IOT Based Real-Time Environmental Monitoring System Using Aurdino and Cloud Service and GSM

Mohd Waseem ali siddique[1], Mohammed Abdullah ali[2], Mir ahmed Uddin ali khan[3]
[1,2,3]*Member, Department of electronics and communication engineering, ISL Engineering College, Bandlaguda, Hyderabad 500005*

***Abstract*—Internet of Things (IoT) is expected to play a major role in our lives through pervasive systems of sensor networks encompassing our environment. These systems are designed to monitor vital physical phenomena generating data which can be transmitted and saved at cloud from where this information can be accessed through applications and sms, further actions can be taken. This paper presents the implementation and results of an environmental monitoring system which employs sensors for temperature and humidity of the surrounding area. This data can be used to trigger short term actions such as remotely controlling heating or cooling devices or long term statistics. The sensed data is uploaded to cloud storage and an Android application accesses the cloud and presents the results to the end users. The system employs Arduino UNO board, DHT11 sensor, Air quality sensor, rain fall sensor, ESP8266 Wi-Fi module, which transmits data to open IoT API service ThingSpeak where it is analyzed and stored. An Android application is developed which accesses the cloud and displays results for end users via REST API Web service. The experimental results show the usefulness of the system**

Keywords- Internet of things, cloud service , GSM, Arduino UNO , LCD display.

I. INTRODUCTION

Internet of Things (IoT) is expected to revolutionize our world by enabling us to monitor and control vital phenomena in our environment through the use of devices capable of sensing, processing and wirelessly transmitting data to remote storage like cloud which stores, analyzes and presents this data in useful form. From the cloud this information can be accessed through various front end user interfaces such as web or mobile applications, depending upon suitability and requirements. Internet lies at the heart of this transformation playing its role in efficient, reliable and swift communication of data from devices to the cloud and from the cloud to the end users. In this new paradigm, the concept of the typical end system or host in the Internet is modified and hosts comprise of devices or things hence the name Internet of Things. The “things” are capable of sensing and transmitting data such as temperature, pressure, humidity, noise, pollution, object detection, patient vitals etc.

II. RELATED WORK

Lately, IoT has emerged as an area which has gained immense interest of both venture capitalists and tech giants, resulting in a plethora of research activities and business initiatives. Some of the applications which have garnered attention include smart grid, smart city, smart wearable devices and smart home. Almost all of the various IoT applications involve some kind of sensors and transducers normally attached to a microcontroller along with wired or wireless transmission to either a local database or a remote cloud which transforms raw data into useful information which can be effectively utilized. The research and development activities comprise techniques of fabrication of smart objects or devices, suitable wireless technologies, development boards, designing network protocols, applications and much more. In the context of our project, we explored recent work accomplished in the development of useful and interesting applications using low cost development boards such as Raspberry Pi and Arduino. Some of the popular applications developed using these boards include home automation, patient monitoring systems and weather and environmental monitoring systems. In [1], temperature, humidity, light intensity, gas leakage, sea level and rain intensity are measured and the data are sent wirelessly to ThingSpeak using Arduino UNO. This work focuses considerably on MATLAB visualization and analysis.

III. HARDWARE DESIGN

The central unit is a microcontroller (Arduino UNO), and acts as the main processing unit for the entire system, it interfaces with the sensor chip at the input for receiving temperature and humidity readings and interfaces with the Wi-Fi module at the output to send the received data to the cloud over the Internet. The microcontroller polls the sensor to retrieve data and sends it over the Internet to ThingSpeak cloud for analysis.

A. Microcontroller

The central hardware component of our system is the microcontroller which interfaces with other components of the system. Since the system comprises of temperature and humidity monitoring for which a single sensor interface is required and no local storage of data therefore we selected Arduino UNO microcontroller which serves our purpose well due to its simplicity, robustness and low cost [6]. Figure 1 shows a picture of Arduino UNO microcontroller used in our system [6]. This microcontroller board is based on the ATmega328P. It has 14 digital input/output pins, 6 analog input pins, a USB connection, 16 MHz quartz crystal, a power jack, and a reset button. It can be powered with a battery. It is programmable with the Arduino IDE (Integrated Development Environment) via a type B USB cable

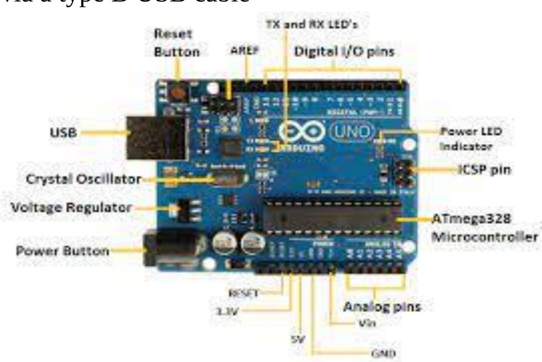

Fig. 1. Arduino UNO microcontroller board

B. Sensors

We selected temperature and humidity for environmental monitoring and prefered a single sensor with both sensing capabilities instead of separate sensors for each parameter. For this reason, we selected DHT11 composite sensor chip which gives readings for both temperature and humidity at the same time, it has high reliability and excellent long-term stability. It has small dimensions, low cost, good quality, fast response, strong anti-interference ability, digital signal output, and precise calibration. It can be easily interfaced with Arduino UNO board with the help of DHT library and connecting wires. Figure 2 shows a picture of the DHT composite sensor which we used in our system. It has temperature range from 0 to 50°C and humidity range from 20 to 90%RH. It has 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 the pin 2 of the Arduino. Then we installed the DHT library and run the code for getting it started.

C. WiFi Module

In order to upload sensor readings from DHT11 to the open source cloud ThingSpeak, Arduino UNO interfaces at the output with WiFi module ESP8266 (Figure 3). It is a low cost WiFi microchip with 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 WiFi 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 ESP8266 WiFi module as TCP client and send data to ThingSpeak server which is an open IoT platform to visualize and analyze live data from sensors.

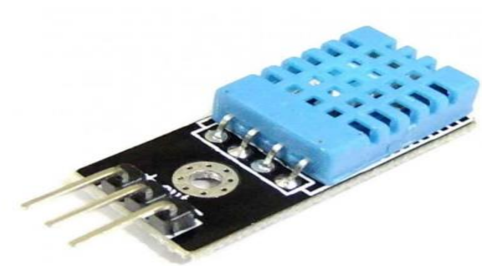

Fig. 2. DHT11 composite sensor for temperature and humidity

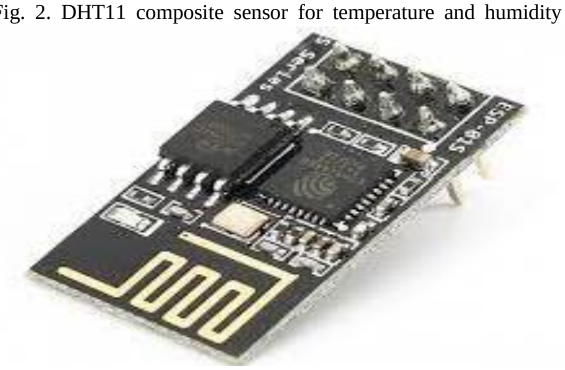

Fig. 3. ESP8266 WiFi module

D. Rain Fall Sensor

Rain sensors for irrigation systems are available in both wireless and hard-wired versions, most employing hygroscopic disks that swell in the presence of rain and shrink back down again as they dry out — an electrical switch is in turn depressed or released by the hygroscopic disk stack, and the rate of drying is typically adjusted by controlling the ventilation reaching the stack. However, some electrical type sensors are also marketed that use tipping bucket or conductance type probes to measure rainfall. Wireless and wired versions both use similar mechanisms to temporarily suspend watering by the irrigation controller — specifically they are connected to the irrigation controller's sensor terminals, or are installed in series with the solenoid valve common circuit such that they prevent the opening of any valves when rain has been sensed.

Some irrigation rain sensors also contain a freeze sensor to keep the system from operating in freezing temperatures, particularly where irrigation systems are still used over the winter.

LM393 Comparator:

The LM393 series are dual independent precision voltage comparators capable of single or split supply operation. These devices are designed to permit a common mode range−to−ground level with single supply operation. Input offset voltage specifications as low as 2.0 mV make this device an excellent selection for many applications in consumer, automotive, and industrial electronics.

Fig. 4. Rain Fall Sensor

Features:

• Wide Single−Supply Range: 2.0 Vdc to 36 Vdc
• Split−Supply Range: ±1.0 Vdc to ±18 Vdc
• Very Low Current Drain Independent of Supply Voltage: 0.4 mA
• Low Input Bias Current: 25 nA
• Low Input Offset Current: 5.0 nA
• Low Input Offset Voltage: 5.0 mV (max) LM293/393
• Input Common Mode Range to Ground Level
• Differential Input Voltage Range Equal to Power Supply Voltage
• Output Voltage Compatible with DTL, ECL, TTL, MOS, and CMOS Logic Levels
• ESD Clamps on the Inputs Increase the Ruggedness of the Device without Affecting Performance
• NCV Prefix for Automotive and Other Applications Requiring Unique Site and Control Change Requirements; AEC−Q100 Qualified and PPAP Capable

E. Air Quality Sensor

Air quality sensors are devices used to detect contaminants in the air. This includes particulates, pollutants and noxious gases that may be harmful to human health. They are used in applications like air quality monitoring, gas detection in industry, combustion controllers and oxygen generators in aircraft. Volatile organic compound (VOC) sensors are available that are capable of detecting volatile chemicals and odorous pollutants.

Fig.5. Air Quality Sensor

A common device for detecting particulate matter in the air is the household fire alarm. Smoke is detected using two types of sensor: ionization detectors and photoelectric detectors. Ionization sensors work by using a small radioactive source (Americium-241) produces alpha particles at a constant rate. Smoke particles enter the ionization chamber and interrupt the small leakage current between electrodes. This is detected, setting off the alarm.

IV. HARDWARE BLOCK DAIGRAM

Figure 4 shows the hardware block diagram for our system. The figure shows the flow of the system functionality where DHT11 gives live readings of temperature and humidity, Rain detection, air quality reading simultaneously to the microcontroller which sends these reading through the Wi-Fi module over the Internet to the ThingSpeak cloud.

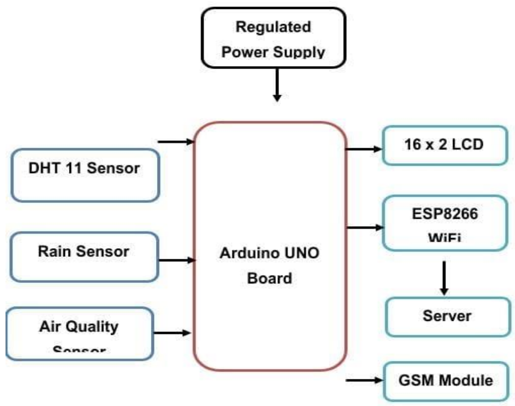

Fig.6. Block diagram of environmental monitoring system

V. EXPERIMANTAL RESULT

The complete design of our environmental monitoring system is shown in Figure 4. The implemented design is shown in Figure 6 which shows the integration of all Engineering, Technology & Applied Science Research Vol. 8, No. 4, 2018, 3238-3242 3241 www.etasr.com Zafar et al.: An IoT Based Real-Time Environmental Monitoring System Using Arduino and Cloud Service hardware components in working conditions. DHT11 and ESP8266 are connected to Arduino UNO. DHT11 and ThingSpeak are interfaced using the Arduino IDE. The Android application connects with ThingSpeak and displays the sensed data. Table I shows the details of the system hardware, cloud and display.

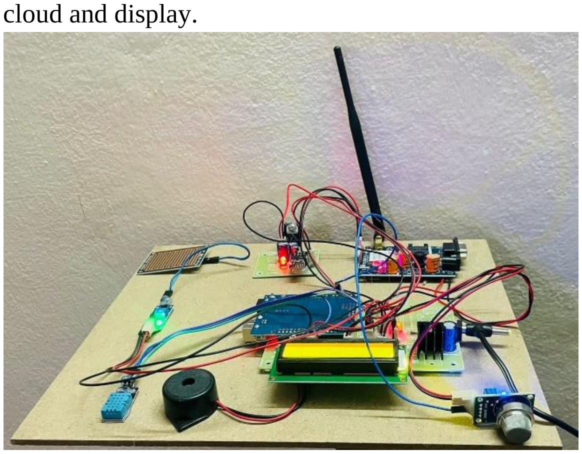

Fig.7. Output Result

VI. CONCLUSION

This paper presents an environmental monitoring system for real-time monitoring of temperature and humidity and rain detection and air quality of surrounding environment. The sensed data is sent through Wi-Fi to the cloud where both real-time data and its graphical analyses can be viewed. This system can be extended to implement a home automation system where the monitored values of temperature and humidity can be used to trigger some action and control the devices for heating or cooling via the mobile application. This system is a crucial step in understanding the IoT applications development and implementation and serves as a building block for a number of useful innovations in this direction.

VII. 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, Arduino Uno Rev 3 Overview, available at: https://store.arduino.cc/arduino-uno-rev3

[7] https://thingspeak.com

[8] https://developer.android.com