Geetanjali Journal of Computer Applications
ISSN: 3048-7277
Volume 01 Issue 01, June 2024

# Use of IoT Access to Create an Architectural Framework for Vehicle Monitoring Systems

Mr. Firdos Sheikh[1], Mr. Sandeep Boardia[2], Mr. Devendra Suthar[3]
*Associate Professor, Department of Computer Science and Engineering, Poornima University, Jaipur, Rajasthan[1]*
*Assistant Professor, Department of Computer Application, Geetanjali Institute of Technical Studies, Udaipur, Rajasthan[2]*
*Assistant Professor, Department of Computer Application, Aravali Institute of Technical Studies, Udaipur, Rajasthan,[3]*
[1] firdos.sheikh@gmail.com, [2]sandeep11d@gmail.com , [3]trishaenterprises.udaipur@gmail.com

***Abstract*— As time goes on, it is becoming more apparent that the Internet of Things (IoT) is pervasive in a variety of sectors, such as the monitoring of traffic, medical treatment, and industrial processes. How to manage a growing number of connected devices is a fundamental and difficult challenge that arises as the Internet of Things ecosystem expands. Due to the intricacy of the situation, it is of the utmost importance to enable standard access to the Internet of Things. By using system-on-chips (SoCs) and field programmable gate arrays (FPGAs), this study introduces a novel Internet of Things access architecture that provides a means for unified access to the Internet of Things for a variety of low-speed and high-speed devices of varying speeds. In order to accommodate the ever-evolving needs of Internet of Things deployment, this design places an emphasis on the capacity to be extensible and to be personalised. Based on the IEEE 1451.2 standard, we have adjusted our design to include an automobile monitoring system, and we have experimentally proven the usefulness of this modification. By demonstrating the system's capability to perform exceptionally well in real-world scenarios, we demonstrate that it is applicable in a practical setting.***

***Keywords*—IoT, Network, Vehicle Monitoring, FPGA, SoCs, Traffic Monitoring, automobiles.**

## I. INTRODUCTION

While Professor Ashton was doing his first research on radio frequency identification (RFID) at the Massachusetts Institute of Technology Auto-ID in 1999, the Internet of Things (IoT) emerged as a cutting-edge technology. In the time that has passed since then, it has evolved into an essential component that plays a significant part in determining the composition of services in a wide variety of applications. The Internet of Things makes it possible for many components in our environment to communicate with one another in a seamless manner by allowing connections to be made over the Internet.

As can be seen in Figure 1, the present architecture of the Internet of Things (IoT) is composed of three primary layers: the viewpoint layer, the network layer, and the application layer simultaneously. Within the context of the Internet of Things, the perception layer is made up of various intelligent terminals, sensors, and actuators. RFID tags are also included in this layer. These components serve the purpose of connecting to the framework's infrastructure in a connector capacity. It is the responsibility of the network layer to ensure that the transfer of data between human users and linked devices is carried out without any interruptions. There is a large variety of applications that may be enabled by the application layer, which highlights the adaptability and diversity of the technology behind the Internet of Things [1, 2].

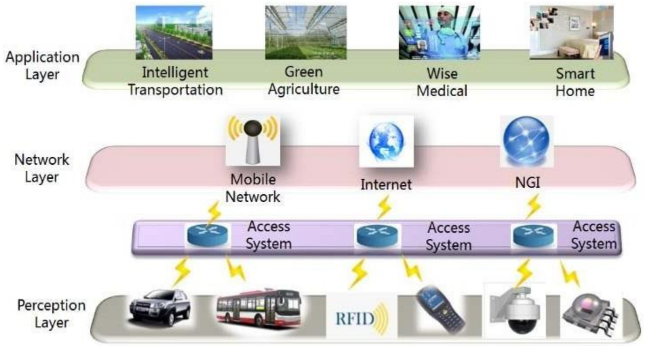

**Fig. 1. The IoT architecture comprises three layers**

It is primarily via the facilitation of connection between various "things" that the Internet of Things (IoT) differentiates itself from the conventional Internet. Because there are so many different kinds of devices involved, ensuring that everyone has equitable access to the Internet of Things (IoT) is a difficult feat to accomplish. Even if there are a great number of systems and interface tools available on the market, the most of them are developed for specific environments and can only handle a limited number of devices that have specialist interfaces. With the intention of overcoming this challenge, we have developed and implemented a unique Internet of Things access architecture that is capable of managing a broad variety of Internet of Things devices. Through the use of the IEEE 1451.2 standard, we are able to better support the utilisation of a wide range of sensors, actuators, and transducers in order to improve the process of data collection [4]. In addition, we have established this design by using FPGA and SoC technologies, which enabled us to achieve comfortable programmability while simultaneously reducing the amount of power that was used.

## II. PROPOSED ARCHITECTURE

The Internet of Things (IoT) is made possible by our cutting-edge framework, which makes it possible for "things" to connect to the IoT without any difficulty. Within the Internet of Things (IoT) network, our cutting-edge design enables comprehensive collection, analysis, retention, and transfer capabilities across all kinds of devices and equipment operating within the network. Because of its versatility, this architecture may be used in a broad variety of Internet of Things scenarios, such as real-time monitoring, the collection of environmental data, and asset management.

When it comes to connecting and interacting with a wide variety of sensors, actuators, and transducers, our design is significantly dependent on the implementation of the IEEE 1451.2 standard as a reference point. The many requirements that are defined by this standard include the specifications for the sensor interface as well as the techniques that are used to gather data [5, 6]. Our use of FPGA technology in the construction of the whole system was done with the intention of achieving the highest possible level of efficiency from the system's hardware resources. Optimising functionality while reducing resource consumption is the goal of this design, which has a large number of distinct intellectual property cores. Moreover, the use of System-on-Chip (SoC) technology allows for the primary component of the system to be condensed into a single Field-Programmable Gate Array (FPGA) chip, therefore enhancing both the efficiency and the expandability of the system.

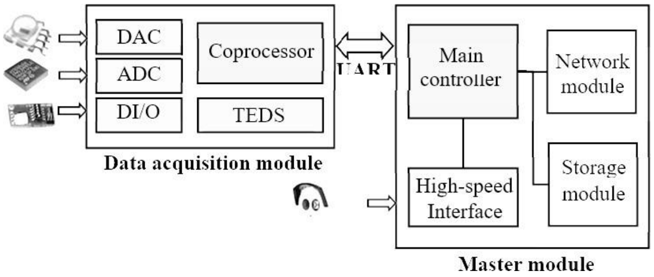

**Fig.2: Proposed Architecture**

It is clear from looking at Figure 2 that the system is made up of two primary components: the master module and the data collecting module. Through the use of the universal asynchronous receiver/transmitter (UART) interface, these two components engage in communication with one another. Access to a number of sensors and the collection of data on the environment are both the responsibilities of the data collection module. Through the use of an analog-to-digital converter (ADC) and a digital-to-analog converter (DAC), the device is able to effectively accommodate both analogue and digital signal inputs. Coprocessors that are based on IP cores are the ones that are accountable for the administration and processing of signals across the whole module. The IEEE 1451.2 standard makes use of a transducer electronic data sheet, also known as a TEDS, in order to identify the kind of sensors, actuators, and transducers, as well as their working and inherent characteristics. Within the confines of this architecture, the requirements are carried out together with being kept in block RAM (BRAM) [7].

The following is a diagrammatic representation of a modular data collection and control programme. This figure is broken up into two primary portions, which are referred to as the "Data acquisition module" and the "Master module." All of the components that are included in the Data acquisition module are essential for converting and processing a wide variety of signals. These components include a Digital-to-Analog Converter (DAC), an Analog-to-Digital Converter (ADC), a Digital Input/Output (DIO), a coprocessor, and a Transducer Electronic Data Sheet (TEDS). A primary controller, a network module, and a storage module are the three components that make up the "Master module," which is coupled to these components. It would seem that the primary controller is responsible for managing duties and the flow of data between the modules. On the other hand, the network module is responsible for facilitating communication across a network, and the storage module is in charge of managing it. A high-speed interface connects the master module and the data collection module, indicating that the master module has the capability to transport data quickly, which is critical for real-time applications.

Through its role as the central hub of the system, the master module is accountable for essential operations such as the communication across the network and the storage of data stored locally. Additionally, it is equipped with specialist connections that have been developed expressly for the purpose of connecting to fast equipment such as digital cameras. The master module is driven by an ARM Cortex central processing unit (CPU), which is well acknowledged for its exceptional performance, amazing durability, and remarkable stability.

By using this architectural style, the system is able to readily interface with a large variety of sensors, actuators, and high-speed devices that are present in the Internet of Things infrastructure. The master module is able to concentrate on more complex duties, such as maintaining network connections and storing data locally, since the specialist data collection module is intended to gather data in a short amount of time. The overall efficiency of the system is optimised, and its versatility for the management of various Internet of Things applications is increased, thanks to the division of responsibilities.

## III. APPLICATION IN VEHICLE MONITORING

The implementation of "intelligent traffic" has become an essential component in people's day-to-day lives as a result of advancements in technology. It is possible that there are other products that are equivalent to this one; nevertheless, these products often lack the characteristics of generality, expandability, and reusability [9], [10]. This study describes the creation and implementation of a cutting-edge and intelligent vehicle monitoring system that makes use of the architecture that was suggested. In Figure 3, we see an illustration of the typical situation in which the programme is called upon. One of the most important components of the system is the vehicle monitoring terminal, which is responsible for collecting a wide range of real-time environmental data, including video, environmental, and location specifications. The processed data is then sent to the server over 4G connection once it has been sent. Using the client system, users have the ability to quickly and conveniently access a broad variety of information.

A vehicle monitoring system uses a centralised server to collect and evaluate various types of data sent from automobiles over a 4G network. The system gathers video, environmental, and positional data from many types of vehicles, such as buses, cars, and trucks, via a vehicle monitoring station. Once collected, the data is sent to a server, where it is typically processed, stored, and potentially analysed to monitor the vehicle's performance, location, and other significant attributes. Customers may access this data via internet-connected devices, allowing for remote monitoring and management of vehicle fleets.

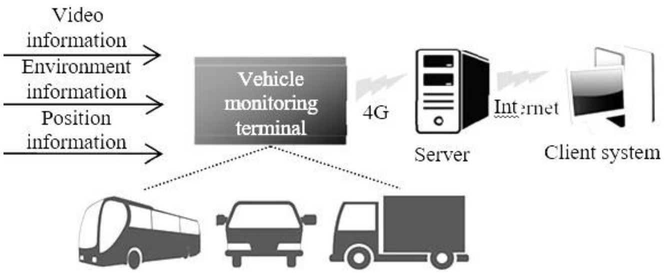

**Fig.3: A Possible Application Scenario for Vehicle Monitoring**

A dual-core ARM Cortex-A9 MPCore-based processing system (PS) and Xilinx programmable logic are both components of the Xilinx Zynq-7000 system-on-chip (SoC) processor, which is the foundation upon which the hardware system is constructed. A broad variety of input/output peripherals, as well as on-chip memory and connections to external memory, are included in the ARM PS. There are several FPGA resources that make up the programmable logic (PL), and these resources may be adapted to accommodate a variety of devices and interfaces [8].

As a coprocessor, we make use of the MicroBlaze IP core in this application. We also make full use of the resources provided by the FPGA in order to achieve the highest possible power efficiency and performance performance. Through the use of a UART connection, the ARM processor and the coprocessor are able to interact with one another. As an additional feature, the board provides support for both the power supply and the DDR memory [11]. A graphical depiction of the hardware block diagram may be seen in Figure 4, which can be consulted.

Our system is equipped with an on-board 12-bit analog-to-digital converter (ADC), which enables the attachment of up to 17 analogue input channels from the outside. The seamless connection between the system and the sensors that create analogue signals is made possible as a result of this. The creation of a variety of digital signal interfaces for digital sensors, such as UART, GPIO, and IIC, is accomplished via the use of specialised FPGA IP cores. Two general-purpose input/output (GPIO) interfaces that are specifically built for temperature and humidity sensors have been developed and used by us for this particular instance. Additionally, we have built a UART interface that is exclusive to the GPS module. This interface is utilised for navigation purposes alone [12]. For the purpose of storing TEDS data, a 64KB BRAM is created with the help of the resources provided by the FPGA.

Through the use of the extensive assortment of input/output peripherals that are made available by ARM, we directly utilise USB and Ethernet as high-speed interfaces. This application comprises the incorporation of digital cameras onto the board via the use of Ethernet, which enables the transit of data to be conducted more efficiently [13]. Furthermore, the system is able to create a connection with a 4G or Wi-Fi module via the use of a USB port, which enables the user to effortlessly access the network wirelessly [14]. Within the framework of the system architecture, the broad use of both analogue and digital interfaces ensures the availability of flexible connection capabilities and effective data exchange capabilities.

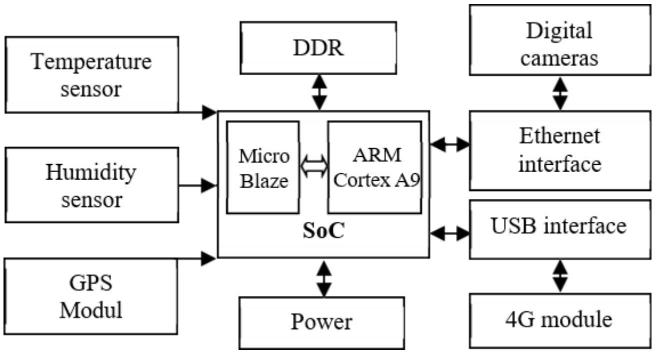

**Fig.4: The process of designing the hardware for monitoring systems for vehicles**

A diagrammatic illustration of the architecture of an embedded system, illustrating the many components and the relationships between them that are included inside a System on Chip (SoC). The operation is consolidated into the System-on-Chip (SoC), which combines a MicroBlaze central processing unit (CPU) and an ARM Cortex A9 processor, both of which are responsible for carrying out computational activities. A number of peripheral devices and sensors are connected to the System-on-Chip (SoC). These include sensors that measure temperature and humidity, a GPS module, DDR memory for data storage, digital cameras, and communication interfaces such as Ethernet, USB, and a 4G module. The components of the system, when combined, make it possible to perform a wide range of functions, including as detecting the environment, processing data, communicating with others, and managing media [15]. Because of its adaptability, the system is appropriate for applications in a variety of industries, including embedded multimedia systems, mobile computing, and the Internet of Things (IM).

Within the context of this application for monitoring vehicles, the terminal serves as a central hub for the collection, analysis, and transmission of environmental data. Using a strategic approach, we have divided the system into two distinct components, namely the data collecting module and the master module. This has allowed us to improve both the performance and the power efficiency of the platform [16]. To be more specific, we have included the sophisticated Standalone system into the data gathering module, while the master module makes use of the robust capabilities offered by the Linux kernel.

By using this architectural framework, we ensure that the system will function at its highest possible level while simultaneously reducing the amount of power that is used. Through the use of the lightweight Standalone system, the separation of functions make it possible to handle data collection operations in a quick and efficient manner. In addition to this, it provides the master module with the advanced capabilities of the Linux kernel, which enables it to do comprehensive data processing and transmission simultaneously.

In order to ensure that the data collecting module is in accordance with the IEEE 1451.2 standard, we have designed a specific software system that has been constructed with great care. This system guarantees that the collection, processing, and transmission of sensor data are carried out without issue. During the process of initialization, the system goes through the steps of commencing operations and then enters a standby state, where it waits for instructions from the master module. This is shown in Figure 5. Whenever the module is presented with a request signal, it promptly determines the nature of the request and promptly carries out the procedures that are required [17]. This module is responsible for effectively directing sensor data requests to the appropriate sensor, carrying out data retrieval activities, and finally transmitting the findings to the master module. In a similar manner, if there are requests for TEDS data, the module promptly directs them to the appropriate TEDS repository and promptly transmits the information that is required to the primary module. Additionally, we have included a self-testing mechanism in order to do a comprehensive evaluation of the overall condition of the module, therefore ensuring that it is reliable and maintains its performance integrity [18]. This operational structure allows for seamless integration and efficient data exchange between the master modules and the data collecting modules, resulting in a robust and responsive vehicle monitoring system that satisfies the standards of the industry.

The master module serves as the central control unit, which is responsible for keeping an eye on the administration of the system, as well as the numerous data processing operations and the seamless movement of data. The PetaLinux operating system, which includes a broad variety of device drivers and libraries that are intended exclusively for application development, has been included into our system in order to facilitate the streamlining of these processes that are vital. In accordance with this framework, we have established a specialist data acquisition agent that is accountable for receiving sensor data from the data acquisition module. This ensures that the data collection process is both efficient and reliable [19]. A video capture agent has been built in order to retrieve video data from digital cameras, which has resulted in an improvement in the monitoring capabilities of the system. We are now working on developing a transmission and management unit that will make the process of installing, maintaining, and transferring data in wireless networks more transparent and straightforward. Within the system, this component will play a significant role in ensuring that communication is not only seamless but also efficient and that data flows effectively. Figure 6 illustrates the software architecture of the master module, which displays the intricate network of agents and functions that were developed to improve the functionality and performance of the system.

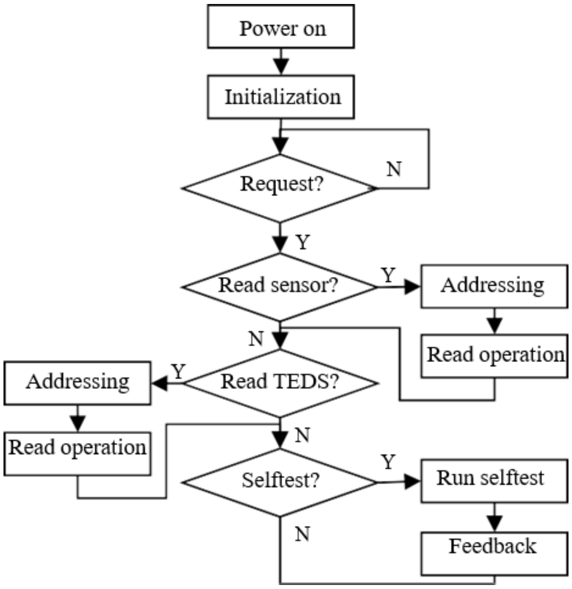

**Fig.5: The software's workflow inside the data acquisition module**

Beginning with the activation procedure, the flowchart illustrates the successive processes that are involved in the operation of a system. When the power is switched on, the system begins the process of running through its starting procedure. Immediately after the system is started up, it checks to see whether there is a request [20]. The method is terminated if there is no request; if there is a request, it determines whether or not the request is to get data from a sensor using confirmation. After the system has been given a request to read a sensor, it will first carry out an addressing procedure, and then it will continue with the read operation. It checks to see whether it is essential to read TEDS, which stands for transducer electronic data sheet, unless the request is to read a sensor at the same time [21]. In the event that the response is correct, the operational procedures for addressing and reading will continue to be the same. The system will determine whether or not a self-test has to be carried out if there are no TEDS that need to be read. When it is determined that a self-test is required, it is carried out, and then the system provides feedback on the results. It is also possible for the system to deliver direct feedback in the event that a self-test is not necessary.

A Framework for

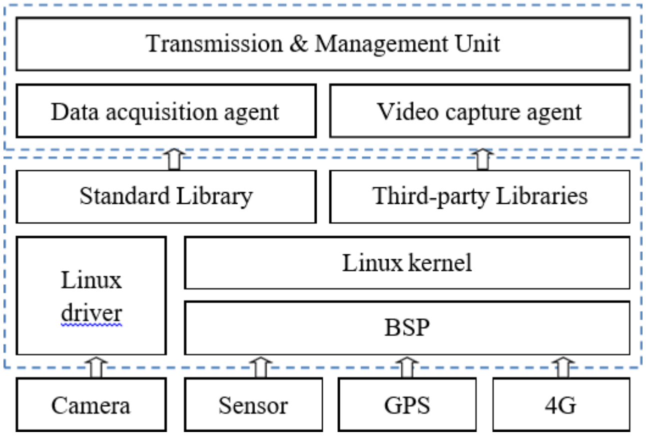

**Fig.6: The Architectural Design of the Master Module Software**

When it comes to handling transmission and administration chores, the architecture of an embedded system is largely geared to address these concerns. "Data acquisition agent" and "Video capture agent" are the two components that make up the "Transmission & Management Unit" at the very top of the hierarchy. Data and video streams are the respective responsibilities of these agents, who are responsible for collecting and processing them. Both the "Standard Library" and the "Third-party Libraries" are used by these agents in order to carry out their respective responsibilities. It is the "Linux kernel," which serves as the primary operating system, that is located at the very bottom of this software layer. It is accompanied by the "Linux driver" and the "BSP" (Board Support Package), both of which are essential for managing the hardware that is below it. For example, a "Camera," "Sensor," "GPS," and "4G" module are some of the hardware components that make up the lowest half of the system. The drivers that connect these components to the system make it possible for the system to carry out a variety of duties, including video recording, environment sensing, location tracking, and cellular communications, among others. The configuration in question is a complex and sophisticated system that was developed for the purpose of ensuring reliable data handling and smooth networking.

## IV. Evolution

Within the context of our real-world situation, we have successfully built and executed vehicle monitoring terminals, which have been installed on about one hundred buses that are now in operation. Every bus is outfitted with either five or six digital cameras, which are seamlessly integrated with the vehicle monitoring terminal. The number of cameras is determined by the kind of vehicle being monitored. In addition, GPS and 4G antennas have been strategically placed outside of the vehicles in order to enhance the capabilities of location tracking and data transmission. The vehicle monitoring terminal is shown in Figure 7, which may be consulted for better understanding. For the purpose of ensuring that the whole system functions well, we have put in place a robust server infrastructure as well as multi-client software. It is the responsibility of the server system to ensure that all of the data and information that is generated by the terminals is efficiently sent to the user clients. The TCP/IP protocol is responsible for facilitating the transfer of sensor data, whilst the Response Transfer Protocol (RTSP) protocol is responsible for delivering visual data. For the purpose of providing support for a large number of terminals, the server has been set up with a public IP address. Additionally, it has been designed with sufficient delivery bandwidth and robust real-time capabilities to effectively handle access from a lot of different users. On the client side, we have developed real-time video monitoring programmes and sensor data display programmes by using the application programming language C#. In addition to this, we have provided users and traffic controllers with a comprehensive monitoring and management platform that we have built. As can be seen in Figure 8, this platform has matrix displays that provide enhanced visualisation and control capabilities. Users are provided with the capacity to monitor in real time, and traffic controllers are provided with efficient management tools, all thanks to this all-encompassing strategy, which ensures the seamless running and administration of the vehicle monitoring system.

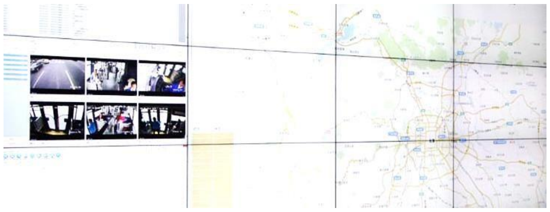

**Fig.7: A platform for the management and monitoring efforts**

Real-time and comprehensive monitoring of several buses is provided by the vehicle monitoring system. This monitoring includes the monitoring of variables such as temperature and humidity, as well as live location tracking and streaming video data. A user-friendly visual interface is used to display the vast amount of information, which assists in the effective management of traffic dispatch operations and the monitoring of security. At the same time, all of the information that has been acquired is stored safely in the database of the server, which ensures that it will be accessible in the future for purposes of analysis and reference. Since the system has been in operation for a considerable amount of time, it has consistently shown a high level of stability and consistently produced the results that were anticipated, so demonstrating its reliability and efficiency in real-world scenarios.

Using the intelligent interface design that is suggested, the system displays outstanding compatibility and scalability, which enables it to integrate with a wide variety of sensors, actuators, and other Internet of Things (IoT) devices in a smooth manner. This system's inherent adaptability makes it possible to easily integrate it with a broad variety of hardware components, which allows it to cater to the particular requirements of individual users and makes it easier to adapt to a variety of application scenarios. The vehicle monitoring system is a prime example of the effectiveness of intelligent interface design. It offers a solution that is both adaptable and dependable, allowing for the monitoring and management of transportation in a variety of scenarios in real time.

## V. Conclusion

In this study, an Internet of Things (IoT) access architecture is presented. This design makes use of the IEEE 1451.2 standard in order to facilitate communication amongst the many devices that are part of the IoT ecosystem. Our design ensures that there is continuous access to a broad variety of sensors, actuators, and high-speed devices, which satisfies the requirements for real-time applications that are present in Internet of Things environments. Through the strategic use of FPGA and SoC technologies, the system is able to deliver large processing capabilities, robust interoperability, and scalability, while simultaneously maximising the utilisation of hardware resources and the consumption of power. Our design has been subjected to performance validation within the context of an automobile monitoring system that has been put into action in real-world scenarios. Despite this, there are still interesting avenues that need more exploration. In the case of some Internet of Things devices, for instance, the current system permits wired communication; nevertheless, the provision of wireless access for other devices offers a substantial challenge. In our future attempts, we will make it a priority to resolve this component by working towards enhancing the flexibility of the system and making it possible to accommodate a wider variety of Internet of Things devices in a seamless manner.

## REFERENCES

[1] Bhatti, G., H. Mohan, and R. Raja Singh, Towards the future of ssmart electric vehicles: Digital twin technology. Renewable and Sustainable Energy Reviews, 2021. 141.

[2] Friha, O., et al., Internet of Things for the Future of Smart Agriculture: A Comprehensive Survey of Emerging Technologies. IEEE/CAA Journal of Automatica Sinica, 2021. 8(4): p. 718-752.

[3] Radoglou-Grammatikis, P., et al., A compilation of UAV applications for precision agriculture. Computer Networks, 2020. 172.

[4] Sishodia, R.P., R.L. Ray, and S.K. Singh, Applications of remote sensing in precision agriculture: A review. Remote Sensing, 2020. 12(19): p. 1-31.

[5] Du, R., et al., The Sensable City: A Survey on the Deployment and Management for Smart City Monitoring. IEEE Communications Surveys and Tutorials, 2019. 21(2): p. 1533-1560.

[6] Gupta, H., et al. iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments. in Software - Practice and Experience. 2017.

[7] Boursianis, A.D., et al., Internet of Things (IoT) and Agricultural Unmanned Aerial Vehicles (UAVs) in smart farming: A comprehensive review. Internet of Things (Netherlands), 2022. 18.

[8] Yang, Y., et al., Internet of things for smart ports: Technologies and challenges. IEEE Instrumentation and Measurement Magazine, 2018. 21(1): p. 34-43.

[9] Liu, L., X. Guo, and C. Lee, Promoting smart cities into the 5G era with multi-field Internet of Things (IoT) applications powered with advanced mechanical energy harvesters. Nano Energy, 2021. 88.

[10] Musleh, A.S., G. Yao, and S.M. Muyeen, Blockchain Applications in Smart Grid-Review and Frameworks. IEEE Access, 2019. 7: p. 86746-86757.

[11] Garg, S., et al., Edge Computing-Based Security Framework for Big Data Analytics in VANETs. IEEE Network, 2019. 33(2): p. 72-81.

[12] Gao, J., et al., A Blockchain-SDN-Enabled Internet of Vehicles Environment for Fog Computing and 5G Networks. IEEE Internet of Things Journal, 2020. 7(5): p. 4278-4291.

[13] Medvedev, A., et al. Waste management as an IoT-enabled service in smart cities. in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2015.

[14] Popescu, D., et al., Advanced UAV–WSN system for intelligent monitoring in precision agriculture. Sensors (Switzerland), 2020. 20(3).

[15] Lagkas, T., et al., UAV IoT framework views and challenges: Towards protecting drones as “things”. Sensors (Switzerland), 2018. 18(11).

[16] Said Mohamed, E., et al., Smart farming for improving agricultural management. Egyptian Journal of Remote Sensing and Space Science, 2021. 24(3): p. 971-981.

[17] Ali, S.S. and B.J. Choi, State-of-the-art artificial intelligence techniques for distributed smart grids: A review. Electronics (Switzerland), 2020. 9(6): p. 1-28.

[18] Azar, J., et al., An energy efficient IoT data compression approach for edge machine learning. Future Generation Computer Systems, 2019. 96: p. 168-175.

[19] Kanan, R., O. Elhassan, and R. Bensalem, An IoT-based autonomous system for workers’ safety in construction sites with real-time alarming, monitoring, and positioning strategies. Automation in Construction, 2018. 88: p. 73-86.

[20] Malekloo, A., et al., Machine learning and structural health monitoring overview with emerging technology and high-dimensional data source highlights. Structural Health Monitoring, 2022. 21(4): p. 1906-1955.

[21] A. Kumar, V. Srivastava, M. K. Singh, and G. P. Hancke, "Current Status of the IEEE 1451 Standard-Based Sensor Applications," IEEE Sensors Journal, vol. 15, pp. 2505-2513, 2015.