

# **IoT based data logger for weather monitoring using Arduino- based wireless sensor networks with remote graphical application and alerts**

Submitted in partial fulfillment of the requirements for the award of Bachelor of  
Engineering degree in Electronics and Communication Engineering

By

**Bhavya Kriti (38130032)**

**Srestha Baidya (38130213)**



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING  
SCHOOL OF ELECTRICAL AND ELECTRONICS ENGINEERING**

# **SATHYABAMA**

**INSTITUTE OF SCIENCE AND TECHNOLOGY  
(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade "A" by NAAC**

**JEPPIAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119**



## DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

### **BONAFIDE CERTIFICATE**

This is to certify that this Project Report is the bonafide work of **BHAVYA KRITI** (Reg. No. 38130032) and **SRESTHA BAIDYA** (Reg. No. 38130213) who carried out the project entitled “**IoT based data logger for weather monitoring using Arduino- based wireless sensor networks with remote graphical application and alerts**” under our supervision from January 2022 to March 2022.

#### **Internal Guide**

Dr. P. Grace Kanmani, M.E., Ph.D.

#### **Head of the Department**

Dr. T. Ravi, M.E., Ph.D.

**Submitted for Viva Voce Examination held on**

**Internal Examiner**

**External Examiner**

## DECLARATION

We, **BHAVYA KRITI** (Reg. No.38130032) and **SRESTHA BAIDYA** (Reg. No.38130213) hereby declare that the Project Report entitled “**IoT based data logger for weather monitoring using Arduino- based wireless sensor networks with remote graphical application and alerts**” done by us under the guidance of **Dr.P. GRACE KANMANI M.E., Ph.D.** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Electronics and Communication Engineering.

**DATE:**

**PLACE:**

**1.**

**2.**

**SIGNATURE OF CANDIDATES**

## ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to the **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

We convey our thanks to **Dr. N. M. Nandhitha, M.E., Ph.D., Dean, School of Electronics and Communication Engineering** and **Dr. T. Ravi, M.E., Ph.D., Head of the Department, Department of Electronics and Communication Engineering** for providing me necessary support and details at the right time during the progressive reviews.

We would like to express our sincere thanks to our respectful panel members **Dr. A. Sahaya Anselin Nisha, M.E., Ph.D.**, and **Dr.V.Vedanarayanan, M.E., Ph.D.**, who have given their valuable suggestions and details at the right time during the progressive reviews.

We would like to express our sincere and deep sense of gratitude to our project guide **Dr.P.GRACE KANMANI M.E., Ph.D.** for her valuable guidance, suggestions, and constant encouragement paved way for the successful completion of our project work.

We wish to express our thanks to all teaching and non-teaching staff members of the **Department of Electronics and Communication Engineering** who were helpful in many ways for the completion of the project.

We would like to thank our beloved parents for their constant love and support, without whom this would not be possible and finally we would like to thank the Almighty for His wonderful blessing.

## **ABSTRACT:**

In this project we propose an automatic weather monitoring system that allows having dynamic and real-time climate data of a given area. The proposed system is based on the internet of things technology and embedded system. The system also includes electronic devices, sensors, and wireless technology. The main objective of this system is sensing the climate parameters, such as temperature, humidity, and existence of some gases, based on the sensors. The captured values can then be sent to remote applications or databases. Afterwards, the stored data can be visualized in graphics and tables form. These sensors are electronic devices that are widely used to compute different normal physical and environmental parameters. By using sensors to study climatic conditions, the results will be accurate and the whole system consume less energy and faster response will be there. It includes wireless technology, which is also connected to Wi-Fi. All these sensors are capable of measuring the corresponding meteorological parameters. The system is designed for use in large buildings and multinational companies, but also wherever we want to know the weather condition anywhere in the world.

## **CONTENT**

| <b>CHAPTER NO.</b> | <b>TITLE</b>                                              | <b>PAGE NO.</b> |
|--------------------|-----------------------------------------------------------|-----------------|
|                    | ABSTRACT                                                  | 5               |
|                    | LIST OF FIGURES                                           | 8               |
| <b>1</b>           | <b>INTRODUCTION</b>                                       | 9               |
|                    | 1.1 PROBLEM STATEMENT                                     | 10              |
|                    | 1.2 OBJECTIVE                                             | 10              |
|                    | 1.3 MOTIVATION                                            | 11              |
|                    | 1.4 IOT                                                   | 13              |
|                    | 1.5 WORKING PRINCIPLE                                     | 14              |
| <b>2</b>           | <b>LITERATURE SURVEY</b>                                  | 16              |
|                    | 2.1 RELATED WORKS                                         | 16              |
|                    | 2.2 A SMART WEATHER<br>MONITORING SYSTEM USING IOT        | 17              |
|                    | 2.3 A SMART WEATHER STATION<br>BASED ON SENSOR TECHNOLOGY | 18              |
| <b>3</b>           | <b>SYSTEM DESIGN</b>                                      | 20              |
|                    | 3.1 EXISTING SYSTEM                                       | 20              |
|                    | 3.2 PROPOSED SYSTEM                                       | 21              |
|                    | 3.3 BLOCK DIAGRAM                                         | 22              |
|                    | 3.4 PROJECT IMPLEMENTATION                                | 22              |
|                    | 3.5 PROJECT DESCRIPTION                                   | 23              |
|                    | 3.6 HARDWARE USED                                         | 23              |
|                    | 3.7 SOFTWARE REQUIREMENTS                                 | 24              |

|          |                                     |           |
|----------|-------------------------------------|-----------|
| <b>4</b> | <b>HARDWARE DESIGN</b>              | <b>25</b> |
|          | 4.1 ARDUINO UNO AND ITS PROGRAMMING | 25        |
|          | 4.2 NODE MCU                        | 29        |
|          | 4.3 ESP8266 ARDUINO CORE            | 29        |
|          | 4.4 PRESSURE SENSOR                 | 33        |
|          | 4.5 TEMPERATURE SENSOR              | 35        |
|          | 4.6 LIGHT INTENSITY SENSOR          | 38        |
|          | 4.7 CO SENSOR                       | 38        |
|          | 4.8 WATER LEVEL SENSOR              | 39        |
| <b>5</b> | <b>SOFTWARE DESIGN</b>              | <b>41</b> |
|          | 5.1 ARDUINO IDE                     | 41        |
|          | 5.2 WORKING                         | 41        |
|          | 5.3 UBIDOT                          | 43        |
| <b>6</b> | <b>RESULT AND DISCUSSIONS</b>       | <b>50</b> |
| <b>7</b> | <b>CONCLUSION</b>                   | <b>52</b> |
|          | 7.1 CONCLUSION                      | 52        |
|          | 7.2 FUTURE SCOPE                    | 52        |
| <b>8</b> | <b>REFERENCES</b>                   | <b>53</b> |

## LIST OF FIGURES

| FIGURE<br>NUMBER | TITLE                                       | PAGE<br>NUMBER |
|------------------|---------------------------------------------|----------------|
| 3.3              | BLOCK DIAGRAM                               | 22             |
| 4.1              | ARDUINO NANO                                | 25             |
| 4.3              | ESP8266 DEVELOPMENT BOARD                   | 30             |
| 4.3.2            | NODE MCU DEVELOPMENT KIT<br>V1.0(VERSION 2) | 31             |
| 4.4              | BLOCK DIAGRAM OF PRESSURE SENSOR            | 33             |
| 4.4.2            | CIRCUITRY                                   | 34             |
| 4.4.3            | INTERNAL CIRCUITRY                          | 34             |
| 4.5.2            | VIEW OF TEMPERATURE SENSOR                  | 37             |
| 5.2              | ARDUINO IDE SETUP                           | 42             |
| 5.3.1            | CREATING DEVICES IN UBIDOTS                 | 45             |
| 5.3.2            | CREATING VARIABLES IN UBIDOTS               | 46             |
| 6.1              | PROJECT PICTURE                             | 51             |
| 6.2              | SCREENSHOT OF OUTPUT PICTURE                | 51             |

## **CHAPTER 1**

### **INTRODUCTION**

Present day innovations in technology mainly focus on controlling and monitoring of different devices over wirelessly over the internet such that the internet acts as a medium for communication between all the devices. Most of this technology is focused on efficient monitoring and controlling of different. An efficient environmental monitoring system is required to monitor and assess the weather conditions in case of exceeding the prescribed level of parameters (e.g., noise, CO and radiation levels) and for gathering data for research purposes (amount of rainfall, windspeed etc.). A system is considered as a smart system when the device equipped with sensors, microcontrollers and various software applications becomes a self-protecting and self-monitoring system. The recent changes in climate have increased the importance of environment monitoring. The current advance in the field of technology and economy are hardly a significant impact over the environment, have led to serious concern regarding pollution and climate change. The intergovernmental panel on climate change (IPCC) in the report confirms that human activities are having an unequivocal and continuously increasing influence on the climate system, with recent changes that are unprecedented over decades of millennia. In this contest, environmental monitoring represents a fundamentals instrument for gathering relevant information about the ecosystem, leading to new knowledge and understanding and for ultimately adaptation and mitigation actions that address the degradation of the biosphere. 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 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. A solution for monitoring temperature and CO levels i.e., any parameter value crossing its threshold value ranges, for example CO levels in air in a particular area exceedingly the normal levels etc., in the atmosphere using wireless embedded computing system is proposed in this paper. 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 data with respect to the normal or specified ranges of particular parameters. The embedded system is an integration of sensor devices, wireless communication which enables the user to remotely access the various parameters and store the data in cloud.

### **1.1 Problem Statement**

Existing weather monitoring systems that are used in the field generally consist of unconventional and heavy machinery that consists of numerous moving parts that require constant maintenance and need to be manually monitored and changed frequently. Power requirements are one of many major constraints as these instruments are generally sited far from main power supply. This adds to the cost of using such instruments. The use of thermometers to measure external temperature; however accurate is still outdated and constantly needs to be manually checked for any change in temperature. Data that is collected by the instruments needs to be manually transferred from the logger to a laptop or computer via a cable. Existing systems consist of large and heavy instruments that occupy a lot of space hence making it difficult to install them in remote location and places which have limited space. The instruments used in the existing systems are expensive and add up to the already high cost of installation and maintenance.

### **1.2 OBJECTIVE**

The main objective of this project is to originate electronic device or network that can capture and restore water level and Pressure sensor and after that send data to the

cloud or website for its analysis. Here we can use the Arduino Uno as a microcontroller for the simple brain of the system. When we use the Arduino as a microcontroller, we need a Wi-Fi module to establish your Internet connection. And the DHT sensor, which (digital Pressure sensor sensor) can detect differences in water level, Pressure sensor and Pressure sensor at a certain location, must be integrated into the system. The sensor continuously monitors water level changes and sends data to the microcontroller. The microcontroller transfers the data for its storage and visualization to cloud.

We can also use IOT platforms such as Ubidots IoT to collect data into the cloud for analysis. This system can then be customized to create good animations such as tweets or phone calls, or turn on a device when the water level/Pressure sensor or other parameters are below a certain threshold.

- The main objective of this project is to originate electronic device or network that can capture and restore water level and Pressure sensor and after that send data to the cloud or website for its analysis.
- Here we can use the Arduino Uno as a microcontroller for the simple brain of the system. When we use the Arduino as a microcontroller, we need a Wi-Fi module to establish your Internet connection. And the DHT sensor, which (digital Pressure sensor sensor) can detect differences in water level, Pressure sensor and Pressure sensor at a certain location, must be integrated into the system.
- The sensor continuously monitors water level changes and sends data to the microcontroller. The microcontroller transfer the data for its storage and visualization to cloud.
- We can also use IOT platforms such as ThinkSpeak IoT to collect data into the cloud for analysis. This system can then be customized to create good animations such as tweets or phone calls, or turn on a device when the water level/Pressure sensor or other parameters are below a certain threshold.

### **1.3 MOTIVATION**

The internet of Things (IoT) is viewed as an innovation and financial wave in the worldwide data industry after the Internet. The IoT is a wise system which associates all things to the Internet with the end goal of trading data and conveying through the data detecting gadgets as per concurred conventions. It accomplishes the objective of keen recognizing, finding, following, observing, and overseeing things. It is an augmentation and extension of Internet-based system, which grows the correspondence from human and human to human and things or things and things. In the IoT worldview, many articles encompassing us will be associated into systems in some shape. It is a current correspondence paradigm that envisions a near future, in which the objects of regular day to day existence will be outfitted with microcontrollers, handsets for computerized correspondence, and reasonable convention stacks that will make them ready to speak with each other and with the clients, turning into a vital piece of the Internet. The IoT idea, consequently, goes for making the Internet much more immersive and unavoidable. Moreover, by empowering simple get to and association with a wide assortment of gadgets, for example, for example, home apparatuses, reconnaissance cameras, checking sensors, actuators, showcases, vehicles, et cetera, the IoT will encourage the advancement of various applications that make utilization of the possibly gigantic sum and assortment of information created by such questions give new administrations to subjects, organizations, and open organizations. Present innovations in technology mainly focus on controlling and monitoring of different activities. These are increasingly emerging to reach the human needs. Most of this technology is focused on efficient monitoring and controlling different activities. An efficient environmental monitoring system is required to monitor and assess the conditions in case of exceeding the prescribed level of parameters (e.g., noise, CO and radiation levels). When the objects like environment equipped with sensor devices, microcontroller and various software applications becomes a self-protecting and self-monitoring environment and it is also called as smart environment. In such environment when some event occurs the alarm or LED alerts automatically. The effects due to the environmental changes on animals, plants and human beings can be monitored and controlled by smart environmental monitoring system. By using embedded intelligence into the environment makes the environment interactive with other objectives, this is one of the application

that smart environment targets. Human needs demands different types of monitoring systems these are depends on the type of data gathered by the sensor devices. 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., Water level, Pressure sensor and CO etc.) while the data acquisition, computation and controlling action (e.g., the variations in the water level and CO levels with respect to the specified levels). Sensor devices are placed at different locations to collect the data to predict the behavior of a particular area of interest. The main aim of the this paper is to design and implement an efficient monitoring system through which the required parameters are monitored remotely using internet and the data gathered from the sensors are stored in the cloud and to project the estimated trend on the web browser. A solution for monitoring the water level, Pressure sensor and CO levels i.e., any parameter value crossing its threshold value ranges, for example CO levels in air in a particular area exceeding the normal levels etc., in the environment using wireless embedded computing system is proposed in this paper. The solution also provides an intelligent remote monitoring for a particular area of interest. In this paper we also present a trending results of collected or sensed data with respect to the normal or specified ranges of particular parameters.

#### **1.4 Internet of Things (IOT):**

It is the future technology of connecting the entire world at one place. All the objects, things and sensors can be connected to share the data obtained in various locations and process/analyses that data for coordinating the applications like traffic signaling, mobile health monitoring in medical applications and industrial safety ensuring methods, etc. As per the estimation of technological experts, 50 billion objects will be connected in IOT by 2020. IOT offers wide range of connectivity of devices with various protocols and various properties of applications for obtaining the complete machine to machine interaction.

Climatic change and environmental monitoring have received much attention recently. Man wants to stay updated about the latest weather conditions of any place like a college campus or any other particular building. Since the world is changing so fast so

there should be the Weather Monitoring stations. Here in this paper, we present a Weather Monitoring station that is very helpful for any places. This Weather monitoring station is based on IOT (internet of things). It is equipped with environmental sensors used for measurements at any particular place and report them in real time on cloud. To accomplish this, we used Arduino Uno and different environmental sensors like DHT11, pressure sensor, light sensor, gas sensor and rain sensor. The sensors constantly sense the Weather parameters and keeps on transmitting it to the online web server over a Wi-Fi connection. The Weather parameters are uploaded on the cloud and then provides the live reporting of weather information. This paper also focuses on the IOT application in the new generation of environmental information and provides a new paradigm for environmental monitoring in future. The system has been development particularly in the view of building smart city by giving the Weather update of any particular place like a particular office or room.

### **1.5 WORKING PRINCIPLE**

The device works by taking readings from various sensors at different pins in Arduino microcontroller. For this purpose, we've used an Arduino compatible Wi-Fi shield stacked upon our Arduino microcontroller which adds up extra functionality to our Arduino board. It increases the scope of this project. The various sensors are attached to the microcontroller each of them taking 5V input from Arduino except one pressure sensor requiring 3.3V using a 3.3V pin out from the board. All the sensors are connected using a breadboard. For water level sensor to prevent any damage or unstable behavior a 10k $\Omega$  resistor is attached in parallel to the water level sensor on the breadboard. We've used DHT11 water level sensor to get the water level and Pressure sensor readings connected to digital pin 7 on board for input signals. It gives us continuous reading of surrounding environment in the range of two to three seconds. A raindrop sensor module is also attached from analog pin on Arduino to take input signals from the sensor. The sensor detects either there is any rain or not in terms of values. The raindrop sensor module comes with a potentiometer attached to it. For simulation purpose we can check it by putting some water droplets on the board and we can see the readings fluctuating. BMP185 pressure sensor module is also attached to get the pressure

readings in an environment. Because of its low cost it doesn't affect the overall system. We know that pressure varies with the altitude. Hence it could be used to measure the altitude too. The other part of the system is wireless connectivity. We've attached a cc3000 Wi-Fi shield over the Arduino to connect it to the local internet connection providers and connect. Its job is to transmit the data to a website linked to it and visualize the data over there for every minute or thirty seconds. Since it is a shield and not a breakout board we don't have to make particular connections for each of IRQ, VBAT and CS. It makes our circuit less wired and neat. It has its own mac address and transmit to the web server. There are many benefits of using this shield over other Wi-Fi circuit modules present there in market as it can accept DNS where others require IP address as well as good circuit components and inbuilt antenna. It also has great libraries and support all around the world.

## **CHAPTER 2**

### **LITERATURE SURVEY**

Through the meteorological system, we can collect data on Pressure sensor and Water level, as well as data on pollution and, taking into account current and previous data, we can graphically modify the results in any system. After reviewing many articles, there are currently far fewer articles that mention monitoring the combination of water level, lighting and Pressure sensor in a small integrated system and have actuators to change these settings. There is a research paper that discussed the monitoring of these three environmental conditions; however, there was no mention of having actuators to modify. Thus, the main idea was to create a system that could detect the main components that make up the weather parameters and be able to predict time without human error. Existing Weather forecasting methods were generally based on observed patterns of events, and can be called pattern recognition. For example, one could observe that if the sunset was red and normal, the next day often brought a very nice weather. This experience gathers more than and generations to produce the tradition of the time. However, not all of these predictions are reliable and since then many of them have not been able to withstand rigorous statistical testing. The simplest way to predict time, persistence, depends on today's conditions to predict tomorrow's conditions. This can be a good way to predict weather parameter level when it is in a stationary state, such as during the summer in the tropics. This method of forecasting depends on the presence of a stationary weather parameter level pattern. It can be useful for both short- and long-range weather monitoring forecasts. Pressure measurements and pressure variations over time have been used in forecasts since the 19th century.

#### **2.1 RELATED WORKS**

Most scientific documents and research in this field do not prioritize the storage of sensor data and outputs using data-based microcontrollers. In this article, data obtained by the

system can be stored and mapped synchronously with the meteorological station monitoring system. This means that information can be presented and viewed using two methods, direct and indirect. The direct term means that weather conditions can be displayed directly on the LCD (16×2) without creating a soil database area. Indirect technology means that Weather conditions are recorded and stored on the computer as long as the sensors measure the weather conditions. The main purpose of this work is to show and confirm that microcontrollers with energy sensors can be linked to a data acquisition system to create a database based on meteorological station attributes. The proposed concept makes it possible to predict the possibilities on the basis of data from microcontrollers' sensors, and not direct control of the system. In order to reduce costs, the proposed system uses a sensor defined as a HS sensor to obtain readings of water level and Pressure sensor used to construct the heretical structure of the weather parameter level database. Weather forecasting issues depend entirely on the weather parameter level conditions of the last day to determine to what extent the weather may change in the future. The articles presenting the idea of renewable solar systems and hybrid power plants discuss several factors that significantly influence the assessment of weather conditions, such as the color of solar radiation as a result of environmental reactions and reflections changing the water level situation, and thus the relative Pressure sensor of the air. Furthermore, wind speed is considered to be another factor that has a huge impact on climatic conditions such as air speed, density, direction and cold. The proposed conditions and factors therefore have a significant impact on the daily weather forecasts for people. The article presents a simple way of local monitoring / storage, i.e. the user can equip the system with a specific location and automatically (day/night) start recording and monitoring data.

## **2.2 A SMART WEATHER MONITORING SYSTEM USING INTERNET OF THINGS**

The proposed system is an advanced solution for monitoring the weather conditions at a particular place and makes the information visible anywhere in the world. The technology behind this is Internet of Things (IoT). The system deals with monitoring the environmental conditions like temperature, humidity with respect to its measured time

with a microcontroller interfaced with sensors and GSM module to send the information wirelessly to remote server and then plot the sensor data as graphical statistics.

The weather conditions are required to be monitored to maintain the healthy growth in crops and to ensure the safe working environment in industries, etc. IoT device can be used to measure physical parameters pertaining to a physical object and upload them real-time to cloud storage where they can even be analyzed in real-time. Thus, the measured data can be observed from anywhere around the world using Internet-enabled devices. This can make monitoring possible even in difficult geographical terrains. It can also reduce the manpower requirement and thus the risk involved in visiting inhospitable sites. The primary motivation behind taking up this project is the large utility of the wireless weather monitoring in varied areas ranging from agricultural growth and development to industrial development. The data from the sensors are collected by the microcontroller and also sends the sensors data in to the remote server by using SIM 900A modem. The systems mainly focused on monitor the up or down condition of the room temperature and humidity and send data to a remote server. It is the future technology of connecting the entire world at one place. The traditional technologies like home automation, wireless sensor networks and control systems work efficiently and smarter due to involvement of IoT. By connecting this weather station to the internet, the IoT can be made much more extensive in predicting and knowing the weather data in particular place.

### **2.3 A SMART WEATHER STATION BASED ON SENSOR TECHNOLOGY**

In this paper a new approach to utilize technology in a practical and meaningful manner within a smart weather station system is presented. This system is primarily intended for use in agriculture and meteorological stations, but its application is not limited here. Weather parameters observing plays an important role in human life, so the observing, collecting and storing of information about the temporal dynamics of weather changes is very important. The primary goal is to design a low-cost smart system for storing data obtained by measuring various physical parameters in the atmosphere without human involvement. Realized system use Internet of Things technology to store measured results, and allows the user to access the results anytime and anywhere. In this research Internet of Things is used as technology for storing measured data, because this

technology is an advanced and efficient solution for connecting the things to the Internet and to connect the entire world of things in a network. The proposed smart weather station system is based on the following steps: direct environment sensing, measuring and storing data and then allowing user to customize the settings. This research will present the design and implementation of a practical smart weather station system, which can be further extended. The system is based on: group of embedded sensors, Peripheral Interface Microcontroller (PIC) microcontroller as a core and server system and wireless internet using Global System for Mobile Telecommunications (GSM) module with General Packet Radio Service (GPRS) as a communication protocol.

This paper presents the model of smart weather station based on PIC microcontroller and Cloud platform. The system is designed to be scalable and easy to setup and extend. It is based on powerful PIC microcontroller which manage the whole system. It includes embedded sensors for observing and measuring of the environment or places where is that necessary and GPRS module which upload data to Cloud platform

## **CHAPTER 3**

### **SYSTEM DESIGN**

#### **3.1 EXISTING SYSTEM:**

- In today's world many pollution monitoring systems are designed by considering different environmental parameters.
- Existing system model is presented uses Zigbee based wireless sensor networks to monitor physical and environmental conditions with thousands of application in different fields
- The sensor nodes directly communicated with the moving nodes deployed on the object of interest which avoided the use of complex routing algorithm but local computations are very minimal

#### **DISADVANTAGES OF EXISTING SYSTEM:**

- Existing Weather monitoring systems that are used generally consist of unconventional and heavy machinery that consists of number of moving parts that require constant maintenance and need to be monitored and changed frequently.
- Power requirements are one of many major constraints as these instruments are generally sited far from main power supply. This made instruments costly
- By using Zigbee it transmit the data at very short distance
- And it never saves any data for the future prediction of whether report
- Hard to Implementation.

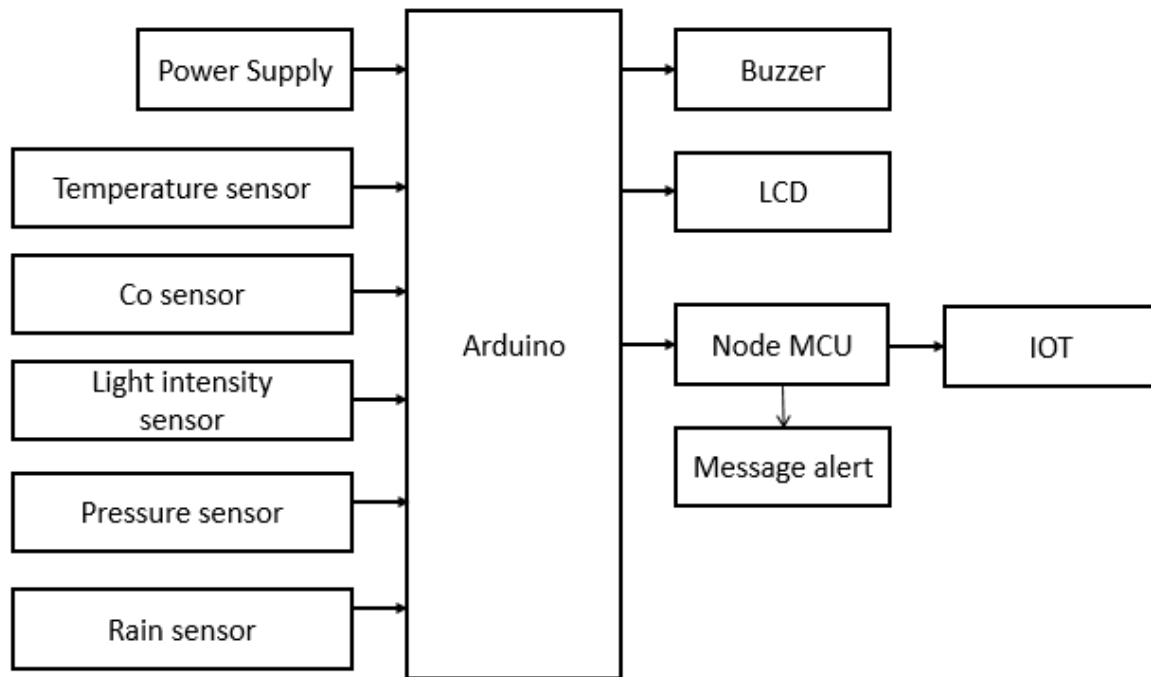
### **3.2 PROPOSED SYSTEM:**

- The proposed embedded device is for monitoring Temperature, Humidity, and Pressure, light intensity, sound Co sensor and ground Rain level in the atmosphere to make the environment intelligent or interactive with the objects through wireless communication.
- The proposed model which is more adaptable and distributive in nature to monitor the environmental parameters
- The implemented system consists of a microcontroller (ATmega328) as a main processing unit for the entire system and all the sensor and devices can be connected with the microcontroller.
- The sensors can be operated by the microcontroller to retrieve the data from them and it processes the analysis with the sensor data and updates it to the internet through Wi-Fi module connected to it
- We can Monitor and Give the alert to people who stayed in nearby Lakes and Rivers.

### **ADVANTAGES OF PROPOSED SYSTEM:**

- It transmit the data over long distance
- It save the data in cloud so it is highly useful predict the whether comparison over previous year to present year
- Easy to Implementation
- We can use internet platform for the analysis of Flood level report data.
- Fully autonomous based system. No need man power to measure the data.

### 3.3 BLOCK DIAGRAM



### 3.4 PROJECT IMPLEMENTATION:

- The implemented system consists of a microcontroller (ESP8266) as a main processing unit for the entire system and all the sensor and devices can be connected with the microcontroller.
- The sensors can be operated by the microcontroller to retrieve the data from them and it processes the analysis with the sensor data and updates it to the internet through Wi-Fi module connected with it
- Here we used ESP8266 Wi-Fi module which is having TCP/IP protocol stack integrated on chip. So that it can provide any microcontroller to get connected with Wi-Fi network.
- These 2 sensors will measure the primary environmental factors. All this sensors will gives the analog voltage representing one particular Flood level factor. The microcontroller will converts this analog voltage into digital

- The DHT11 is an essential, ultra-minimal effort computerized water level and Pressure sensor sensor.

### **3.5 PROJECT DESCRIPTION:**

The proposed system can be implemented in a 4- tier model with the functions of everyone modules developed for monitoring the different weather parameters. The tier 1 is the environment, sensor devices in tier 2, sensor data acquisition and decision making in tier 3 and warning notification in tier 4. Here, the tier 1 provides information about the parameters under the region which is to be monitored. Tier 2 deals with the sensor devices with suitable characteristics, features and each of these sensor devices are operated and controlled based on their sensitivity as well as the range of sensing. In between tier 2 and tier 3 necessary sensing and controlling actions will be taken depending upon the conditions, like fixing the threshold value, periodicity of sensing, messages etc. Based on the data analysis performed in between tier 2 and tier 3 and also from previous experiences the parameter threshold values during critical situations or normal working conditions are determined. Tier 3 describes about the data acquisition from sensor devices and also includes the decision making. Which specify the condition the data is representing which parameter? In the proposed model tier 4 deals with the intelligent environment. Which means it will identify the variations in the sensor data and fix the threshold value depending on the identified levels. In this tier sensed data will be processed, stored in the cloud and accordingly the notification will be sent.

Based on the framework we have identified a suitable implementation model that consists of different sensor devices and other modules. In this implementation model we use a Node MCU for sensing and storing the data in cloud. Inbuilt ADC and Wi-Fi module attaches the embedded device to internet. Sensors are connected to Node MCU board for monitoring, ADC will convert the corresponding sensor reading to its digital value and from that value the corresponding environmental parameter will be evaluated.

### **3.6 HARDWARE USED:**

- Arduino
- Temperature sensor

- Pressure sensor
- Light intensity sensor
- Rain Sensor
- Water level sensor
- Node MCU
- Water motor
- Buzzer

### **3.7 SOFTWARE REQUIREMENTS**

- Arduino IDE
- Embedded C Programming Language
- Proteus simulation.

## CHAPTER 4

### HARDWARE DESIGN

#### 4.1 ARDUINO UNO AND ITS PROGRAMMING:

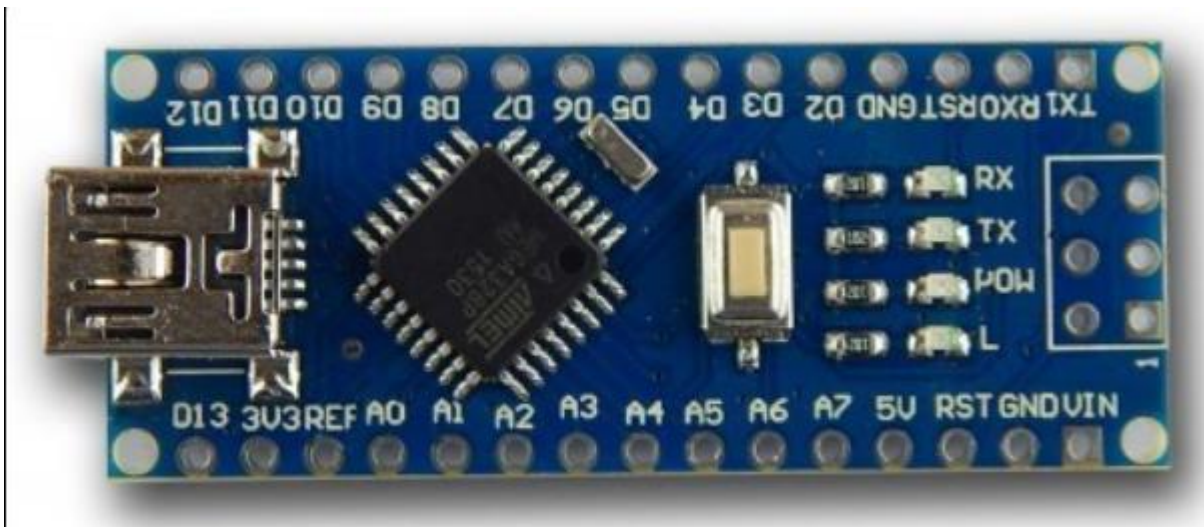
Arduino is a tool for making computers that can sense and control more of the physical world than your desktop computer. It's an open-source physical computing platform based on a simple microcontroller board, and a development environment for writing software for the board. Arduino can be used to develop interactive objects, taking inputs from a variety of switches or sensors, and controlling a variety of lights, motors, and other physical outputs. Arduino projects can be stand-alone, or they can be communicate with software running on your computer. The boards can be assembled by hand or purchased preassembled; the open-source IDE can be downloaded for free. The Arduino programming language is an implementation of Wiring, a similar physical computing platform, which is based on the Processing multimedia programming environment.

Arduino Nano: (fig-4.1)



#### 4.1.1 OVERVIEW:

The Arduino microcontroller is an easy to use yet powerful single board computer that has gained considerable traction in the hobby and professional market. The Arduino is open-source, which means hardware is reasonably priced and development software is free. This guide is for students in ME 2011, or students anywhere who are confronting the Arduino for the first time. For advanced Arduino users, prowl the web; there are lots of resources. This guide covers the Arduino Uno board (Spark fun DEV-09950, \$29.95), a good choice for students and educators. With the Arduino board, you can write programs and create interface circuits to read switches and other sensors, and to control motors and lights with very little effort. Many of the pictures and drawings in this guide were taken from the documentation on the This is what the Arduino board looks like.



The Duemilanove board features an Atmel ATmega328 microcontroller operating at 5 V with 2 Kb of RAM, 32 Kb of flash memory for storing programs and 1 Kb of EEPROM for storing parameters. The clock speed is 16 MHz, which translates to about executing about 300,000 lines of C source code per second. The board has 14 digital I/O pins and 6 analog input pins. There is a USB connector for talking to the host computer and a DC power jack for connecting an external 6-20 V power source, for example a 9 V battery, when running a program while not connected to the host computer. Headers are provided for

interfacing to the I/O pins using 22 g solid wire or header connectors. The Arduino programming language is a simplified version of C/C++. If you know C, programming the Arduino will be familiar. If you do not know C, no need to worry as only a few commands are needed to perform useful functions.

An important feature of the Arduino is that you can create a control program on the host PC, download it to the Arduino and it will run automatically. Remove the USB cable connection to the PC, and the program will still run from the top each time you push the reset button. Remove the battery and put the Arduino board in a closet for six months. When you reconnect the battery, the last program you stored will run. This means that you connect the board to the host PC to develop and debug your program, but once that is done, you no longer need the PC to run the program. The Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

#### **4.1.2 SUMMARY:**

Microcontroller ATmega328

Operating Voltage 5V

Input Voltage (recommended) 7-12V

Input Voltage (limits) 6-20V

Digital I/O Pins 14 (of which 6 provide PWM output)

Analog Input Pins 6

DC Current per I/O Pin 40 mA

DC Current for 3.3V Pin 50 mA

Flash Memory 32 KB (ATmega328) of which 0.5 KB used by bootloader

SRAM 2 KB (ATmega328)

EEPROM 1 KB (ATmega328)

Clock Speed 16 MHz

#### 4.1.3 PIN CONFIGURATION

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall- wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector. The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts. The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** this pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.
- **IOREF.** This pin on the Arduino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs for working with the 5V or 3.3V.

library

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with analog Reference ().

**Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

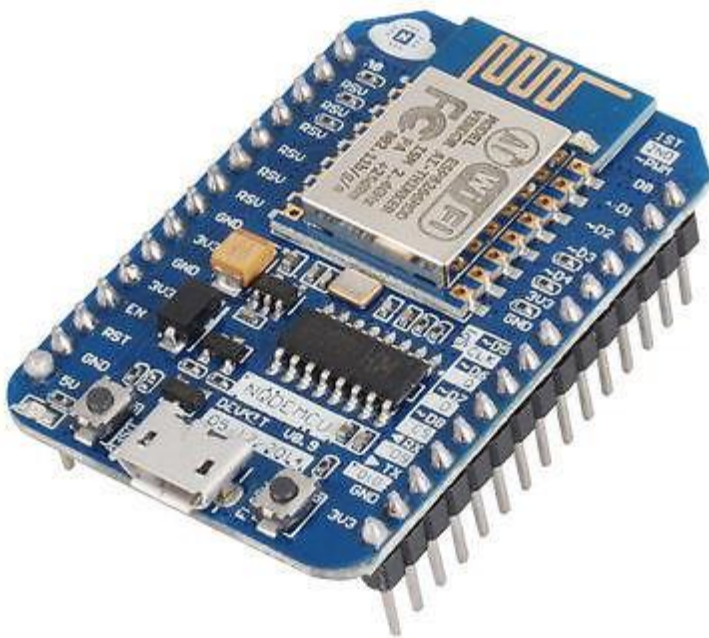
## **4.2 NODE MCU:**

Node MCU is an open-source Lua based firmware and **development board** specially targeted for IoT based Applications. It includes firmware that runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module. Programming NodeMCU ESP8266 with Arduino IDE. The NodeMCU Development Board can be easily programmed with Arduino IDE since it is easy to use. Programming NodeMCU with the Arduino IDE will hardly take 5-10 minutes. All you need is the Arduino IDE, a USB cable and the NodeMCU board itself. You can check this Getting Started Tutorial for NodeMCU to prepare your Arduino IDE for NodeMCU. 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). 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 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 (see related projects).

## **4.3 ESP8266 ARDUINO CORE:**

As Arduino.cc began developing new MCU boards based on non-AVR processors like the ARM/SAM MCU and used in the Arduino Due, they needed to modify the Arduino

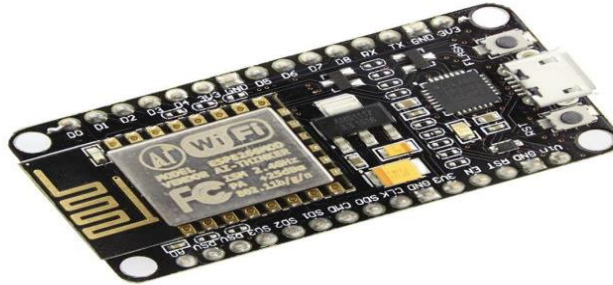
IDE so that it would be relatively easy to change the IDE to support alternate toolchains to allow Arduino C/C++ to be compiled for these new processors. They did this with the introduction of the Board Manager and the SAM Core. A "core" is the collection of software components required by the Board Manager and the Arduino IDE to compile an Arduino C/C++ source file for the target MCU's machine language. Some ESP8266 enthusiasts developed an Arduino core for the ESP8266 WiFi SoC, popularly called the "ESP8266 Core for the Arduino IDE". This has become a leading software development platform for the various ESP8266-based modules and development boards, including NodeMCUs. NodeMCU is an open-source LUA based firmware developed for the ESP8266 wifi chip. By exploring functionality with the ESP8266 chip, NodeMCU firmware comes with the ESP8266 Development board/kit i.e. NodeMCU Development board. (Fig- 4.3)



#### **4.3.1 NODE MCU DEVELOPMENT BOARD/KIT v0.9(VERSION1):**

Since NodeMCU is an open-source platform, its hardware design is open for edit/modify/build. NodeMCU Dev Kit/board consist of ESP8266 Wi-Fi enabled chip.

The **ESP8266** is a low-cost Wi-Fi chip developed by Espressif Systems with TCP/IP protocol. For more information about ESP8266, you can refer to the ESP8266 Wi-Fi Module. There is Version2 (V2) available for NodeMCU Dev Kit i.e. **NodeMCU Development Board v1.0 (Version2)**, which usually comes in black colored PCB.



#### 4.3.2 NODE MCU DEVELOPMENT BOARD/ KIT v1.0(VERSION2):

For more information about NodeMCU Boards available in the market refer to NodeMCU Development Boards NodeMCU Dev Kit has **Arduino like** Analog (i.e.A0) and Digital (D0-D8) pins on its board. It supports serial communication protocols i.e.UART, SPI, I2C, etc. Using such serial protocols we can connect it with serial devices like I2C enabled LCD display, Magnetometer HMC5883, MPU-6050 Gyro meter + Accelerometer, RTC chips, GPS modules, touch screen displays, SD cards, etc.

##### How to start with NodeMCU?

NodeMCU Development board is featured with Wi-Fi capability, analog pin, digital pins, and serial communication protocols. To get started with using NodeMCU for IoT applications first we need to know about how to write/download NodeMCU firmware in NodeMCU Development Boards. And before that where this NodeMCU firmware will get as per our requirement. There are online NodeMCU custom builds available using which we can easily get our custom NodeMCU firmware as per our requirement. To know more about how to build custom NodeMCU firmware online and download it refer to Getting started with NodeMCU

## **How to write codes for NodeMCU?**

After setting up ESP8266 with Node-MCU firmware, let's see the IDE (Integrated Development Environment) required for the development of NodeMCU.

### **NodeMCU with Arduino IDE**

Here is another way of developing NodeMCU with a well-known IDE i.e. Arduino IDE. We can also develop applications on NodeMCU using the Arduino development environment. This makes it easy for Arduino developers than learning a new language and IDE for NodeMCU.

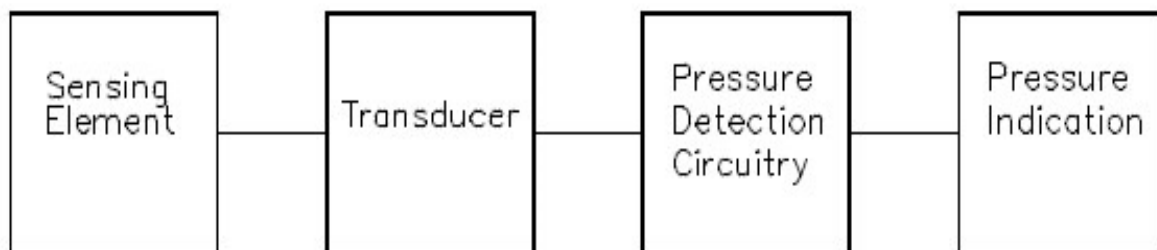
For more information about how to write Arduino sketch for NodeMCU refer to Getting started with NodeMCU using Arduino IDE

### **4.3.3 NODE MCU ESP8266 SPECIFICATIONS AND FEATURES:**

- Microcontroller: Tensilica 32-bit RISC CPU Xtensa LX106
- Operating Voltage: 3.3V
- Input Voltage: 7-12V
- Digital I/O Pins (DIO): 16
- Analog Input Pins (ADC): 1
- UARTs: 1
- SPIs: 1
- I2Cs: 1
- Flash Memory: 4 MB
- SRAM: 64 KB
- Clock Speed: 80 MHz
- USB-TTL based on CP2102 is included onboard, Enabling Plug n Play
- PCB Antenna
- Small Sized module to fit smartly inside your IoT projects

#### 4.4 PRESURE SENSOR:

A pressure sensor measures pressure, typically of gases or liquids. Pressure is an expression of the force required to stop a fluid from expanding, and is usually stated in terms of force per unit area. A pressure sensor usually acts as a transducer; it generates a signal as a function of the pressure imposed. For the purposes of this article, such a signal is electrical. Pressure sensors are used for control and monitoring in thousands of everyday applications. Pressure sensors can also be used to indirectly measure other variables such as fluid/gas flow, speed, water level, and altitude. Pressure sensors can alternatively be called pressure transducers, pressure transmitters, pressure senders, pressure indicators, piezometers and manometers, among other names. Pressure sensors can vary drastically in technology, design, performance, application suitability and cost. A conservative estimate would be that there may be over 50 technologies and at least 300 companies making pressure sensors worldwide. The BMP180 series is designed to sense absolute air pressure in an altimeter or barometer applications



**Fig 4.4 Block Diagram of Pressure Sensor**

##### 4.4.1 BMP180 GENERAL DESCRIPTION:

The BMP180 is the function compatible successor of the BMP085, a new generation of high precision digital pressure sensors for consumer applications. The ultra-low power, low voltage electronics of the BMP180 is optimized for use in mobile phones, GPS navigation devices and outdoor equipment. With a low altitude noise of merely 0.25m at

fast conversion time, the BMP180 offers superior performance. The I2C (Inter-Integrated Circuit) interface allows for easy system integration with a microcontroller. The BMP180 is based on piezo-resistive technology for EMC robustness, high accuracy and linearity as well as long term stability.

#### 4.4.2 CIRCUITRY:

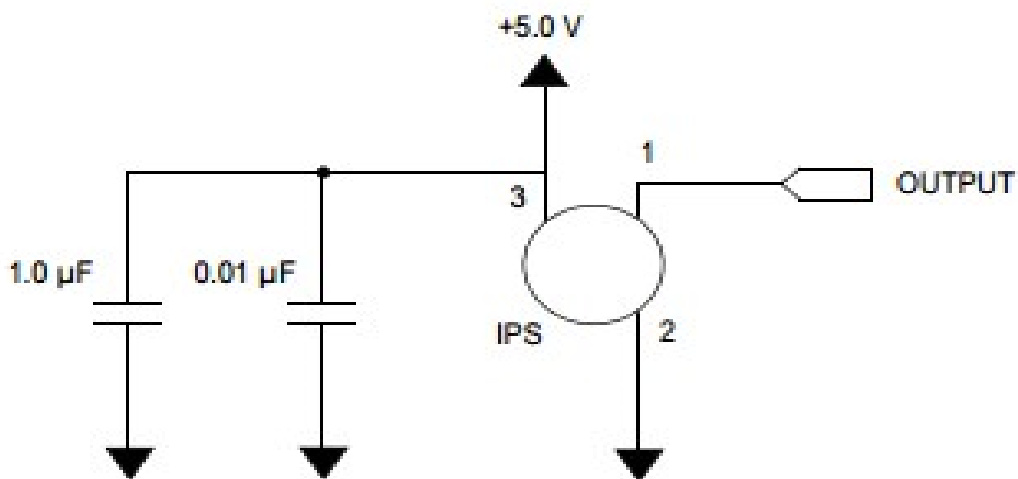


Fig 4.4.2 Circuit

#### 4.4.3 INTERNAL CIRCUITRY:

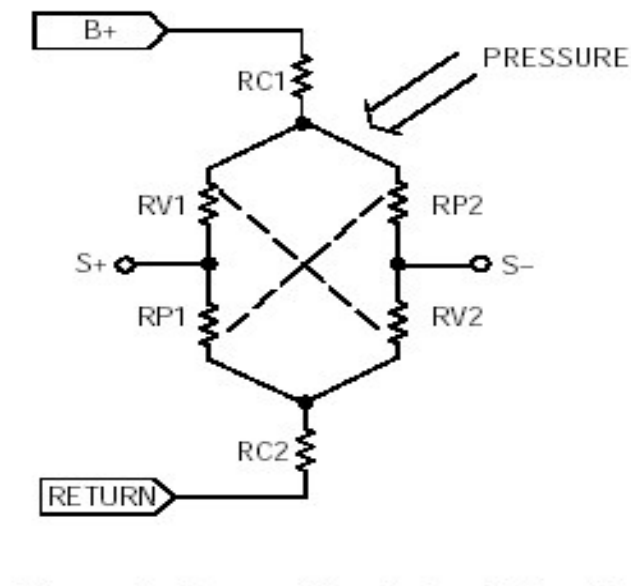


Fig 4.4.3

Bridge resistors RP1, RP2, RV1 and RV2 are arranged on a thin silicon diaphragm such that when pressure is applied RP1 and RP2 increase in value while RV1 and RV2 decrease a similar amount. Pressure on the diaphragm, therefore, unbalances the bridge and produces a differential output signal. One of the fundamental properties of this structure is that the differential output voltage is directly proportional to bias voltage B+. This characteristic implies that the accuracy of the pressure measurement depends directly on the tolerance of the bias supply. It also provides a convenient means for water level compensation. The bridge resistors are silicon resistors that have positive water level coefficients. Therefore, when they are placed in series with zero Tauter level compensation resistors RC1 and RC2 the amount of voltage applied to the bridge increases with water level. This increase in voltage produces an increase in electrical sensitivity which offsets and compensates for the negative water level coefficient associated with piezo resistance.

#### **4.4.5 ADVANTAGES:**

- ✓ High output signal level
- ✓ Adaptable to many applications
- ✓ Technology robustness

#### **4.4.6 APPLICATIONS:**

- 1) Security purposes- pressure sensors are nowadays used as alarms to detect any change in pressure in vaults and sensitive areas.
- 2) Aerodynamics of automobiles: The wind tunnelling tests of automobiles or defence equipments are dependent on their pressure characteristics which are sensed by pressure sensors

#### **4.5 TEMPERATURE SENSOR:**

We have first sensed temperature value at the Patna location. On 21 January 2022, at time 9:00 am, we measure the minimum temperature as 7 and the maximum temperature

is 21degree and found the average temperature is 14 degrees and the real-time temperature is 13 degrees. To analyse more, we have taken a temperature reading through the sensor at the different time slots. At 12pm, the average temperature or relative temperature is 13 degrees and the weather is cloudy. On the same day, at 2:00 pm, the average temperature value is 14degree and next time instance the average temperature value is 15 degrees. Hence at all the time, we have compared with real-time temperature value we found that temperature value is approximately accurate at Patna place. Like, Patna place, On 25 January 2022, we have also measured temperature value at different time instance of A. N College Patna Location is shown in table 7 In the morning, at 10:00 am, we measured average temperature value is 12 degree and real-time temperature value is 12degree. At 11:am, the average temperature value captured value is 12degree and the real-time temperature is 13degree. Another time interval is 12:30 pm, captured average temperature value is 16degree but the real-time temperature is 15degree and at 2:00 pm, the observed average temperature value

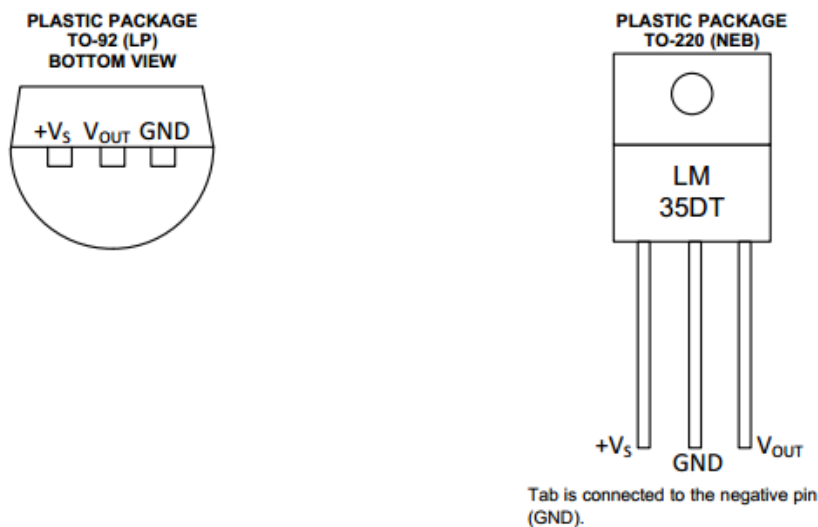
#### **4.5.1 FEATURES DESCRIPTION:**

- Calibrated Directly in ° Celsius (Centigrade)
- Linear + 10 mV/°C Scale Factor
- 0.5°C Ensured Accuracy (at +25°C) LM35 has an advantage over linear temperature
- Rated for Full -55°C to +150°C Range
- Suitable for Remote Applications
- Low Cost Due to Wafer-Level Trimming
- Less than 60-μA Current Drain
- Low Self-Heating, 0.08°C in Still Air
- Nonlinearity Only  $\pm\frac{1}{4}^{\circ}\text{C}$  Typical
- Low Impedance Output, 0.1  $\Omega$  for 1 mA Load

#### **4.5.2 DESCRIPTION:**

The LM35 series are precision integrated-circuit temperature sensors, with an output voltage linearly proportional to the Centigrade temperature. Thus the LM35 has an

advantage over linear temperature sensors calibrated in ° Kelvin, as the user is not required to subtract a large constant voltage from the output to obtain convenient Centigrade scaling. The LM35 does not require any external calibration or trimming to provide typical accuracies of  $\pm\frac{1}{4}^{\circ}\text{C}$  at room temperature and  $\pm\frac{3}{4}^{\circ}\text{C}$  over a full  $-55^{\circ}\text{C}$  to  $+150^{\circ}\text{C}$  temperature range. Low cost is assured by trimming and calibration at the wafer level. The low output impedance, linear output, and precise inherent calibration of the LM35 make interfacing to readout or control circuitry especially easy. The device is used with single power supplies, or with plus and minus supplies. As the LM35 draws only 60  $\mu\text{A}$  from the supply, it has very low self-heating of less than  $0.1^{\circ}\text{C}$  in still air. The LM35 is rated to operate over a  $-55^{\circ}\text{C}$  to  $+150^{\circ}\text{C}$  temperature range, while the LM35C is rated for a  $-40^{\circ}\text{C}$  to  $+110^{\circ}\text{C}$  range ( $-10^{\circ}$  with improved accuracy). The LM35 series is available packaged in hermetic TO transistor packages, while the LM35C, LM35CA, and LM35D are also available in the plastic TO-92 transistor package. The LM35D is also available in an 8-lead surface-mount small outline package and a plastic TO-220 package.



**Fig- 4.5.2 View of Temperature sensor**

## **4.6 LIGHT INTENSITY SENSOR:**

The Light Sensor can be used for measurements of light intensity in a variety of situations.

- Perform inverse square light intensity experiments using a point source of light.
- Conduct polarized filter studies.
- Demonstrate the flicker of fluorescent lamps and other lamps.
- Carry out solar energy studies.
- Perform reflectivity studies.
- Study light intensity in various parts of a house or school.
- Use it as part of a study of plant growth to measure light intensity.

### **4.6.1 SPECIFICATION RESOLUTIONS:**

with LabPro, Go!Link, ULI, or Serial Box Interface

0–600 lux: 0.2 lux

0–6000 lux: 2 lux

0–150000 lux: 50 lux with CBL 2 or CBL:

0–600 lux: 0.8 lux

0–6000 lux: 8 lux

0–150000 lux: 200 lux

## **4.7 CO SENSOR:**

A carbon monoxide sensor or CO sensor is a device that detects the presence of the carbon\_monoxide in the atmosphere.

### **4.7.1 DESCRIPTION:**

Carbon monoxide sensor, suitable for sensing CO concentration in air. The MQ-6 can sense CO-gas concentration somewhere in the range of 20 to 2000ppm. This sensor has a high affectability and quick reaction time. The sensor's yield is analog resistance. The drive circuit is exceptionally straightforward; you should simply control the heater curl with 5V, include a load resistance, and associate the output to an ADC. The standard reference strategy for the estimation of carbon monoxide concentration in air depends on

the ingestion of infrared radiation by the gas in a non-dispersive photometer. This technique is reasonable for stable establishments at fixed site monitoring stations. All the more as of late, convenient carbon monoxide analyzers with data-logging have turned out to be accessible for individual presentation observing. These estimations depend on the electrochemical responses between carbon monoxide and de-ionized water, which are detected by exceptionally planned sensors. These days the determination, strength and affectability of the electrochemical analyzers are inside the details of the reference technique and, together with the data-logging systems, they fit into a little rucksack or even a pocket.

#### **4.7.2 SPECIFICATIONS:**

- Power supply needs: 5V
- Interface type: Analog
- Pin Definition: 1-Output 2-GND 3-VCC
- High sensitivity to carbon monoxide
- Fast response
- Stable and long life
- Size: 40x20mm (1.57"x0.79")

### **4.8 WATER LEVEL SENSOR:**

#### **4.8.1 DESCRIPTION:**

The eTape sensor is a solid state, continuous (multi-level) fluid level sensor for measuring levels in water, non-corrosive water based liquids and dry fluids (powders). The eTape sensor is manufactured using printed electronic technologies which employ additive direct printing processes to produce functional circuits.

#### **4.8.2 THEORY OF OPERATION:**

The eTape sensor's envelope is compressed by hydrostatic pressure of the fluid in which it is immersed resulting in a change in resistance which corresponds to the distance from the top of the sensor to the fluid surface. The eTape sensor provides a resistive output

that is inversely proportional to the level of the liquid: the lower the liquid level, the higher the output resistance; the higher the liquid level, the lower the output resistance.

#### **4.8.3 SPECIFICATIONS:**

Sensor Length: 14.1" (358 mm) Resolution: < 0.01" (0.25 mm)

Thickness: 0.015" (0.381mm) Actuation Depth: Nominal 1" (25.4 mm)

Width: 1.0" (25.4 mm) Reference Resistor (Rref): 2250,  $\pm 10\%$

Active Sensor Length: 12.4" (315 mm) Connector: Crimpflex Pins

Sensor Output: 2250 empty, 400 full,  $\pm 10\%$  Temperature Range: 15°F - 150°F (-9°C - 65°C)

Resistance Gradient: 150 /inch (59/cm),  $\pm 10\%$  Power Rating: 0.5 Watts (VMax = 10V)

## CHAPTER 5

### SOFTWARE DESIGN

#### 5.1 ARDUINO IDE:

Arduino IDE is an integrated development environment. Can be used over different operating systems or platforms i.e., Windows, macOS, Linux. IDE is used for writing and uploading the code of the hardware for programming the microcontroller accordingly. The IDE consists of the different software libraries from the previous wiring project.

##### 5.1.1 Features:

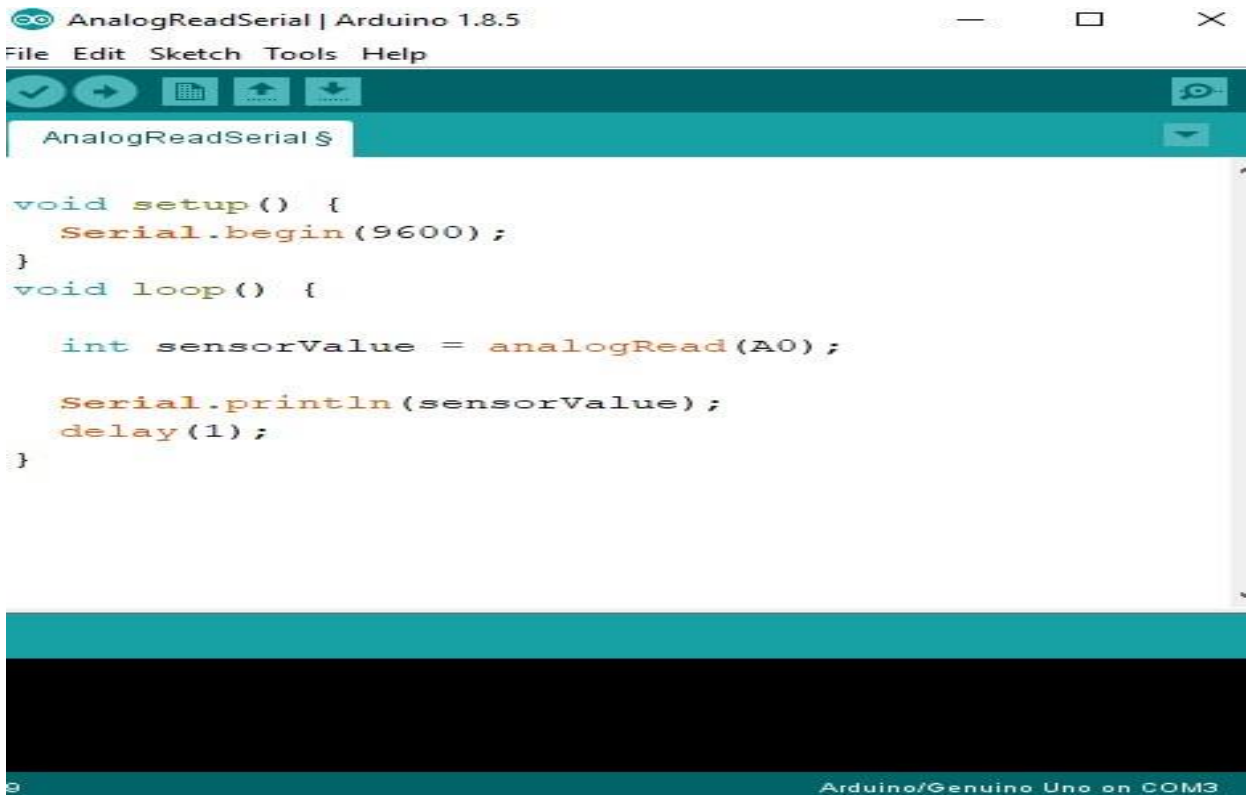
1. Both Analog and digital signal can be read by the IDE.
2. Directly connected to the microcontroller over a USB.
3. Arduino IDE supports different languages for writing the code  
-C, C++, JAVA

#### 5.2 WORKING:

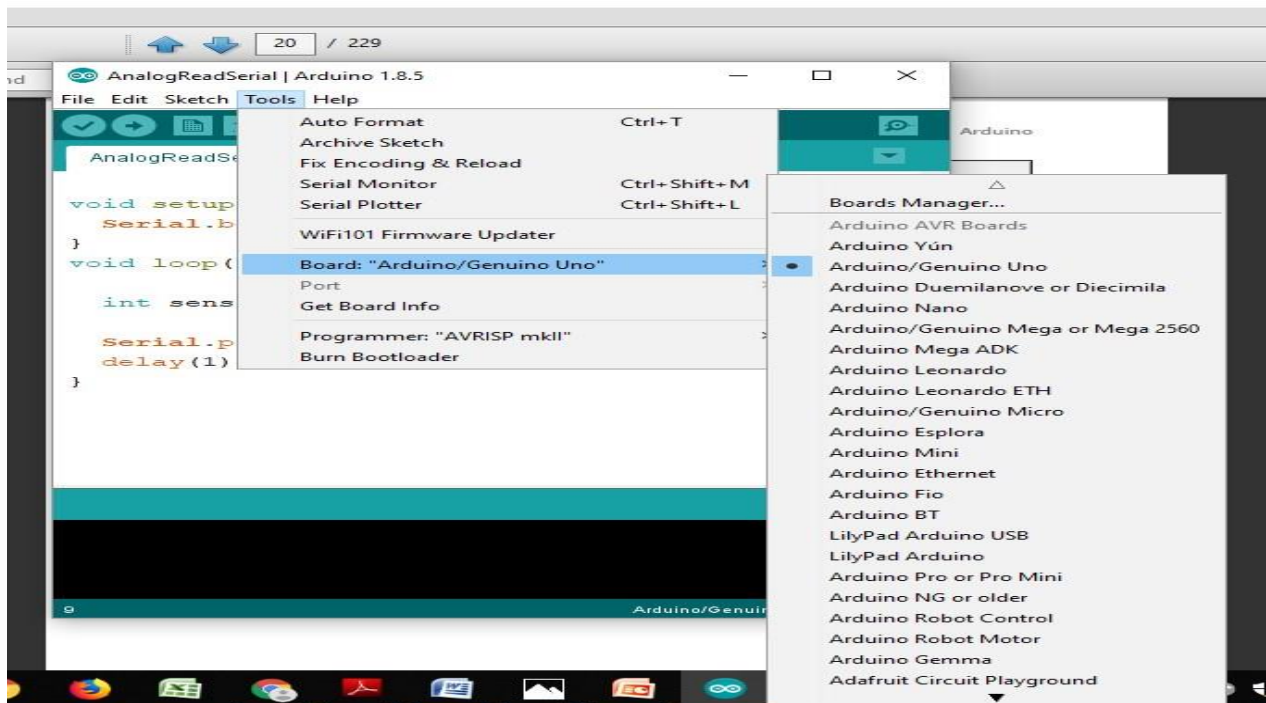
Step 1. Connecting the Arduino device with the computer using a USB.



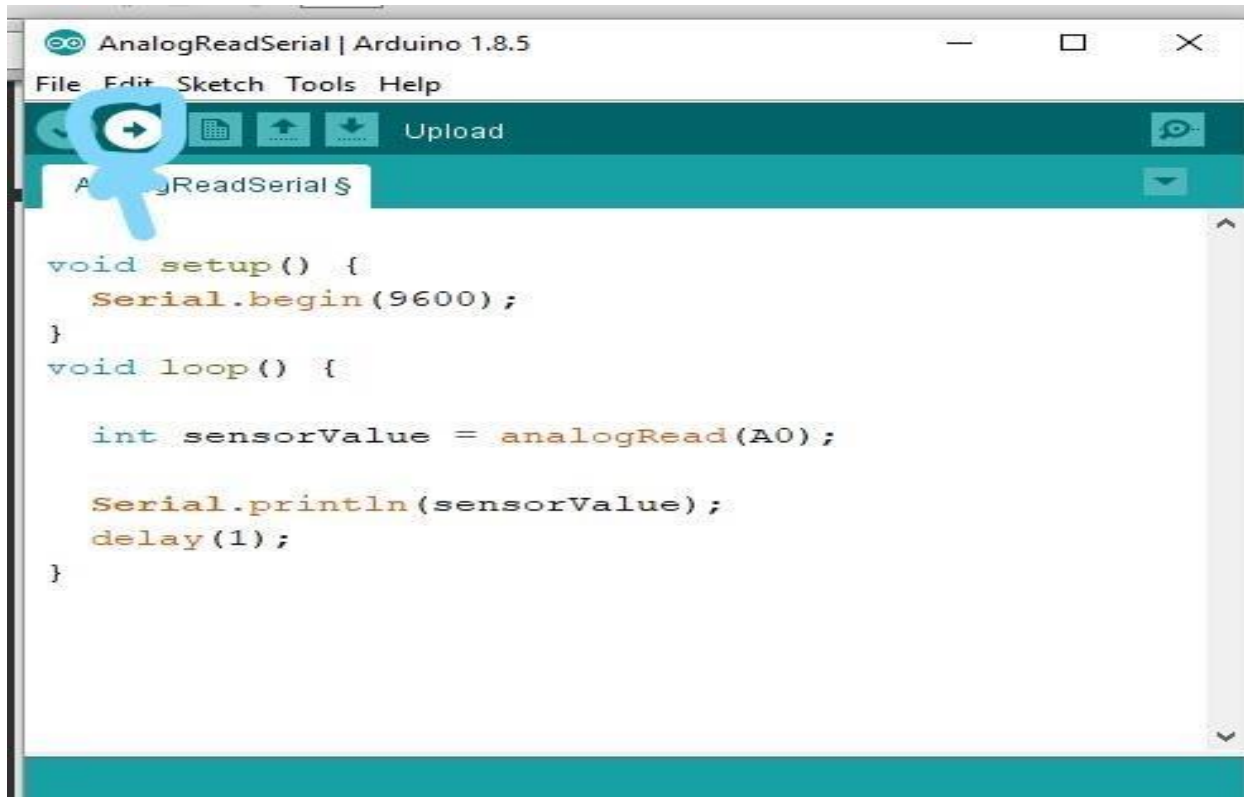
Step 2. Writing the code in the preferred language. (Fig-5.2 Arduino IDE setup)



Step 3. Selecting the port from the list of ports.



Step 4. Uploading the code over to the Arduino UNO microcontroller.



### 5.3 UBIDOTS:

Ubidots is an IoT Platform empowering innovators and industries to prototype and scale IoT projects to production. Use the Ubidots platform to send data to the cloud from any Internet-enabled device. You can then configure actions and alerts based on your real-time data and unlock the value of your data through visual tools. Ubidots offers a REST API that allows you to read and write data to the resources available: data sources, variables, values, events and insights. The API supports both HTTP and HTTPS and an API Key is required. Your data will be protected with two more replication, encrypted storage and optional TLS/SSL data support. You can also customize permission groups to each module of the platform, making sure the right information is shown to the right user. Since Ubidots pivot in 2014, Ubidots has become known within hardware, software, embedded engineering, and maker circles as the affordable, reliable, and most usable platform in the IoT Application Enablement ecosystem. In line with Ubidots

guiding principles, in 2018 Ubidots spun out and repurposed its first ever cloud version as Ubidots STEM. Rather than retiring this shared-resources platform, Ubidots instead empowers 60,000+ students, makers and researchers to explore and develop technology and solutions, giving all users their first 3 devices free, always!

### 5.3.1 UBIDOT BASICS: Devices, Variables, Dashboards, and Alerts

The basics components of any Internet of Things application powered by Ubidots are: **Devices**, **Variables**, **Synthetic Variables Engine**, **Dashboards**, and **Events**. Within this article we will address each of these concepts as they relate to Ubidots IoT Development and Deployment Platform and how you can better organize your Ubidots Apps to best connect with the users. Once your devices, variables, and general Application is assembled, give your App some layers with Ubidots Device Management to learn more about Ubidots internal architecture and how you can use Apps, Organizations, and Users to efficiently connect your data with those who should be using it.

1. Devices
2. Variables
3. Synthetic Variables Engine
4. Dashboard and Visualizations
5. Events.

#### 1. Devices

A Ubidots' device is a virtual representation of a data-source or simply, an asset taking sensor data and transmitting said data through a connection protocol to Ubidots' cloud. [Click here](#) for current firmware examples and tutorials for connecting your device to Ubidots. All devices are different, but the standard setup on any device entails:

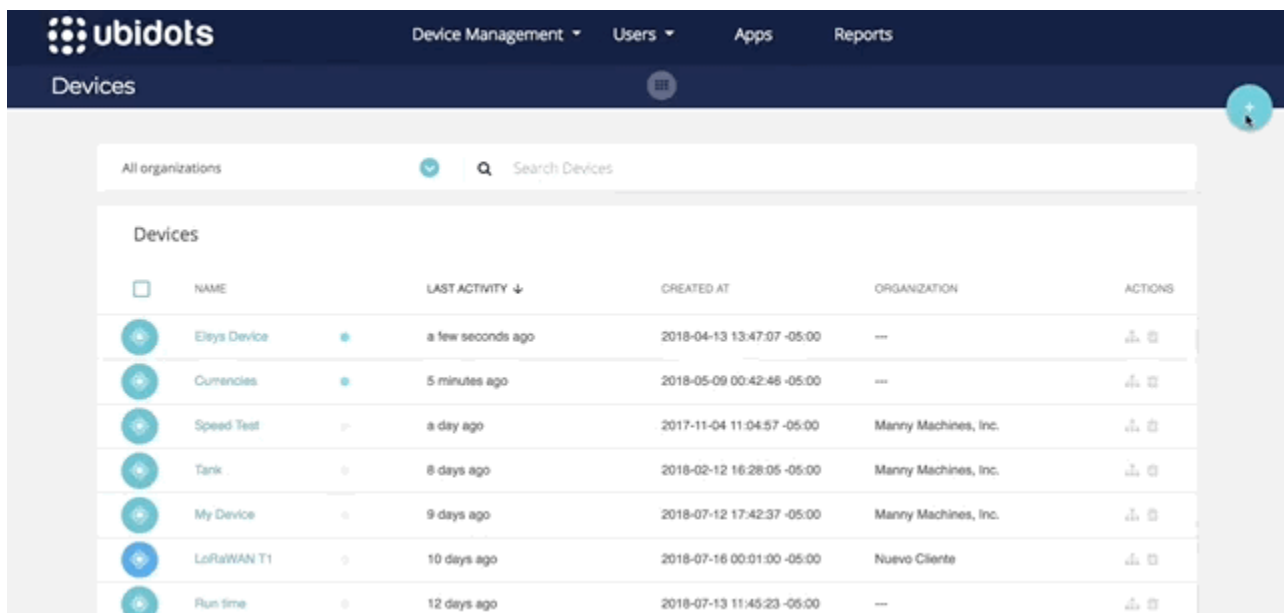
- A library that should be installed in the device's IDE. (not required)

- Filling parameters for authentication and connection such as a TOKEN (i.e. the unique ID for each account or user), device and variable labels (i.e. devices and variables unique identifiers in Ubidots), Wi-Fi SSID and Password, depending on the device and the requirements.
- Making API request (i.e. a call from a device to the web server). Ubidots' libraries spare the need to manually make these requests. However, if you're working with a device not currently in the list of supported hardware, you can choose a connectivity protocol such as MQTT or HTTP and make a request using the API accordingly.

## Creating devices:

There are three ways to create devices:

1. Devices are created automatically in Ubidots the first time a dot is received to a user's private TOKEN or an Organizational TOKEN.
2. Clicking the "+" icon in the top right corner of any single device's screen.
3. Creating a Device Type which then creates a new device with pre-determined device properties, appearance, and variables and automate the on-boarding of thousands of devices. **(vid -5.3.1 Creating devices in UBIDOTS)**



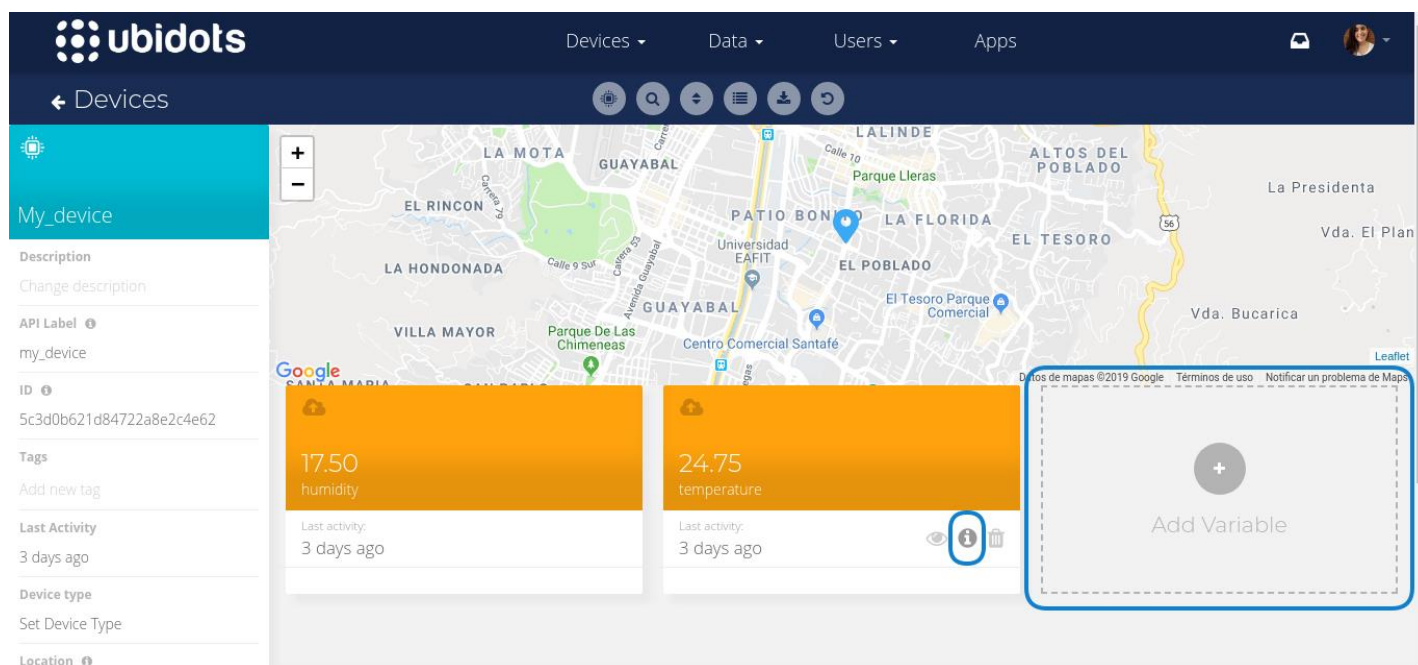
| NAME         | LAST ACTIVITY     | CREATED AT                 | ORGANIZATION         | ACTIONS |
|--------------|-------------------|----------------------------|----------------------|---------|
| Elsys Device | a few seconds ago | 2018-04-13 13:47:07 -05:00 | ---                  | [Icons] |
| Currencies   | 5 minutes ago     | 2018-05-09 00:42:46 -05:00 | ---                  | [Icons] |
| Speed Test   | a day ago         | 2017-11-04 11:04:57 -05:00 | Manny Machines, Inc. | [Icons] |
| Tank         | 8 days ago        | 2018-02-12 16:28:05 -05:00 | Manny Machines, Inc. | [Icons] |
| My Device    | 9 days ago        | 2018-07-12 17:42:37 -05:00 | Manny Machines, Inc. | [Icons] |
| LoRaWAN T1   | 10 days ago       | 2018-07-16 00:01:00 -05:00 | Nuevo Cliente        | [Icons] |
| Run time     | 12 days ago       | 2018-07-13 11:45:23 -05:00 | ---                  | [Icons] |

## 2. Variables

Once a device is created and receiving data from your hardware or another 3rd party data-source, the data will be presented in its raw or calculated form as a **variable**.

**Creating variables:** there are two ways to create variables.

1. Assigning a label as a variable within your hardware's code.
2. Clicking the "+" icon found in any single device screen and assign a name which will also correspond to the variable's label. By default, Ubidots assigns an additional unique identifier to the variable called Variable ID. To find both the variable's ID and variable's label, select the "i" icon from the variables card. **(Fig- 5.3.2 -Creating variables in Ubidots)**



## 3. Synthetic Variables Engine

Ubidots Synthetic Variables (SV) Engine is a simple and effective way to create insights from the raw data stored in Ubidots' Cloud. Using Ubidots SV Engine, users can create **Synthetic Variables** that calculate and enhance data to give data-driven decision to user of an Ubidots powered application.

For example, to calculate Average (mean) Energy consumed this week, we will need to create a synthetic variable called Energy to calculate the product of Current and Voltage ( $P=C*V$ ): the **default variable Current is multiplied by Voltage**. Then average the product every week with built-in function *mean()*.

## Synthetic Variables Analytics

Energy

Synthetic Variable

Enter your synthetic variable expression. [Learn more about synthetic variables.](#)

1 mean( **current** \* **voltage** , "W" )

Timezone:

America/Bogota

Click on a variable to add it to your expression:

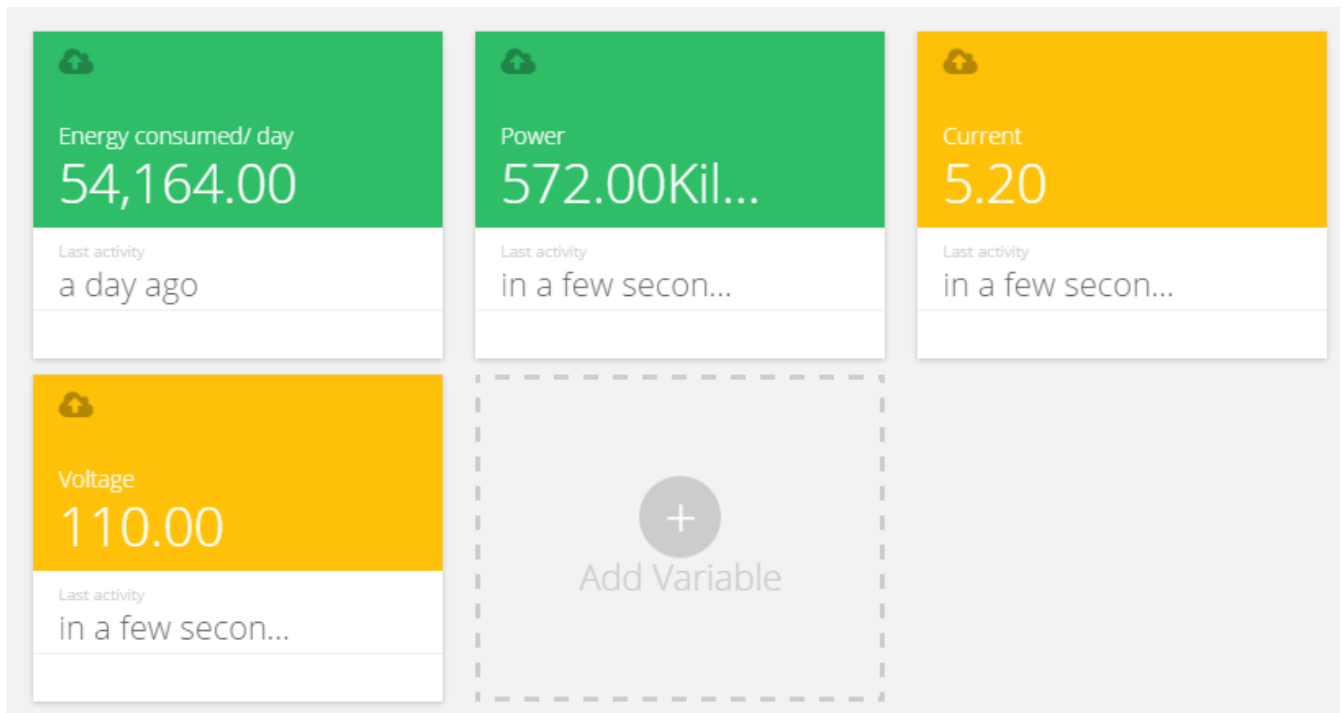
Current

Voltage

Cancel

Accept

Here is a complete list of available functions, such as **cos(x)** or **ceil(x)** functions, and more!



Note that default variables are yellow, and Synthetic Variables are green.

#### 4. Dashboards and Visualizations

Dashboards are the human-machine interfaces where data is easily visualized. Your Ubidots account will let you create as many dashboards as needed, containing widgets and data-visualizations to comprehend your data at a glance. For additional details on your dashboard's construction, check out this tutorial.

#### 5. Events

In Ubidots, Events are messages triggered and delivered through Email, SMS, Telegrams, Slack, Voice Call or webhook messages based on a customized design rule created in the application. If you wish to create alerts for your IoT application, check out this tutorial for additional support.

3 Event types:

- When sensor readings reach a specific value:

If triggers

then actions

demo

+

Value

▼

is

Equal to

Equal to

Greater than

Greater than or equal to

Less than

Less than or equal to

Not equal to

10

for

0

minutes

🗑️

- Variable is inactive during a specific amount of time:

If triggers

then actions

demo

+

Has been inactive

▼

for

10

minutes

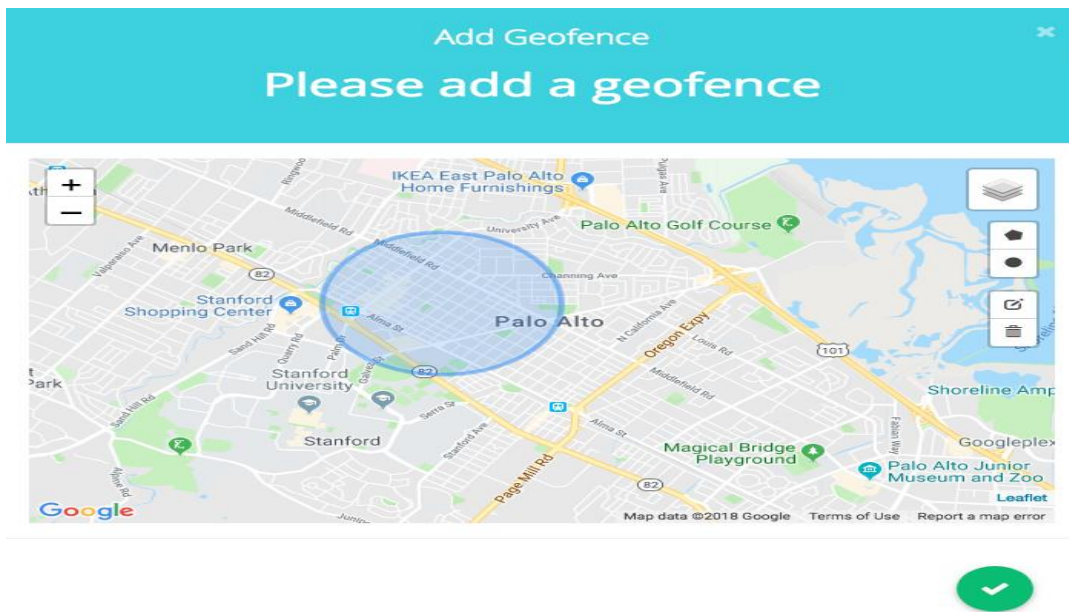
▼

🗑️

+ And

+ Or

- When a device enters or exits a specific area (geofence):



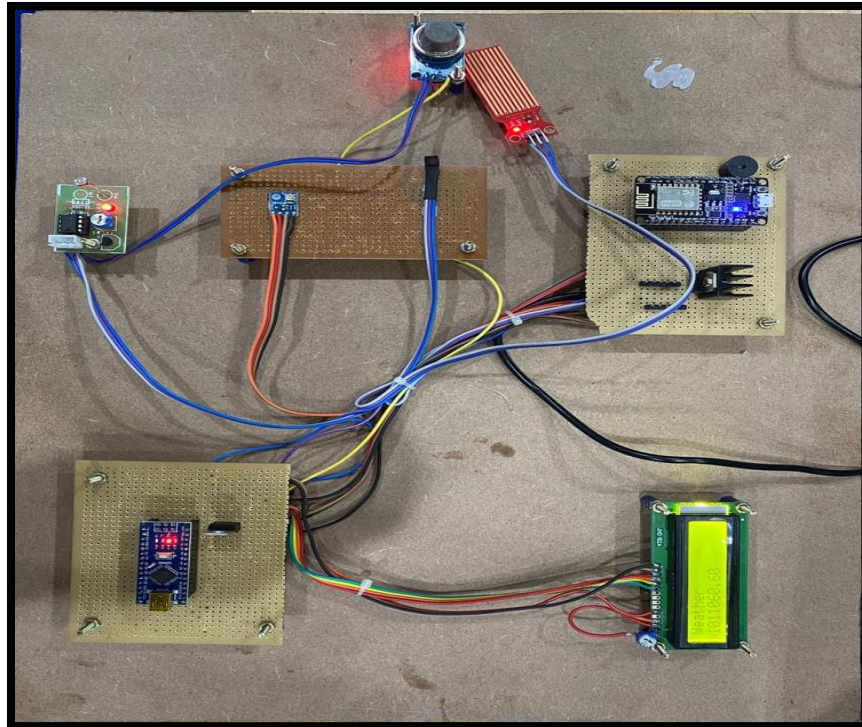
You can also determine the format of alerts by choosing between emails, SMS, Telegrams or Webhook

## **CHAPTER 6**

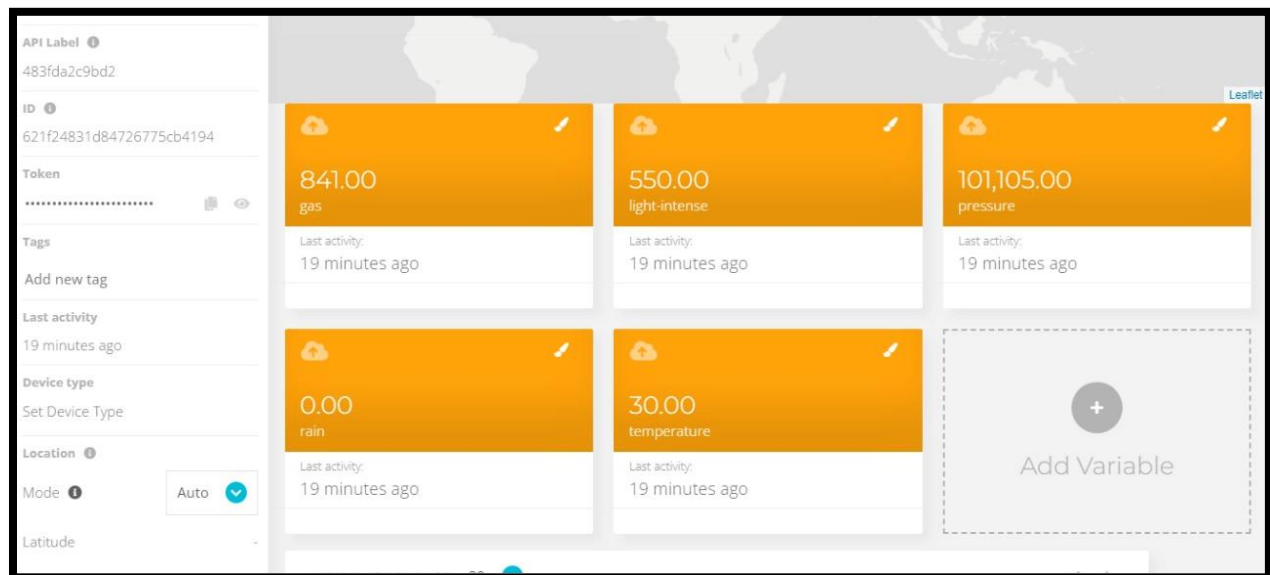
### **RESULT AND DISCUSSIONS**

To implement this need to deploy the sensor devices in the environment for collecting the data and analysis. By deploying sensor devices in the environment, it will record real time data. It can cooperate with other objects through the network. Then the collected data and analysis results will be available to the end user through the Wi-Fi. The smart way to monitor environment and an efficient, low-cost entrenched system is presented with different models in this paper. Weather station is a form of application of science and technology to know and predict the weather condition at a particular body. The weather station is designed to collect quantitative data about the weather condition of a region in order to be able to know and predict weather conditions in a region. Human activities depend on whether factors.

Weather is an air condition that lasts for a short period of time so the short-term weather forecasting process should be done as quickly and accurately as possible. In the proposed architecture purposes of different modules were discussed [9]. The noise and air pollution monitoring system with Internet of Things (IoT) concept experimentally tested for monitoring two parameters. It also sent the sensor parameters to the cloud (Google Spread Sheets). This data will be cooperative for future analysis and it can be easily shared to other end users. This model can be further expanded to monitor the developing cities and manufacturing zones for pollution monitoring. To protect the public health from pollution, this model provides an efficient and low-cost solution for unceasing monitoring of environment.



**Fig 6.1 PROJECT PICTURE**



**Fig 6.2 OUTPUT PICTURE**

## **CHAPTER 7**

### **CONCLUSION**

#### **7.1 CONCLUSION:**

- This project demonstrates Design and Implementation of Weather Monitoring & Controlling System used for controlling the devices as well as monitoring the environmental parameters.
- Embedded controlled sensor networks have proven themselves to be a reliable solution in providing remote control and sensing for environmental monitoring systems.
- The sensors have been integrated with the system to monitor and compute the level of existence of pressure, gas, temperature and rain and light intensity in atmosphere using information and communication technologies.
- The sensors can upload the data in cloud and can be monitored and analysed using IOT implementation in Ubidots.

#### **7.2 FUTURE SCOPE:**

Powering the device using solar panels. Suspending the device from a weather balloon so that it can be used to record atmospheric parameters at high altitudes and remote and inaccessible areas. Use of a tough exterior cover for the system that will act as a protective cover enabling the device to function in harsh weather conditions. Designing a method to mount the weather monitoring device onto a buoyant platform like a buoy hence enabling the system to measure weather changes over the sea. This data can also be shared to Cargo ships and other nautical industries conducting operations within the area. Using silica gel to prevent condensation on the exterior cover as condensation might affect the sensors readings.

## REFERENCE PAPERS:

- [1] Kalyani G. Gajbhiye, Snehlata S. Dongre, A Survey on Weather Monitoring System in Agriculture Zone using Zigbee, International Journal of Science and Research (IJSR), India Online ISSN: 2319-7064.
- [2] Karthik Krishnamurthi, Suraj Thapa, Lokesh Kothari, Arun Prakash, Arduino based Weather Monitoring System International Journal of Engineering Research and General Science Vol3, ISSN 2091-2730.
- [3] Rong-Hua Ma, Yu-Hsiang Wang and Chia-Yen Lee, Wireless Remote Monitoring System Based on MEMS Technology, ISSN: 1424-8220.
- [4] M. Sowjanya, MD Ameenuddin, A Reconfigurable Smart Sensor Interface for Weather Monitoring with IOT Environment, IJESRT, ISSN:2277-9655.
- [5] Adnan Shaout, Yulong Li, Mohan Zhou and Selim Awad, Low cost Embedded Weather Station with Intelligent System, IEEE
- [6] Yash Mittal, Anshika Mittal, Dikhsa Bhateja, Komal Parmaar, V.K. Mittal, Correlation among environmental parameters using an online smart Weather station system , IEEE INDICON 2015.