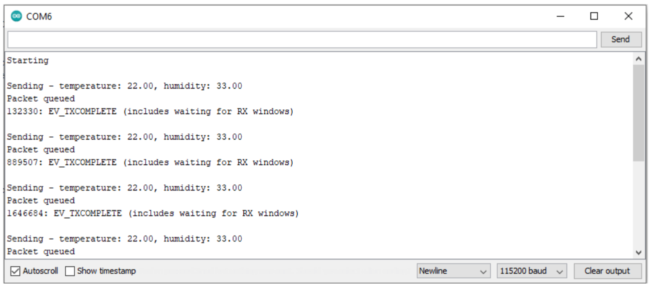

Figure 4. TH values detected by the end device in Arduino IDE serial monitor

Moreover, the results from Figure 5 show the online uplink messages on TTN. From the time section on the left side of the uplink messages on the application page, the application server received the uplink message at about 36 to 37 seconds intervals. Compared to the initial 30 seconds in the end device reception interval, we can observe a 6 to 7 seconds delay in the timing.

Figure 5. TH values on the TTN application server

The difference in time is reasonable when the network strength, encryption-decryption delay, and the complete data transmission processes are considered. For sensor data to be successfully sent to the application server, it has to travel through the device, the gateway, the network server, and finally, the application server. Despite all the observations, the results clearly show the exact sensed data, which is now on the internet and can be distantly accessed from every device with network access ranging from mobile phones, tablets, or computers.

Furthermore, Figure 6 presents the Raspberry Pi gateway ID and the status of the single-channel packet forwarder. The program detected the Semtech SX1276 chipset on which the RFM95W radio module was built. LoRa uses six spreading factors (SF7 – SF12) to enhance connectivity and transmission. The Lora radio module uses the SF7 on the 868MHz frequency. The SF7 is the ideal SF for regular use due to its improvements in features. A high SF number means increasing the transmission range but reducing payload size and increasing the energy consumption. Also, the packet data spends more time in the air making it more susceptible to attack.

The gateway ID is very relevant because it is used to register the gateway on TTN. It serves as the online gateway device identifier. In Figure 7, the gateway is connected and broadcasting to TTN. We can observe that the gateway IDs are the same for the sender (the gateway) and receiver (the online network server).

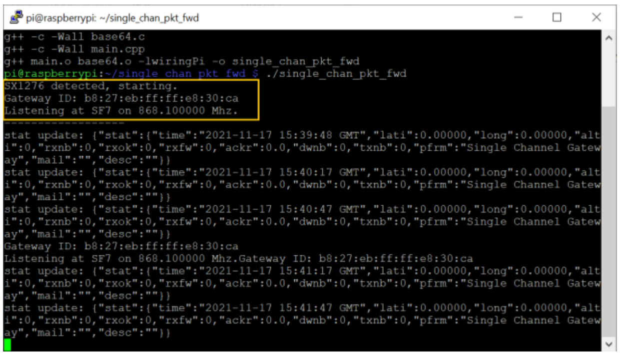

Figure 6. The gateway ID and the status of the single-channel packet forwarder

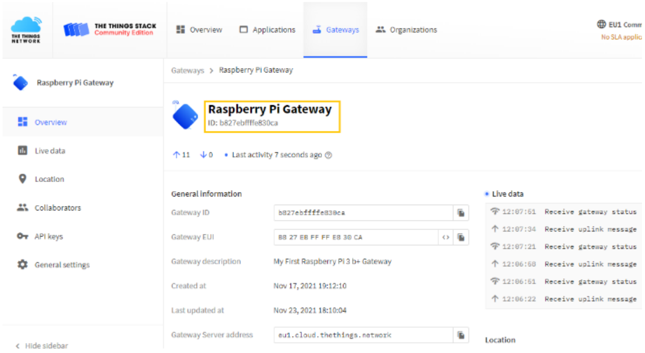

Figure 7. The connected gateway on TTN

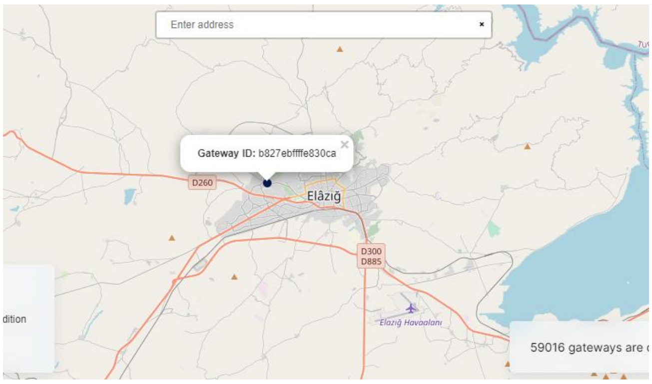

Figure 8. The map of the Raspberry Pi gateway on TTN

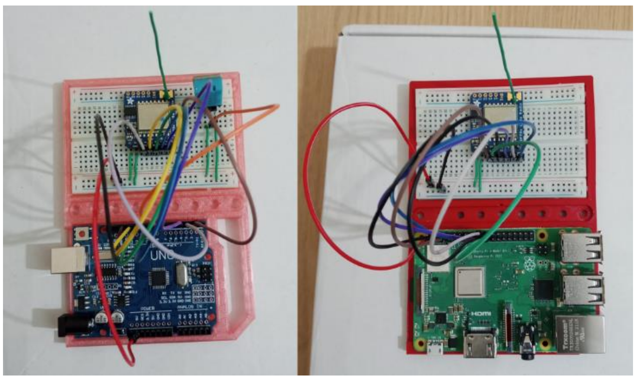

Figure 9. Complete end device and gateway implementation

TTN provides a map for tracking the number of registered gateways on its website using the gateway ID. This way, all gateways operating within the geographic location can be identified. The implemented Raspberry Pi gateway in this system was also seen on the map with its exact ID. To the best of our knowledge, this gateway is the first deployed LoRa gateway in the whole of Elazığ, with very few others in Turkey. Figure 8. shows the map with the only registered gateway below. Furthermore, the complete system implementation consisting of the end node and gateway is presented in Figure 9.

Nevertheless, the successful realization of the proposed system is not without challenges and limitations. The first limitation of the system is related to the use of the single-channel packet forwarder in transmitting data to TTN. The single-channel packet forwarder is not fully LoRaWAN compatible as it only sends the packet on a single channel. However, LoRaWAN supports data transmission in eight (8) channels. The system can reliably operate on earlier TTN versions, but it has less support on the new TTN version 3, affecting reliable data transfer and overall performance. Also, developers of wiringPi have deprecated the library due to the reduced usage of the C and BASIC programming language that it was developed to operate. Many people are turning to other programming languages such as Python, Java, etc. Because of this, only the old version of the library is available and can pose security challenges due to a lack of updates. In the future, LoRa-based projects are recommended to be deployed on the 8-channel gateways for full support, reliable data transmission, and improved performance. In addition, a severe challenge was encountered regarding the headless connection between the Raspberry board and wireless network while developing the Raspberry Pi gateway. The gateway successfully connected to the network initially, after which the network details must be manually inserted in the boot folder every time the board is restarted. This issue was overcome by changing the regular class 10, 16 GB microSD memory card with a better one (32 GB Samsung Evo Plus microSD card class 10).

6. CONCLUSIONS

This paper proposed a design and implementation of a THMS based on LoRa and LoRaWAN using a Raspberry Pi gateway with an Arduino-based end device to monitor the TH values using LoRa radio modules. The sensor data was later transmitted to the TTN network for remote access. The TH values of 22.0°C and 33.0% displayed by the Arduino serial monitor and the TTN application server clearly show that sensor values can be effectively transmitted with the LoRaWAN over long distances with minimal power consumption. The Raspberry Pi implementation of the gateway highly reduced the cost of unnecessary gateway deployment. Therefore, this paper can assist many researchers with sufficient knowledge to build many projects based on the LPWAN technologies. It also contributed to using cost-effective components to deploy long-range and low-power devices on the internet, which provide IoT solutions to the problem of TH monitoring. For future research using the LPWAN technologies, we recommend the deployment of fully compatible gateways that support 8-channel frequency transmission. Doing this will minimize any potential problems like security breaches, loss in data transmission, etc., due to the level of support and regular upgrade they receive from the manufacturers and TTN. Moreover, the use of compatible microprocessors in developing the end nodes is highly recommended to eliminate the need for manual registration on the application server. Manual registration of end devices may lead to the wrong insertion of the device details and MAC version.

REFERENCES

[1] Xiao, J., Li, J.T. (2020). Design and implementation of intelligent temperature and humidity monitoring system based on ZigBee and WiFi. Procedia Computer Science, 166: 419-422. http://dx.doi.org/10.1016/j.procs.2020.02.072

[2] Baykara, M., Abdullah, S. (2020). Designing a securable smart home access control system using RFID cards. Journal of Network Communications and Emerging Technologies (JNCET), 10(12).

[3] Gherbi, C. (2021). Internet of things and heterogeneous networks technologies: Concepts, challenges and perspectives. Ingénierie des Systèmes d'Information, 26(4): 403-408. https://doi.org/10.18280/isi.260408

[4] Izang, A.A., Ajaegbu, C., Ajayi, W., Omotunde, A.A., Enike, V.O., Ifidon, B.O. (2022). Radio frequency identification based student attendance system. Ingénierie des Systèmes d'Information, 27(1): 111-117. https://doi.org/10.18280/isi.270113

[5] Salleh, A., Ismail, M.K., Mohamad, N.R., Abd Aziz, M.A., Othman, M.A., Misran, M.H. (2013). Development of greenhouse monitoring using wireless sensor network through ZigBee technology. International Journal of Engineering Science Invention, 2(7): 6-12.

[6] Nasution, T.H., Muchtar, M.A., Seniman, S., Siregar, I. (2019). Monitoring temperature and humidity of server room using Lattepanda and ThingSpeak. Journal of Physics: Conference Series, 1235(1): 012068. http://dx.doi.org/10.1088/1742-6596/1235/1/012068

[7] Polonelli, T., Brunelli, D., Bartolini, A., Benini, L. (2018). A lorawan wireless sensor network for data center temperature monitoring. In International Conference on Applications in Electronics Pervading Industry, Environment and Society, pp. 169-177. http://dx.doi.org/10.1007/978-3-030-11973-7_20

[8] "Sensmax," "Temperature and humidity monitoring for greenhouses." https://sensmax.eu/solutions/temperature-and-humidity-monitoring-for-greenhouses/, accessed on Apr. 15, 2022.

[9] "Sony," "What is the maximum communication range of the Bluetooth connection? | Sony USA." https://www.sony.com/electronics/support/articles/00014085, accessed on Apr. 15, 2022.

[10] Gu, F., Niu, J., Jiang, L., Liu, X., Atiquzzaman, M. (2020). Survey of the low power wide area network technologies. Journal of Network and Computer Applications, 149: 102459. http://dx.doi.org/10.1016/j.jnca.2019.102459

[11] "Lora Developers," "LPWAN Technologies | DEVELOPER PORTAL." https://lora-developers.semtech.com/documentation/tech-papers-and-guides/lpwan-technologies, accessed on Dec. 07, 2021.

[12] "LoRa Alliance," "RP2-1.0.3 LoRaWAN® Regional Parameters - LoRa Alliance®." https://lora-alliance.org/resource_hub/rp2-1-0-3-lorawan-regional-parameters/, accessed on Dec. 08, 2021.

[13] "Makerdemy," "Resources - Makerdemy," Jan. 23, 2020. https://makerdemy.com/resources/, accessed on Dec. 22, 2021.

[14] Gao, Z., Jia, Y., Zhang, H., Li, X. (2012). A design of temperature and humidity remote monitoring system based on wireless sensor network technology. In 2012 International Conference on Control Engineering and Communication Technology, pp. 896-899. http://dx.doi.org/10.1109/ICCECT.2012.223

[15] Guan, W., Wang, C., Cai, Y., Zhang, H. (2016). Design and implementation of wireless monitoring network for temperature-humidity measurement. Journal of Ambient Intelligence and Humanized Computing, 7(1): 131-138. http://dx.doi.org/10.1007/s12652-015-0314-7

[16] "Britannica," "integrated circuit - Microprocessor circuits | Britannica." https://www.britannica.com/technology/integrated-circuit/Microprocessor-circuits#ref837117, accessed on Dec. 23, 2021.

[17] Chang, C.Y., Hung, S.S. (2012). Implementing RFIC and sensor technology to measure temperature and humidity inside concrete structures. Construction and Building Materials, 26(1): 628-637. http://dx.doi.org/10.1016/j.conbuildmat.2011.06.066

[18] Wang, Y., Chi, Z. (2016). System of wireless temperature and humidity monitoring based on Arduino Uno platform. In 2016 Sixth International Conference on Instrumentation & Measurement, Computer, Communication and Control (IMCCC), pp. 770-773. http://dx.doi.org/10.1109/IMCCC.2016.89

[19] Bellido-Jiménez, J.A., Gualda, J.E., García-Marín, A.P. (2021). Assessing new intra-daily temperature-based machine learning models to outperform solar radiation predictions in different conditions. Applied Energy, 298: 117211. http://dx.doi.org/10.1016/j.apenergy.2021.117211

[20] Qadeer, K., Ahmad, A., Qyyum, M.A., Nizami, A.S., Lee, M. (2021). Developing machine learning models for relative humidity prediction in air-based energy systems and environmental management applications. Journal of Environmental Management, 292: 112736. http://dx.doi.org/10.1016/j.jenvman.2021.112736

[21] Jung, S., Jeoung, J., Hong, T. (2022). Occupant-centered real-time control of indoor temperature using deep learning algorithms. Building and Environment, 208: 108633. http://dx.doi.org/10.1016/j.buildenv.2021.108633

[22] Kreuzer, D., Munz, M., Schlüter, S. (2020). Short-term temperature forecasts using a convolutional neural network—An application to different weather stations in Germany. Machine Learning with Applications, 2: 100007. http://dx.doi.org/10.1016/j.mlwa.2020.100007

[23] Parvez, F.A., Chaitanya, J., Yara, T. (2021). A development architecture of Lora gateway communication technology using artificial intelligence. Materials Today: Proceedings. http://dx.doi.org/10.1016/j.matpr.2021.01.747

[24] Bernard, A., Dridi, A., Marot, M., Afifi, H., Balakrichenan, S. (2021). Embedding ML algorithms onto LPWAN sensors for compressed communications. In 2021 IEEE 32nd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), pp. 1539-1545. http://dx.doi.org/10.1109/PIMRC50174.2021.9569714

[25] Ouameur, M.A., Caza-Szoka, M., Massicotte, D. (2020). Machine learning enabled tools and methods for indoor localization using low power wireless network. Internet of Things, 12: 100300. http://dx.doi.org/10.1016/j.iot.2020.100300
[26] Sanchez-Iborra, R. (2021). LPWAN and embedded machine learning as enablers for the next generation of wearable devices. Sensors, 21(15): 5218. http://dx.doi.org/10.3390/s21155218
[27] Loukatos, D., Fragkos, A., Arvanitis, K.G. (2021). Exploiting voice recognition techniques to provide farm and greenhouse monitoring for elderly or disabled farmers, over Wi-Fi and LoRa interfaces. In Bio-Economy and Agri-production, pp. 247-263. http://dx.doi.org/10.1016/B978-0-12-819774-5.00015-1
[28] "Mobilefish.com - LoRa/LoRaWAN tutorial." https://www.mobilefish.com/developer/lorawan/lorawan_quickguide_tutorial.html, accessed on Jul. 15, 2022.
[29] Semtech. (2020). LoRa and LoRaWAN. Semtech Tech. Pap., no. December 2019, pp. 1-17. https://lora-developers.semtech.com/library/tech-papers-and-guides/lora-and-lorawan/, accessed on Dec. 28, 2021.
[30] Saban, M., Aghzout, O., Medus, L.D., Rosado, A. (2021). Experimental analysis of IoT networks based on LoRa/LoRaWAN under Indoor and outdoor environments: Performance and limitations. IFAC-Papers OnLine, 54(4): 159-164. http://dx.doi.org/10.1016/j.ifacol.2021.10.027
[31] "Arduino IDE," "Software | Arduino." https://www.arduino.cc/en/software, accessed on Dec. 28, 2021.
[32] "Raspberry Pi OS," "Raspberry Pi Documentation - Raspberry Pi OS." https://www.raspberrypi.com/documentation/computers/os.html, accessed on Dec. 28, 2021.
[33] "Raspberry Pi," "Raspberry Pi OS – Raspberry Pi." https://www.raspberrypi.com/software/, accessed on Dec. 28, 2021.
[34] PuTTY, "Download PuTTY - a free SSH and telnet client for Windows." https://www.putty.org/, accessed on Dec. 28, 2021.
[35] "Semtech Lora-net," "Lora-net/packet_forwarder: A LoRa packet forwarder is a program running on the host of a LoRa gateway that forwards RF packets receive by the concentrator to a server through a IP/UDP link, and emits RF packets that are sent by the server." https://github.com/Lora-net/packet_forwarder, accessed on Dec. 28, 2021.
[36] "Radio Head," "RadioHead: RadioHead Packet Radio library for embedded microprocessors." https://www.airspayce.com/mikem/arduino/RadioHead/, accessed on Dec. 28, 2021.
[37] "Arduino LMIC," "matthijskooijman/arduino-lmic: This library is deprecated, see the README for alternatives." https://github.com/matthijskooijman/arduino-lmic, accessed on Dec. 29, 2021.
[38] "Raspberry Pi," "Buy a Raspberry Pi 3 Model B+ – Raspberry Pi." https://www.raspberrypi.com/products/raspberry-pi-3-model-b-plus/, accessed on Dec. 29, 2021.
[39] "Arduino Uno," "Arduino Uno Rev3 — Arduino Official Store." https://store.arduino.cc/products/arduino-uno-rev3/, accessed on Dec. 29, 2021.
[40] "Adafruit LoRa." "Adafruit RFM95W LoRa Radio Transceiver Breakout - 868 or 915 MHz [RadioFruit]: ID 3072: $19.95: Adafruit Industries, Unique & fun DIY electronics and kits." https://www.adafruit.com/product/3072, accessed on Dec. 29, 2021.
[41] Seneviratne, P. (2019). Introduction to LoRa and LoRaWAN.
[42] "Adafruit DHT11," "Overview | DHT11, DHT22 and AM2302 Sensors | Adafruit Learning System." https://learn.adafruit.com/dht, accessed on Dec. 29, 2021.
[43] Github SCPF. "tftelkamp/single_chan_pkt_fwd: Single Channel LoRaWAN Gateway." https://github.com/tftelkamp/single_chan_pkt_fwd, accessed on Dec. 29, 2021.
[44] The Things Network. "Overview - Console - The Things Network." https://eu1.cloud.thethings.network/console/, accessed on Dec. 29, 2021.
[45] Arduino LMIC. https://github.com/matthijskooijman/arduino-lmic/blob/master/examples/ttn-abp/ttn-abp.ino, accessed on Dec. 29, 2021.