2021 6th International Conference for Convergence in Technology (I2CT)
Pune, India. Apr 02-04, 2021

Design and Development of Wireless Sensor Network based data logger with ESP-NOW protocol

Koushik M S
RNS Institute of Technology
Bangalore, India
mskoushik98@gmail.com

Srinivasan M
Dayananda Sagar Academy of Technology and Management
Bangalore, India
srinivasanm329@gmail.com

Lavanya R
Dayananda Sagar Academy of Techno0logy and Management
Bangalore, India
lavanya4520@gmail.com

Alfred S
Dayananda Sagar Academy of Technology and Management
Bangalore, India
salfredpeter789@gmail.com

Dr. Sreenivasa Setty
Dananda Sagar Academy of Technology and Management
Bangalore, India
sairamsst@gmail.com

***Abstract* —This paper presents Wireless Sensor Network (WSN) based data logger system for collecting data accurately from agricultural field. Agriculture is the backbone of India. Though many technological advances have been made in various fields, not much of it is implemented in the field of agriculture. One of the fundamental aspects that we can improve in agricultural field is by getting more data from soil, atmosphere, plants, etc accurately. By collecting more data we can apply machine learning and prediction algorithms to figure out the amount of fertilizer, water and other manures required for crops to get better yield. But in developing countries like India, cost of the technology is quite not affordable for implementing in large scale. Hence an effective and low cost real time monitoring system is needed, to monitor and collect data accurately. Wireless Sensor Network (WSN) based data logger system with ESP-NOW protocol developed by us aims at doing just that and the project is discussed here.**

Keywords—wireless sensor network, ESP-NOW, agriculture, Internet of Things, electronic device, firmware, firebase, sensor node, sink node, development

I. INTRODUCTION

One of the most important sectors of Indian economy is Agriculture; 18% of India's Gross Domestic Production (GDP) is accounted by agriculture and Indian agriculture provides employment to 50% of the country's workforce [1]. As India's population is increasing, the demand for food is also increasing. This creates a demand for more land to farm and cultivate food products. To strengthen any country, advancement of technology must be used effectively not only in defense sector but also to improve agricultural produce. Though there is advancement in weather prediction with help of AI (Artificial intelligence) and ML (Machine Learning), implementing a technology for agriculture in a larger scale is very difficult. Mainly because of these 3C's: Cost of implementing the technology; Connectivity of technology and Complexity of the technology. Hence we built a suitable prototype trying to circumvent afore mentioned problems. This project is easily scalable in terms of land area, number of users and location.

II. OUTLINE OF THE PROJECT

Our project works on the concept of wireless sensor network (WSN) where a sensor node collects data from different sensors like humidity, temperature, and moisture content of the soil. A sink node (or base station or master node) which collects the data from multiple similar sensor nodes and sends to a cloud database where the data can be visualized and processed for prediction or to automate and control real-time machineries.

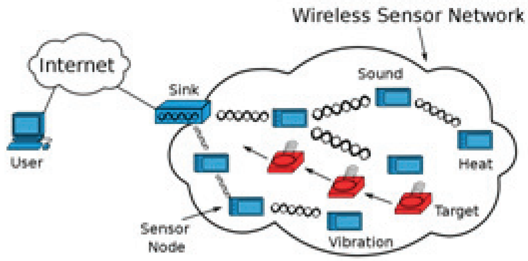


Fig. 1. Overview of Wireless Sensor Network (WSN)

The above figure ("Fig. 1") outlines the working of Wireless Sensor Network (WSN). Sensor node which is made up of various sensors is connected to a sink node to communicate with users over the internet.

Advantage of our device is sink node is independent of the number of sensor nodes, so it is very easy to increase the type/number of sensors without any modification to the sink node. For the communication part, there is a proprietary protocol based on IEEE 802.11 vendor-specific action frames called ESP-NOW [2]. After the data is transmitted from the sensor node, the sensor node can be inactive. After the time interval set by the user has elapsed, the data communication process will repeat. Using this protocol, the sensor node consumes only 170mA while sending the data and during idle state the device consume 80mA, so with the current implementation the device can run for 54.922 hours theoretically or for approx. 49 hours practically.

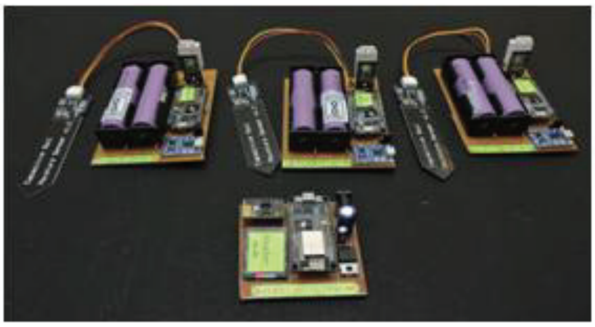


Fig. 2. IoT based Agriculture system*(overview)*

"Fig. 2" shows the complete picture of our working prototype of WSN based data logger which consists of sensor nodes and sink node

The sink node consists of a master receiver (NodeMCU) which receives the data from sensor nodes and transmits this data to a Wi-Fi module (ESP8266-01) via UART, which in turn transfers the data to the internet. The advantage of having these two separate systems is that it helps to keep the system highly modular. Incase this WSN based data logger needs to be installed in a remote area where internet is inaccessible, the master receiver can still receive data from the sensor node without any modification and instead of transferring the data to internet, we can transfer the data to a local server or a Raspberry Pi which can do the required computation locally and show the real-time data to the user via a local network or a physical dashboard.

The sink node also supports local storage via a SD card. This helps when there is no possible way to transfer the collected data; we can manually access the SD card and see the data (not real time) or transfer the data where the computation can be done.

Our sensor nodes consist of two sensors and a microcontroller. Two batteries are used to power the circuit. Although sensor node can contain any sensor the user wishes, we have used DHT22 sensor and capacitive moisture sensor.

III. HARDWARE IMPLEMENTATION

A. Sensor node

We used NodeMCU for the microcontroller of the sensor node which has integrated Wi-Fi and many common additional peripherals like SPI, UART and I2C which most of the sensors use to communicate. Along with this, two 18650 2200mAh batteries are used to power the circuit and a TP4056 charging circuit is used to protect the lithium ion battery from over/under charging as well as from short circuiting. For the sake of proof of concept, we have used very simple sensors which are easily available. DHT22 sensor, which can measure the temperature and humidity of the surrounding and send it to a microcontroller via one wire communication. Then to measure the soil moisture we used capacitive moisture sensor which outputs the data in analog form.

Fig. 3. Hardware of sensor node

"Fig. 3" shows the one of the sensor nodes collecting data. The sensor node contains afore mentioned sensors.

The sensor node contains two 18650 lithium ion rechargeable battery connected in parallel to give storage of 4400mAh. These batteries are charged through a TP4056 module which acts as both charging circuit as well as lithium ion battery protection circuit for the batteries.

Reasons why we choose DHT22 and capacitive moisture sensor:

- DHT22 can measure both temperature and humidity simultaneously and can read data more accurately and is less prone to error than DHT11 sensor.
- Capacitive moisture sensor is used instead of any regular resistive based moisture sensor because the former prevents the corrosion of the sensor and gives more accurate readings.

The sensor node is also modular. Many more sensors can be added and with very minimal change in communication part of the code, sensor node can transfer the data to the master (sink) node without a hitch.

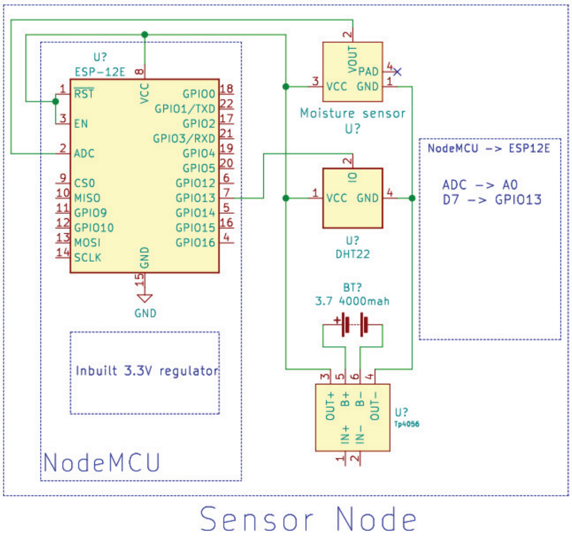


Fig. 4. Circuit diagram of sensor node

The circuit diagram of the sensor node as shown in "Fig. 4". All other sensor nodes are constructed similar to this. For getting data over a large area many sensor nodes are required. We were able to construct a total of three sensor nodes.

B. Sink node

For the sink node, NodeMCU is used as microcontroller because the ESP NOW protocol works only between the family of ESP which is developed by Espressif. Along with the microcontroller a ESP8266-01 is used to transfer the collected data from the microcontroller to the firebase database. Even though the Wi-Fi is integrated with NodeMCU an external Wi-Fi module is used because if there is a situation where Wi-Fi communication is not possible, we can very easily switch over to other communication type [GPS, LORA, ZIGBEE, etc] since we used UART communication between the microcontroller and the Wi-Fi module.

If there is any interruption while sending the data to the firebase real-time database, there is a SD card connected to the microcontroller through the SPI bus which stores the data offline. Later this data can be directly analysed or simply imported to the firebase (real-time database) since the data is stored in a JSON format. A simple regulator circuitry is also added for ease of use with variety of multiple DC power supply varying from 6V-32V. Though we recommend using 6V because greater the voltage more power will be dissipated by the linear regulator IC7805.

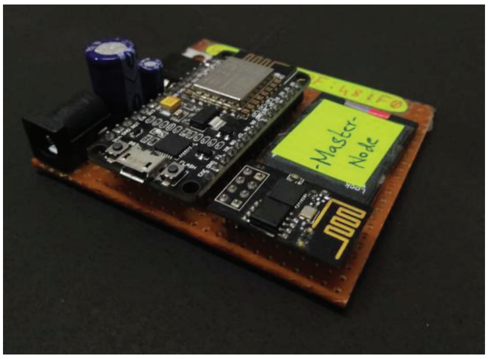


Fig. 5. Hardware of sink node

"Fig. 5" shows the hardware of sink node. This node is the master node. It collects the data from all the sensor nodes and writes in the SD-card as well as transmits to firebase.

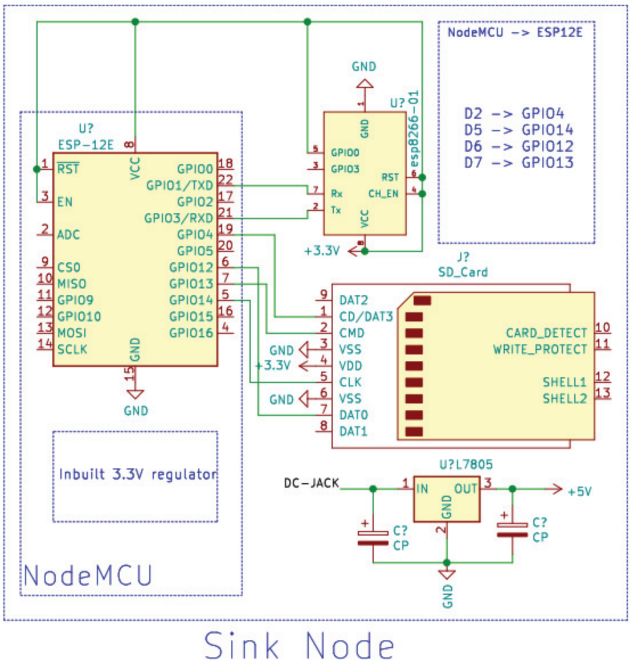


Fig. 6. Circuit diagram of sink node

The above figure ("Fig. 6") shows circuit diagram of sink node which depicts the connections amongst NodeMCU, SD-card, ESP8266 and Voltage regulator. For powering the circuit, a DC power supply is used, which is then regulated using a 7805IC which can handle a range of power from 6-32V.

The master is constantly powered so, it can receive data anytime from any sensor node and these sensor nodes can be programmed to send data at any interval.

IV. Battery Life Calculation and Battery Management

In the experiment performed data was sent once every one hour from the sensor node to the master node and the current consumed by the device was measured to be 170mA for 40ms during transmission of data and when the device was idle, it consumed 80mA.

The average current consumed by sensor node can be calculated using the formula,

$$I_{Avg} = \frac{(I_r \, X \, T_r)}{3600} + \frac{(I_i \, X \, T_i)}{3600} \quad mA$$

Where,

I_{Avg} is Average current consumption

I_r is Run time current consumption

T_r is Running time in seconds

I_i is Idle time current consumption

T_i is Idle time in seconds

And 3600 is the number of seconds in one hour.

$$I_{Avg} = \frac{(0.170 \, X \, 0.040)}{3600} + \frac{(0.080 \, X \, 3599.960)}{3600} \quad mA$$

$$\boxed{I_{Avg} = 80.114 \text{ mA}}$$

And the total time the sensor node will operate once it's fully charged is given by Run Time. It can be calculated by using the formula,

$$T_r = \frac{(Ba \, X \, I_{Avg})}{3600}$$

Where,

T_r is Run time in seconds

Ba is Battery Capacity in mAh

I_{Avg} is Average current consumption

$$T_r = \frac{(4400 \, X \, 80.114)}{3600}$$

$$\boxed{T_r = 54.922 \text{ hours}}$$

Albeit the battery can run the sensor node for 54.922 hours theoretically, but in practical experimental setup the battery lasts for 49 hours (approx.). If we can introduce the light sleep within NodeMCU where it turns off the Wi-Fi the current consumption of the whole system drops to 10mA, so theoretically the battery can run the sensor node for 435.356 hours i.e. approx. 18 days and 3 hours. With little more optimization, the circuit can control the sensor current and by adding deep sleep mode to NodeMCU the battery life can be prolonged further.

V. Firmware Implementation

The coding for this prototype was done using an Arduino IDE with a ESP8266 core (ESP8266 core is explained in detail in [3]) installed along with it. To read the sensor data in the sensor node Adafruit DHT sensor library for DHT sensor and a simple analog read was used for the Moisture sensor. To send data from sensor node to sink node ESP NOW was implemented [4].

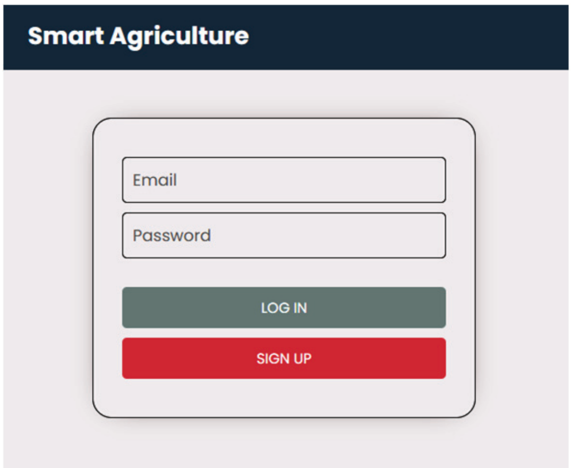
A structure (struct) was created in the code for both sensor node and sink node which has exact data types and size of the structure. This structure is passed with data from the sensor node, along with the mac address and a ID to find from which sensor node the data is received. Once the data is received, the data is parsed inside a JSON, then the JSON object is sent to firebase using Firebase-ESP8266 by Mobizt [5]. The dynamic nature of adding more nodes is taken care by Sink node, as it can recognize different nodes, it creates a new JSON Object with a different node name and then this data is pushed to firebase.

VI. Website Development

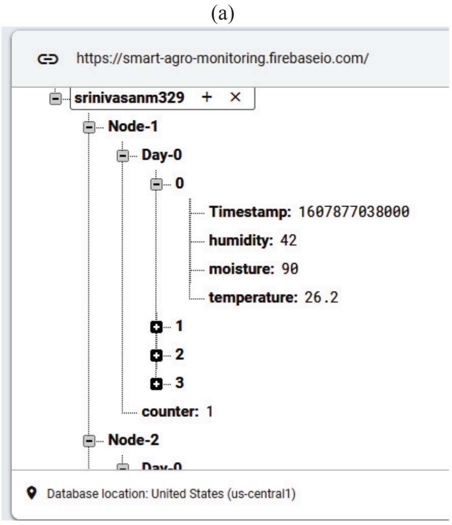
An informative dashboard is presented to every user that Logs-In, which is associated with the user Id in the hardware device. This dashboard contains data/graphs of all sensor data that is accumulated by the sensors and sent by the sink node. These data can be used to make agricultural choices like what crops would grow better or machine learning algorithms can be applied to predict the climate change or can be used to calculate the required fertilizers and amount of water.

A. Web Technology used

The Web Application is built using HTML, CSS and Vanilla Javascript. To display the graphs and data Chart.js library which gives easy access to plot, change or decorate the graph which is on the dashboard.

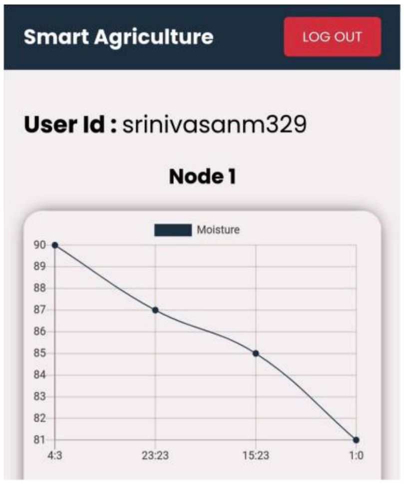


(a)

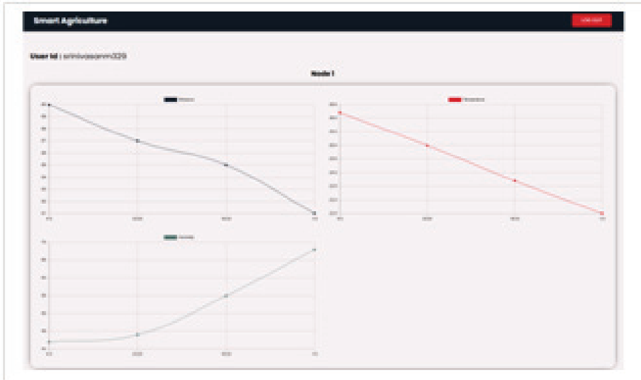


(b)

Fig. 7. Image shows (a) Login page (b) Firebase Real-time database



(a)



(b)

Fig. 8. Graph of the collected data in (a) Mobile view (b) Desktop View

“Fig. 7” shows the log in page and the collected data under each node, every day and every hour in the firebase.

“Fig. 8” shows the graph in both mobile view and desktop view.

B. Firebase

Firebase is a platform developed by Google for creating mobile and web applications[6]. Firebase was chosen since it is easier to build mobile application from the existing database and hosting is simpler as well.

Firebase Authentication makes it simple and secure for the user to Login using email and view their statistics. This can also be scaled easily to other login methods, like logging in using Facebook or phone number or any other.

Real-time Database is used to store the stream of data that is coming from the sensors. Since the database is real time, any change in the data will be instantly reflected on the dashboard.

VII. Result

The result of the experiment performed showed good potential to be adapted in real world. The data obtained after performing the experiment is as tabulated below.

TABLE I. TABULATION OF COLLECTED DATA ON TESTING

Display of collected data	Node - 1			
	Day - 0			
	At 4:30pm	At 5:30pm	At 6:30pm	At 7:30pm
Timestamp (UNIX Epoch seconds)	1607877030	1607880630	1607884230	1607887830
Humidity (in %)	42	44	55	68
Moisture (in %)	90	87	85	81
Temperature (in ^{o}C)	26.2	25	23.7	22.5

The collected humidity of surrounding, moisture content of the soil and ambient temperature is as tabulated in “Fig.9”. The data was collected from sensor node – 1 on the first day (day 0). The sensor adds timestamp to record the time at which this data was collected. The timestamp is of Unix Epoch system. It describes a specific point in time. It can be translated as the respective column heading. Like how the data is tabulated from first node, similarly data can be received from the rest of the nodes and can be sent to the master node.

The firmware developed has very dynamic nature of adjusting to multiple nodes. As the number of nodes in hardware grows or addition of different sensors in the existing sensor node can be handled by the master node with very little or almost no changes to be made in the firmware. Accessing the data by the user is also very flexible as Firebase Hosting is used, which can be accessed all over the world.

VIII. CONCLUSION

Conclusion can be drawn that advancement in the field of agriculture is not only vital with India’s growing population but is also long overdue. Hence our system aims at monitoring and collecting data accurately which can be further subjected to any form of AI or ML algorithms [7] to make use of the collected data. By using Wireless Sensor Network (WSN) based data logger system with ESP-NOW protocol our project was able to collect data in a dynamic and flexible way. This project is not only efficient but also quite practical.

ACKNOWLEDGMENT

We would like to thank Dr. Sreenivasa Setty sir for guiding our project. In addition, we thank our families and friends for their support.

REFERENCES

[1] Bhatnagar R., Gohain G.B. (2020) Crop Yield Estimation Using Decision Trees and Random Forest Machine Learning Algorithms on Data from Terra (EOS AM-1) & Aqua (EOS PM-1) Satellite Data. In: Hassanien A., Darwish A., El-Askary H. (eds) Machine Learning and Data Mining in Aerospace Technology. Studies in Computational Intelligence, vol 836. Springer, Cham. https://doi.org/10.1007/978-3-030-20212-5_6

[2] https://www.espressif.com/en/products/software/esp-now/resources

[3] https://arduino-esp8266.readthedocs.io/en/latest/

[4] https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/network/esp_now.html

[5] https://github.com/mobizt/Firebase-ESP8266

[6] https://en.wikipedia.org/wiki/Firebase

[7] R. Singh, S. Srivastava and R. Mishra, "AI and IoT Based Monitoring System for Increasing the Yield in Crop Production," 2020 International Conference on Electrical and Electronics Engineering (ICE3), Gorakhpur, India, 2020, pp. 301-305, doi: 10.1109/ICE348803.2020.9122894.

[8] Chetan Dwarkani M, Ganesh Ram R, Jagannathan S and R. Priyatharshini, "Smart farming system using sensors for agricultural task automation," 2015 IEEE Technological Innovation in ICT for Agriculture and Rural Development (TIAR), Chennai, 2015, pp. 49-53, doi: 10.1109/TIAR.2015.7358530.

[9] J. Lee, Y. Su and C. Shen, "A Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee, and Wi-Fi," IECON 2007 - 33rd Annual Conference of the IEEE Industrial Electronics Society, Taipei, 2007, pp. 46-51, doi: 10.1109/IECON.2007.4460126.

[10] Das, Aditya & Popa, Dan & Ballal, Prasanna & Lewis, Frank. (2009). Data-logging and supervisory control in wireless sensor networks. IJSNet. 6. 13-27. 10.1504/IJSNET.2009.028022.

[11] I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[12] I. Mat, M. R. Mohd Kassim, A. N. Harun and I. M. Yusoff, "Smart Agriculture Using Internet of Things," 2018 IEEE Conference on Open Systems (ICOS), Langkawi Island, Malaysia, 2018, pp. 54-59, doi: 10.1109/ICOS.2018.8632817.

[13] D. Vasisht, Z. Kapetanovic, J.-h. Won, X. Jin, R. Chandra, A. Kapoor, et al., "Farmbeats: An IoT Platform for Data-driven Agriculture", Proc. of the 14th USENIX Conference on Networked Systems Design and Implementation (NSDI'17), pp. 515-528, 2017.

[14] D. Yukhimets, A. Sych and A. Sakhnenko, "Designing a Method for Constructing Distributed Open ACS Based on the ESP-NOW Wireless Protocol," 2020 International Russian Automation Conference (RusAutoCon), Sochi, Russia, 2020, pp. 642-647, doi: 10.1109/RusAutoCon49822.2020.9208135.

[15] Pertab Rai and Murk Rehman, "ESP32 Based Smart Surveillance System", 2nd Int. Conf. on Computing Mathematics and Engineering Technologies, 2019.

[16] M. Mafuta, M. Zennaro, A. Bagula, G. Ault, H. Gombachika and T. Chadza, "Successful deployment of a wireless sensor network for precision agriculture in Malawi", Int. J. Distrib. Sensor Netw., vol. 2013, Apr. 2013.

[17] Nattapol Kaewmard and Saiyan Saiyod, "Sensor data collection and irrigation control on vegetable crop using smart phone and wireless sensor networks for smart farm", IEEE Conference on Wireless sensors (ICWiSE), pp. 106-112, 2014.

[18] H. Sahota, R. Kumar and A. Kamal, "A wireless sensor network for precision agriculture and its performance", Wireless Commun. Mobile Comput., vol. 11, no. 12, pp. 1628-1645, Dec. 2011.

[19] A.-u. Rehman, A. Z. Abbasi, N. Islam and Z. A. Shaikh, "A Review of Wireless Sensors and Networks' Applications in Agriculture", Computer Standards & Interfaces, vol. 36, no. 2, pp. 263-270, 2014.

[20] Vinayak N. Malavade and Pooja K. Akulwar, "Role of IoT in Agriculture", IOSR Journal of Computer Engineering, 2016.

[21] S. Raj, S. Sehrawet, N. Patwari and K. C. Sathiya, "IoT based model of automated agricultural system in India," 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 2019, pp. 88-93, doi: 10.1109/ICO