

# ATMOSPHERIC MONITORING USING ARDUINO BOARDS AND LOW COST SENSORS

**Amit Kumar Bhargava**

*Department of Electrical & Electronics Engineering, (India)*

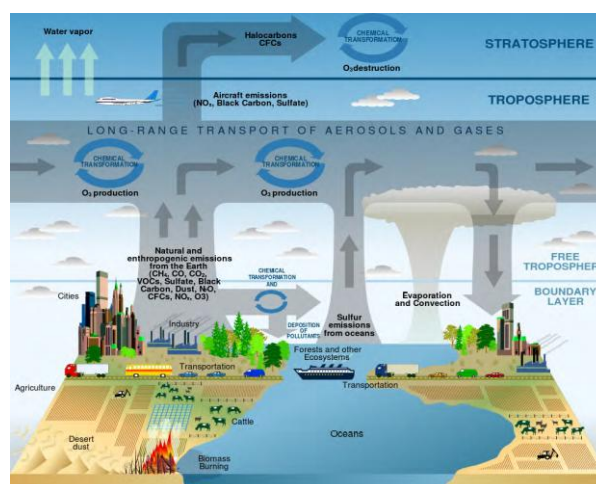
## ABSTRACT

*In the old days, when people wanted to know what was in the atmosphere, they used chemically-treated filter paper, and hung it in a breeze. The chemicals reacted with whatever was in the air and would respond by changing color or they bubbled the atmosphere through water and measured the different compounds that resulted as gas dissolved in water. This kind of work could only be performed in a dedicated chemistry lab. This paper is aimed to explore possibility of using low cost sensors available in the market along with open source hardware and software for automatic meteorological observations.*

**Keywords:** Atmosphere, Arduino Board, Photometer, Sensor, Liquid Crystal Display

## I INTRODUCTION

1.1 Weather observation in the field of meteorology is getting modernized from manual observations using conventional equipments to modern sensors based digital automatic observation equipments. At present variety of sensors, microcontrollers or open source hardware and software's are available in the market at very competitive prices and can be explored to make cost effective automatic weather stations.



**Fig.1:** Atmosphere

1.2 In the old days, when people wanted to know what was in the atmosphere, they used chemically-treated filter paper, and hung it in a breeze. The chemicals reacted with whatever was in the air and would respond by

changing colour. Or they bubbled the atmosphere through water and measured the different compounds that resulted as gas dissolved in water. This kind of work could only be performed in a dedicated chemistry lab. Fortunately for us, we can now purchase a small, complete atmospheric laboratory for less than \$10, in the form of an electronic sensor. These sensors detect different substances in the atmosphere by measuring the changing resistance of a film made of oxides.

## II SENSORS

A **sensor** is a transducer whose purpose is to sense (that is, to detect) some characteristic of its environs. It detects events or changes in quantities and provides a corresponding output, generally as an electrical or optical signal,

A good sensor obeys the following rules:

- (a) Is sensitive to the measured property only.
- (b) Is insensitive to any other property likely to be encountered in its application.
- (c) Does not influence the measured property.

### 2.1 How Sensors Work?

2.1.1 Oxygen in the atmosphere removes electrons from the tin dioxide film, which decreases its conductivity (and increases its resistance). When other types of gases, particularly those that are chemically reducing, touch the tin dioxide film, electrons are injected into the material. This *increases* the conductivity (and *lowers* the resistance) of the tin dioxide layer. You can use your Arduino to measure that change in resistance.

2.1.2 It's important to keep in mind that tin dioxide sensors tend to be broadly selective. While certain sensors may be marketed as being "alcohol" sensors or "carbon monoxide" sensors, they actually respond to more than just alcohol or carbon monoxide, respectively; they respond to a wide family of similar gases. Manufacturers can make the tin dioxide-based gas sensors more selective by adding various catalysts into the sensor head, or by using external filters. The datasheet provided with each sensor explains more completely how to adjust the sensitivity of each sensor for various gases.

2.1.3 In case of atmospheric elements like pressure, temperature, humidity the sensors utilize the property of electrical conductivity. There are metals whose resistivity changes when subjected to change in temperature or humidity and these metals play an important role in designing meteorological sensors.

2.1.4 In case of pressure sensors linear variable differential transformer technology (LVDT) is utilized for measuring accurate atmospheric pressure. Here a pressure responsive element is directly coupled to the core of a linear LVDT. An LVDT is an electro-mechanical device that produces an electrical output that is linearly proportional to the displacement of a linear core. It consists of a primary coil with two secondary coils placed on either side of the primary coil. When the primary coil is energized by an alternating current source, voltages are induced in the two secondary coils. A movement of the core leads to changes in magnetic coupling and

producing net output signal from connected secondary's. To form pressure transducers, the core displacement of LVDT is produced by the movement of the metallic pressure responsive diaphragm.

### **III ELEMENTS WE CAN MONITOR?**

3.1 There are electronic sensors for a wide range of atmospheric elements and gases. There are easy-to-use inexpensive sensors on the market to detect pressure, temperature, humidity, rainfall, carbon monoxide, carbon dioxide, liquid petroleum gas, butane, propane, methane (natural gas), hydrogen, ethyl alcohol, benzene, volatile organic compounds, ammonia, ozone, hydrogen sulphide, and more.

3.2 It's not unreasonable to expect that it won't be long before cheap sensors hit the market that can detect nitrogen oxides and other contaminants. All of these gases count as pollutants; in varying concentrations, all of them can be harmful.

### **IV ARDUINO**

#### **4.1 History**

Arduino started in 2005 as a project for students at the Interaction Design Institute Ivrea in Ivrea, Italy. At that time program students used a "BASIC Stamp" at a cost of \$100, considered expensive for students. Massimo Banzi, one of the founders, taught at Ivrea.<sup>[4]</sup> The name "Arduino" comes from a bar in Ivrea, where some of the founders of the project used to meet. The bar, in turn, has been named after Arduin of Ivrea, who was the margrave of Ivrea and King of Italy from 1002 to 1014.<sup>[5]</sup>

Colombian student Hernando Barragán created the Wiring development platform which served as the basis for Arduino. Following the completion of the Wiring platform, its lighter, less expensive versions were created and made available to the open-source community; associated researchers, including David Cuartielles, promoted the idea. The Arduino's initial core team consisted of Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, and David Mellis

#### **4.2 Hardware**

An Arduino board historically consists of an Atmel 8-, 16- or 32-bit AVR microcontroller (although since 2015 other makers' microcontrollers have been used) with complementary components that facilitate programming and incorporation into other circuits. An important aspect of the Arduino is its standard connectors, which lets users connect the CPU board to a variety of interchangeable add-on modules known as *shields*. Some shields communicate with the Arduino board directly over various pins, but many shields are individually addressable via an I<sup>2</sup>C serial bus—so many shields can be stacked and used in parallel. Prior to 2015 Official Arduinos had used the Atmel megaAVR series of chips, specifically the ATmega8, ATmega168, ATmega 328, ATmega 1280, and ATmega 2560 and in 2015 units by other manufacturers were added. At a conceptual level, when using the Arduino integrated development environment, all boards are programmed over a serial connection. Its implementation varies with the hardware version.

### 4.3 Software

Arduino programs may be written in any programming language with a compiler that produces binary machine code. Atmel provides a development environment for their microcontrollers, AVR Studio and the newer Atmel Studio. The Arduino project provides the Arduino integrated development environment (IDE), which is a cross-platform application written in Java. It originated from the IDE for the Processing programming language project and the Wiring project. It is designed to introduce programming to artists and other newcomers unfamiliar with software development. It includes a code editor with features such as syntax highlighting, brace matching, and automatic indentation, and provides simple one-click mechanism for compiling and loading programs to an Arduino board. A program written with the IDE for Arduino is called a "sketch".

## V THE BENEFITS OF ARDUINO INCLUDE

5.1 A single-board computer that is deliberately designed to be used by people who are not experts in electronics, engineering, or programming.

5.2 It is inexpensive, cross-platform (the Arduino software runs on Windows, Mac OS X, and Linux), and easy to program.

5.3 Both Arduino hardware and software are open source and extensible.

5.4 Arduino is also powerful: despite its compact size, it has about as much computing muscle as one of the original navigation computers from the Apollo program, at about 1/35,000 the price .

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
Length	68.6 mm
Width	53.4 mm
Weight	25 g

Fig.2: Microcontroller Features

## VI ARDUINO BOARDS

### 6.1 Arduino Diecimilla

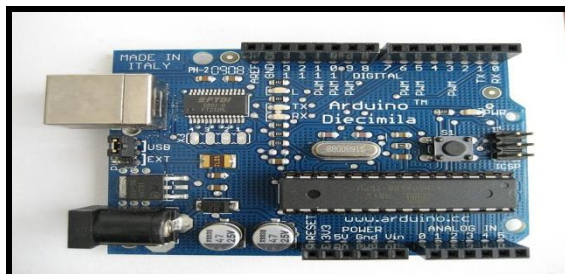


Fig.3

### 6.2 Arduino Duemilanove

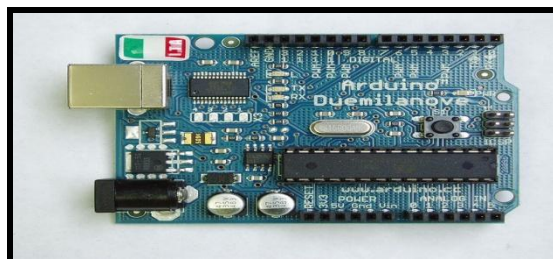


Fig.4

### 6.3 Arduino Uno

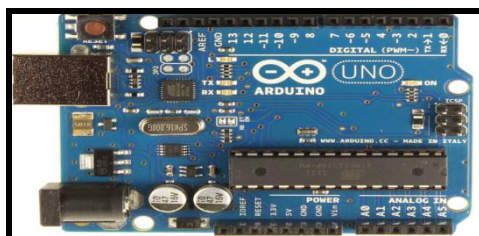


Fig.5

### 6.4 Arduino Mega

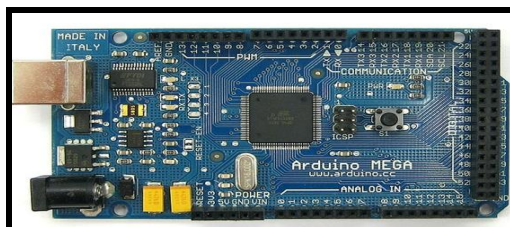


Fig.6

### 6.5 Arduino Leonardo

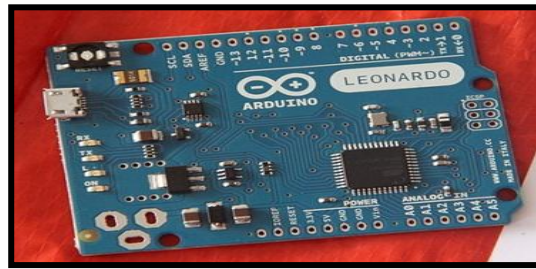


Fig.7

### 6.6 Arduino Nano



Fig.8

### 6.7 Arduino Lilypad

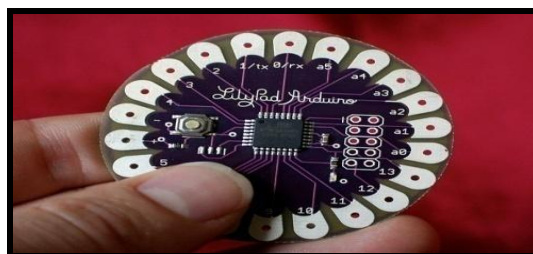


Fig.9

## VII FEATURES OF ARDUINO

7.1 An open source design. The advantage of it being open source is that it has a large community of people using and troubleshooting it. This makes it easy to find someone to help you debug your projects.

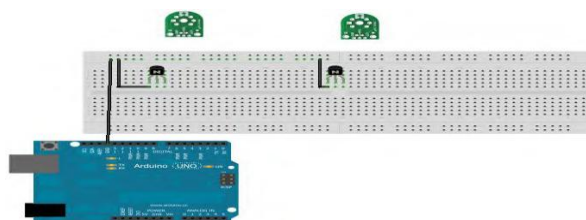


Fig.10

7.2 An easy USB interface. The chip on the board plugs straight into your USB port and registers on your computer as a virtual serial port. This allows you to interface with it as through it were a serial device. The benefit of this setup is that serial communication is an extremely easy (and time-tested) protocol, and USB makes connecting it to modern computers really convenient.

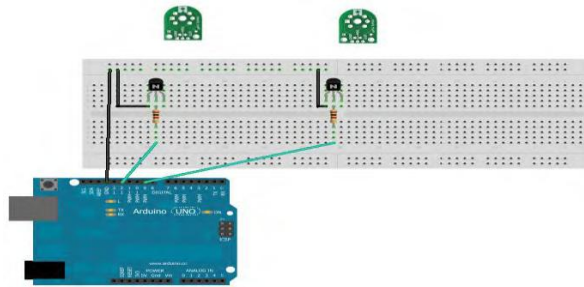


Fig.11

Very convenient power management and built-in voltage regulation. You can connect an external power source of up to 12v and it will regulate it to both 5v and 3.3v. It also can be powered directly off of a USB port without any external power.

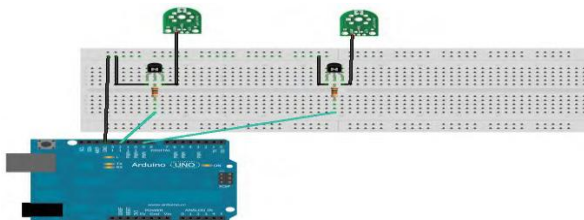


Fig.12

7.3 An easy-to-find, and dirt cheap, microcontroller "brain." The ATmega328 chip retails for about \$2.88 on Digikey. It has countless number of nice hardware features like timers, PWM pins, external and internal interrupts, and multiple sleep modes.

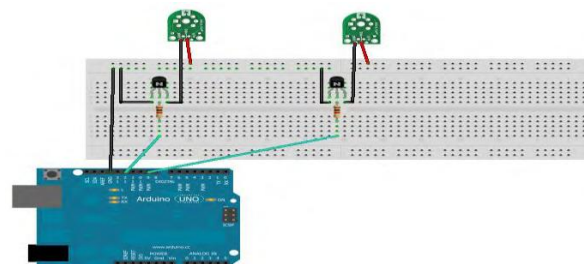


Fig.13

7.4 A 16 mhz clock. This makes it not the speediest microcontroller around, but fast enough for most applications. 32 KB of flash memory for storing your code. 13 digital pins and 6 analog pins. These pins allow you to connect external hardware to your Arduino. These pins are key for extending the computing capability of the Arduino into the real world. Simply plug your devices and sensors into the sockets that correspond to each of these pins and you are good to go.

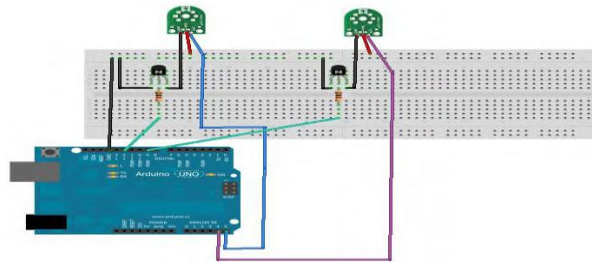


Fig.14

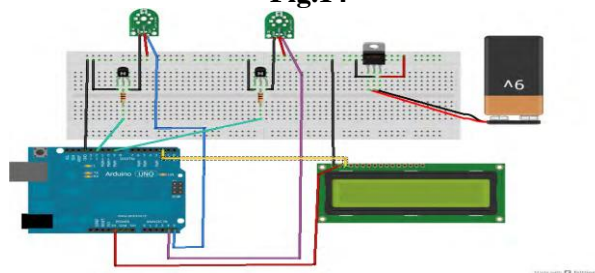


Fig.15

7.5 An ICSP connector for bypassing the USB port and interfacing the Arduino directly as a serial device. This port is necessary to boot load your chip if it corrupts and can no longer talk to your computer. to say that the lightweight, cheap electronic gadgets in our lives—handheld cell phones, computers, digital cameras, flat screen TVs, microwave ovens, cable television, touchtone phones, simple portable AM/FM radios, essentially anything more complicated than a flashlight—would be impossible without the transistor.

**VIII** The first thing you notice when you look at a transistor is that unlike almost every other electronic device we've seen so far, a transistor has three terminals. We can control the voltage between two of the terminals by applying a specific electric current or voltage to the third terminal. The three terminals are the base, the collector, and the emitter. The base is the controller; voltage applied here determines whether or not electricity flows from the collector to the emitter. The collector is the “source” of the electrical current, and the emitter is the output. If we were to send varying levels of current from the base, we can regulate the amount of current flowing from the collector to the emitter. This is how a transistor acts as an amplifier: a very low signal coming into the base is repeated at a much larger voltage provided by the collector. When we use a transistor as a switch, the circuitry is even simpler. A transistor switch is either fully on or fully off. A small data signal to the base determines whether the transistor is switched on or off. When it is switched on, current flows between the



ground and the collector. This simple setup lets us use Arduino to turn on components that have a separate power supply.

Build the Gadget		
Amount	Part Type	Properties/(Assembly Code)
2	1 k $\Omega$ resistor	Package THT; tolerance 5%; bands 4; resistance 1 k $\Omega$ ; pin spacing 400 mil (R1 & R2)
1	Voltage regulator, 5 V	Package TO220 [THT]; voltage 5 V (U1)
2	NPN-transistor	Package TO92 [THT]; type NPN (Q1 & Q2)
1	Arduino UNO R3	(Arduino1)
1	LCD screen	Character type, 16 pins (LCD1)
1	Battery block 9 V	(VCC1)

## IX DISPLAYING AND STORING YOUR DATA

You can connect Arduino to components that display data, as well as those that store data for later use.

### X LIQUID CRYSTAL DISPLAYS

10.1. Liquid crystal displays (LCDs) are cheap and easy ways to display data, status, warnings, and other messages from Arduino. They come in many different colours: you can buy LCDs with amber characters on a black back-ground, black characters on a green background, yellow characters on a blue background, and other colour combinations. Some LCDs have two rows of 16 characters, others four rows of 20 characters, and other display combinations are available as well. But for our uses, the biggest differences in LCDs involve the way they handle data. The most basic (and least expensive) LCDs make you do all the data handling. They can take up as many as 10 digital data pins (most Arduino only have 13), and might even require you to design your own characters. Some makers love doing stuff like that, but others just want to plug in a device and have it work.

10.2 For our uses, we've decided to go with a serial-controlled LCD, one in which a small microprocessor attached to the LCD takes care of all the data and character management. It's more expensive, but also much easier to use. All we need to do is ground the device, give it some power, and feed it data. Step seven of the build explains how to connect the LCD to the tropospheric gas detector.

### XI LED PHOTOMETER

A photometer is a device that measures one or more qualities of light. Most photometers consist of an interference filter and a photo detector. The interference filter is a coloured piece of plastic or glass that filters out nonessential colors, letting through only the precise wavelengths of light we're interested in studying. A photo detector behind that filter monitors the intensity of the light that makes it through the filter.

A couple of things to keep in mind: -

11.1 Although LEDs can work as detectors, they were not built for this purpose. Whereas an ideal detector for our needs would detect only a very narrow range of wavelengths, and be perfectly centered around the wavelengths we wish to study, LEDs can detect light over a fairly wide range of wavelengths. That's why we've tested the LEDs: to know their detection peaks, so that we know which wavelengths of light we're measuring.

11.2 Also, the current generated by an LED can change with the temperature of the LED; for this reason, we advise you to keep the gadget at room temperature, and make all outside measurements relatively quickly, before the gadget heats up (or cools down) too much.

**XII** Even allowing for these shortcomings, however, it is perfectly possible to get scientifically valid data by using the LED detectors to measure the upper atmosphere.

### **XIII BUILD THE GADGET**

Amount	Part Type	Properties
1	Piezo Speaker	
1	Blue LED, 5 mm	Package 5 mm [THT]; leg yes; color Blue
1	Green LED, 5 mm	Package 5 mm [THT]; leg yes; color Green
1	Red LED, 5 mm	Package 5 mm [THT]; leg yes; color Red
1	Arduino	Processor ATmega; variant Arduino
1	LCD screen	Type character; pins 16

There are two sequences of steps that are equally essential to creating the LED photometer: building it, and calibrating it.

13.1 Connect the LEDs to Arduino. We've found it's easiest to do this as follows:

13.1.1 Connect the negative (i.e., shorter wire) of each LED to the GND rail of the breadboard.

13.1.2 Connect the positive (i.e., longer wire) of each LED to the breadboard.

13.1.3 Using jumpers, connect each LED to the appropriate Analog pin on Arduino our Arduino sketch expects the red LED to be plugged into A1, the green LED into A2, and the blue LED into A3.

13.2 Connect the speaker. Plug the positive lead into Arduino digital pin 7 and the negative pin into Arduino GND. Many speakers don't list the polarity of their connection, so don't worry too much about this; just plug it in!

13.3 Connect the LCD unit to Arduino. The data line connects to Arduino digital pin 5, the power line connects to Arduino 3.3 v pin, and GND line connects to GND.

### **XIV USING LED PHOTOMETER**

14.1 One of the oldest bits of weather lore is "Red sky at night, sailor's delight. Red sky in morning, sailors take warning." The sky is often red, or reddish orange, at sunrise and sunset. Reds, pinks, and oranges dominate at these times of day because the distance from your eye to the sun is greater at these times than when the sun is

directly overhead. When sunlight travels through the atmosphere—and especially when it travels that extra distance from the horizon—an effect known as *Rayleigh scattering* takes place.

14.2 As atmospheric particles get larger, they tend to scatter all wavelengths of light equally and indiscriminately. This explains why clouds, which are made of relatively large water droplets, are white. Of course, molecules don't just scatter wavelengths of light. Certain molecules, such as water vapor, carbon dioxide, and ozone, also absorb specific wavelengths of light. To a measuring instrument on the ground, there is less light at those specific wavelengths than there is at other nearby wavelengths, because some of that light has been absorbed. By measuring these "holes" where wavelengths of light are absent, you can get a very good estimate of how much water vapor, ozone, and some other substances are in the atmosphere above your head.

## XV CONCLUSION

Efforts have been made to explore the possibility of utilizing various low cost sensors available in the market for meteorological purpose. If prove successful these low cost sensors making a very economic and effective network of automatic observatories across area of interest. Selectivity of gas sensors, i.e its ability to detect specific gas present in the atmosphere, For that we have to choose either of following approach: -

- 15.1 Select application specific sensor.
- 15.2 Multiple sensors for particular meteorological element could provide good opportunity for selecting the best and most sensitive sensor
- 15.3 Calibration of sensor
- 15.4 Sensitivity of whole device has to be high, so that it can detect very small changes as well.
- 15.5 Power backup for whole device, which has to be used as a standalone system in application.

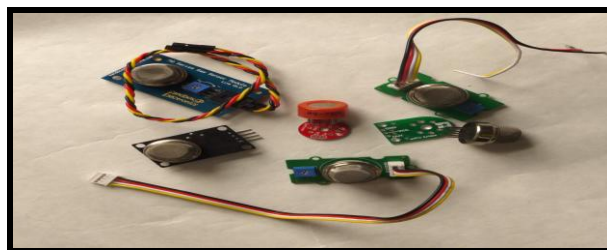


Fig.16- Types of Sensors

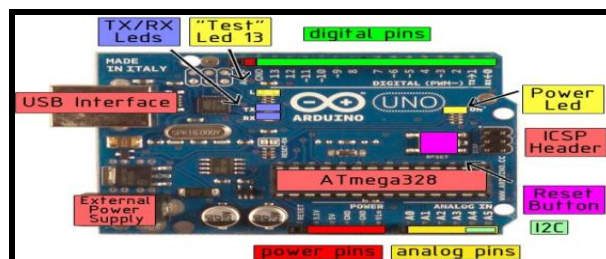


Fig.17- Arduino-Uno ports and specifications

#### REFERENCES

- [1] <https://www.tinkercad.com/>
- [2] <https://circuits.io/>
- [3] <https://www.arduino.cc/>
- [4] <https://circuitdigest.com/microcontroller-projects/iot-air-pollution-monitoring-using-arduino>.

#### Journal Papers:

- [5] Riteeka Nayak, Malaya Ranjan Panigrahy, Vivek Kumar Rai and T Appa Rao: IOT based air pollution monitoring system Vol-3, Issue-4, 2017.
- [6] Navreetinder Kaur, Rita Mahajan and Deepak Bagai: Air Quality Monitoring System based on Arduino Microcontroller Vol. 5, Issue 6, June 2016.
- [7] Palaghat Yaswanth Sai: An IoT Based Automated Noise and Air Pollution Monitoring System Vol. 6, Issue 3, March 2017.

#### Books:

- [8] Exploring Arduino: Tools and Techniques for Engineering Wizardry by Jeremy Blum 1<sup>st</sup> edition.