

A Novel Environmental Monitoring System for Real Time using Arduino and Node JS

Subhadra Kompella, Vikas B, Y Srija Reddy



Abstract: The Internet of Things (IoT) is playing a vital role in our daily lives in order to accomplish tasks by incorporating the use of sensor networks encompassing our environment. The systems developed to observe the physical tendency that creates data and the created data is stored in the cloud. The stored information is then utilized for designing applications for controlling necessary actions. This paper describes the application and experimentation of a system composed of sensors for monitoring temperature and humidity of the area surrounding. This observed information is used to perform ephemeral actions such as, controlling the electronic gadgets for warming or cooling that takes more time. The recorded data is loaded to cloud for storage and further accessed through an Android application and displays the results to the mobile-users. The framework presented in this paper employed Arduino UNO board, DHT11 sensor, ESP8266 Wi-Fi module, which makes information to open IoT based API administration ThingSpeak through which it is investigated and kept aside. An Android application is created which gets to the cloud and shows results for end-clients through REST API Web administration. The experimental results shown prove the effectiveness of the system.

Keywords: paper describes the application and experimentation of a system composed of sensors for monitoring temperature and humidity of the area surrounding.

I. INTRODUCTION

The development of Internet of Things (IoT) has radically changed the world through the use of electronic sensors and several such devices for controlling and monitoring environment at critical situations. These sensors and wireless devices are capable of sensing, storing and transmitting data to store into logical pools remotely namely a cloud thereby analyzes and presents required data in a way to survive the purpose of the end user. This information in the cloud is available to different end users provided through an interface for several mobile applications as per the requirement. Internet plays a major role in this transformation for an effective, reliable and swift transformation of data from devices to the

cloud and to the end-users. The concept behind this prototype is that an end system constitutes number of devices or things at the host end, thus the title "Internet of Things". The devices referred to the things are capable of performing activities like sensing and transmission of information like temperature, humidity, pollution rates etc. The concept of IoT is effectively applied where the manual involvement is not frequently possible such as environmental monitoring.

Environmental monitoring applications require transient data as input through which necessary controlling actions are responded by the system through sensors like remote controlling. This paper provides the details of implementation and experimentation results of an environmental monitoring system. The proprietary component of the system is the Arduino UNO board that acts as an interface by accepting temperature and humidity as input readings obtained via monitoring sensor DHT11 and the output obtained by sensing the data is transmitted through ESP8266 Wi-Fi module to remote cloud storage open IoT API ThingSpeak. With the help of ThingSpeak, MATLAB analysis is carried out about the data and generates a trigger. A mobile application is designed for an Android operating system where data is extracted from ThingSpeak which would be displayed from any part in the world. The application developed is a low-cost economical system that insights the implementation of a complete IoT application involving several actions namely sensing and wireless transmission to store in cloud and information extraction from the cloud via the developed application. This paper also presents a detailed study of the deployment of Arduino board along with its interfacing of input and output modules implemented by employing sensors and Wi-Fi module, the utilization of ThingSpeak open-source API and ends with the demonstration of the application. The experimental results provided at the end of this paper shows the monitoring and statistical analysis of temperature and humidity levels from any location in the world. This implementation of this application can be further extended to enable remote controlling of appliances through sensing data.

II. RELATED WORK

Recently, Internet of Things is an emerging technology that gained the most potential interest at both venture capitalists and technology giants, resulting in a surplus research activities which may lead to many business initiatives. There are a lot of applications that could be developed using Internet of Things which will be a part in designing metropolitan cities. These applications include smart grid, smart city, smart wearable devices, and smart home.

Manuscript published on January 30, 2020.

* Correspondence Author

Dr. K. Subhadra*, Received Doctorate, JNT University, Hyderabad, India.

Vikas B, Assistant Professor Department of CSE In GITAM. (Deemed to be University) University in Visakhapatnam, Andhra Pradesh, India.

Y Srija Reddy, Pursuing B.Tech Department of CSE, GITAM (Deemed to be University) University in Visakhapatnam, Andhra Pradesh, India.

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](https://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC-BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

A Novel Environmental Monitoring System for Real Time using Arduino and Node JS

Each of these applications constitute sensors and transducers generally embedded in a microcontroller along with wired or wireless transmission integrated with either a local database or to a remote cloud that translates input data into useful information which can be utilized in an efficient way. The research in the field of IoT App development through fabricating smart devices via implementing wireless technologies, designing boards, enabling network protocols, developing web applications to name a few.

The implementation part of this paper accomplishes the recent development in the context of IoT through designing a mobile based application in a lesser cost by using boards such as Raspberry Pi and Arduino. These boards have already taken its shape in the development of applications like home automation, patient monitoring systems, and weather and environmental monitoring systems. In the paper[1], measurements of temperature, humidity, light intensity, gas leakage, sea level, and rain intensity are considered and the input is transmitted wirelessly through ThingSpeak using Arduino UNO maintained using MATLAB. Authors in paper[2] also have monitored environmental conditions namely temperature, relative humidity, light intensity and CO2 level using sensors and LPC2148 microcontroller where the data was again sent to the ThingSpeak cloud. In contrast, this paper is implemented with Arduino UNO developing the system which is user friendly, low cost and less complex to process smaller applications. Authors in [3] narrated an IoT based application to monitor real-time weather using Raspberry Pi which is difficult compared to Arduino due to Python language and Raspbian operating system. In paper[4], authors designed weather monitoring system incorporating Arduino to import data from multiple sensors. The same is presented in[5], that includes designing and development a wireless sensor network system using Raspberry Pi and Arduino by employing an Xbee module to implement the IEEE 802.15.4 standard for data collection from multiple sensor nodes at a base station (Raspberry Pi). The authors in [5] mentioned that their project can also be applied for larger applications, despite in the current scenario, the system lacks cloud connectivity.

III. EXPERIMENTAL SETUP

A. Hardware Design

Arduino UNO is the central unit which acts as the main processing unit for the whole system. This is interfaced with the sensor chip to receive temperature and humidity as input readings. There by, integrated with the Wi-Fi module to produce output over receiving data from cloud. The microcontroller then initiates the sensor to receive data and transmit over the Internet via ThingSpeak cloud for further analysis.

B. Microcontroller

The core part of the system is the microcontroller, a central hardware component that interfaces with other components of the system. As the developing application requires a single sensor for monitoring temperature and humidity where no data is locally stored, Arduino UNO is selected as microcontroller which serves our purpose well due to its simplicity, robustness and low cost. This microcontroller board is developed on ATmega328P comprising 14 digital input/output pins, 6

analogue input pins, a USB connection, 16 MHz quartz crystal, a power jack, and a reset button. The microcontroller receives power through a battery, and implemented using Arduino IDE (Integrated Development Environment) via a type B USB cable.



Figure 1: Arduino UNO

C. Sensors (DHT11)

The implementation in this paper is performed using a single sensor to monitor temperature and humidity for environmental monitoring. This purpose is served with the DHT11 composite sensor chip, hence is included in the system for reading temperature and humidity at the same time. The outstanding characteristic feature of this sensor is its high reliability and long-term stability. This is also popularly used because this is highly economical with smaller dimensions, quick response, strong anti-interference ability, digital signal output, and precise calibration. Also this is easily interfaced with the Arduino UNO board. The figure shows a picture of the DHT composite sensor which we used in our framework. It can read temperature ranging from 0 to 50°C and humidity ranges from 20 to 90%RH. It has a signal transmission range of 20m. To interface it with Arduino UNO, we connected the Ground and VCC of the DHT11 sensor with the Ground and 5V of the Arduino. Then we connect the Data pin of the DHT11 to pin 2 of the Arduino. Then we installed the DHT library and run the code for getting it started.

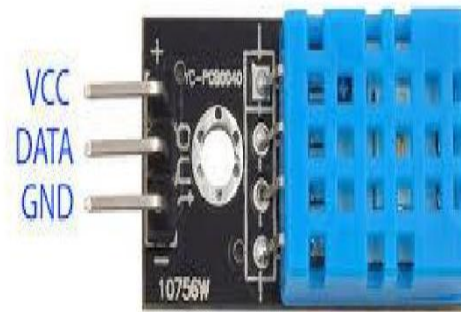


Figure 2: DHT11 Sensor

D. Wi-Fi Module

In order to upload sensor readings from DHT11 to the open-source cloud ThingSpeak, Arduino UNO interfaces at the output with Wi-Fi module ESP8266. It is a low-cost Wi-Fi microchip with a full TCP/IP stack. It works on the 3.3V that is provided by Arduino UNO in our system. The module is configured through AT commands and needs the required sequence to be used as a client.

The module can work as both client and server. It gets an IP on being connected to Wi-Fi through which the module and then communicates over the Internet. After testing our ESP8266 module, we connected it with Arduino UNO and then programmed Arduino UNO to configure the ESP8266 Wi-Fi module as TCP client and send data to ThingSpeak server which is an open IoT platform to visualize and analyse live data from sensors.



Figure 3:ESP8266 Wi-Fi module

E. Hardware Block Diagram

The below figure picturizes the basic hardware block diagram for our system. The figure describes about the flow of the system functionality where DHT11 gives live record values of temperature and humidity parallelly to the microcontroller which transfers these values to ThingSpeak through the Wi-Fi module

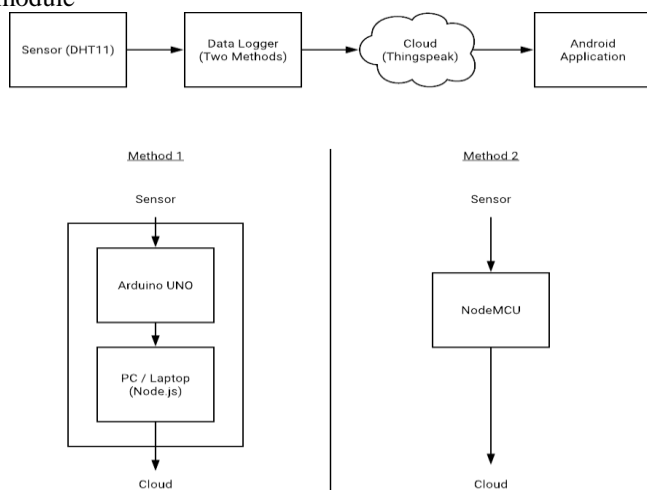


Figure 4:Hardware Block Diagram

IV.SOFTWARE IMPLEMENTATION

V.Software plays as a key role in functioning of hardware.

VI.Two ways of software development are: initialization and configuration of hardware,developing mobile application which benefits client interface

Software for Initialization and Configuration of Hardware
Arduino works to gather data from sensors and transfer data to cloud. Assimilation can be done in later stages after examination of hardware elements.InitiallyAT commands were used in desired pattern to initialize to functionalise it as user.Thus Wi-Fi Module ESP8266 is characterized to TCP/IP

client.A software is written for retrieving the sensor data from DHT11 which generates temperature and humidity values.After this data had got updated in the cloud we use IoT analytics (like ThingSpeak to aggregate, visualize and analysis of live streams).ThingSpeak generates instantaneous visualisation of present data in the cloud which had got updated.IP is used to transfer data to cloud through WIFI module.After API connection to ThingSpeak has been done, A key called API key is used to view the results over a channel. Thus before storing the data and displaying it in channel we first write the API key

Android Application

Thus the process of developing Android application through Aurdino studio is done through JAVA.This developed application interact with the microcontroller through cloud(ThingSpeak).Manipulation on channels (Update,Create,Clear,Delete) can be done through REST API methods like GET,POST,PUT,DELETE.A request to ThingSpeak is sent through JavaScript Object Notation with help of REST API Web Services and Channel ID and Field Number in its limits.The response will be reached in JSON format and produce tables in Application through JSON parser. The users execute the application and thus will be able to monitor the temperature and humidity readings of particular area. Application designed is portable,efficient and uses GUI.The end users observes results in graphical forms from cloud as shown in previous section.

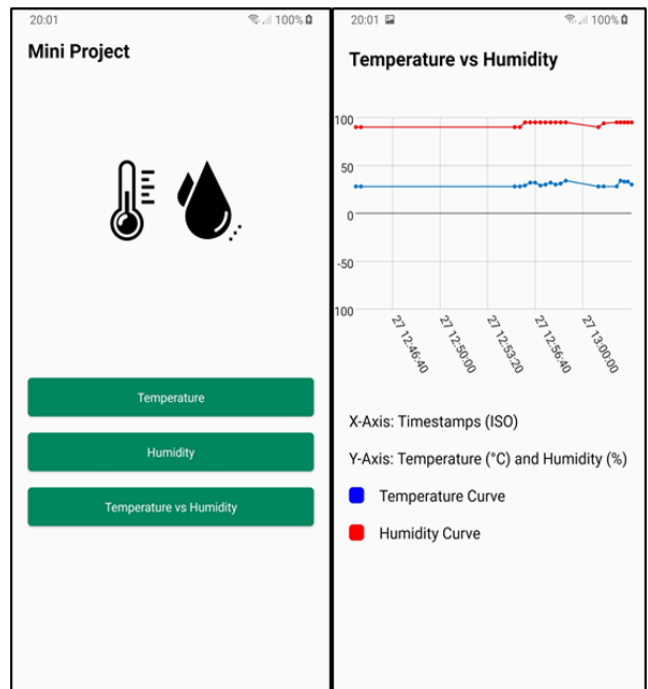


Figure 5:Android User Interface

VII.PROJECT METHODOLOGIES

Initial Setup

Step 1: ThingSpeak Setup for Temperature and Humidity Monitoring

Sign on ThingSpeak(If there is an account which is already signed up just login with id and password.

A Novel Environmental Monitoring System for Real Time using Arduino and Node JS

View www.thingspeak.com to create an account. Now move to sign up option in case if you don't have account and thus sign in with your details. After this verify your E-mail id .

Step 2: Create a Channel for Your Data

After Creation Of account create a new channel through button named New Channel. After this enter Name and title to the data to be uploaded in that specific channel. Here we have titled it as mini project. Now we have to add the attributes temperature and humidity in Field 1 and Field 2 respectively. The box next to the Field option can be used if we want to add another attribute as field and enter the description of data. To save your details click on save channel button.

Step 3: API Key

ThingSpeak uses a special API Key to find the location to store the data and navigate the activity when it is called. To upload DHT11 data by having unique API key , press the API key Button in ThingSpeak site.

Click on "API

Method 1: Environment sensing using Arduino UNO and NodeJS

The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc.

The board consists of various digital and analogue input/output pin sets that may be connected with various shields and circuits. The board consists of 14 Digital, 6 Analog pins, which are flexible with type B USB cable connected with ArduinoIDE. The Arduino board can accept the voltages between 7-20 volts but it is powered only by an external 9-volt which is similar to Arduino Nano and Leonardo

Circuit and working

Initial Setup

Step 1: ThingSpeak Setup for Temperature and Humidity Monitoring

Same as mentioned above.

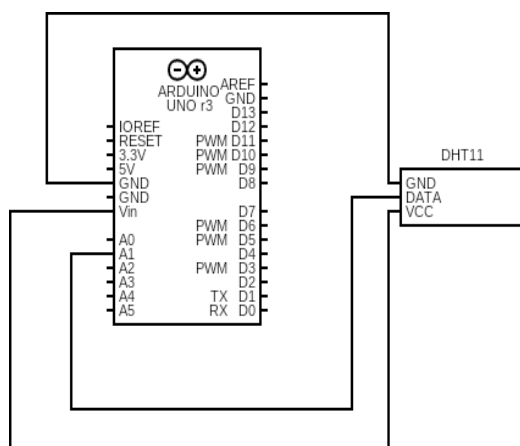


Figure 6: Interfacing Arduino UNO and DHT11

Figure 6: The diagram defines the humidity and temperature circuit which is designed with DHT11 sensor and NodeJS Arduino. As shown above the information regarding humidity and temperature is sent to digital pin 1 of NodeMCU through sensor. At uniform intervals of time, with help of ESP8266

Wi-Fi module the temperature and humidity values are recorded in the cloud from node. Thus the temperature and humidity can be viewed graphically on ThingSpeak from the cloud. The serial port establishes the interaction between Arduino board and PC. The interception of data bits takes place through Node JS and then it is stored in short-term buffer which is then rebuilt with wild cards and it. The communication between the Arduino board and the PC is established through a serial port. NodeJS is used to intercept the data bits coming from Arduino through the serial port and stored in the temporary buffer which is then reformatted with pre-defined patterns(wild-cards) and is non parallelly placed in cloud by using HTTPS protocol. Thus the temperature and humidity can be viewed graphically on ThingSpeak from the cloud

Construction and Testing

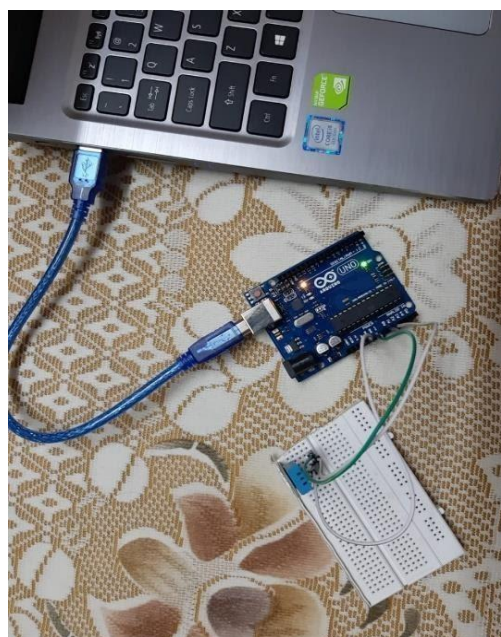


Figure 7: Hardware Implementation of method 1

After storing the DHT library is used by the Arduino program, if it is not initially there in Arduino IDE, follow Sketch→Includelibrary→Managelibraries→Install DHT Sensor library. Then compile program and update in an Arduino by IDE. Note: Wifi modem and internet connection should be working properly code, the Temperature and Humidity data are placed on ThingSpeak .thus we can see it graphically in window of your channel privately.

Method 2: Environment sensing using NodeMCU

ESP8266 Wi-Fi transceiver is a well known Wi-Fi modules used in IoT based Applications. Here we have used it to establish a connection with ThingSpeakIoT cloud Platform.

In this project, we transfer the Temperature and Humidity sensor data to ThingSpeak with the use of DHT11.

By this process we can observe DHT11 sensor's temperature and humidity data across internet bwith help of ThingSpeak IoT server, and the data which is logged and graph can be seen on timeline dashboard over internet using ThingSpeak IOT server, and we can view the logged data and graph over time on the ThingSpeak dashboard.

NodeMCU retrieves the present temperature and humidity from DHT11 and passes it to ThingSpeak server for live monitoring from any part of world.

Circuit and working

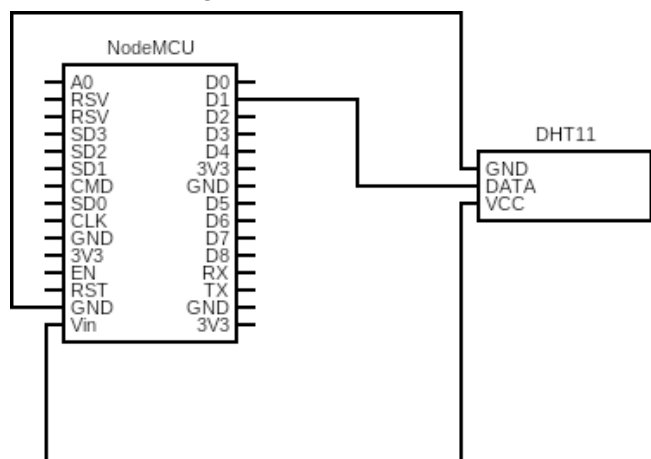


Figure 8: Interfacing NodeMCU and DHT11

Circuit diagram for monitoring humidity and temperature is shown in Fig. above. It is built around NodeMCU, DHT11 sensor and ESP8266 Wi-Fi module. As shown above the information regarding humidity and temperature is sent to digital pin 1 of NodeMCU through sensor. At uniform intervals of time, with help of ESP8266 Wi-Fi module the temperature and humidity values are recorded in the cloud from node. Thus the temperature and humidity can be viewed graphically on ThingSpeak from the cloud.

Construction and Testing



Figure 9: Hardware Implementation of method 2

As described in section 5.1 we start ThingsSpeak and when the procedure is completed, we connect the hardware as displayed in fig9. The Arduino program works on DHT library, if it is not presented it follows the previously mentioned process for installation and follow the respective process of compilation.. And you can able to see the uploaded data from serial port of Arduino IDE.

VIII. EXPERIMENTAL RESULTS

The entire design of environmental monitoring system is shown in Figures. The design which is used is shown in picture which describes the integration of all hardware components during working conditions. DHT11 and ThingSpeak are interfaced using the Arduino IDE. The sensed

data is viewed when Android Application gets connected with ThingSpeak. The results of the experiment are shown here i.e., the data required for the user is shown through mobile application and graphical depiction of humidity and temperatures records.

Tabular representation of Temperature and Humidity

The below tabular data represents the sample of temperature and humidity readings observed at different time intervals recorded using DHT11 sensor. The data is transmitted to the ThingsSpeak cloud where it is automatically converted into a graph as shown in later result. This tabular data is extracted from JSON object file generated by the ThingsSpeak cloud service. The scope of this paper is to only get this data over Internet remotely. In practise this setup can be integrated with other home sensing application to remotely trigger an action based on patterns observed in the readings.

Table 1: Temperature and Humidity readings

Time	Temperature	Humidity
27-09-2019 18.12	90	28
27-09-2019 18.13	90	28
27-09-2019 18.13	90	28
27-09-2019 18.13	90	28
27-09-2019 18.14	90	28
27-09-2019 18.14	90	28
27-09-2019 18.25	90	28
27-09-2019 18.25	90	28
27-09-2019 18.25	95	29
27-09-2019 18.26	95	32
27-09-2019 18.26	95	32
27-09-2019 18.27	95	29
-09-2019 18.26	95	32
27-09-2019 18.26	95	32
27-09-2019 18.27	95	29

Graphical Record of Temperature Monitoring

The Figure 10. displays the data of temperature observed over a period of time. The graph here makes changes in temperature for an interval of 15 seconds. Here the graph is drawn among temperature and Date.

In the process of testing accuracy, we have changed the temperature through lightening systems therefore we observed a raise in graph followed with average temperature reading

A Novel Environmental Monitoring System for Real Time using Arduino and Node JS

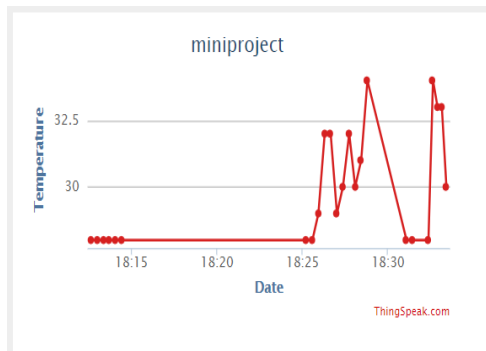


Figure 10: Record of Temperature Monitoring

The Figure 11. displays the data of humidity observed over a period of time. The graph here makes changes in temperature for an interval of 15 seconds. Here the graph is drawn among humidity and Date. In the process of testing accuracy, we have changed the temperature through lightening systems, therefore we observed a fall in graph followed with average humidity reading

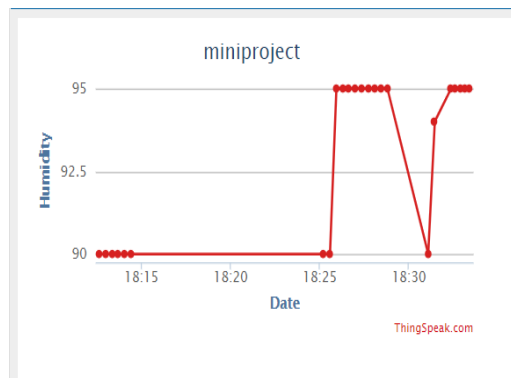


Figure 11: Record of Humidity Monitoring

IX. CONCLUSION

This paper defines an environmental monitoring system for real-time monitoring of temperature and humidity of the surrounding environment. The data which is collected or sensed is passed to cloud by wi-fi where we can see data and graphical analysis. An Android application is developed for the client who can monitor the environment of the area where the hardware is placed and executed using a smartphone. The designed application is used as basic home automation system where the observed values of temperature and humidity can be used to operate some action and control the devices through the mobile application. Thus designing of this system is an important step in understanding the IoT application development and implementation and works as a foundation for many useful innovations in this direction.

REFERENCES

1. S. Pasha, "ThingSpeak based sensing and monitoring system", International Journal of New Technology and Research, Vol. 2, No. 6, pp. 19-23, 2016
2. K. S. S. Ram, A. N. P. S. Gupta, "IoT based data logger system for weather monitoring using wireless sensor networks", International Journal of Engineering Trends and Technology, Vol. 32, No. 2, pp. 71-75, 2016
3. S. D. Shewale, S. N. Gaikwad, "An IoT based real-time weather monitoring system using Raspberry Pi", International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering", Vol. 6, No. 6, pp. 4242-4249, 2017

4. R. Ayyappadas, A. K. Kavitha, S. M. Praveena, R. M. S. Parvathi, "Design and implementation of weather monitoring system using wireless communication", Vol. 5, No. 5, pp. 1-7, 2017
5. S. Ferdoush, X. Li, "Wireless sensor network system design using Raspberry Pi and Arduino for environmental monitoring application", Procedia Computer Science, Vol. 34, pp. 103-110, 2014
6. Arduino, ArduinoUnoRev3 Overview, available at: <https://store.arduino.cc/arduino-uno-rev3>
7. <https://thingspeak.com>
8. <https://developer.android.com>

AUTHORS' PROFILE



Dr. K. Subhadra, Has Received Doctorate From JNT University, Hyderabad In 2017 And Masters From Andhra University. She Is Actively Engaged In Research In The Area Of Big Data Analytics And Algorithms. She Has A Teaching Experience Of About 13 Years.



Vikas B, Is Currently Working As Assistant Professor In CSE In GITAM. He Is Member Of IAENG And IEEE-CS And His Research Interests Include Deep Learning, Data Mining, Bioinformatics, Information Security, And Data Sciences



Y Srima Reddy, is Currently Pursuing B.Tech CSE From GITAM Deemed To Be University. Enthusiast In Programming And Developing Projects.