# An Approach to Design a Multi-Protocol Gateway Device for Internet of Things System

Thanh Bang Le*+, Hoang-Gia Vu*, Hai Duong Nguyen* and The Son Vu*

*Department of Microprocessor Engineering*
*Faculty of Radio-Electronics*
*Le Quy Don Technical University*
Hanoi, Vietnam
+Corresponding author email: banglt@lqdtu.edu.vn

*Abstract—* **Due to the expansion of the Internet of Things (IoT) system in both the number of devices and the connection technologies, new extremely high criteria for data processing speed, bandwidth and security have been developed. One of the solutions to achieve those requirements is to use a device that integrates multiple communicating protocols (IoT gateway) at the edge of the IoT system. The paper introduces an approach to design this device based on an embedded computing platform that connects different protocols of ZigBee and LoRa sensor networks. The experimental results have proved the feasibility of the proposed method.**

*Keywords—IoT gateway, ZigBee, LoRa, IoT system*

## I. INTRODUCTION

The rapid-expanding of the Internet of Things (IoT) system nowadays leads to an increasing demand for connectivity between sensor devices using different communicating protocols such as Bluetooth, NFC, RFID, Wi-Fi, ZigBee, LoRa…into one integrated sensor network. There are many solutions for combining these networks in which one approach is to employ a gateway device that connects multi-communication protocols at the edge of the IoT system. This device acts as a bridge (or IoT gateway) between IoT sensor nodes and the Internet environment. In this case, if data are intelligently aggregated and analyzed at the edge of network, the amount of forwarding data to the cloud computing will be notably reduced. As the results, the response time and network cost will also be optimized. Therefore, the requirement for IoT gateway devices to be researched, designed, and manufactured is always a top priority.

In recent years, there have been many studies related to hardware design for IoT gateway devices based on different platforms such as embedded platforms [1], [2], [3], software-defined radio [4], or FPGA technology [5]. The typical trend of designing is to combine multi-communication protocols from different sensor networks into a common protocol located in a single device and take advantage of cloud computing to process the collected data [6], [7], [8]. In [9], the authors implemented a multi-protocol gateway that combined Wi-Fi/BLE and ZigBee into one sensor network for smart home and building automation applications. A hybrid architecture is suggested for real-time monitoring of oil pipelines using 2.4 GHz-based ZigBee and LoRa communication with a cloud server in [10]. Although LoRa and ZigBee adopt distinct modulation techniques, cross-technology communication is developed to exchange data between two networks [11]. However, due to the heterogeneity and complexity characteristics of the IoT system, the designing platform is selected highly dependent on each application to be deployed. To support research and educational training purposes, the paper presents an approach to implementing the IoT gateway device focused on the integration function for the ZigBee/ IEEE 802.15.4 and LoRa sensor networks using an embedded computer platform. The experimental results of the design proved the feasibility of the proposed solution.

The rest of the paper is organized as follows: Section II introduces the architecture of the IoT gateway device in the IoT system. The suggested method and testing results in designing are shown in Section III. Finally, the conclusion is presented in Section V.

## II. THE IoT GATEWAY ARCHITECTURE IN THE IoT SYSTEM

In general model of the IoT system, sensing data are transmitted to the cloud computing center for further analysis and processing, and then forwards the results to the user's terminal as shown in Figure 1.
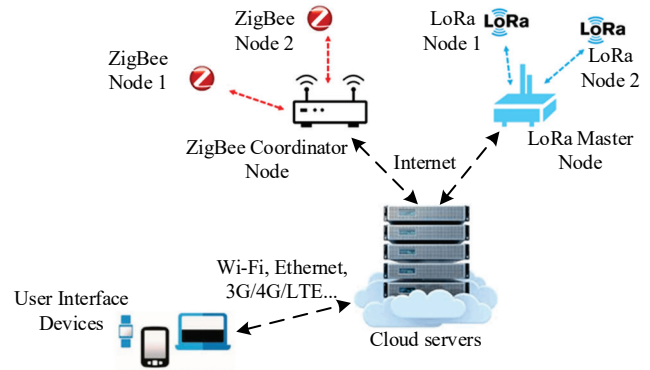


Fig. 1. The model of data center using cloud computing for two ZigBee and LoRa sensor networks.

In Fig.1, although cloud servers are capable of performing a wide range of processing tasks, they are sometimes placed in very remote locations from end devices, which might cause transmission delays. In addition, as more and more IoT devices are connected recently, the data sent to cloud computing centers are increasingly dense. As consequently, the transmitting bandwidth is significantly consumed, yielding a slow speed and a "bottleneck" situation. To overcome these issues, edge computing solution is introduced. The method refers to the aspect of computing infrastructure that locates close to the data source, which means that essential data are processed at the source instead at the centralized cloud servers. This approach allows the bandwidth of the system not to be congested when the number of data suddenly increases and thus the data transfer rate will be improved. In the edge computing model, the IoT Gateway device is responsible for providing local connection standards such as NFC, BLE, Wi-Fi, ZigBee, LoRa, 3G/4G LTE…Therefore, the basic functions of the IoT Gateway device may include: protocols conversion, pre-processing data, local storage, network connection to the data center.

Figure 2 shows the edge computing model associated with the IoT Gateway device.
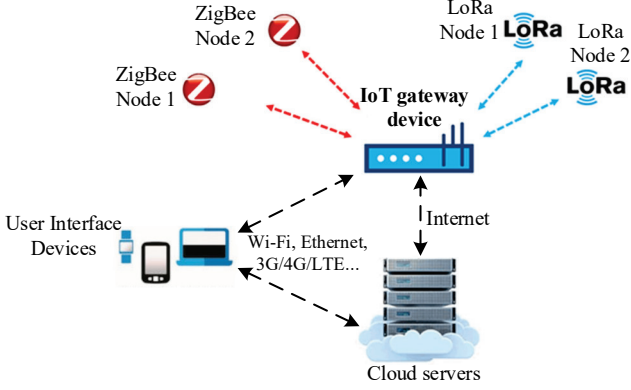


Fig. 2. Edge computing model combined with IoT Gateway device for ZigBee and LoRa sensor networks.

The reference architecture for IoT gateway devices has not yet been specified as a common standard currently. Each manufacturer defines this architecture in terms of hardware and software based on the application. Figure 3 depicts an IoT gateway architecture in terms of functional modules.
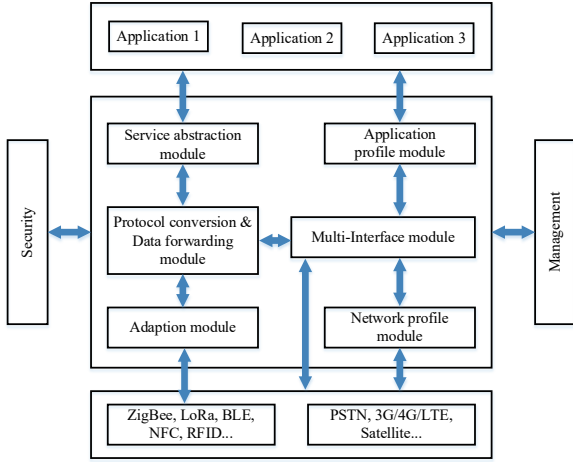


Fig. 3. The reference architecture of the IoT Gateway device [12].

## III. THE APROACH TO DESIGN AN IOT GATEWAY DEVICE

As previous described in Section II, IoT gateway devices play an important role at the edge of the IoT system. Section III presents an approach to design this device which integrates the communication protocols between ZigBee/IEEE 802.15.4 and LoRa sensor networks. The proposed model is shown in Figure 4.
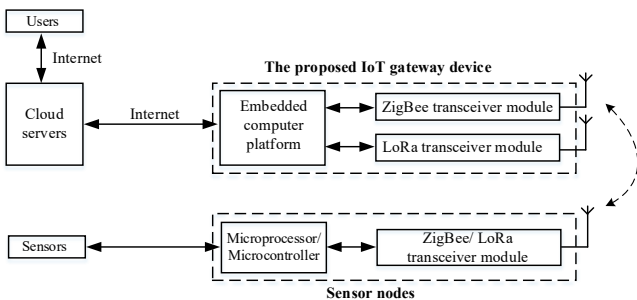


Fig. 4. The model of the proposed IoT gateway device in the IoT system.

According to the introduced model, the IoT gateway device performs the task of collecting data from both LoRa and ZigBee networks, arranging and then encapsulating data to transmit to database over the internet environment. The conversion function of the device is demonstrated through the exchange of data from LoRa, ZigBee protocols to Ethernet/ Wi-Fi protocols (TCP/IP or UDP) connected to the cloud server. The design process includes the following steps: creating two sensor networks, implementing the IoT device, building database, designing website and setting the appropriate warnings for the users.

### A. Implementation of the ZigBee sensor network

The ZigBee sensor network is constructed on a ZigBee coordinator node and two ZigBee end-device nodes in a star topology. Each sensor node is designed on an experimental board as shown in Figure 5. This board contains an ARM microprocessor (BeagleBone Black), a ZigBee CC2530 transceiver module and a DHT11 sensor.
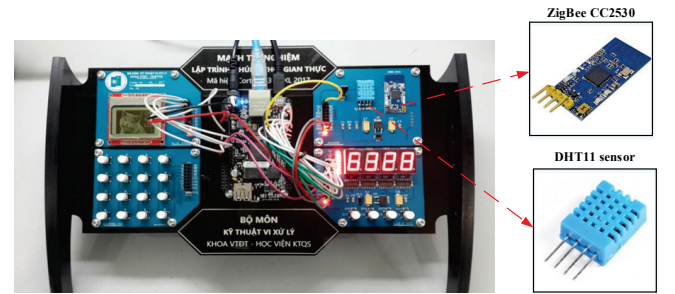


Fig. 5. Structure of the ZigBee sensor node.

The ZigBee coordinator node in this study is built on the embedding computer platform (Raspberry Pi). To transmit data in the ZigBee network, the time multiplexing method is utilized. This approach aims to reuse frequencies, save computing resources and extend the end-devices in the simple way. To implement this method, it is required the nodes must be synchronized with each other by updating their current time using the NTP (Network Time Protocol) protocol with an accuracy of $10\ ms$. Therefore, a time frame will be divided into time slots in which one slot and the guard interval is equal to one second as can be illustrated in Figure 6.
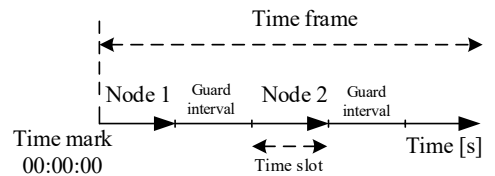


Fig. 6. The time multiplexing method applied to the ZigBee network.

The current time of a sensor node is then determined in the time frame by the following equation:

$$T_n = t_0 \ mod(m * t_d) \qquad (1)$$

where $mod(.)$ is the modulo operator; $T_n$ is the current time of the examined node $n$; $t_0$ is the time relative to the time mark 00:00:00; $m$ is total number of time slots and $t_d$ is the size of a time slot.

In case the value of $T_n$ is in the range calculated by the equation (2), the node $n$ will be ready to transmit its collected data to the ZigBee coordinator.

$$\begin{cases} T_n \geq t_d * (n-1) \\ T_n \leq t_d * (n-0.6) \end{cases} \quad (2)$$

where $n$ is the node ID (Identifier) of the examined node.

The data frame format contains a node ID field, a humidity value field, a temperature value field, and a null character to indicate the end of the transmitted string. This structure is depicted in Figure 7.

| Node ID | Humidity | Temperature | \0 |
|---------|----------|-------------|-----|

Fig. 7. The structure of the data frame format in the ZigBee network.

To transmit the collected data in the designed network, the following algorithm is applied to the sensor nodes.
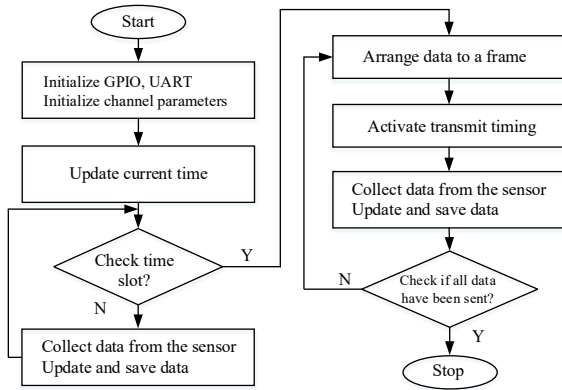


Fig. 8. The transceiving algorithm applied to the ZigBee sensor node.

The ZigBee coordinator then receives sensor data from end-device nodes as shown in Figure 9.



Fig. 9. The receiving data is observed at the ZigBee coordinator node (Raspberry Pi).

### B. Implementation of the LoRa sensor network

The star topology is utilized in the LoRa network in which all sensor nodes are arranged around and exchange data with the master node as depicted in Figure 10.
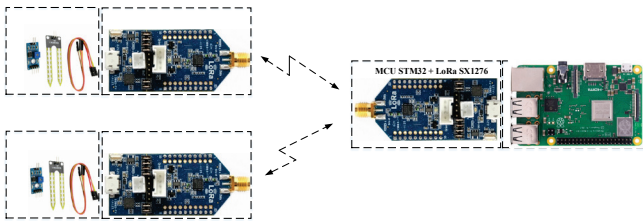


Fig. 10. Star network model applied to the LoRa sensor network.

Each LoRa node contains a soil moisture sensor and an environmental sensor DS18B20. The MCU STM32L07x initially collects the sensor data in analogue form via GPIO pins, performs the analogue-to-digital conversion and then transmits data to the master node via the LoRa SX1276 transceiver module [13].

In the introduced LoRa network, the polling method is employed for transmitting data purposes in which the master node sequentially sends the ID values of the component nodes. In case one node receives the data frame, it will compare this ID to the pre-agreed identifier, if it is correct, this node will transmit its data to the master node. This process is described as follows:
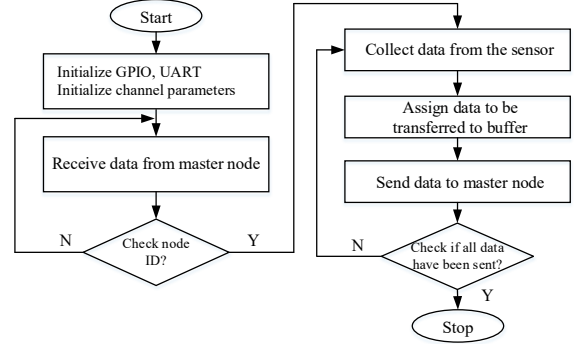


Fig. 11. The data processing flowchart in the LoRa sensor network.

The structure of data frame which exchanges between the master node and the sensor node is shown as follows:

| Node ID | Integer part of temperature value | Fractional part of temperature value | Soil moisture value |
|---------|-----------------------------------|--------------------------------------|---------------------|

Fig. 12. The structure of the data frame format in the ZigBee network.

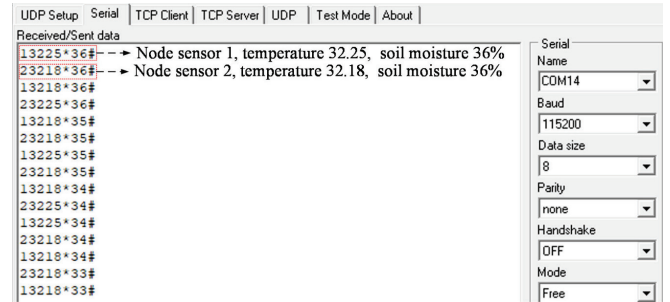Figure 13 depicts the observed data at the LoRa master node.



Fig. 13. The sensing data observed at the LoRa master node.

### C. Implementation of the IoT gateway device

In recent studies, there are multiple techniques to design the IoT gateway device depending on which IoT application to be deployed. For the research and educational purposes, the embedded computer platform is chosen due to the following reasons: Embedded computer is more cost-effective than a multi-function computer; Its processing performance is relatively high and suitable for specialized applications; It supports multiple communication standards such as GPIO, RS-232, USB, SPI, BLE, Ethernet, Wi-Fi...It has a compact size. As a result, these platforms are widely applied in both wireless sensor networks and the IoT systems. There are several types of embedded computers in use today. In this study, the Raspberry Pi Model 3B Plus is selected to implement the IoT gateway device, which carries out the

function of converting communication protocols for two wireless sensor networks as shown in Figure 14. Both ZigBee CC2530 and LoRa SX1276 modules are connected directly to this device. Therefore, it performs as a coordinator node for the ZigBee network and the master node for the LoRa network concurrently. The technical characteristics of module Raspberry Pi Model 3B Plus are described according to the document [14].
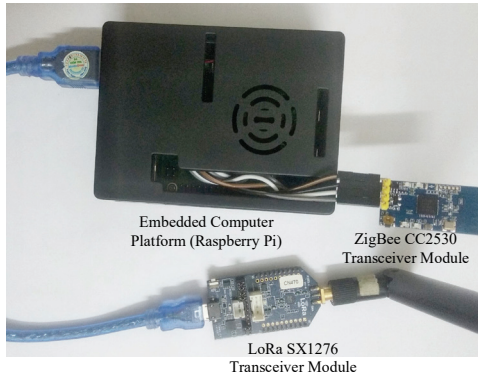


Fig. 14. The embedded computer platform (Raspberry Pi) for designing the IoT gateway device.

To process data received from ZigBee and LoRa networks, this device is programmed by using multi-thread solution. This technique allows two data streams to be received and processed independent of each other at the same time as described as the algorithm in Figure 15.
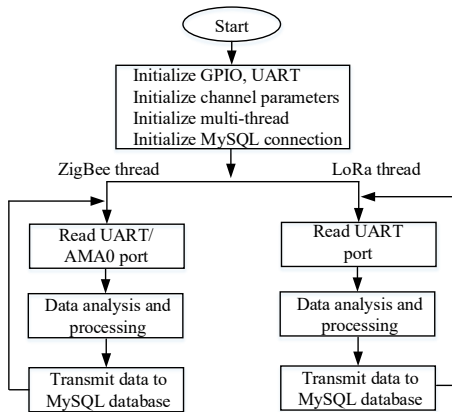


Fig. 15. Data processing algorithm in the suggested IoT gateway device.

The collected data are processed firstly at the designed device and then sent to the database (MySQL) located on the cloud server for further analysis and processing purposes. The structure of data frame is described as following figure.

| Node ID | Humidity | Temperature | Longitude | Latitude |
|---------|----------|-------------|-----------|----------|

Fig. 16. The structure of data frame sent from IoT gateway device to MySQL database.

As shown in Figure 16, alongside the previously specified node ID, humidity, temperature fields, there are also latitude and longitude fields assigned to sensor nodes in the data frame structure for the network management purpose. However, these coordinate fields are optional parameters at the current stage of this study, which are entered into the database to check the functionality of the website prior attaching the geolocation module to the sensor node. Figure 17 shows the collected data from two ZigBee nodes and two LoRa nodes at the IoT gateway device.
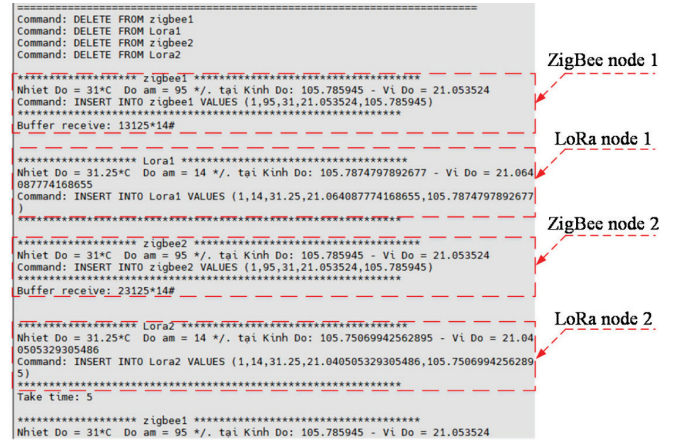


Fig. 17. The collected data at the IoT gateway device after integrating two ZigBee and LoRa networks.

The results of data processing and analyzing then will be shown to the users through the web interface. Therefore, the website is designed with the following functions: Displaying the received data; Analyzing statistical parameters of the data; Sending email alerts; Locating the sensor nodes on the map. Figure 18 illustrates the algorithm for updating the database and mechanism to detect whether the temperature exceeds the threshold (for example $40^0C$), then automatically sends an email to alert the user:
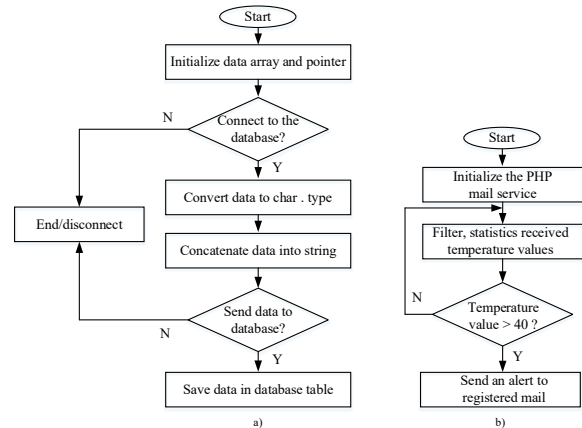


Fig. 18. The algorithm a) updates database and b) automatically sends warning emails to the user.

Figure 19 shows the data collected from two ZigBee and LoRa sensor networks on the website.
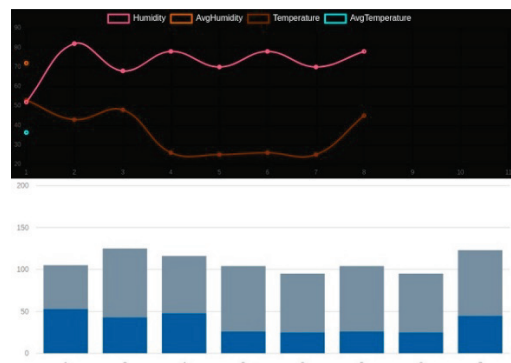


Fig. 19. The obtained data are shown in grid and column format on the website dashboard.

In case the temperature value exceeds the set threshold, the system will automatically send an email alert to the user.
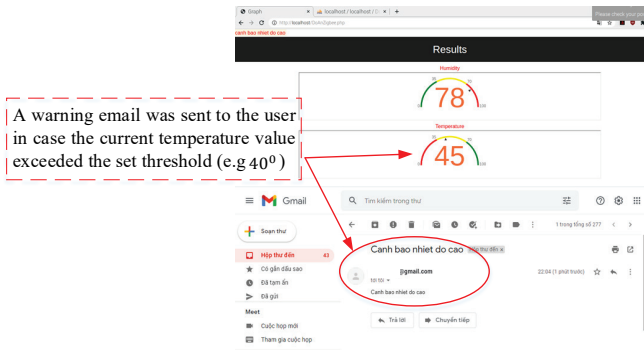


Fig. 20. The mechanism to detect temperature exceeding a set threshold and send an email alert to the user.

The testing results have shown that the designed IoT gateway performs the basic function of a device that integrates multi-communication protocols from ZigBee and LoRa sensor networks, and then sends collected data to the database for further analysis and processing. By using an embedded computing platform like Raspberry Pi in this study, the designed IoT gateway device not only successfully combines two typical sensor technologies, ZigBee and LoRa, but is also available to provide additional wireless communication interfaces such as Bluetooth, Wi-Fi or wired connections, for example, Ethernet and USB. Therefore, this device will be able to handle the requirements for expanding new sensor networks in the future.

To evaluate the operability of the IoT gateway device after design, two measurements were carried out. The first experiment was conducted by setting one LoRa sensor node at the distances of $100\,m$, $200\,m$... up to $1500\,m$ far from the IoT gateway device under the condition of existing at least one straight line-of-sight propagation between these devices. The end node sent its ID 100 times in the cases: the SF (Spreading Factor) was set to 7, 9 and 12. In these cases, the node parameters were determined as follows: working frequency $433\,MHz$, bandwidth $500 kHz$, output power $20\,dBm$ in transmission. The IoT gateway device calculated the number of correct ID packets received and then determined the packet success rate as shown in Figure 21.
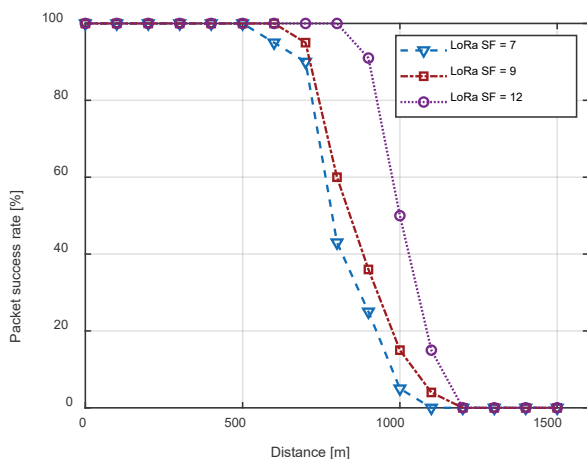


Fig. 21. The relation between packet success rate and distance with different SF values of LoRa network.

Theoretically, the distance between receiver and transmitter LoRa SX1276 module can be up to $1.5\,km$. However, this module used in the paper does not have a power amplifier IC, so the maximum experimental distance is only $900\,m$ corresponding to the SF=12. Figure 21 also indicates that with the same packet success rate, the larger the SF, the longer the communication distance of the system will be. This is the advantage of the spread spectrum technique when applied to the LoRa transceiver module.

The second experiment was carried out by setting one ZigBee sensor node in home conditions with a distance $25\,m$ from the IoT gateway device. Module ZigBee CC2530 was set at $2.4\,GHz$ frequency band and output power $4\,dBm$. The received data will also be evaluated to check the reliability of the communication algorithm introduced in Section *III A*. The result described in Figure 22 shows that in some cases the temperature and humidity data are in some situations (at minutes 49 and 1260) due to the missing synchronization between the node and the IoT gateway. Therefore, the time multiplexing algorithm needs to be further improved in the future to meet the requirements of reliable transmission.
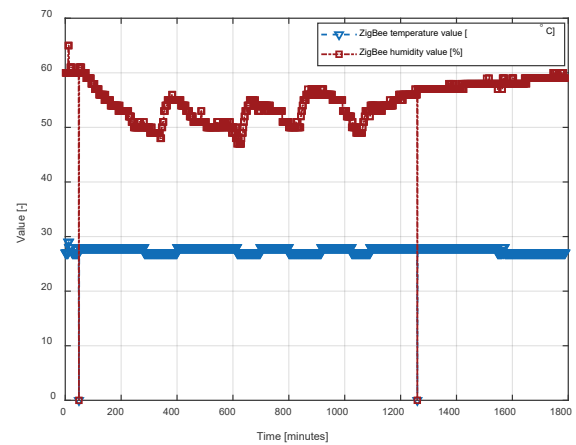


Fig. 22. The description of the capability to transmit data from the ZigBee sensor node to the IoT gateway device.

## IV. CONCLUSION

The IoT gateway is a key component operating at the edge of the IoT system. It plays as a device that connects different sensor networks and converts distinct communication protocols into one common protocol for transmission over the Internet environment. The paper presents an approach to design this device based on the embedded computing platform. The proposed solution takes advantage of Raspberry Pi's low cost, high performance and multi-interface support to deploy this device in a combination of short-range ZigBee and long-range LoRa sensor networks to provide IoT applications. Experimental results have proved the feasibility of the proposed design method. However, the device is only used on a basic level and has limited functionality. In future, the research will continue to improve the hardware system to enhance energy saving, improve reliability, increase transmission distance, and also add remote management software for the device.

## REFERENCES

[1] K. Kanitkorn, and R. Siripokarpirom, "A multi-protocol IoT gateway and WiFi/BLE sensor nodes for smart home and building automation: Design and implementation." 2019 10th International Conference of Information and Communication Technology for Embedded Systems (IC-ICTES). IEEE, 2019.

[2] Y. Xiaofeng and L. Wang, "Design and implementation of IOT gateway based on embedded μTenux operating system." International Journal of Grid and Utility Computing 8.1 (2017): 22-28.

[3] A. Glória, F. Cercas and N. Souto, "Design and implementation of an IoT gateway to create smart environments." Design and implementation of an IoT gateway to create smart environments (2017): 568-575.

[4] C. Gavrilă et al, "Reconfigurable IoT Gateway Based on a SDR Platform." 2018 International Conference on Communications (COMM). IEEE, 2018.

[5] A. Stanciu et al, "Securing the IoT gateway based on the hardware implementation of a multi pattern search algorithm." 2017 International Conference on Optimization of Electrical and Electronic Equipment (OPTIM) & 2017 Intl Aegean Conference on Electrical Machines and Power Electronics (ACEMP). IEEE, 2017.

[6] F. Jalali et al, "Cognitive IoT gateways: automatic task sharing and switching between cloud and edge/fog computing." Proceedings of the SIGCOMM Posters and Demos. 2017. 121-123.

[7] O. Salman et al, "Edge computing enabling the Internet of Things." 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT). IEEE, 2015.

[8] K. Seong-Min, C. Hoan-Suk, and W. Rhee, "IoT home gateway for auto-configuration and management of MQTT devices." 2015 IEEE Conference on Wireless Sensors (ICWiSe). IEEE, 2015.

[9] A. I. Ali, S. Z. Partal, S. Kepke and H. P. Partal, "ZigBee and LoRa based Wireless Sensors for Smart Environment and IoT Applications," 2019 1st Global Power, Energy and Communication Conference (GPECOM), 2019, pp. 19-23, doi: 10.1109/GPECOM.2019.8778505.

[10] R. Singh, M. Baz, C.L. Narayana, M. Rashid, A. Gehlot, S.V Akram, S.S Alshamrani, D. Prashar and Al Ghamdi, "Zigbee and Long-Range Architecture Based Monitoring System for Oil Pipeline Monitoring with the Internet of Things". Sustainability 2021, 13, 10226. https://doi.org/10.3390/su131810226.

[11] J. Shi et al "LoRaBee: Cross-Technology Communication from LoRa to ZigBee via Payload Encoding". 1-11. 10.1109/ICNP.2019.8888145.

[12] C. Hao, J. Xueqin and L. Heng, "A brief introduction to IoT gateway," IET International Conference on Communication Technology and Application (ICCTA 2011), 2011, pp. 610-613, doi: 10.1049/cp.2011.0740.

[13] https://lora-alliance.org/lora_products/lora-sip-module-s76s/

[14] https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-Model-Bplus-Product-Brief.pdf