Journal of Technology Informatics and Engineering (JTIE)
Vol. 3 No. 2 August 2024
E-ISSN: 2987-9434; P-ISSN: 2988-0343, Pages 117-137

Innovative Data Delivery in IoT: Master-Slave Method and LoRa Communication

Hendry[1*], Danny Manongga[1]
[1]*Dept. Computer Science, Satya Wacana Christian University, Salatiga, Indonesia*
Email: hendry@uksw.edu, danny.manongga@uksw.edu
Orcid: 0000-0002-7387-2622, 0000-0002-7430-8740
*Corresponding author

Abstract
This study explores the implementation of multi-node sensor data delivery using the master-slave method in Long Range (LoRa) communication. By utilizing sensor nodes as slaves that gather and transmit data to a master node, which then processes and consolidates the data before sending it to a central server, the system aims to enhance efficiency and reliability in data collection across widespread locations. Experimental results demonstrate that the master-slave method reduces latency and increases data transmission speed. This approach also highlights the potential for significant improvements in the Internet of Things (IoT) applications, which require extensive communication range and low power consumption. Key challenges such as data capacity, synchronization, interference, power consumption, and security were addressed with solutions including bandwidth management, time synchronization, frequency hopping, power optimization, and data encryption. This research indicates that the master-slave method in LoRa communication can significantly optimize the performance and reliability of sensor networks, making it suitable for various sectors such as agriculture, healthcare, environmental monitoring, and industry.

Keywords: *Data Delivery, Sensor Nodes, Master-Slave Method.*

I. INTRODUCTION

In the current digital era, the need for reliable and efficient communication systems is increasingly urgent, especially in the Internet of Things (IoT) context. The IoT links numerous intelligent devices that gather and exchange data to boost operational efficiency, lower expenses, and elevate the quality of life. One of the key technologies in IoT is Long Range (LoRa), which offers a wide communication range and low power consumption, making it suitable for applications requiring long-distance data transmission with limited power (Khutsoane et al., 2017; Rawat et al., 2020). However, data delivery from multiple sensor nodes distributed across various locations presents challenges, especially regarding efficiency and reliability. The master-slave method is one approach that can address these challenges (Shu et al., 2018). In a master-slave system, sensor nodes act as slaves that collect data and send it to the master node. The master then manages, combines, and sends the data to a central server for further analysis (Li et al., 2014). This method improves the reliability and efficiency of data delivery, reduces latency, and optimizes resource usage. With implementing the master-slave method in LoRa communication (Faradisa et al., 2023), data collection systems can become more effective and efficient, enabling

Received on 02 June, 2024; Revised on 07 June, 2024; Accepted on August 16, 2024. Published on August 21, 2024.
Doi: 10.51903/jtie.v3i2.179

broader and more diverse applications in the field of IoT. This technology is highly relevant in various sectors such as agriculture, healthcare, environment, and industry, where real-time monitoring and data collection are crucial (Elijah et al., 2018).

Implementing data delivery from multiple sensors using the master-slave method in LoRa communication encounters several significant challenges (Pagano et al., 2023). The limited bandwidth of LoRa can cause congestion when many sensors send data simultaneously. Synchronization between the master and multiple slave nodes can become complex, especially as the number of nodes increases. The frequencies used by LoRa can be affected by other sources, reducing the reliability of data transmission. Sensors using batteries may experience decreased battery life if communication is continuous. Furthermore, without adequate encryption, data transmitted via LoRa is vulnerable to eavesdropping and manipulation. To address these challenges, several solutions have been proposed. Implementing timing or slot time algorithms can ensure that only one node sends data at a time, reducing the likelihood of congestion. Using protocols like Network Time Protocol (NTP) ensures all nodes have synchronized time, thereby reducing the possibility of data collisions. Frequency hopping techniques can be utilized to reduce interference and increase the reliability of data transmission. Power consumption optimization can be achieved by implementing sleep mode on sensor nodes when not transmitting data and using more efficient data collection techniques to save power. Additionally, end-to-end encryption can protect data transmitted over the LoRa network, preventing unauthorized access (Pagano et al., 2023). Using this approach, data delivery from multiple sensors through the master-slave method in LoRa communication can be optimized to enhance reliability, efficiency, and security. This research focuses on exploring and implementing the master-slave method for multi-node sensor data delivery via LoRa communication and evaluating the system's performance under various operational conditions. The anticipated results aim to significantly contribute to the advancement of more efficient and dependable IoT communication systems.

II. LITERATURE REVIEW

A. Data Transfer

Data delivery is the process of transferring information from one location to another using various techniques and technologies according to environmental needs and conditions (Hu et al., 2018). The main components in data delivery include the data source, which is the starting point where data is generated, such as sensors, computers, or other devices that collect and produce information. Transmission media, either wired (such as copper wire or fiber optics) or wireless (such as radio waves or infrared), serve as the physical or logical path used to send data from the source to the destination. Communication protocols, such as TCP/IP, HTTP, and MQTT, are sets of rules that govern how data is encoded, transmitted, and received, ensuring that data is sent and received correctly and in the proper order. Modulation and coding are often necessary to convert digital data into analog signals or to change data into a specific format for transmission. Interference and error handling mechanisms, such as error detection and correction and redundancy methods, are employed to ensure data integrity during transmission (Ali et al., 2023). Security is critical for protecting data from unauthorized access, and encryption is commonly used to transform data into a format that cannot be read without the correct encryption key. Synchronization between the sender and receiver is crucial to prevent data conflicts and ensure that data is received in the correct order. In systems that use battery power, such as wireless sensors, optimizing power consumption is important to ensure longer operation without frequent recharging. Network topology, the structure or layout of a data network, affects the efficiency and reliability of data delivery, with common topologies including star, ring, bus, and mesh, each having its advantages and disadvantages. Effective bandwidth management techniques, such as Quality of Service (QoS), are used to allocate and optimize the transmission capacity within a network, ensuring that critical applications receive priority in bandwidth usage (Esmailpour & Nasser, 2011; Son & Buyya, 2019). Efficient and secure data delivery is crucial for many modern applications, from the IoT and wireless sensors to computer communication and telecommunications. By understanding and managing these elements, data can be delivered quickly, reliably, and securely.

B. Sensor Node

Sensor nodes are essential components in a sensor network system, playing a crucial role in collecting, processing, and transmitting data from the surrounding environment (Deng et al., 2019). Each sensor node includes several main components, such as the sensor itself, which is responsible for detecting and measuring specific physical or chemical phenomena like temperature, humidity, pressure, light, and motion. The data produced by the sensor is converted into electrical signals for further processing. A processor serves as the data processing unit that receives signals from the sensor, processes the data, and prepares it for transmission to the network. This unit can perform various functions, including data processing, compression, and encryption. The communication module allows the sensor node to communicate with other nodes or a control center via various data transmission technologies such as RF (Radio Frequency), Wi-Fi, Bluetooth, Zigbee, or LoRa (Long Range). The power source, typically batteries, powers the sensor node, though energy can also be drawn from the environment through solar panels or kinetic energy. Energy efficiency is critical as sensor nodes are often placed in hard-to-reach locations where battery replacement is difficult. Sensor nodes measure environmental parameters or specific objects and convert these measurements into processable data. This initial function, known as sensing, is vital for capturing accurate and relevant information. The collected data undergoes preliminary processing where it is filtered, compressed, and analyzed to detect patterns and anomalies, thus reducing the volume of data that needs to be transmitted. After processing, the data is transmitted to other nodes or a central control unit using available communication networks, ensuring that the information reaches its intended destination for further action or analysis. Sensor nodes also coordinate with each other, exchanging information and collaborating to enhance data collection and processing efficiency. This interaction is crucial for maintaining a coherent and synchronized network, especially in complex systems where multiple nodes work together to monitor and manage various parameters. Sensor nodes are employed in a wide range of applications, demonstrating their versatility and importance (Landaluce et al., 2020).

In environmental monitoring, they track temperature, humidity, and air quality, providing valuable data for environmental studies and climate change research. In smart agriculture, these nodes monitor soil and weather conditions, enhancing irrigation efficiency and overall agricultural management. The healthcare sector benefits from sensor nodes through real-time monitoring of patient health parameters, enabling remote diagnosis and treatment. In smart homes, they control and monitor household devices such as thermostats, lighting systems, and security apparatuses, contributing to home automation and energy management. In the industrial sector, sensor nodes play a pivotal role in monitoring machinery and production processes, facilitating predictive maintenance, and boosting operational efficiency. Sensor nodes face several challenges, such as battery life, as these nodes often operate in remote or hard-to-reach locations. To extend battery life, techniques such as sleep mode and energy harvesting are utilized, optimizing power usage and reducing the need for frequent battery replacements. Communication reliability is another critical concern, especially in environments with potential interference and signal range limitations. Solutions such as frequency hopping and mesh networks help overcome these issues by enhancing signal robustness and ensuring reliable data transmission. Additionally, data security is paramount to protect the integrity of transmitted information. Implementing data encryption and strong authentication mechanisms helps safeguard against security threats, ensuring that the data remains confidential and unaltered during transmission. By understanding the components, functions, applications, challenges, and solutions associated with sensor nodes, we can develop efficient, reliable, and secure sensor network systems tailored to various modern application needs.

C. Long Range (LoRa)

LoRa communication is a wireless technology tailored for IoT applications that necessitate long-range communication and low power consumption. This technology utilizes the chirp spread spectrum (CSS) radio modulation to enable data transmission over distances of several kilometers, depending on the physical environment and transmission conditions (Bizon Franco De Almeida et al., 2021; Gbadoubissa et al., 2023; Maleki et al., 2023; Pasolini, 2022). The main advantage of LoRa lies in its ability to transmit data over long distances, overcoming physical obstacles such as buildings and vegetation without requiring complex cellular network infrastructure. This makes it suitable for applications such as environmental monitoring, smart agriculture, asset management, and smart city infrastructure where IoT devices are spread over large and remote areas (Ayaz et al., 2019; Maraveas et al., 2022). Operating within global Industrial Scientific and Medical (ISM) frequency bands—like 433 MHz, 868 MHz (in Europe), and 915 MHz (in North America)—LoRa provides flexibility in implementation according to local regulations and environmental conditions. Furthermore, the technology benefits from an active community dedicated to developing standards and hardware, which aids in the deployment and innovation of IoT solutions (Bhuiyan et al., 2021; Čolaković & Hadžialić, 2018; Vermesan et al., 2014). With its combination of long-range communication, low power consumption, and cost-effective implementation, LoRa has become a leading choice for IoT applications that require reliable and efficient communication across various scales and applications.

D. Master-Slave method

The Master-Slave method is an approach in computing and communication systems where multiple devices or entities interact hierarchically. In this system, the device acting as the "master" controls and coordinates the activities of one or more "slave" devices (El Moursi et al., 2014; Townsend & Guertin, 1999). The primary role of the master device is to manage the activities of the slave devices by sending instructions or requests to execute, stop, or modify specific tasks. The master also collects data from the slave devices and manages the resources required to run the system as a whole. On the other hand, slave devices carry out the tasks assigned by the master and send the results back to the master (Caldognetto & Tenti, 2014; Townsend & Guertin, 1999). The main advantage of the Master-Slave method is its ability to divide tasks into smaller, more manageable parts. This not only increases the overall efficiency of the system but also reduces resource conflicts and allows for better grouping of tasks according to the capabilities and needs of each device. This method is commonly applied in various fields of computing and technology, such as network communication, database systems, and industrial control. For example, in an IoT network, one master device can control the operations of many sensor devices distributed across different locations (Bansal & Kumar, 2020; Kopetz & Steiner, 2022; Qiu et al., 2018). Thus, the Master-Slave method provides a robust framework for organizing interactions and managing resources among devices within a system, facilitating effective coordination and efficient resource utilization.

III. METHODS

This research involves an implementation with a design approach, meaning the focus is on implementing a new system or solution based on a pre-planned design. The research process follows several stages, including needs analysis, system design, implementation, testing, and evaluation, as shown in Figure 1. Initially, a needs analysis is conducted to understand the system requirements that must be met. Based on this analysis, system design is carried out, which includes architecture design, technology selection, and detailed system specifications. Once the design is complete, the next step is implementation, where the planned system or solution is built according to the established specifications. After implementation, testing is performed to ensure that the system operates well and meets the expected requirements. This testing includes evaluating the functionality, performance, and security of the system. Finally, an evaluation is conducted to assess the success of the implementation based on the criteria established beforehand. By using this design-based implementation approach, this research aims to produce significant contributions in the form of systems or solutions that can be applied and provide benefits in relevant contexts.

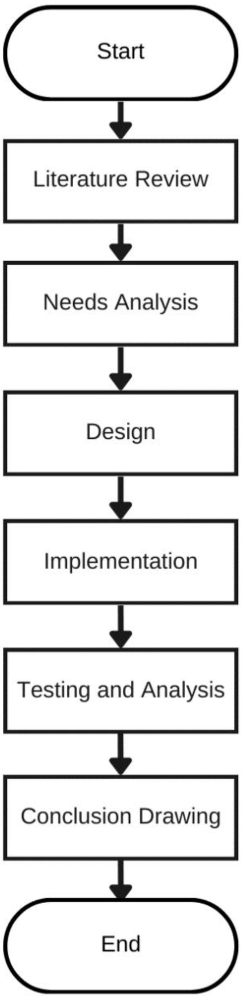

Figure 1. Design Approach

IV. RESULT/FINDINGS AND DISCUSSION

A. *Design System*

The system architecture for multi-node sensor data transmission using the master-slave method with LoRa communication comprises several critical components and layers that collaboratively collect and transmit data. As illustrated in Figure 2, each sensor node is outfitted with sensors that gather environmental data, such as temperature, humidity, and pressure. These sensor nodes also contain a LoRa module for transmitting data to the master node. The master node, also equipped with a LoRa module, receives data from the sensor nodes. It includes a processing unit, such as a microcontroller or a small computer, which manages and processes the data received from the sensor nodes and can send commands back to them.

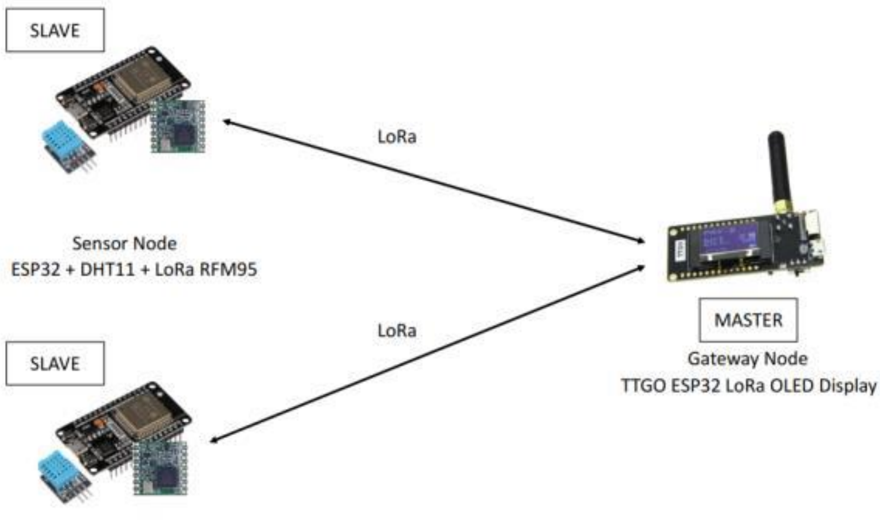

Figure 2. General Overview of System Architecture

The master-slave communication method is central to this system design. The master node acts as the main controller, managing the timing of data transmission from each sensor node. It sends commands or polling requests to the sensor nodes to request data, and the sensor nodes, functioning as slaves, only send data when requested by the master node. This approach helps reduce data transmission conflicts and increases the efficiency of LoRa bandwidth usage. LoRa communication technology, as illustrated in Figure 3, supports long-range data transmission with low power consumption, making it suitable for remote sensor applications. The network topology often used in this setup is a star topology, with the master node at the center directly connected to multiple sensor nodes. Data from the master node can be forwarded to a gateway and then to a central server for further processing and analysis, enabling effective and reliable data transmission over long distances.

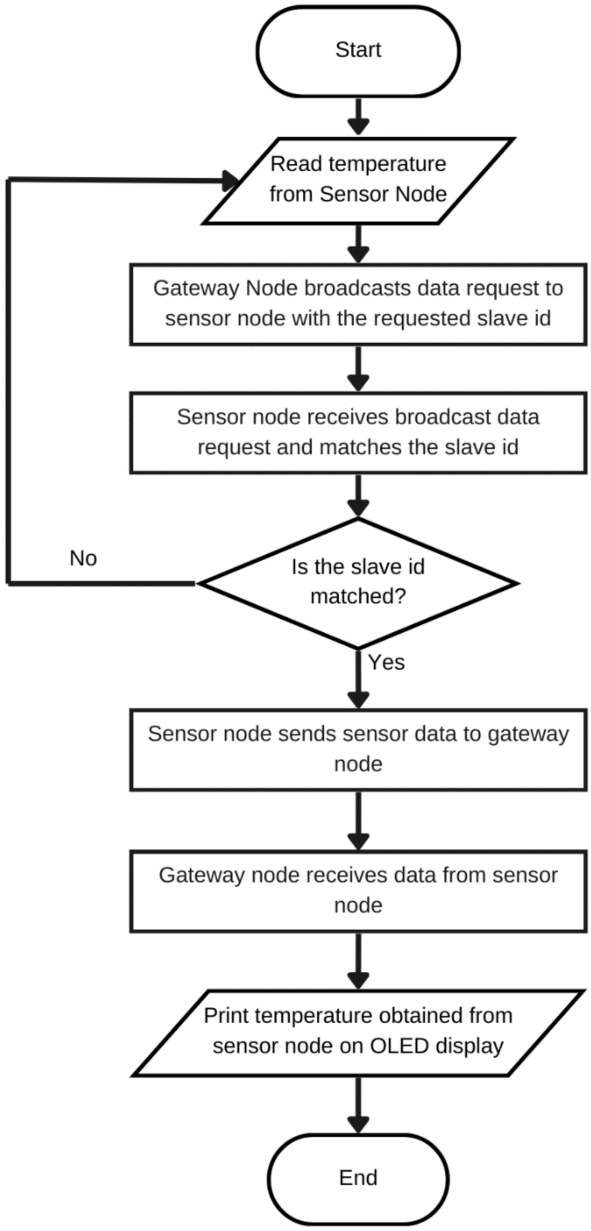

Figure 3. Flowchart of System Flow

The system workflow design, detailed in Figure 4, involves identifying user needs, mapping information flow, and organizing the sequence of operations to ensure the system operates efficiently and effectively. This design ensures the system can handle various operational conditions and deliver reliable performance. Figures 5 and 6 provide a detailed view of the sensor node architecture and the flowchart of the sensor node design workflow. These illustrations demonstrate the steps and components involved in designing and implementing sensor nodes, highlighting the importance of an efficient and robust design to ensure reliable data collection and transmission.

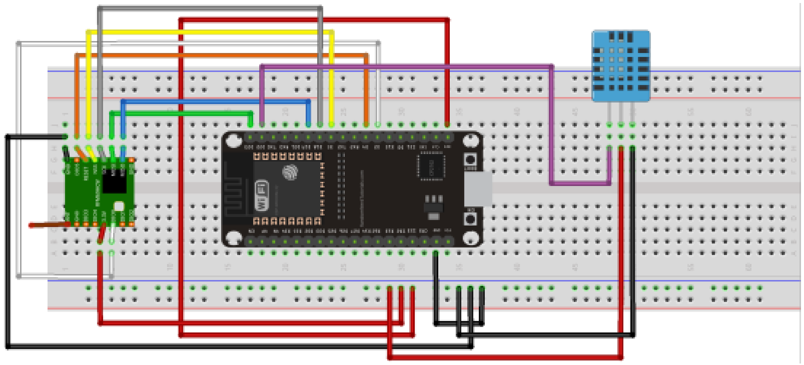

Figure 4. Sensor Node Architecture

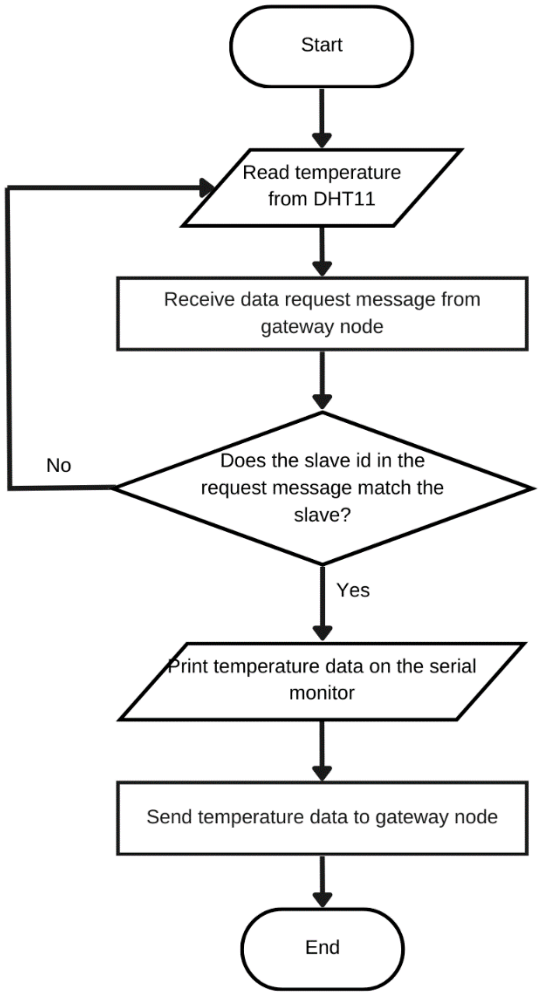

Figure 5. Flowchart of Sensor Node Design Workflow

The gateway node design, shown in Figure 7, bridges the communication between the sensor nodes and the central system or cloud. It ensures seamless data transmission and processing, playing a crucial role in the overall system efficiency. The operational steps implemented in the gateway node as a master are visualized in Figure 8, which helps in understanding the process flow and the interaction between the master and the sensor nodes. Figures 9 and 10 illustrate the actual implementation of the sensor nodes and the gateway node, showcasing the practical setup used in this research. This includes the hardware configurations and the data flow between the nodes and the master. Through this detailed system design, the research aims to demonstrate the effectiveness of the master-slave method in optimizing data transmission and coordination among multiple sensor nodes using LoRa communication. This design provides a scalable and reliable solution for various IoT applications, enhancing the overall performance and efficiency of sensor networks.

B. System Workflow Design

The system workflow design involves formulating the steps and the sequence of tasks needed to perform specific functions within the system. This process is illustrated in Figure 4 and includes several crucial stages to ensure efficient and effective system operation. Initially, the analysis is conducted to identify and understand user requirements and the system's objectives. This analysis helps in determining the necessary features and functionalities that the system must provide. Following the analysis, process modeling is carried out to represent existing business or operational processes using flowcharts or diagrams. This stage helps in visualizing the current processes and identifying areas for improvement or optimization. Based on the insights gained from process modeling, the system design phase begins. During this phase, the system elements, including hardware, software, and interactions between components, are designed to meet the identified requirements.

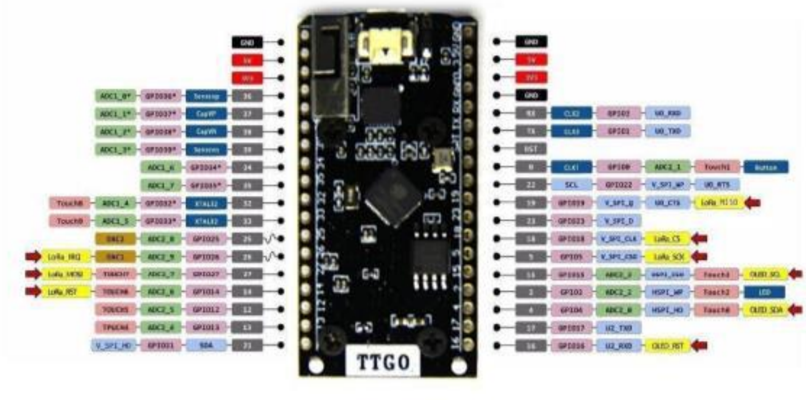

Figure 6. Pinout TTGO ESP32 LoRa OLED

After the design is finalized, the subsequent step is testing and validation. This phase ensures that the designed workflow meets the specified requirements and functions correctly through thorough and rigorous testing. This includes testing the system's functionality, performance, and security to ensure it operates as expected under various conditions.

Implementation follows, where the system is developed and deployed based on the created design. This stage involves setting up the hardware and software components and integrating them into a cohesive system. Finally, maintenance is performed to monitor and update the system continuously, ensuring it adapts to changing needs and operates smoothly over time. Figure 5 provides a detailed view of the flowchart of the sensor node design workflow, illustrating the steps and procedures involved in designing a sensor node. This includes starting the design process, determining the parameters to be measured, selecting the appropriate sensor type, creating the electronic circuit design, developing the code for data collection and transmission, and testing the prototype sensor node.

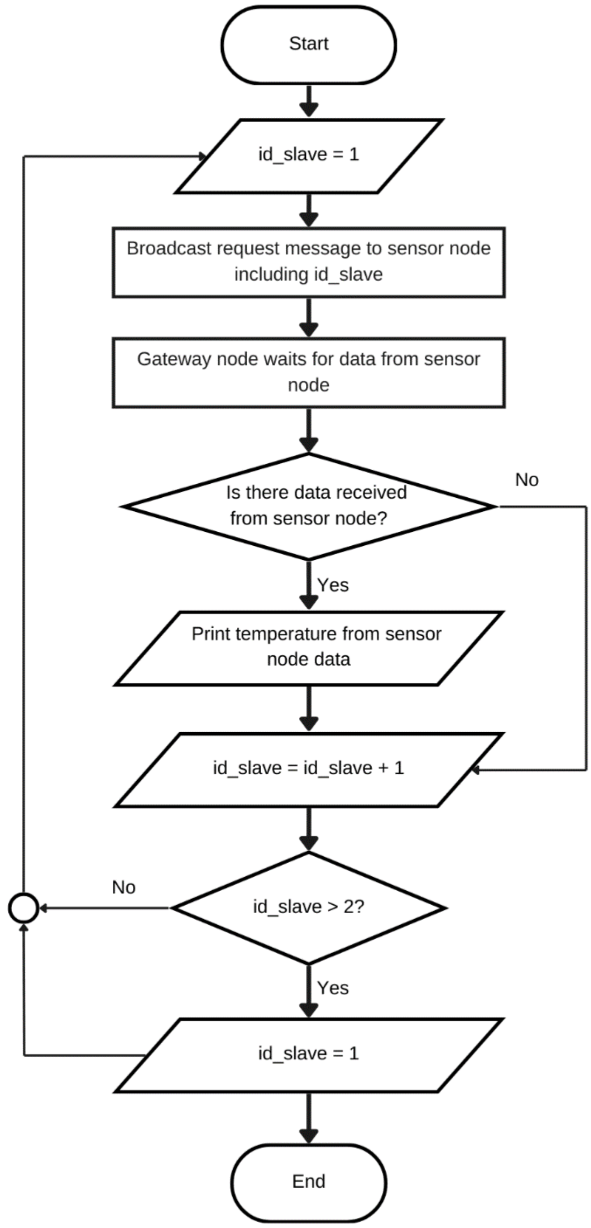

Figure 7. Flowchart Gateway node

Journal of Technology Informatics and Engineering (JTIE)
Vol. 3 No. 2 August 2024
E-ISSN: 2987-9434; P-ISSN : 2988-0343, Pages 117-137

C. Sensor Node Architechture

The design of a sensor node involves creating a device capable of collecting, processing, and transmitting data from its surrounding environment. The key steps in designing a sensor node are depicted in Figure 5, which outlines the workflow and the components involved. Initially, the appropriate sensor type is selected based on the parameters to be measured, such as temperature, humidity, or pressure. Following this selection, the next step involves designing the electronic circuit, which incorporates the sensor, microcontroller, and communication module. The communication module, such as a LoRa module, allows the sensor node to transmit data over long distances with low power consumption. Next, code development is carried out to enable the sensor node to collect, process, and transmit data effectively. This step includes writing software to control the sensor, process the collected data, and manage communication with the master node. Testing the prototype sensor node is crucial to ensure it meets performance expectations and operates reliably under various conditions. The prototype undergoes rigorous testing to validate its functionality, data accuracy, and communication reliability.

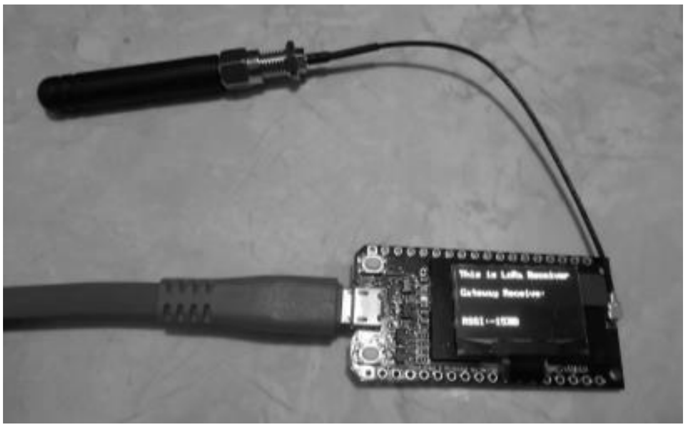

Figure 8. Microcontroller Gateway node

Figure 6 provides a detailed view of the sensor node architecture, illustrating the components and their interactions within the sensor node. The design also includes an efficient power solution, such as batteries or alternative energy sources like solar panels, to meet the power needs of the node. Energy efficiency is a critical consideration, as sensor nodes are often deployed in remote or hard-to-reach locations where frequent battery replacement is impractical. The flowchart of the sensor node design workflow, shown in Figure 7, visualizes the steps involved in designing and implementing the sensor node. It starts with determining the parameters to be measured, selecting the appropriate sensor, designing the electronic circuit, developing the code, and testing the prototype. After evaluating the prototype, necessary adjustments are made to the design based on the test results. The final design is then implemented, ensuring that the sensor node operates effectively and efficiently.

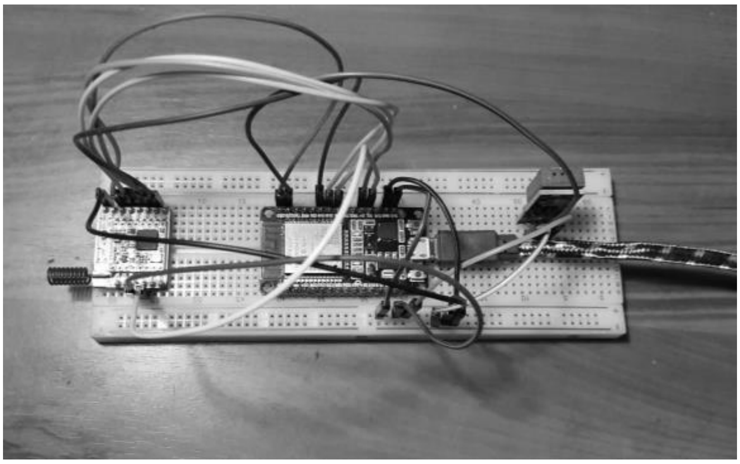

Figure 9. Implementation of the Node Sensor

D. Gateway Node Design

The design of a gateway node involves developing a device that acts as a bridge between sensor nodes and the central system or cloud. The primary steps in designing a gateway node are depicted in Figure 6, which outlines the components and their interactions. The gateway node must receive data from multiple sensor nodes and transmit this data to a central server or cloud for further processing and analysis. The key components of the gateway node include a microprocessor, communication modules (e.g., Wi-Fi, LoRa, Zigbee), and network interfaces. These components are selected based on the type of data to be received and transmitted, as well as the communication protocols to be used. The design process starts with determining the required functions and specifications of the gateway node, such as data handling capacity and communication range. Next, the necessary circuitry and components are designed to connect the sensor nodes to the central system. This includes developing the software to receive, process, and transmit data from the sensor nodes to the central system or cloud. Thorough testing is performed to ensure that the gateway node can accurately and efficiently collect, process, and transmit data.

Figure 7 illustrates the operational steps implemented in the gateway node as a master. In this context, the gateway node orchestrates the data transmission process, ensuring that data from various sensor nodes are collected and integrated efficiently. This process involves sending requests to sensor nodes to gather data, receiving the data, processing it, and then transmitting the processed data to the central system or cloud for further analysis and storage. As shown in Figure 8, the gateway node implementation demonstrates how the hardware manages the data transmission process from the sensor nodes, highlighting its critical role in ensuring an efficient

Journal of Technology Informatics and Engineering (JTIE)
Vol. 3 No. 2 August 2024
E-ISSN: 2987-9434; P-ISSN : 2988-0343, Pages 117-137

and reliable sensor network communication ecosystem. The diagram of the gateway node microcontroller visualizes the hardware setup and the data flow between the sensor nodes and the master.

E. Implementation Result

Implementing the data transmission method using the master-slave system for sensor nodes and the gateway node involves configuring the gateway node as the master, which oversees the data transmission process from the sensor nodes. The microcontroller of the gateway node, as depicted in Figure 8, is crucial in managing data transmission from the sensor nodes, ensuring efficient and reliable communication within the sensor network. The devices are programmed using pre-prepared code by the authors with the Arduino IDE software. This code ensures that the devices operate according to the functional requirements that have been tested. On the sensor node side, this implementation uses two sensor nodes to collect environmental temperature data. Each sensor node is equipped with a temperature sensor to monitor the surrounding temperature conditions. The data read by the sensor nodes is sent to the gateway node according to the requests received from the gateway node.

In this system, the sensor nodes are responsible for taking temperature readings of the environment and transmitting them to the gateway node based on the requests received from the gateway node's query messages. The implementation of the sensor nodes is shown in Figure 9, which illustrates the setup and the data flow from the sensor nodes to the gateway node. By implementing this master-slave method, the research ensures that data collection and transmission are managed efficiently, reducing the likelihood of data collisions and optimizing the use of the LoRa communication bandwidth. This setup enables reliable and scalable data transmission, making it suitable for various IoT applications that require extensive coverage and low power consumption. The implementation details provided in Figures 8 and 9 demonstrate the practical setup used in this research, highlighting the hardware configurations and the interactions between the sensor nodes and the gateway node. This thorough implementation approach ensures that the system operates effectively under various conditions, providing reliable data transmission and processing capabilities.

F. Testing Result

The testing phase consists of functional and non-functional testing to ensure that the system operates according to the established specifications and meets performance expectations.

1. *Functional Testing*

Functional testing is conducted to verify that the system performs its intended functions accurately. This involves ensuring that each sensor node can transmit data accurately and promptly to the gateway node via the LoRa communication protocol. The integration between sensor nodes and the gateway node is tested to confirm effective communication and coordination in data transmission. The system's performance is measured concerning data transmission latency, response speed, and data handling capacity to ensure it operates efficiently under various load conditions. Additionally, the system's response to data transmission requests from the gateway node to the sensor nodes is evaluated. The results of the functional testing are summarized in Table 1, which shows that all tests were successful and the system met the functional requirements.

Table 1. Functional Testing (tools)

No.	Code	Dec.
1.	MSL-01	Success
2.	MSL-02	Success
3.	MSL-03	Success
4.	MSL-04	Success
5.	MSL-05	Success

2. *Non-Functional Testing*

Non-functional testing focuses on aspects of the system that are not directly related to core functions but significantly impact the overall performance, security, and reliability. This includes performance testing to measure the system's response time, data processing speed, and ability to handle different workloads. Security testing examines the system's resistance to threats such as cyber-attacks, penetration, and other vulnerabilities. Compatibility testing ensures that the system can operate seamlessly with other devices and platforms. Load tolerance testing assesses the system's capacity to manage an increasing number of users or data volume without significant performance degradation. Additionally, system availability is tested to ensure high uptime and minimal unplanned downtime.

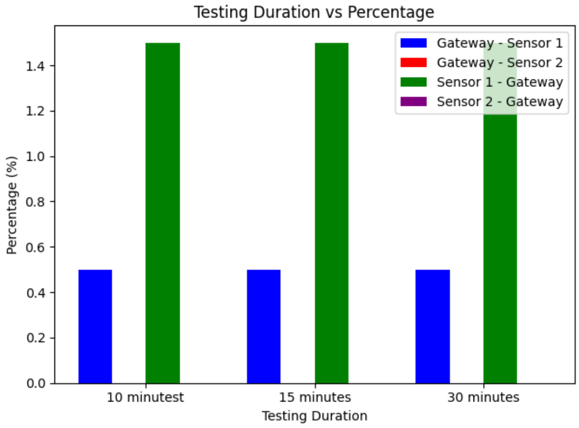

Figure 10. Graph of Packet Loss Percentage at 50 Meters Distance Non-Functional Testing

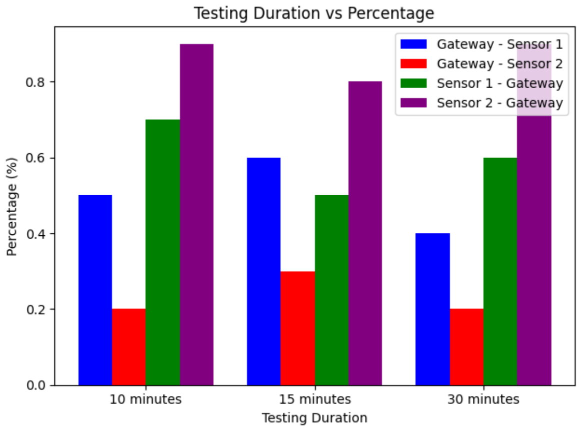

Figure 11. Graph of Packet Loss Percentage at 100 Meters Distance Non-Functional Testing

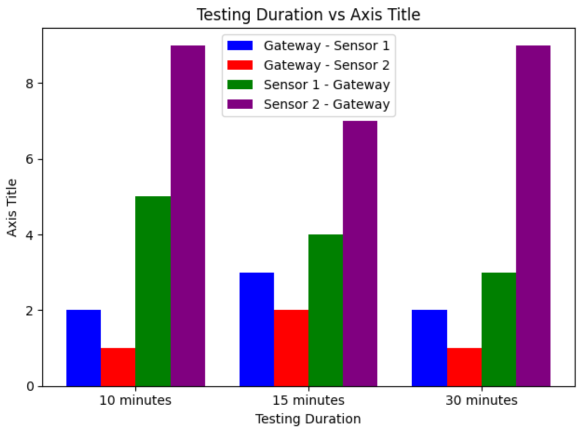

Figure 12. Graph of Packet Loss Percentage at 200 Meters Distance Non-Functional Testing

The impact of distance on data transmission reliability is illustrated in Figures 10, 11, and 12. These figures show the graphs of packet loss percentage at various distances (50 meters, 100 meters, and 200 meters). The results indicate that increasing the distance often results in a higher percentage of data packet loss due to potential interference and signal attenuation. Factors such as electromagnetic interference, changes in atmospheric conditions, or physical obstructions between the sender and receiver can negatively affect the quality of data transmission. Efforts to address these issues include using communication technologies like LoRa, which are designed for reliable long-range transmission with low power consumption. Optimal infrastructure setup and appropriate frequency selection are critical in reducing data packet loss in long-distance wireless communication environments.

V. CONCLUSION AND RECOMMENDATION

The conclusion on implementing multi-node sensor data transmission using the master-slave method in LoRa communication is as follows: Utilizing the master-slave method in LoRa communication effectively optimizes coordination among sensor nodes for managing multi-node sensor data transmissions. With one node acting as the master that manages communication and data transmission from the slave nodes, this approach not only improves the efficiency of resource utilization but also enhances the quality of monitoring the environment or system being observed.

A. Implications of the Master-Slave Method

Implementing multi-node sensor data transmission using the master-slave method in LoRa communication offers several significant benefits. It enhances spectrum efficiency by effectively managing spectrum usage and preventing data collisions between communicating sensor nodes. Data transmission management is improved with the master-slave configuration, allowing for better prioritization and scheduling of transmissions. This method also enhances data security and integrity because the master node oversees the coordination and validation of data received from sensor nodes. This method supports system scalability, allowing new sensor nodes to be added easily without disrupting overall system performance. Despite its significant benefits, the implementation of the master-slave method can increase complexity in system setup and programming, requiring careful attention to design and management.

Considering these implications, the implementation of the master-slave method in LoRa communication will enhance the performance and reliability of widely distributed sensor networks. This research has shown that this approach effectively optimizes data transmission and coordination among multiple sensor nodes, making it ideal for various IoT applications that require extensive communication range and low power consumption. The findings of this study significantly contribute to the development of more efficient and reliable IoT communication systems.

B. Future Recommendations

For future research on the implementation of data transmission from multiple sensor nodes using the master-slave method in LoRa communication, several ideas or approaches can be considered. Performance analysis of the LoRa master-slave system should be evaluated, including aspects such as latency, throughput, and power consumption. Optimization of communication protocols can help reduce overhead and improve data transmission efficiency in LoRa networks. Security should be further studied to protect master-slave implementations in LoRa networks from potential threats. Scalability research should explore the system's ability to handle a large number of sensor nodes and manage complex network configurations. Energy efficiency analysis should be conducted to find strategies to improve power usage in sensor nodes and the LoRa master-slave system. Lastly, integration with other technologies, such as IoT platforms or cloud services, should be explored to expand the functionality and data analysis capabilities of the sensors.

REFERENCES

Ali, M. M., Hashim, S. J., Chaudhary, M. A., Ferré, G., Rokhani, F. Z., & Ahmad, Z. (2023). A Reviewing Approach to Analyze the Advancements of Error Detection and Correction Codes in Channel Coding with Emphasis on LPWAN and IoT Systems. *IEEE Access*, *11*, 127077–127097. https://doi.org/10.1109/ACCESS.2023.3331417

Ayaz, M., Ammad-Uddin, M., Sharif, Z., Mansour, A., & Aggoune, E. H. M. (2019). Internet-of-Things (IoT)-based smart agriculture: Toward making the fields talk. *IEEE Access*, *7*, 129551–129583. https://doi.org/10.1109/ACCESS.2019.2932609

Bansal, S., & Kumar, D. (2020). IoT Ecosystem: A Survey on Devices, Gateways, Operating Systems, Middleware and Communication. *International Journal of Wireless Information Networks*, *27*(3), 340–364. https://doi.org/10.1007/S10776-020-00483-7

Bhuiyan, M. N., Rahman, M. M., Billah, M. M., & Saha, D. (2021). Internet of Things (IoT): A Review of Its Enabling Technologies in Healthcare Applications, Standards Protocols, Security, and Market Opportunities. *IEEE Internet of Things Journal*, *8*(13), 10474–10498. https://doi.org/10.1109/JIOT.2021.3062630

Bizon Franco De Almeida, I., Chafii, M., Nimr, A., & Fettweis, G. (2021). Alternative Chirp Spread Spectrum Techniques for LPWANs. *IEEE Transactions on Green Communications and Networking*, *5*(4), 1846–1855. https://doi.org/10.1109/TGCN.2021.3085477

Caldognetto, T., & Tenti, P. (2014). Microgrids operation based on master-slave cooperative control. *IEEE Journal of Emerging and Selected Topics in Power Electronics*, *2*(4), 1081–1088. https://doi.org/10.1109/JESTPE.2014.2345052

Čolaković, A., & Hadžialić, M. (2018). Internet of Things (IoT): A review of enabling technologies, challenges, and open research issues. *Computer Networks*, *144*, 17–39. https://doi.org/10.1016/J.COMNET.2018.07.017

Deng, F., Yue, X., Fan, X., Guan, S., Xu, Y., & Chen, J. (2019). Multisource Energy Harvesting System for a Wireless Sensor Network Node in the Field Environment. *IEEE Internet of Things Journal*, *6*(1), 918–927. https://doi.org/10.1109/JIOT.2018.2865431

El Moursi, M. S., Zeineldin, H. H., Kirtley, J. L., & Alobeidli, K. (2014). A dynamic master/slave reactive power-management scheme for smart grids with distributed generation. *IEEE Transactions on Power Delivery*, *29*(3), 1157–1167. https://doi.org/10.1109/TPWRD.2013.2294793

Elijah, O., Rahman, T. A., Orikumhi, I., Leow, C. Y., & Hindia, M. N. (2018). An Overview of Internet of Things (IoT) and Data Analytics in Agriculture: Benefits and Challenges. *IEEE Internet of Things Journal*, *5*(5), 3758–3773. https://doi.org/10.1109/JIOT.2018.2844296

Esmailpour, A., & Nasser, N. (2011). Dynamic QoS-based bandwidth allocation framework for broadband wireless networks. *IEEE Transactions on Vehicular Technology*, *60*(6), 2690–2700. https://doi.org/10.1109/TVT.2011.2158674

Faradisa, I. S., Palevi, B. R. P. D., Widodo, K. A., Sotyohadi, Megawati, C. D., & Cholis, M. N. (2023). QoS Evaluation of Wireless Sensor Networks for Toxic Gas Monitoring in Volcanic Areas: A Test-Bed Study of LoRa Communication using Master Slave TDD Method. *Proceeding - COMNETSAT 2023: IEEE International Conference on Communication, Networks and Satellite*, 39–46. https://doi.org/10.1109/COMNETSAT59769.2023.10420761

Gbadoubissa, Z., Ari, A., Caputo, S., Biotti, L., Mucchi, L., Edinio Zacko Gbadoubissa, J., Adamou Abba Ari, A., Radoi, E., & Mourad Gueroui, A. (2023). M-Ary Direct Modulation Chirp Spread Spectrum for Spectrally Efficient Communications. *Information 2023, Vol. 14, Page 323*, *14*(6), 323. https://doi.org/10.3390/INFO14060323

Hu, Z. Z., Tian, P. L., Li, S. W., & Zhang, J. P. (2018). BIM-based integrated delivery technologies for intelligent MEP management in the operation and maintenance phase. *Advances in Engineering Software*, *115*, 1–16. https://doi.org/10.1016/J.ADVENGSOFT.2017.08.007

Khutsoane, O., Isong, B., & Abu-Mahfouz, A. M. (2017). IoT devices and applications based on LoRa/LoRaWAN. *Proceedings IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*, *2017-January*, 6107–6112. https://doi.org/10.1109/IECON.2017.8217061

Kopetz, H., & Steiner, W. (2022). Internet of Things. *Real-Time Systems*, 325–341. https://doi.org/10.1007/978-3-031-11992-7_13

Landaluce, H., Arjona, L., Perallos, A., Falcone, F., Angulo, I., & Muralter, F. (2020). A Review of IoT Sensing Applications and Challenges Using RFID and Wireless Sensor Networks. *Sensors 2020, Vol. 20, Page 2495*, *20*(9), 2495. https://doi.org/10.3390/S20092495

Li, F., Ooi, B. C., Özsu, M. T., & Wu, S. (2014). Distributed data management using MapReduce. *ACM Computing Surveys (CSUR)*, *46*(3). https://doi.org/10.1145/2503009

Maleki, A., Nguyen, H. H., Bedeer, E., & Barton, R. (2023). *A Tutorial on Chirp Spread Spectrum for LoRaWAN: Basics and Key Advances*. https://doi.org/10.1109/OJCOMS.2024.3433502

Maraveas, C., Piromalis, D., Arvanitis, K. G., Bartzanas, T., & Loukatos, D. (2022). Applications of IoT for optimized greenhouse environment and resources management. *Computers and Electronics in Agriculture*, *198*, 106993. https://doi.org/10.1016/J.COMPAG.2022.106993

Pagano, A., Croce, D., Tinnirello, I., & Vitale, G. (2023). A Survey on LoRa for Smart Agriculture: Current Trends and Future Perspectives. *IEEE Internet of Things Journal*, *10*(4), 3664–3679. https://doi.org/10.1109/JIOT.2022.3230505

Pasolini, G. (2022). On the LoRa Chirp Spread Spectrum Modulation: Signal Properties and Their Impact on Transmitter and Receiver Architectures. *IEEE Transactions on Wireless Communications*, *21*(1), 357–369. https://doi.org/10.1109/TWC.2021.3095667

Qiu, T., Chen, N., Li, K., Atiquzzaman, M., & Zhao, W. (2018). How can heterogeneous internet of things build our future: A survey. *IEEE Communications Surveys and Tutorials*, *20*(3), 2011–2027. https://doi.org/10.1109/COMST.2018.2803740

Rawat, A. S., Rajendran, J., Ramiah, H., & Rana, A. (2020). LORA (Long Range) and LORAWAN technology for IoT applications in Covid-19 pandemic. *Proceedings - 2020 International Conference on Advances in Computing, Communication and Materials, ICACCM 2020*, *2020-August*. https://doi.org/10.1109/ICACCM50413.2020.9213067

Shu, L., Mukherjee, M., Pecht, M., Crespi, N., & Han, S. N. (2018). Challenges and research issues of data management in IoT for large-scale petrochemical plants. *IEEE Systems Journal*, *12*(3), 2509–2523. https://doi.org/10.1109/JSYST.2017.2700268

Son, J., & Buyya, R. (2019). Priority-Aware VM Allocation and Network Bandwidth Provisioning in Software-Defined Networking (SDN)-Enabled Clouds. *IEEE Transactions on Sustainable Computing*, *4*(1), 17–28. https://doi.org/10.1109/TSUSC.2018.2842074

Townsend, W. T., & Guertin, J. A. (1999). Teleoperator slave - WAM design methodology. *Industrial Robot*, *26*(3), 167–177. https://doi.org/10.1108/01439919910266820

Vermesan, O., Friess, P., Guillemin, P., Sundmaeker, H., Eisenhauer, M., Moessner, K., Arndt, M., Spirito, M., Medagliani, P., Giaffreda, R., Gusmeroli, S., Ladid, L., Serrano, M., Hauswirth, M., & Baldini, G. (2014). Internet of Things strategic research and innovation Agenda. *Internet of Things Applications: From Research and Innovation to Market Deployment*, 7–142. https://doi.org/10.1201/9781003338659-2