INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT (IJSREM)
VOLUME: 05 ISSUE: 07 | JULY - 2021 ISSN: 2582-3930

Weather monitoring system using IoT

Himanshu Singh , Kumar Hritik , Abhyudai Singh Thakur

PROJECT GUIDE : Asst. Prof. S.R. Madkar

Department of Electronics and Communication Engineering

Bharti Vidyapeeth (Deemed to be) University College of Engineering,Pune

Abstract – Internet of Things is one of the new emerging technology. Evolution of IoT made it easier to control and monitor any device from remote location. In our proposed model, we are trying to measure the various weather constraints like temperature, humidity, gas concentration e.t.c with the help of sensors and after processing that sensor data through Node Mcu board we are sending these real time data on ThingSpeak which is an IoT platform provides the feature to store the data in real time mode and display that data. The weather condition can be visualized and analyzed in numerical and graphical way.

KEYWORDS : Internet of Things (IoT), NodeMcu, ThingSpeak,

I. INTRODUCTION

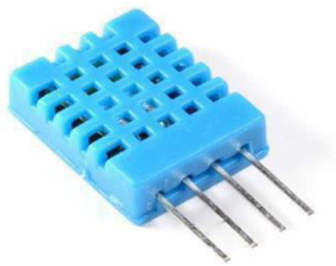
Current day innovations in technology mainly focus on controlling and monitoring of different devices over wirelessly over the internet such that the internet behaves as a medium for communication between all the devices. Most of this technology is focused on efficient monitoring and controlling of different devices. An efficient environmental monitoring system is required to monitor and assess the weather conditions in case of exceeding the prescribed level of parameters (CO concentration and Humidity levels). A system is considered as a smart system when the device equipped with sensors, microcontrollers and various software applications and becomes a self-protecting and self-monitoring system.

Event Detection based and Spatial Process Estimation are the two categories to which applications are classified. Initially the sensor devices are deployed in environment to detect the parameters (e.g., Temperature, Humidity, Pressure, LDR, noise, CO and radiation levels etc.) while the data acquisition, computation and controlling action (e.g., the variations in the noise and CO levels with respect to the quantified levels). Sensor devices are positioned at different locations to collect the data to forecast the behavior of a particular area of interest. The main aim of this paper is to design and implement a resourceful weather monitoring system through which the required parameters are monitored remotely using internet and the data gathered from the devices are stored in the cloud and to project the predictable trend on the web browser. The solution also provides an intelligent remote monitoring for a particular area of interest. In this paper we also current results of collected or sensed

2.SYSTEM OVERVIEW

The implemented system consists of a Node Mcu which is used as the main processing unit for the entire system and all the sensor and devices can be connected with this microcontroller board. The sensors can be operated by the Node Mcu to retrieve the data from them and it processes that data. Let us discuss about the sensors and board used in our model one by one.

DHT 11: The DHT11 is a commonly used Temperature and humidity sensor. The sensor comes with a dedicated NTC to measure temperature and an 8-bit microcontroller to output the values of temperature and humidity as serial data. The sensor is also factory calibrated and hence easy to interface with other microcontrollers.
The sensor can measure temperature from 0°C to 50°C and humidity from 20% to 90% with an accuracy of ±1°C and ±1%.



DHT11 Humidity Sensor consists of 4 pins: VCC, Data Out, Not Connected (NC) and GND. The range of voltage for VCC pin is 3.5V to 5.5V. A 5V supply would do fine. The data from the Data Out pin is a serial digital data

MQ 7: Carbon monoxide (CO) is a very dangerous gas which is odorless, colorless, and tasteless, so it cannot be smelt, seen, or tasted. CO is measured in parts per million (ppm). This Carbon Monoxide (CO) gas sensor MQ 7 detects the concentrations of CO in the air and outputs its reading as an analog voltage. The sensor can measure concentrations of 10 to 10,000 ppm.The sensor can operate at temperatures from -10 to 50°C and consumes less than 150 mA at 5 V.

There are four pins in this sensors : +5V, AOUT, DOUT, and GND. The +5V and GND leads establishes power for the alcohol sensor. The other 2 leads are AOUT (analog output) and DOUT (digital output). How the sensor works is the terminal AOUT gives an analog voltage output in proportion to the amount of carbon monoxide the sensor detects. The more CO it detects, the greater the analog voltage it will output. Conversely, the less CO it detects, the less analog voltage it will output. If the analog voltage reaches a certain threshold, it will send the digital pin DOUT high. Once this DOUT pin goes high, the arduino will

 ISSN: 2582-3930

detect this and will trigger the LED to turn on, signaling that the CO threshold has been reached and is now over the limit. How you can change this threshold level is by adjusting the potentiometer to either raise or lower the level.

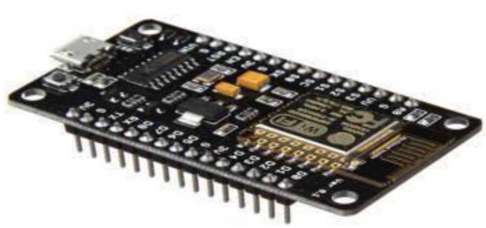
LDR: A Light Dependent Resistor (LDR) is also called a photoresistor or a cadmium sulfide (CdS) cell. It is also called a photoconductor. It is basically a photocell that works on the principle of photoconductivity. The passive component is basically a resistor whose resistance value decreases when the intensity of light decreases. This optoelectronic device is mostly used in light varying sensor circuit, and light and dark activated switching circuits. Some of its applications include camera light meters, street lights, clock radios, light beam alarms, reflective smoke alarms, and outdoor clocks.

Light intensity was measured by using a resistor in series with each LDR making a voltage dividing circuit as shown in Fig. 6. Whenever the intensity of light on LDR changes, its resistance and hence the output voltage are changed; change in intensity is translated into a change in voltage.

Node Mcu: NodeMCU is an open source firmware for which open source prototyping board designs are available. The name "NodeMCU" combines "node" and "MCU" (micro-controller unit).[8]. The term "NodeMCU" strictly speaking refers to the firmware rather than the associated development kits. Both the firmware and prototyping board designs are open source.

The firmware uses the Lua scripting language. The firmware is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. It uses many open source projects, such as lua-cjson and SPIFFS. Due to resource constraints, users need to select the modules relevant for their project and build a firmware tailored to their needs. Support for the 32-bit ESP32 has also been implemented.

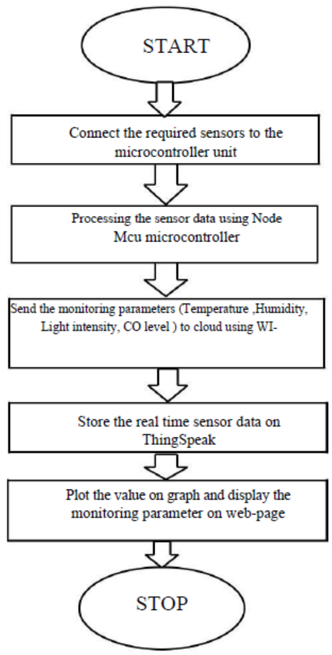
The prototyping hardware typically used is a circuit board functioning as a dual in-line package (DIP) which integrates a USB controller with a smaller surface-mounted board containing the MCU and antenna. The choice of the DIP format allows for easy prototyping on breadboards. The design was initially was based on the ESP-12 module of the ESP8266, which is a Wi-Fi SoC integrated with a Tensilica Xtensa LX106 core, widely used in IoT applications.



ThingSpeak: It is a platform providing various services exclusively targeted for building IoT applications. It offers the capabilities of real-time data collection, visualizing the collected data in the form of charts,

ability to create plugins and apps for collaborating with web services, social network and other APIs. The core element of ThingSpeak is a 'ThingSpeak Channel'. A channel stores the data that we send to ThingSpeak and comprises of the below elements:
8 fields for storing data of any type - These can be used to store the data from a sensor or from an embedded device.
3 location fields - Can be used to store the latitude, longitude and the elevation. These are very useful for tracking a moving device.
1 status field - A short message to describe the data stored in the channel.

Flow Chart

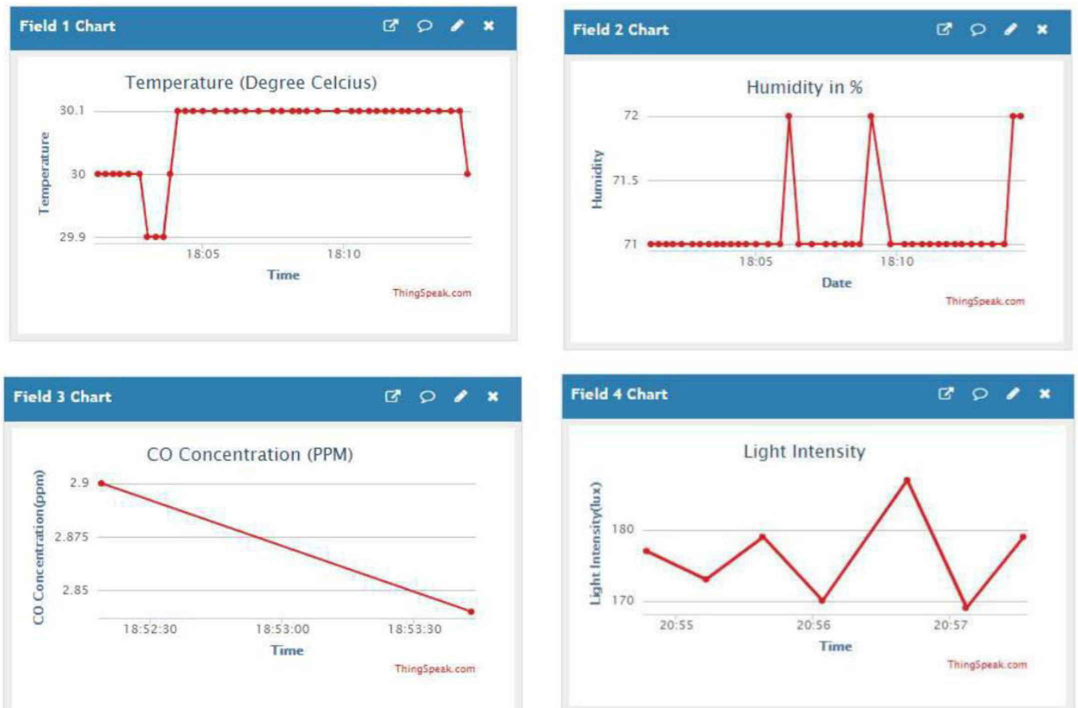


4.OPERATION

To operate this proposed model successfully our first challenge was to gather the various data related to weather constraint. For this we used various sensors like DHT 11 for measuring temperature and humidity, MQ7 for measuring CO (Carbon Monoxide) concentration present in atmosphere and LDR sensor to detect the intensity of light at particular time.
All the above mentioned sensors first senses the surrounding environmental condition. Sensor outputs the signal in analog form. Then these analog signal is being processed in microcontroller unit.
In our case we have used Node Mcu as microcontroller board because it doesn't need external interface for WiFi module instead it has inbuilt ESP8266. When signal comes from sensors to this board then it processes those signal and display that in desired format like temperature in degree celcius or gas in ppm. Of course to convert the analog signal in this format would required certain kind of algorithm and programming. The advantage of using Node Mcu is that it runs on the program which can be compiled on Arduino IDE.
While writing the sketch for module there are two main task that has to be accomplished one is to write the code for reading the sensor data and another is to sending these data on cloud in real time mode. A particular kind of algorithm is applied in case of each sensor to read its analog value and then process it to convert it into digital mode then display that digital data in desired format.
Once this data is collected our main task remains to send it to cloud. Here comes the role of ThingSpeak. It

provides not only the cloud space but also gives a platform where one can display, analyze, visualize their data and can share it publicly. For this it provides each user an unique API key which they have to use while writing the code for data uploading. In that code we also have to mention the SSID and Password of local wireless internet connection. Data is sent on ThingSpeak by means of using the local internet connection and that unique API key. The data is sent on real time mode, while it is on user that on what interval they want to update data on cloud. ThingSpeak also provide the feature to keep your data private and as well as to share it publicly.

5.RESULTS



II. CONCLUSION

This system provides the monitoring access to the temperature, humidity, intensity of light values from anywhere with Internet access. This system is useful for many applications like predicting temperature of environment, agriculture, temperature control of data center and many others. This is the basic of IoT monitoring system and many other sensors can be replaced for the applications of user requirements like car parking monitoring system, security system using PIR sensor power consumption monitoring, soil moisture etc. This system to forecast the parameters based on the historical data. This system can be further expanded to monitor the developing cities and industrial zones for monitoring, collecting the data and analysis. In this system, temperature, humidity, gas concentration, light intensity values are displayed on ThingSpeak web browser in this system. This is not easy for users to access this system. So, mobile application should be created for this system by connecting to ThingSpeak cloud server. In this way, users can monitor the temperature and humidity values easily via mobile applications.

Acknowledgment

We express our deep appreciation and sincere thanks to Prof. Jyoti Morbale for her extensive personal and professional guidance and life in general. She has always been supportive to us during all the time. We are grateful to all of those with whom we have had the pleasure to work during this and other related projects.

REFERENCES

1. Baldonado, M., Chang, C.-C.K., Gravano, L., Paepcke, A.: The Stanford Digital Library Metadata Architecture. Int. J. Digit. Libr. 1 (1997) 108121

2. R. Shete and S. Agrawal, "IoT Based Urban Climate Monitoring using Raspberry Pi," in International Conference on Communication and Signal Processing, 2016, pp. 2008–2012.
3. www.arduino.cc
4. www.nevonprojects.com