Nigerian Research Journal of Engineering and Environmental Sciences 7(2) 2022 pp. 429-435
p ISSN: 2635-3342; e ISSN: 2635-3350

Nigerian Research Journal of Engineering and Environmental Sciences
Journal homepage: www.rjees.com

Original Research Article

Application Interface for the Remote Monitoring of Node Energy Consumption using Wireless Sensor Network for Data Acquisition

***Ehizuenlen, P.E., Apeh, S.T. and Erameh, K.B.**

Department of Computer Engineering, Faculty of Engineering, University of Benin, Edo State, Nigeria.
*prudence.ehizuenlen@uniben.edu; apeh@uniben.edu; kachikwu.erameh@uniben.edu

http://doi.org/10.5281/zenodo.7496600

ARTICLE INFORMATION

Article history:
Received 30 Nov. 2022
Revised 15 Dec 2022
Accepted 16 Dec 2022
Available online 30 Dec, 2022

Keywords:
Wireless sensor network
Remote monitoring
Application interface
Data acquisition
Energy consumption

ABSTRACT

Wireless sensor nodes are mostly deployed in remote locations which are usually difficult for humans to access. This makes it almost impossible for the behavior of the node to be properly monitored particularly regarding the rate of its energy consumption. However, this can be monitored remotely using an application (app) interface domiciled in a remote location for example a base station. This paper describes the design and operation of the app interface for the remote monitoring of node energy consumption using a wireless sensor network for data acquisition. The app interface operates a two-algorithm mode with the custom sensor wireless node continuously obtaining sensed data and forwarding it to the remote location through a wireless sensor network. A context-aware and energy-efficient data acquisition reconfiguration algorithm (CAEEDARA) also known as the power saving algorithm allows adaptive sampling frequency and the interval was developed and run alongside the normal node algorithm which employs fixed sampling frequency and interval. The app interface allows for the switching between operating modes depending on what is most appropriate for the deployment environment. The app interface was tested using a historical dataset obtained for climate change monitoring for three gases namely CO_2, CH_4, and NO_2. The result of the operation showed that the developed app interface was able to efficiently provide real-time information concerning the rate of its energy consumption during data acquisition.

© 2022 RJEES. All rights reserved.

1. INTRODUCTION

A wireless sensor network (WSN) is a network with dedicated sensors that helps in monitoring and keeping records of physical quantities such as temperature, wind, humidity, and pollution levels. Some of the application fields where WSNs are usually deployed include object tracking, agriculture, and environmental monitoring. A WSN comprises three key components which include the sensor node(s), sink, and base station (BS). The sensor node is responsible for the conversion of the detected physical quantities into the appropriate electric signals. These signals are then centrally collated by the sink and analyzed by the BS (Bhardwaj, 2013). Nodes are known to have inherent constraints such as limited energy, limited memory and storage space, and in-network processing with energy management given the topmost priority (Sharma *et al.*, 2013). WSN designs and implementations have become common due to their compact size, low cost of deployment, and ability to communicate wirelessly. Also notable is its ability to operate in a self-organized manner as they are often deployed in remote environments that are partially accessible or completely inaccessible to humans (Wu and Ding, 2021). However, when deployed in such an environment, it is often a challenge to monitor the state or behavior of the node(s), particularly concerning its rate of energy consumption. This concern can be addressed through remote monitoring.

Remote monitoring (RM) can be described as the ability to visualize the status of equipment, performance, and behavior in a network such as a WSN through the use of remote devices known as monitors from a centralized location. Remote devices usually communicate using a combination of wireless and internet of things (IoT) technologies. RM is designed to specify a group of functions that must be communicated to the human personnel or administrator present at the centralized location. RM employs the use of certain network devices such as servers that contain network management applications that act as clients. This allows control of the network using both the servers and applications simultaneously. The RM station can act like it was connected to a single monitor at another point of the network which enables it to gather summarized data obtained from the remote devices (Walker *et al.,* 1993). Implementing a remote monitoring system comes with benefits such as access to real-time data, coverage of a wide area space, ensuring long-term monitoring of the deployment area, fault detection, and system scalability (Jayaprakash, 2019).

Energy is a critical constraint in the design and implementation of sensor nodes as they are mostly battery-powered (Cardei *et al.*, 2002). To achieve a reasonable node lifetime, its power-consuming units which include sensing, computing, and communicating have to be minimized (Ogbiti *et al.,* 2016). Most research focused on conserving energy for the computing and communicating units on the assumption that they consume the most energy (Benini and Micheli, 2002) with sensing being under-explored (Patel and Shah, 2018). However, this assumption does not hold for all applications especially for those with longer acquisition time in comparison with the transmission time, thus resulting in significantly high energy consumption (Alippi *et al.*, 2010). Qi *et al.* (2013) also supported the claim that there are applications that have sensors that can consume more energy during data acquisition (sensing) than in communication. Hence, the need to give serious consideration to the rate with which energy is consumed during data acquisition regardless of how little its contribution is in comparison with communication and computation.

This paper focuses on the design and operation of the app interface for the remote monitoring of node energy consumption using a wireless sensor network for data acquisition.

2. METHODOLOGY

The flow chart employed for the implementation of this work is presented in Figure 1. Figure 1 shows that the design implementation involves two stages namely the design of the custom node and the design of the application interface.

1. Node design: The node architecture was designed using GasCard NG used for the detection of CO_2 and CH_4, and Graphene used for the detection of NO_2 as sensors, MSP430F2272 microcontroller, and a ZigBee communication module. The choice of these components is majorly due to their low power consumption.
2. Application interface design: The development of the app interface was done using Visual Studio 2015 running on a Dell Intel Core i3 processor.

Ehizuenlen, P.E et al. / Nigerian Research Journal of Engineering and Environmental Sciences
7(2) 2022 pp. 429-435

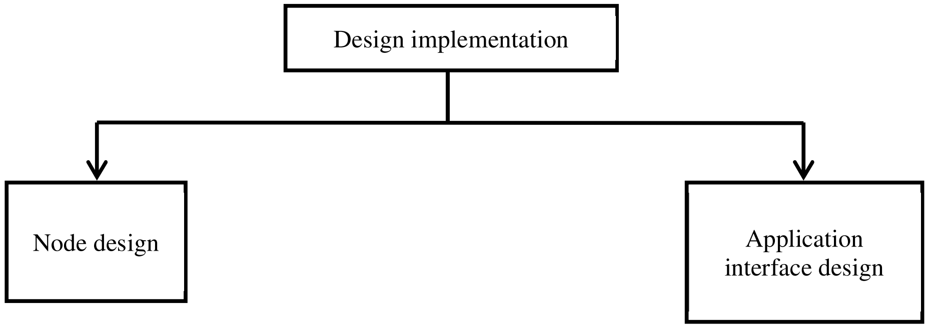

Figure 1: Flow chart for the design implementation

Nine-year historical dataset was obtained from AMiner database for three gases namely; carbon dioxide (CO_2), methane (CH_4), and nitrogen dioxide (NO_2). The dataset was analyzed using RStudio an exploratory data analysis tool to obtain a proper understanding of how the values are spread out in the dataset. The results of the analysis were then used to determine the gas threshold. A context-aware and energy-efficient data acquisition algorithm (CAEEDARA) running on the BS at the remote location was made to communicate with the node so that it can coordinate its behavior during sensing activity. The coordination includes adapting sampling frequency and sampling interval based on a change in input characteristics and its available battery energy (context). A standard algorithm also known as the normal algorithm which operates with a fixed sampling frequency and sampling interval was also made to communicate with the BS. The app interface allows an administrator to switch between algorithms depending on the nature of the behavior of the deployment environment. In both cases, the app interface provides real-time data of the node power rating in watts and rate of power consumption in milliamp hour.

The developed application interface as illustrated in Figure 2 comprises of five (5) modules that provide support for the node operation and they are data, node, microcontroller, base station, and power.

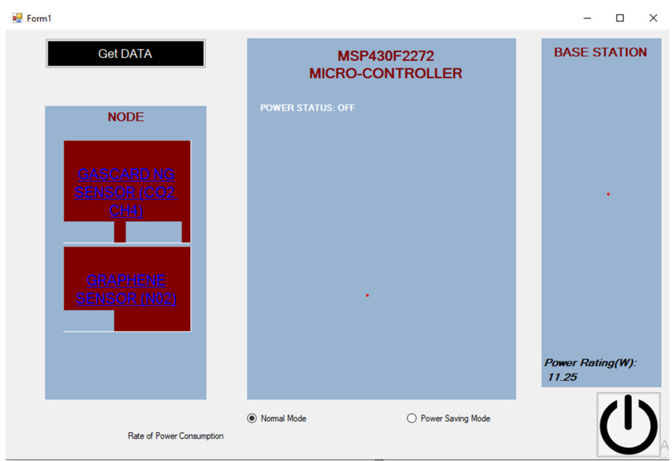

Figure 2: Remote location application interface

The data module provides the relevant data required by the two sensors used in the node design and interfaced with the microcontroller using the developed algorithm provided by the BS from the remote location. At startup, a power of 11.25 W was supplied to the circuit components by the power module, with the administrator at liberty to switch mode (Power Saving or Normal) based on the deployment environment and the nature of the gas being investigated. The application interface allows the administrator to obtain firsthand the rate with which power is consumed operating on either mode without being physically present in the monitoring field.

Ehizuenlen, P.E et al. / Nigerian Research Journal of Engineering and Environmental Sciences
7(2) 2022 pp. 429-435

3. RESULTS AND DISCUSSION

The result of the deployed method is the development of a multi-algorithm remote application interface that allows a switch between two algorithms depending on the behavior of the monitored environment. At startup, the app interface selects the Normal mode as shown in Figure 2. Once data are fed into it from the data module, the mean values obtained from the descriptive analysis of the historical dataset are set on display on the BS module as illustrated in Figure 3.

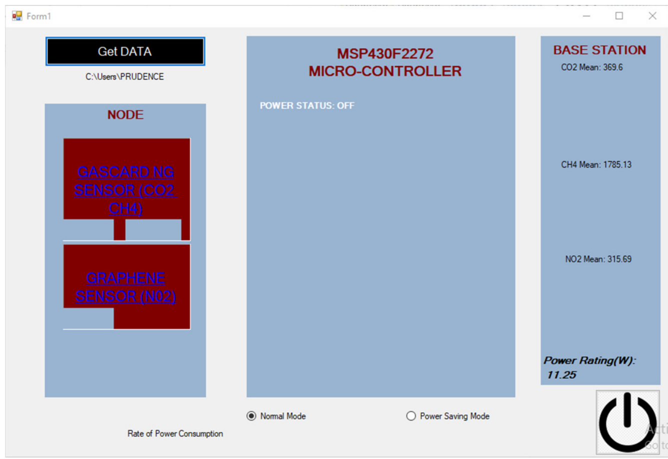

Figure 3: Application interface with mean values on Normal Mode

These mean values stored on the BS serve as the threshold values used in determining the extent of possible deviation within the deployment environment. To initiate a node activity, the power button is turned ON, the sensors continuously obtain sensed data as seen in Figure 4.

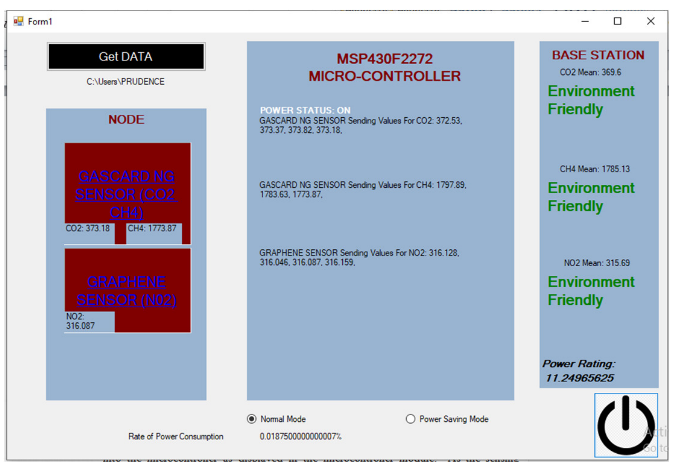

Figure 4: Application interface sending values at iteration startup on Normal Mode

Figure 4 shows that the node activity has begun with the sensors feeding their sensed data into the microcontroller as displayed in the microcontroller module. As the sensing operation continues, the node power begins to drop from its initial startup value of 11.25 W as seen in Figure 2 to 11.24965 W as shown in Figure 3. It can also be observed that the rate of power consumption of the battery energy at this stage of the node operation is said to be about 0.01875% of the node battery energy. The BS module displays the status of the individual gas concerning the mean value to indicate if the gas concentration in the monitored environment is 'Environment Friendly' which implies that the gas concentrations are within the standard limit. The node operation is expected to continue until its battery energy is completely drained, however, due to the dynamic nature of the environment the concentration of the monitored gas varies due to environmental or human factors. In such instances, the developed app notifies the administrator with a change in status as 'Environment Non-Friendly' as shown in Figure 5.

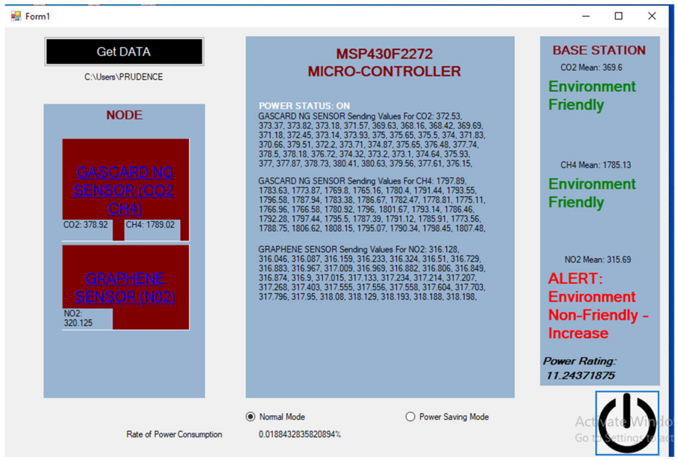

Figure 5: Application interface for an increase in gas concentration

Figure 5 shows that there is an increase in the gas concentration for NO_2, with a drop in the battery power to 11.2429 W and the rate of power consumption at this stage of the operation is about 0.01884% of the battery energy. The increase in gas concentration signals the unsafe state of the monitored environment which has to be addressed if the human presence or continuous stay would be permitted. On the receipt of the Environment non-friendly alert, the administrator immediately proceeds to take appropriate measures to clean the monitored environment and make it safer and habitable. However, on such notice, the BS after some series of sampling shut downsampling from that sensor since it has already been established that the concentration of gas in that said location is above the threshold value predefined. The sensor will only begin to send sampled data alongside the other sensor once the monitored environment has been rid of the excess gas concentration.

Considering an environment where adaptive sampling is most desirable, the app interface can be switched to operate in the power-saving mode as can be seen in Figure 6.

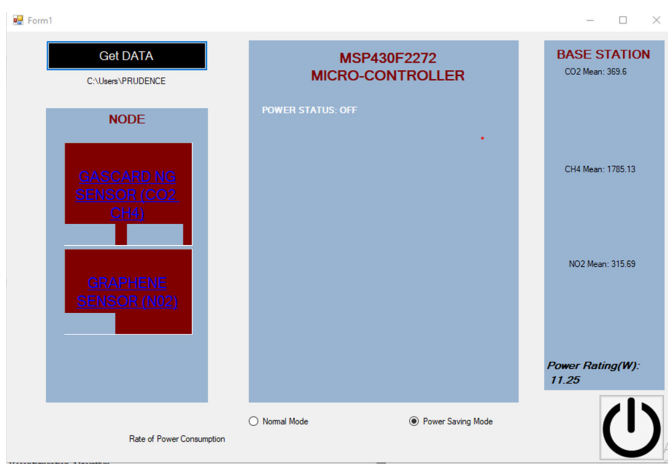

Figure 6: Application interface running on the Power Saving Mode

Like in the normal operating mode, once data are fed into it from the data module, the mean values obtained from the descriptive analysis of the historical dataset were set on display on the BS module. To initiate a node operation, the power button is turned ON, the sensors continuously obtain sensed data as displayed in Figure 7. The sensors will continue to obtain sensed data and forward to the remote location until an unusual increase in gas concentration is detected, which the administrator is expected to address before sampling continues.

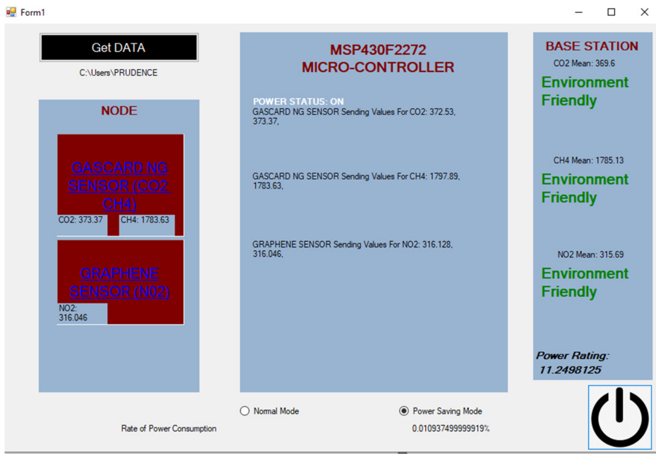

Figure 7: Application interface sending values at iteration startup on Power Saving Mode

4. CONCLUSION

This paper describes the design and operation of an app interface that allows for the remote monitoring of energy consumption during the data acquisition activity of a node. With the developed app interface, a remote administrator is capable of obtaining first-hand information concerning the current power rating of the battery as well as the rate with which energy is being consumed without being physically present in the monitored environment. This eases the administrator of the unsafe and harsh conditions prevalent in most wireless sensor network deployment environments. The developed app interface was validated using historical data obtained from climate change monitoring of CO_2, CH_4, and NO_2 gases. The results obtained from the demonstration of the developed app interface operation show that it can provide the energy consumption information required by the administrator resident at a remote location for appropriate decision-making. A sensor may suddenly fail due to inherent design issues or harsh operating weather conditions within the deployment environment, failing to report the sensed data to the remote location. Based on this, future work will consider reporting outright sensor failure to the remote location for a prompt replacement to ensure continuous environmental monitoring.

5. ACKNOWLEDGEMENT

The authors wish to acknowledge the assistance and contributions of the academic staff of the Department of Computer Engineering, Faculty of Engineering, University of Benin, Benin City toward the success of this work.

6. CONFLICT OF INTEREST

There is no conflict of interest associated with this work.

REFERENCES

Alippi, C., Anastasi, G., Di Francesco, M. and Roveri, M (2010). An Adaptive Sampling Algorithm for Effective Energy Management in Wireless Sensor Networks with Energy-Hungry Sensors. *IEEE Transactions on Instrumentation and Measurement,* 59(2), pp. 335-344.

Benini, L. and Micheli, G.D. (2002). *Dynamic Power Management: Design Techniques and CAD Tools*. Norwell MA, Kluwer, pp. 23-34.

Bhardwaj, S. (2013). Artificial Neural Network for Node Localization in Wireless Sensor Network. *International Journal of Advanced Research in Electrical, Electronics and Electrical and Instrumentation Engineering,* 2(5), pp. 1724-1731.

Cardei, M., Maccallum, D., Cheng, X., Min, M., Jia, X., Li, D., Du, D. and Du, D. (2002). Wireless Sensor Networks with Energy Efficient Organization. Technical Report, Department of Computer Science and Engineering, University of Minnesota, pp. 1-12.

Jayaprakash, K. (2019). Remote Monitoring of Power Plants. *ViewPoint*, pp. 44-47.

Ogbiti, J.T., Ukwuoma, H.C., Danjuma, S. and Ibrahim, M. (2016). Energy Consumption in Wireless Sensor Network. *Computer Engineering and Intelligent Systems*, 7(8), pp. 63-67.

Patel, H. and Shah, V.A. (2018). Sensor Node Design with Dynamic Remote Reconfiguration and Analysis. *International Journal of Applied Engineering Research,* 13(6), pp. 3415-3424.

Qi, X., Keally, M., Zhou, G., Li, Y. and Ren, Z. (2013). AdaSense Adapting Sampling Rates for Activity Recognition in Body Sensor Networks. Proceedings of the 2013 IEEE 19th Real-time and Embedded Technology and Applications Symposium (RTAS), pp. 163-172.

Sharma, S., Bansal, R.K. and Bansal, S. (2013). Issues and Challenges in Wireless Sensor Networks. *International Conference on Machine Intelligence Research and Advancement*, Katra, JK, India. 21-23 December 2013, pp. 58-62.

Walker, H., Law, G., Keene, A.C., Russell, J.P. and Green, D. (1993). *Telecommunication Engineer's Reference Book,* pp. 1-61.

Wu, J. and Ding, X. (2021). Using Wireless Sensor Network to Remote Real-Time Monitoring and Tracking of Logistics Status Based on Difference Transmission Algorithm. *Journal of Sensors*, 2021, pp. 1-10.