CALCUTTA INSTITUTE OF ENGINEERING AND MANAGEMENT

24/1A, Chandi Ghosh Road, Tollygunge, Kolkata,
West Bengal, Kolkata-700040

Bachelor Of Technology

PROJECT REPORT

On

“AIR QUALITY MONITORING SYSTEM USING ARDUINO”

Academic Year: 2016-2020

Instrumentations and Control Engineering Department

A Project Report On

"Air Quality Monitoring System Using Arduino"

Submitted to

Maulana Abul Kalam Azad University Of Technology, West Bengal

BF 142, Sector 1, Salt Lake City, Kolkata: 700064

In Partial fulfillment Of the requirements for the award of the degree of

Bachelor of Technology

In

Instrumentation And Control Engineering

Submitted By

RUPAYAN HALDER

(16504016022)

Under the supervision of

Prof. Sonali Sarkhel

Academic Year: 2016-2020

DEPARTMENT OF

INSTRUMENTATION AND CONTROL ENGINEERING

CALCUTTA INSTITUTE OF ENGINEERING AND MANAGEMENT

KOLKATA-700040, INDIA

JULY-2020

DEPARTMENT OF

INSTRUMENTATION AND CONTROL ENGINEERING

CALCUTTA INSTITUTE OF ENGINEERING AND MANAGEMENT

APPROVAL SHEET

This project report entitled “**AIR QUALITY MONITORING SYSTEM USING ARDUINO** ”by Rupayan Halder is hereby forwarded for approval for the degree of Bachelor of Technology in Instrumentation and Control Engineering**.**

Supervisor(s) **Board Of Examiners**

___________________ ______________________**(External)**

Head of the Department

Mrs. Sonali Sorkhel

Principal

Dr. Nilanjan De

Date: 11/07/2020

Place: Birati,Kolkata

DECLARATION

I the undersigned solemnly declare that the project report "**AIR QUALITY MONITORING SYSTEM USING ARDUINO "** is based on my own work carried out during the course of our study under the supervision of Sonali Sarkhel . I assert the statements made and conclusions drawn are an outcome of my research work. I further certify that the work contained in the report is original and has been done by me under the general supervision of my supervisor.The work has not been submitted to any other Institution for any other degree certificate in this university.We have followed the guidelines provided by the university in writing the report. Whenever we have used materials (data, theoretical analysis, and text) from other sources, we have given due credit to them in the text of the report and giving their details in the references.

Rupayan Halder

(NAME OF THE STUDENT)

ROLL NO:16504016022 REGISTRATION NO:161650110399 of 2016-2017

Date: 11 th july ,2020

ACKNOWLEDGEMENT

The phenomenon remains same that no project ever can be executed proficiently and efficiently without sharing the meticulous ideas, technical expertise and innovative thought put forwarded by the technical and non-technical veterans.

It gives us pleasure in presenting this project report, undertaken by us as per our B. Tech 8th Semester Curriculum "Air Quality Monitoring System using Arduino" on having completed this project.

Very first we would like to thank Prof. Mrs. SONALI SORKHEL ECE DEPT. of CIEM. This project would not have been possible without the guidance and assistance provided to us whenever any difficulty came to our way. We are very personally thankful to our internal guide is Mrs. SONALI SORKHEL.

It is our privilege to express our gratitude to Principal of C.I.E.M college, Kolkata for providing us an excellent environment to complete our work successfully.

RUPAYAN HALDER
(16504016022)

INDEX

Abstract

Air pollution affects our day to day activities and quality of life. It poses a threat to the ecosystem and the quality of life on the planet. Air pollution affects our day to day activities and quality of life. It poses a threat to the ecosystem and the quality of life on the planet. In order to monitor quality of air, The system was developed using the Arduino microcontroller. The air pollution monitoring system was designed to monitor and analyze air quality in real-time of a place . This system implements the combination of an Arduino microcontroller, gas sensors , Output monitoring device & alarming device to sense the air quality of the environment and shows the real condition of air. The system will give the user the indication of the air quality and based on given parameters it will let the user know how much the environmental air is polluted or safe. The result was displayed on the designed hardware's display interface.

CHAPTER-1

Introduction of Project

Project Title: Air Quality Monitoring System Using Arduino

(i) Introduction: Air is one of the essential elements of man's surroundings. The earth's atmosphere is full of air which contains gases such as Nitrogen, Oxygen, Carbon Monoxide and traces of some rare elements. Humans need an atmosphere of air that is free from contaminants. This is very crucial for human life and health. Any change in the natural composition of air may cause grave harm to life forms on earth. Air pollution is the presence of one or more contaminants in the atmosphere such as gases in a quantity that can harm humans, animals and plant. Primary pollutants are released directly into the atmosphere. Secondary pollutants are produced when the primary pollutant reacts with other atmospheric chemicals.

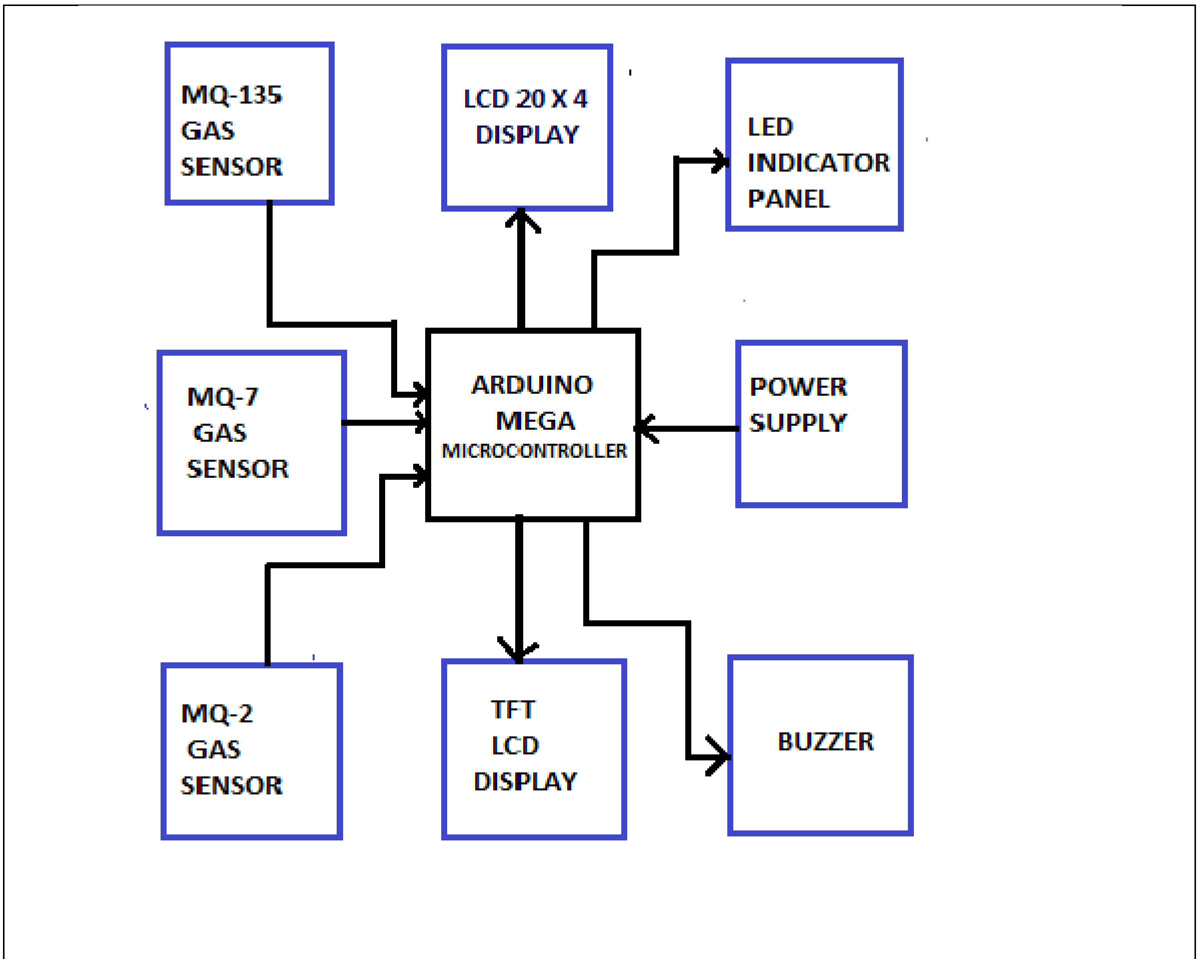
Air is the most useful thing for each and every living thing. Researching on this serious issue this system's main purpose was to estimate the quality of air for people and any other living thing which exist on earth. Very important to know for our living is that how much safe we are now and how the weather and climate has changed for air pollution and it will sustain sound. This system will ease to know the answers for air quality.

Air pollution sensors are devices that can detect the presence of air pollution in the surroundings. The detection is in terms of a ppm value of the pollutants the sensor can detect, and then those ppm values can be used to monitor the environment, based on standard safety values. There are different types of sensors, depending on the usage. The major ones are specialized for one or more of the following components: Ozone, PM, Carbon Monoxide, Sulphur Dioxide and Nitrous Oxide. We have used MQ135 , MQ7, MQ2 Gas Sensors, which are sensitive to gases like Ammonia, Benzene,Carbon Dioxide, Methane, Smoke, LPG, Butane and Carbon Mono Oxide.

(ii)PROJECT OVERVIEW: Whenever we place our system in any place, the system will show the quality of the air in that place, The gas Sensors will detect the number of contaminants like Ammonia, Benzene, Carbon Dioxide, Methane, Smoke, LPG, Butane and Carbon Mono Oxide, then the sensors will send the signal to the Arduino Microcontroller, From Arduino microcontroller, we will send the output values to the Display devices, We will also track the data graphically also. If the Air Quality is very poor in any place and if the quality of air decreases in any place, the buzzer will alert the people to leave that place as soon as possible.

Chapter-2

Block Diagram of The Air Quality Monitoring System using Arduino



Pic1: Block Diagram of Air quality monitoring System

Our project's block diagram contains 9 blocks which are

1. Arduino Mega Microcontroller
2. MQ135 gas Sensor
3. MQ7 gas Sensor
4. MQ2 gas Sensor
5. LCD (20 X 4) display
6. TFT LCD display
7. Buzzer
8. LED indicator panel
9. Power Supply

Chapter-3

Required Hardware and Software

Required Hardware:

1 . Arduino Mega R3 ATmega2650 Board
2. MQ135 gas Sensor
3. MQ7 gas Sensor
4. MQ2 gas Sensor
5. LCD (20 X 4) display
6. TFT LCD display
7. Buzzer
8. 4 LED lights
9. SMPS supply 9V – 2A Arduino compatible or 9V Adapter
10. 10k POT(one)
11. 220 ohm resistor (one)
12. 1 kilo ohm resistor (four)
13. Jumper wires

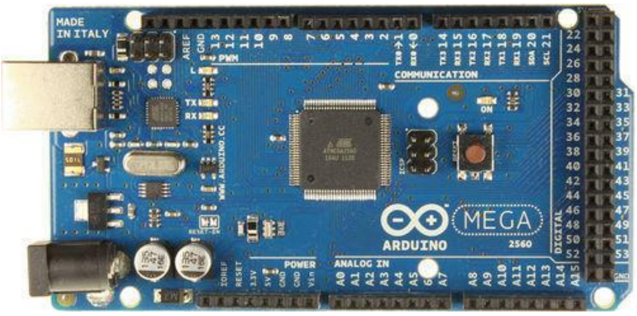
Required Software:

1.Arduino software

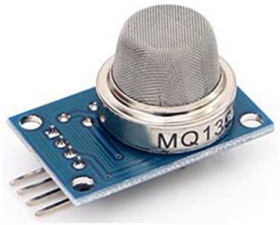
Chapter-4

Description of used components

(i) Arduino Mega R3 ATmega2650 Board: The Arduino Mega 2560 is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button and a flash memory of 256 kb. This gives your projects plenty of room and opportunities maintaining the simplicity and effectiveness of the Arduino platform. The Arduino Mega 2560 is programmed using the Arduino Software (IDE).



(ii)MQ-135 Gas Sensor: The MQ135 Air Quality Sensor consists of a small sensing material whose conductivity is lower in clean air and higher in polluted air, thus making the sensor very useful while detecting dangerous gases. The sensor ionizes the gases which come in its contact, making changes in the resistance of the sensing material.



MQ-135 Sensor Features:

- Fast response and High sensitivity
- Stable and long life
- Operating Voltage is +5V
- Detect/Measure NH3, NOx, alcohol, Benzene, smoke, CO2, etc.
- Analog output voltage: 0V to 5V
- Digital output voltage: 0V or 5V (TTL Logic)
- The Sensitivity of Digital pin can be varied using the potentiometer

(iii)MQ-7 Gas Sensor: The MQ7 is a simple-to-use Carbon Monoxide (CO) sensor suitable for sensing CO concentrations in the air. It can detect CO-gas concentrations anywhere from 20 to 2000ppm. MQ7 is a high sensitivity to carbon monoxide and stable and long-life span.

It make detection by method of cycle high and low temperature, and detect CO when low temperature (heated by 1.5V). The sensor's conductivity is more higher along with the gas concentration rising. When high temperature (heated by 5.0V), it cleans the other gases adsorbed under low temperature.

MQ-7 sensor features:

*Good sensitivity to Combustible gas in wide range.

* High sensitivity to Natural gas.

* Long life and low cost.

* Simple drive circuit.

(iv)MQ-2 Gas Sensor: MQ2 gas sensor is used to detect the presence of LPG, Propane and Hydrogen. It is also used to detect Methane and other combustible steam. It is low cost and suitable for different 15 application. This sensor is sensitive to flammable gas and smoke. Smoke sensor is given 5 volts to power it. Smoke sensor indicates smoke by the voltage that it outputs, more smoke more output. A potentiometer is provided to adjust the sensitivity. SnO2 is the sensor used which is of low conductivity when the air is clean. But when smoke exist sensor provides an analog resistive output based on concentration of smoke. The circuit has a heater. Power is given to heater by VCC and GND from power supply. The circuit has a variable resistor. The resistance across the pin depends on the smoke in air in the sensor. The resistance will be lowered if the content is more and voltage is increased between the sensor and load resistor.

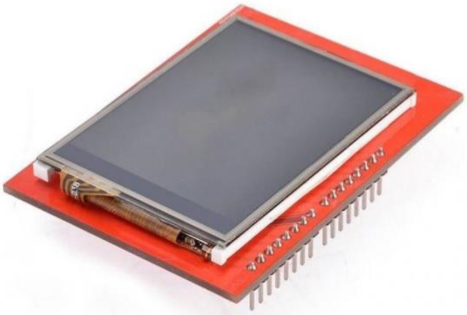
(v)LCD (20x4) Display: An LCD is an electronic display module which uses liquid crystal to produce a visible image. The 20×4 LCD display is a very basic module commonly used in DIYs and circuits. The 20×4 translates o a display 20 characters per line in 4 such lines. In this LCD each character is displayed in 5x7 pixel matrix.

Liquid crystal display technology works by blocking light. Specifically, an LCD is made of two pieces of polarized glass (also called substrate) that contain a liquid crystal material between them. A backlight creates light that passes through the first substrate.

(vi) TFT LCD display: TFT displays are liquid-crystal display modules with thin-film transistor technology. This TFT technology offers full RGB showcasing a range of colors and hues. Get rich colors, detailed images, and bright graphics from an LCD with a TFT screen.

The Monitors using TFT are known as TFT-LCD monitors. The display of TFT monitor has two Glass substrates enclosing a layer of liquid crystal. ... The LCD is placed between these electrodes. When a varying voltage is applied through the electrodes, the liquid crystal molecules align in different patterns.

The Arduino TFT screen is a backlit TFT LCD screen with a micro SD card slot in the back. You can draw text, images, and shapes to the screen with the TFT library. The screen's pin layout is designed to easily fit into the socket of an Arduino.



(vii)Buzzer: A buzzer or beeper is an audio signalling device, which may be mechanical, electromechanical, or piezoelectric (piezo for short). Typical uses of buzzers and beepers include alarm devices, timers, and confirmation of user input such as a mouse click or keystroke.

The buzzer consists of an outside case with two pins to attach it to power and ground. Inside is a piezo element, which consists of a central ceramic disc surrounded by a metal (often bronze) vibration disc.

(viii)Potentiometer: The Potentiometer is an electric instrument used to measure the EMF (electromotive force) of a given cell, the internal resistance of a cell. And also it is used to compare the EMFs of different cells. It can also use as a variable resistor in most of the applications.

The principle of a potentiometer is that the potential dropped across a segment of a wire of uniform cross-section carrying a constant current is directly proportional to its length. The potentiometer is a simple device used to measure the electrical potentials

(ix)Power Adapter: An AC adapter, AC/DC adapter, or AC/DC converter is a type of external power supply, often enclosed in a case similar to an AC plug. Other common names include plug pack, plug-in adapter, adapter block, domestic mains adapter, line power adapter, wall wart, power brick, and power adapter. Adapters for battery-powered equipment may be described as chargers or rechargers (see also battery charger). AC adapters are used with electrical devices that require power but do not contain internal components to derive the required voltage and power from mains power. The internal circuitry of an external power supply is very similar to the design that would be used for a built-in or internal supply.

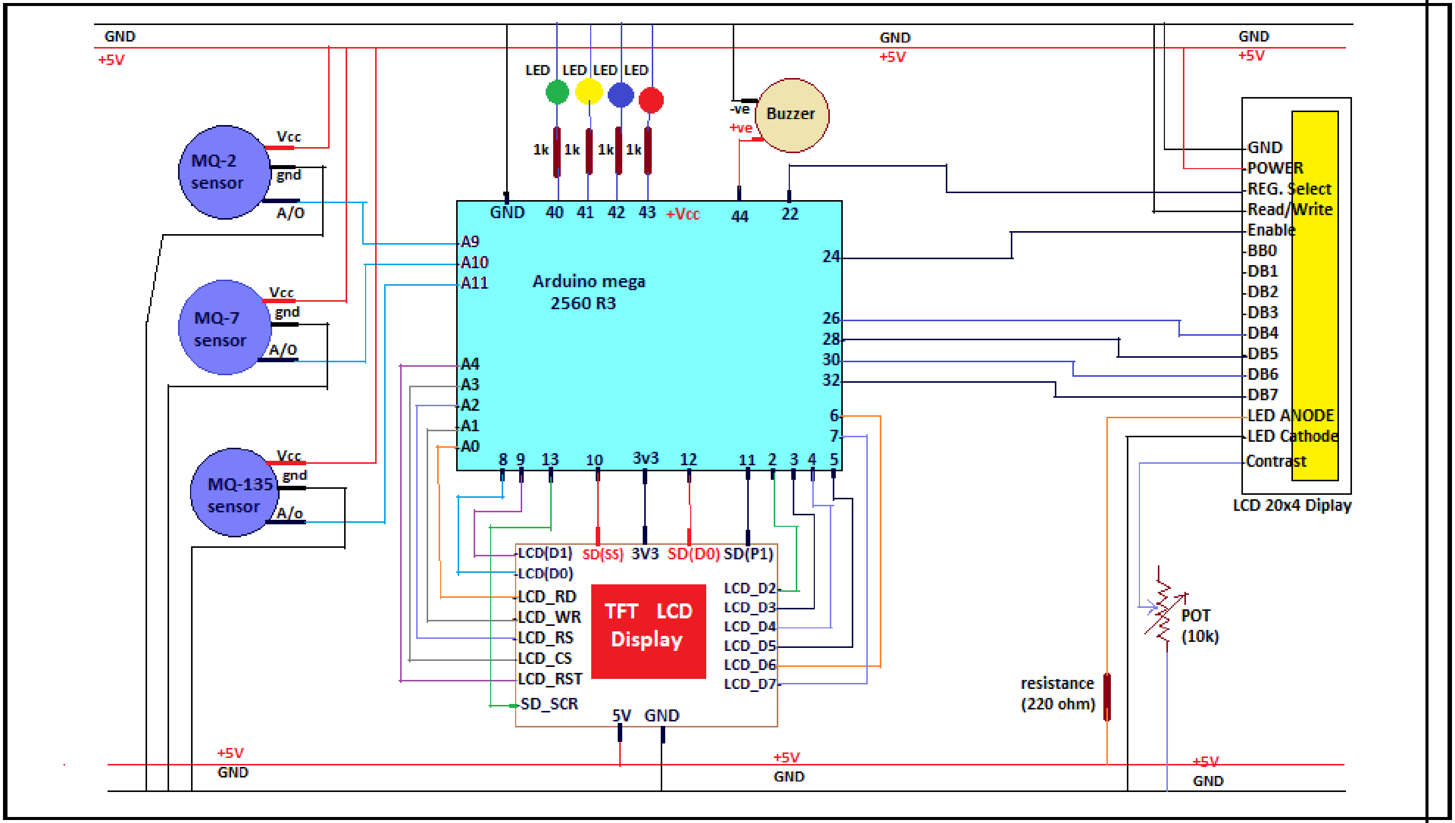
External power supplies are used both with equipment with no other source of power and with battery-powered equipment, where the supply, when plugged in, can sometimes charge the battery in addition to powering the equipment.

(x)LED: In electronics, an LED circuit or LED driver is an electricalcircuit used to power a light-emitting diode (LED). The circuit must provide sufficient current to light the LED at the required brightness, but must limit the current to prevent damaging the LED.

A light-emitting diode (LED) is a two-lead semiconductor light source. It is a basic pn-junction diode, which emits light when activated. When a fitting voltage is applied to the leads, electrons are able to recombine with electron holes within the device, releasing energy in the form of photons.

Chapter-5

Schematic Diagram



Pic2:Schematic diagram of Air Quality Monitoring System

Chapter-6

Circuit Connection Details

Connection Details :

Connection between LCD(20X4) display and Arduino Mega 2560 R3:

Arduino Mega 2560	LCD(20X4) display
+5v	GND
GND	POWER
22	REGISTER SELECT
24	ENABLE
26	DB4
28	DB5
30	DB6
32	DB7

Connection between TFT LCD display and Arduino mega 2560 R3:

TFT LCD display	Arduino mega 2560
5v	5V
GND	GND
LCD_RD	A0
LCD_WR	A1
LCD_RS	A2
LCD_CS	A3
LCD_RST	A4
3V3	3V3
SD_SCR	13
SD_D0	12
SD_P1	11
SD_SS	10
LCD_D1	9
LCD_D0	8
LCD_D7	7
LCD_D6	6
LCD_D5	5
LCD_D4	4
LCD_D3	3
LCD_D2	2

Connection between MQ-135 Gas Sensor and Arduino Mega 2560:

Mq-135	Arduino mega 2560
Vcc	5V
GND	GND
A/O	A11

Connection between MQ-7 Gas Sensor and Arduino Mega 2560:

Mq-7	Arduino mega 2560
Vcc	5V
GND	GND
A/O	A10

Connection between MQ-2 Gas Sensor and Arduino Mega 2560:

Mq-135	Arduino Mega 2560
Vcc	5V
GND	GND
A/O	A9

Connection between Buzzer and Arduino mega 2560 :

Buzzer	Arduino Mega 2560
+ve	44
-ve	GND

Connection between Arduino Mega 2560 R3 and LED panel with other LED panel connections:

Arduino mega Port no.	Resistance value	LED colour	LED Port
40	1K	Green	Anode
41	1K	Yellow	Anode
42	1K	Blue	Anode
43	1K	Red	Anode

LED colour	LED port	Ground
Green	Cathode	GND
Yellow	Cathode	GND
Blue	Cathode	GND
Red	Cathode	GND

LCD(20X4) display Contrast Pin connection with Potentiometer:

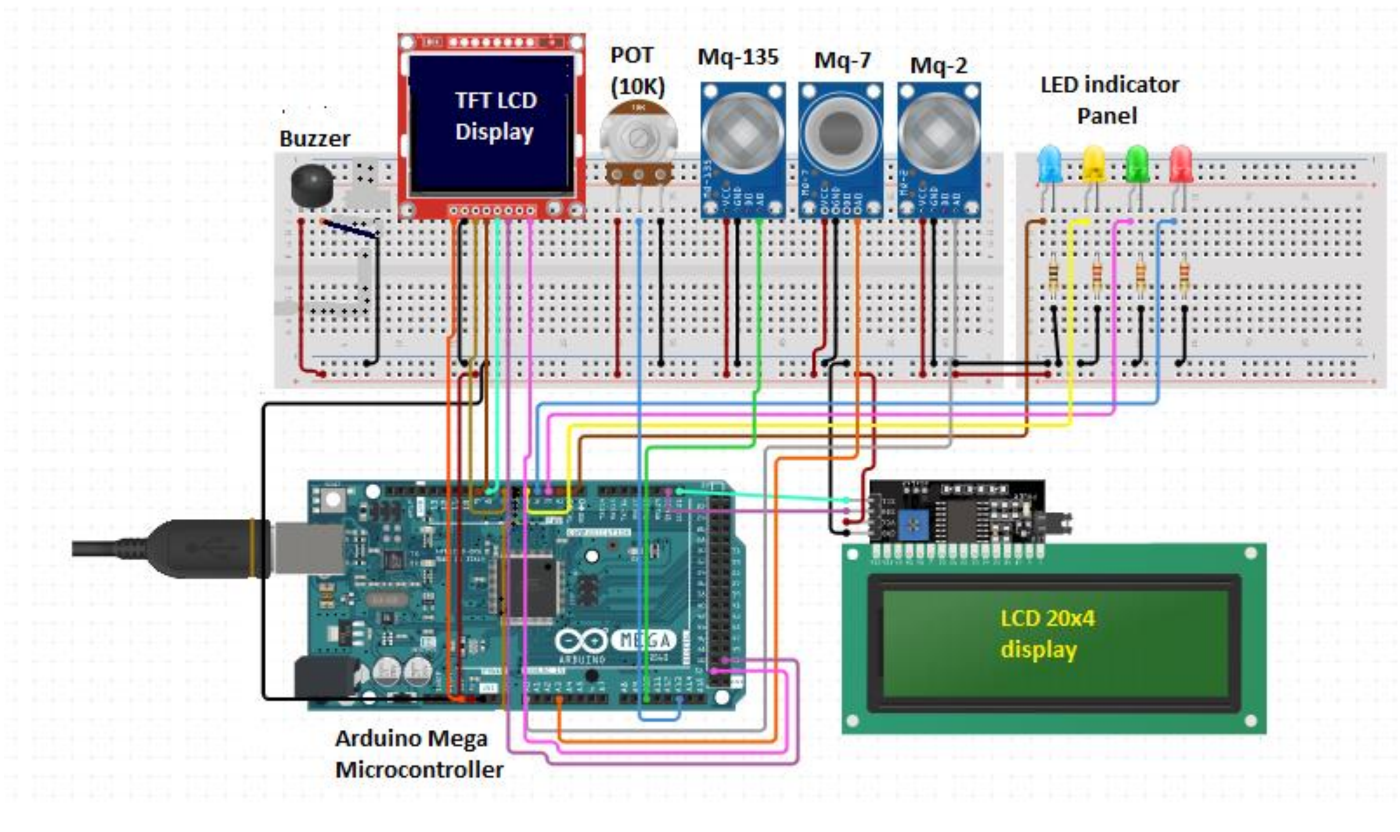
LCD(20X4) display	Potentiometer
Contrast	Pin2(middle pin)

Potentiometer	Ground
Pin 3(end pin)	GND

Chapter-7

Simulation Diagram of the Project

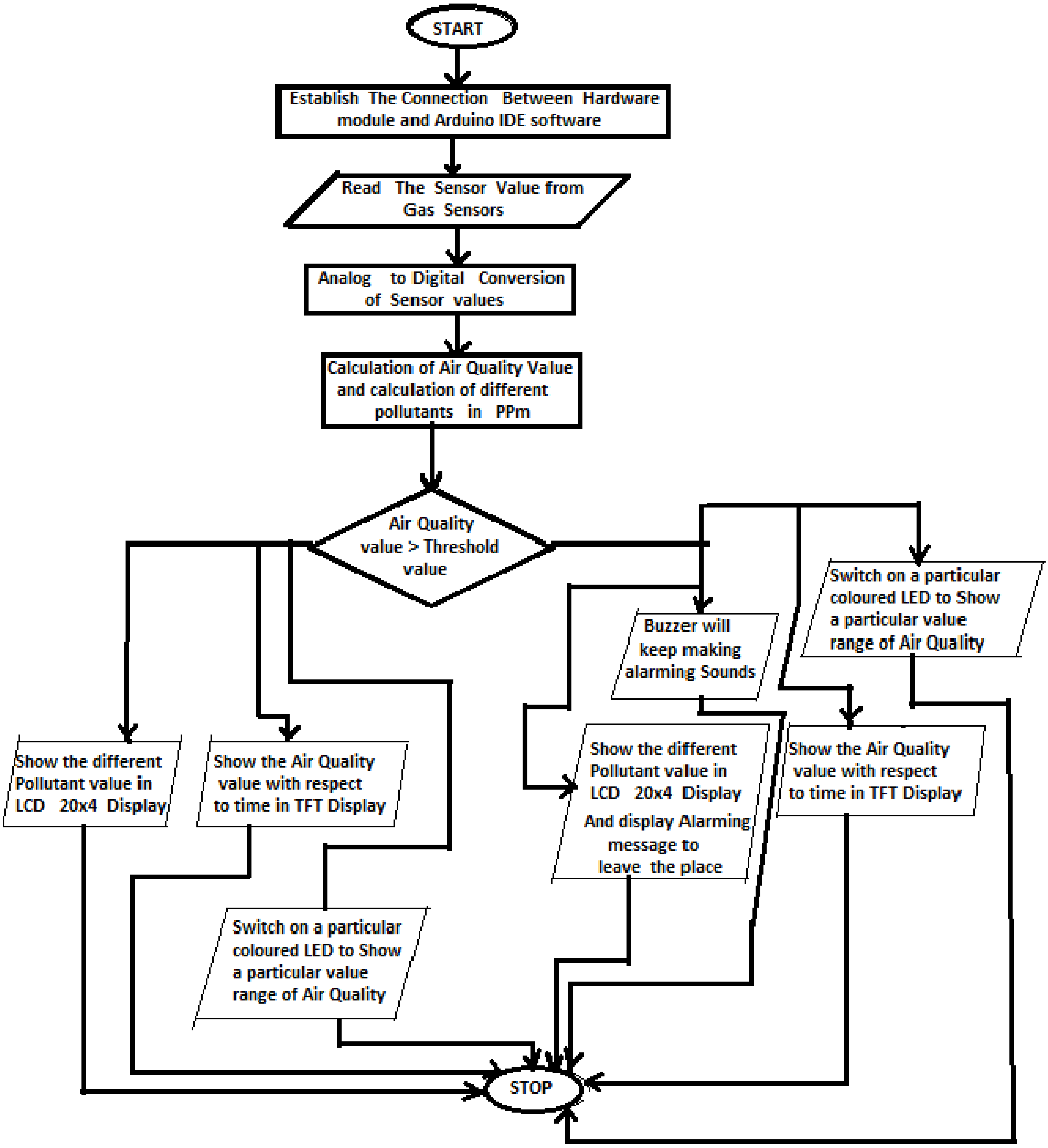
Simulation Diagram:



Pic3:Simulation diagram Of Air Quality Monitoring System

Chapter-8

Flowchart Of The Project


START
Establish The Connection Between Hardware module and Arduino IDE software
Read The Sensor Value from Gas Sensors
Analog to Digital Conversion of Sensor values
Calculation of Air Quality Value and calculation of different pollutants in PPm
Air Quality value > Threshold value
Show the different Pollutant value in LCD 20x4 Display
Show the Air Quality value with respect to time in TFT Display
Switch on a particular coloured LED to Show a particular value range of Air Quality
Buzzer will keep making alarming Sounds
Switch on a particular coloured LED to Show a particular value range of Air Quality
Show the different Pollutant value in LCD 20x4 Display And display Alarming message to leave the place
Show the Air Quality value with respect to time in TFT Display
STOP

Chapter-9
Working Process of The Project

Working Process:

(1) First We will make circuit connections As shown in picture2 which is the schematic diagram of the Project . Then after completing the Hardware module set up work we will establish connection between Arduino IDE and hardware module . Then will write the project program to the Arduino software and upload it to the Arduino Microcontroller.

(2) The Gas Sensors will take the reading of the Pollutants , We used three Gas Sensors here . These sensors are MQ-135 , MQ-7 And MQ-2. Here MQ-135 will help us to find the Pollutants like Benzene,Alcohol,Ammonia And Carbon dioxide. Mq-7 will help us to find the presence of carbon mono-oxide. And Mq-2 will help us to detect the values of Methane, Butane ,LPG and smoke.

(3) These sensors will send us the analog values of to the microcontroller and using code in Arduino program will convert those analog values into PPM and will also calculate the Air Quality value.

(4) We will display the values of the different pollutants in LCD 20x4 display and we will also display the Air Quality value in LCD 20x4 display module.

(5) We will also monitor the change in Air Quality value with respect to time in a graphical mode using TFT LCD module.

(6) The Air quality value is divided into certain limits , If the value falls in one of the those limit then corresponding colour LED will glow for that value and also LCD 20x4 display will also show a specific message which was set for that range of value in the program.

(7) If the Air quality value goes beyond the threshold value , the buzzer will then send us alarming sound signals And LCD 20x4 display module will also show a alarming message to leave the place as soon as possible.

Chapter-10

Air Quality Index (AQI) and Safety Level of elements of Air

(i)Air Quality Index:

An air quality index (AQI) is a number used by government agencies to communicate to the public how polluted the air currently is or how polluted it is forecast to become. As the AQI increases, an increasingly large percentage of the population is likely to experience increasingly severe adverse health effects. Different countries have their own air quality indices, corresponding to different national air quality standards.

There are many elements in the air that become dangerous and harmful for human if they increase in numbers. For our, project purposes we have considered four elements here,CO2,CO,LPG and CH4.They can damage a real good to human if increased immensely in numbers.

The higher the AQI value,The greater the level of air pollution and the greater the health concern. For example, an AQI value of 50 represents good air quality with little potential to affect public health, while an AQI value over 300 represents hazardous air quality.

An AQI value of 100 generally corresponds to the national air quality standard for the pollutant, which is the level EPA has set to protect public health. AQI values below 100 are generally thought of as satisfactory. When AQI values are above 100, air quality is considered to be unhealthy-at first for certain sensitive groups of people, then for everyone as AQI values get higher.

There are six AQI categories, namely Good, Satisfactory, Moderately polluted, Poor, Very Poor, and Severe. The proposed AQI will consider eight pollutants (PM_{10}, $PM_{2.5}$, NO_2, SO_2, CO, O_3, NH_3, and Pb) for which short-term (up to 24-hourly averaging period) National Ambient Air Quality Standards are prescribed.

The air quality index is a piecewise linear function of the pollutant concentration. At the boundary between AQI categories, there is a discontinuous jump of one AQI unit. To convert from concentration to AQI this equation is used:

$$I = \frac{I_{high} - I_{low}}{C_{high} - C_{low}}(C \ - C_{low}) + \ I_{low}$$

Here,

$I =$ the (Air Quality)index,
$I_{high} =$ the index breakpoint corresponding to C_{high}

$C_{high} =$ the concentration breakpoint that is $\geq$ C

$I_{low} =$ the index breakpoint corresponding to C_{low}

$C_{low} =$ the concentration breakpoint that is $\leq$ C

$C =$ the pollutant concentration

(ii)AQI Mechanics :

An individual score (Individual Air Quality Index, IAQI) is assigned to each pollutant and the final AQI is the highest of these six scores. The final AQI value can be calculated with two options either per hour or per 24 hours. The concentrations of pollutants can be measured quite otherwise. If the AQI value is calculated hourly, then SO2, NO2, CO concentrations are measured as average per 24hour, O3 concentration is measured as average per hour and the moving average per 8h, PM2.5 and PM10 concentrations are measured as average per hour and per 24h. If the AQI value is calculated per 24h, then SO2, NO2, CO, PM2.5 and PM10 concentrations are measured as average per 24h, while O3 concentration is measured as the maximum 1h average and the maximum 24h moving average. The IAQI of each pollutant is calculated according to a formula published by the MEP.

The score for each pollutant is non-linear, as is the final AQI score. Thus an AQI of 300 does not mean twice the pollution of AQI at 150, nor does it mean the air is twice as harmful. The concentration of a pollutant when its IAQI is 100 does not equal twice its concentration when its IAQI is 50, nor does it mean the pollutant is twice as harmful. While an AQI of 50 from day 1 to 182 and AQI of 100 from day 183 to 365 does provide an annual average of 75, it does not mean the pollution is acceptable even if the benchmark of 100 is deemed safe. Because the benchmark is a 24-hour target, and the annual average must match the annual target, it is entirely possible to have safe air every day of the year but still fail the annual pollution benchmark.

(iii)Air Quality Index category Chart:

AQI	Associated Health Impacts
Good (0–50)	Minimal impact
Satisfactory (51–100)	May cause minor breathing discomfort to sensitive people.
Moderately polluted (101–200)	May cause breathing discomfort to people with lung disease such as asthma, and discomfort to people with heart disease, children and older adults.
Poor (201–300)	May cause breathing discomfort to people on prolonged exposure, and discomfort to people with heart disease.
Very poor (301–400)	May cause respiratory illness to the people on prolonged exposure. Effect may be more pronounced in people with lung and heart diseases.
Severe (401–500)	May cause respiratory impact even on healthy people, and serious health impacts on people with lung/heart disease. The health impacts may be experienced even during light physical activity.

(iv)AQI Category, Pollutants and Health Breakpoints:

AQI Category (Range)	PM_{10} (24hr)	$PM_{2.5}$ (24hr)	NO_2 (24hr)	O_3 (8hr)	CO (8hr)	SO_2 (24hr)	NH_3 (24hr)	Pb (24hr)	CO_2 (24hr)
Good (0–50)	0–50	0–30	0–40	0–50	0–1.0	0–40	0–200	0–0.5	250-330
Satisfactory (51–100)	51–100	31–60	41–80	51–100	1.1–2.0	41–80	201–400	0.5–1.0	350-600
Moderately polluted (101–200)	101–250	61–90	81–180	101–168	2.1–10	81–380	401–800	1.1–2.0	600-1000
Poor (201–300)	251–350	91–120	181–280	169–208	10–17	381–800	801–1200	2.1–3.0	1000-2000
Very poor (301–400)	351–430	121–250	281–400	209–748	17–34	801–1600	1200–1800	3.1–3.5	2000-6000
Severe (401–500)	430+	250+	400+	748+	34+	1600+	1800+	3.5+	6000+

(v)Used Mathematical Formulas to convert Analog values of Gas sensors into PPM:

(1)MQ-135 calibration:

$$V_{out} = AnalogValue_{MQ135} \times \frac{5}{1023}$$

V_{out} = Analog Value Converted into equivalent digital voltage form

$$R_s = \frac{V_c R_L}{V_{out}} - R_L$$

R_s = Sensor resistance , V_c = Given input voltage , R_L = Load Resistance

R_0 = the resistance of the gas sensor at defined concentration

$R_L = 47k\Omega$ and $R_0 = 76k\Omega$

$$\mathbf{(CO_2)_{PPM} = 114.3544 \times (\frac{R_S}{R_0})^{-2.93599}}$$

$$\mathbf{(NH_3)_{PPM} = 102.694 \times (\frac{R_S}{R_0})^{-2.48818}}$$

(2)MQ-7 calibration:

$$V_{out} = AnalogValue_{MQ7} \times \frac{3.3}{4095}$$

V_{out} = Analog Value Converted into equivalent digital voltage form

$$\mathbf{(CO)_{PPM} = 3.027 \times e^{(1.0698 \times V_{out})}}$$

(3)MQ-2 calibration:

$$V_{out} = AnalogValue_{MQ7} \times \frac{3.3}{4095}$$

V_{out} = Analog Value Converted into equivalent digital voltage form

$$\mathbf{(Methane)_{PPM} = 10.938 \times e^{(1.7742 \times V_{out})}}$$

$$\mathbf{(LPG)_{PPM} = 26.572 \times e^{(1.2894 \times V_{out})}}$$

Chapter-11

Project Program

Program:

```
#define BLACK       0x0000
#define BLUE        0x001F
#define RED         0xF800
#define GREEN       0x07E0
#define CYAN        0x07FF
#define MAGENTA     0xF81F
#define YELLOW      0xFFE0
#define WHITE       0xFFFF
#define LIGHT_GREY    0xBDF7
#define DARK_GREY     0x7BEF
#define ORANGE      0xFBE0
#define BROWN       0x79E0
#define PINK        0xF81F
#include <Adafruit_GFX.h> // Hardware-specific library
#include <MCUFRIEND_kbv.h>
#include <LiquidCrystal.h>
#define LCD_CS A3 // Chip Select goes to Analog 3
#define LCD_CD A2 // Command/Data goes to Analog 2
#define LCD_WR A1 // LCD Write goes to Analog 1
#define LCD_RD A0 // LCD Read goes to Analog 0
#define LCD_RESET A4 // Can alternately just connect to Arduino's reset pin
MCUFRIEND_kbv tft;
const int rs = 22, en = 24, d4 = 26, d5 = 28, d6 = 30, d7 = 32;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
  int valueBlock[500];
  int timeBlock[500];
  int locationBlock[500];
  int valuePos;
  int blockPos;
  float AQI;
  int i;
  int j;
  bool proDebug = 0;
  const int numberOfMarks = 8;
  const int originX = 62;
  const int originY = 200;
  const int sizeX = 270;
  const int sizeY = 150;
  const int deviation = 30;
  const int buzzer = 44;
```

```
   uint8_t aspect=3;
  uint16_t graphColor = BLUE;
  uint16_t pointColor = BLACK;
  uint16_t lineColor = GREEN;

  int sensorValue1;
  int sensorValue2;
  int sensorValue0;
  float ResVal;
  float COppm1;
  float CO2ppm1;
  float NH3ppm1;
  float LPGppm1;
  float CH4ppm1;
   float VrL11;
   float VrL22;
   float VrL33;
  int Vc=5;
  int rL=47;
  float r0=76;
  float rS;
  int VrL1;
  int VrL2;
  int VrL3;
  String graphName = "Air Quality vs Time Graph";
  int graphRange = 1200;
  int markSize = 3;
  int cf=(-75);
  int boxSize = (sizeX / numberOfMarks);
  int mark[] = {(boxSize + deviation), ((boxSize * 2) + deviation), ((boxSize * 3) + deviation), ((boxSize *
4) + deviation), ((boxSize * 5) + deviation), ((boxSize * 6) + deviation), ((boxSize * 7) + deviation),
((boxSize * 8) + deviation)};
  const int minorSizeY = (originY + 10);
  const int minorSizeX = (originX -10);
  int numberSize = (sizeY / 6);
  int number[] = {numberSize, (numberSize * 2), (numberSize * 3), (numberSize * 4), (numberSize * 5),
(numberSize * 6)};
  int numberValue = (graphRange / 6);
  int val[] = {graphRange, (numberValue * 5), (numberValue * 4), (numberValue * 3), (numberValue * 2),
numberValue};
  int SENSOR_GAS2 = A11; //mq-135
  int SENSOR_GAS1 = A10;  //mq-7
  int SENSOR_GAS0 = A9; //mq-2
```

```
void drawHome()
{
  tft.fillScreen(YELLOW);
  delay(500);
  tft.setCursor(60, 10); // set the cursor
  tft.setTextColor(RED); // set the colour of the text
  tft.setTextSize(2); // set the size of the text
  tft.println(" Air  Pollution ");
  tft.setCursor(60, 30); // set the cursor
  tft.setTextColor(RED); // set the colour of the text
  tft.setTextSize(2); // set the size of the text
  tft.println("   Measuring");
  tft.setCursor(60, 50); // set the cursor
  tft.setTextColor(RED); // set the colour of the text
  tft.setTextSize(2); // set the size of the text
  tft.println("   Data graph");
  tft.setCursor(80, 70); // set the cursor
  tft.setTextColor(BLUE); // set the colour of the text
  tft.setTextSize(2); // set the size of the text
  tft.println("  by");
  tft.setCursor(40, 90); // set the cursor
  tft.setTextColor(BLUE); // set the colour of the text
  tft.setTextSize(2); // set the size of the text
  tft.println("    Rupayan Halder");
  tft.setCursor(40, 110); // set the cursor
  tft.setTextColor(BLUE); // set the colour of the text
  tft.setTextSize(2); // set the size of the text
  tft.println("    Arjun  Sen ");
  tft.setCursor(40, 130); // set the cursor
  tft.setTextColor(BLUE); // set the colour of the text
  tft.setTextSize(2); // set the size of the text
  tft.println("    Rukmini  Thakur");
  tft.setCursor(40, 150); // set the cursor
  tft.setTextColor(RED); // set the colour of the text
  tft.setTextSize(2); // set the size of the text
  tft.println("    Guided by ");
  tft.setCursor(10, 170); // set the cursor
  tft.setTextColor(BLUE); // set the colour of the text
  tft.setTextSize(2); // set the size of the text
  tft.println(" Prof. (Dr.) Amitabha Das");
  delay(10000);
  tft.fillScreen(YELLOW);
  delay(500);
}
```

```
void ledlights()
{
  for(j=40;j<=43;j++)
  {
    pinMode(j, OUTPUT);
  }
}
void LCD20into4home()
{
            pinMode(SENSOR_GAS0, INPUT);
            pinMode(SENSOR_GAS1, INPUT);
            pinMode(SENSOR_GAS2, INPUT);
            lcd.begin(20, 4) ;
            lcd.setCursor(3,0);
            lcd.print("Air Quality");
            delay(2000);
            lcd.clear();
            lcd.setCursor(1,0);
            lcd.print("Monitor System");
            lcd.setCursor(1,1) ;
            lcd.print("Produced by");
            delay(2000);
            lcd.clear();
            lcd.setCursor(0,0);
            lcd.print("ARJUN       RUKMINI");
            lcd.setCursor(1,1);
            lcd.print("    RUPAYAN  ");
            lcd.setCursor(3,2);
            lcd.print("Guided by");
            lcd.setCursor(0,3);
            lcd.print("Dr. Amitabha Das");
            delay(5000);
            lcd.clear();
            lcd.print("   System setup");
            delay(1000);
            lcd.clear();
            lcd.print("    Starting...");
            delay(1000);
            lcd.clear();
}
```

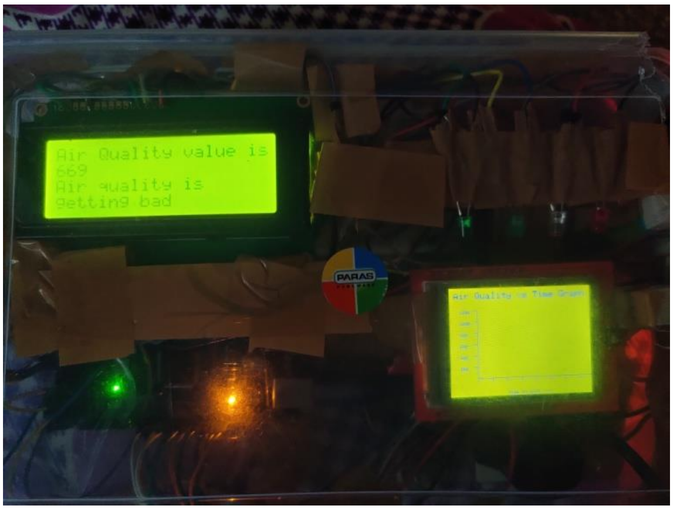
```
void LCD20into4()
{
        lcd.setCursor(0,0);
        lcd.print("CO2Value");
        lcd.setCursor(12,0);
        lcd.print(CO2ppm1);
        lcd.setCursor(16,0);
        lcd.print("ppm");
        delay(250);
        lcd.setCursor(0,1) ;
        lcd.print("NH3Value:");
        lcd.setCursor(12,1);
        lcd.print(NH3ppm1);
        lcd.setCursor(16,1);
        lcd.print("ppm");
        delay(250);
        lcd.setCursor(0,2);
        lcd.print("CO_value:");
        lcd.setCursor(12,2);
        lcd.print(COppm1);
        lcd.setCursor(16,2);
        lcd.print("ppm");
        delay(250);
        lcd.setCursor(0,3) ;
        lcd.print("LPG_value:");
         lcd.setCursor(12,3);
        lcd.print(LPGppm1);
        lcd.setCursor(16,3);
        lcd.print("ppm");
        delay(250);
        lcd.clear();
        delay(300);
        lcd.setCursor(0,0);
        lcd.print("CH4Value");
        lcd.setCursor(12,0);
        lcd.print(CH4ppm1);
        lcd.setCursor(16,0);
        lcd.print("ppm");
        delay(250);
        lcd.setCursor(0,1) ;
        lcd.print("AirQuality:");
        lcd.setCursor(12,1);
        lcd.print(ResVal);
        lcd.setCursor(16,1);
        lcd.print("ppm");
        delay(250);
        lcd.clear();
        delay(300);
    if(ResVal<=50)
    {
        lcd.setCursor(0,0);
        lcd.print("Air Quality value is");
        lcd.setCursor(0,1) ;
        lcd.print(ResVal);
        lcd.setCursor(0,2);
        lcd.print("Air quality is");
        lcd.setCursor(0,3);
        lcd.print(" good ");
       digitalWrite(40, HIGH);
        digitalWrite(41, LOW);
        digitalWrite(42, LOW);
        digitalWrite(43, LOW);
        delay(6000);
    }
    if(ResVal>50 && ResVal<=100)
    {
        lcd.setCursor(0,0);
        lcd.print("Air Quality value is");
        lcd.setCursor(0,1) ;
        lcd.print(ResVal);
        lcd.setCursor(0,2);
        lcd.print("Air quality is");
        lcd.setCursor(0,3);
        lcd.print("Satisfactory");
        digitalWrite(40, LOW);
         digitalWrite(42, LOW);
        digitalWrite(43, LOW);
        digitalWrite(41, HIGH);
        delay(6000);
    }
    if(ResVal>101 && ResVal<=200)
    {
         lcd.setCursor(0,0);
        lcd.print("Air Quality value is");
        lcd.setCursor(0,1) ;
        lcd.print(ResVal);
        lcd.setCursor(0,2);
```

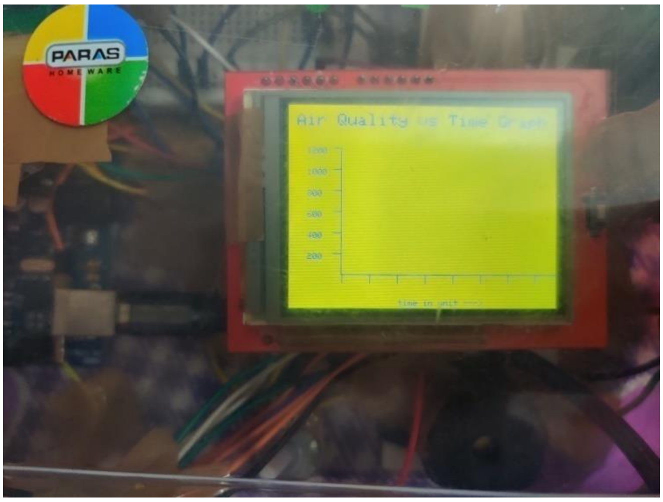
```
          lcd.print("Air quality is");
          lcd.setCursor(0,3);
          lcd.print("Moderately polluted");
          digitalWrite(40, LOW);
          digitalWrite(41, LOW);
          digitalWrite(43, LOW);
          digitalWrite(42, HIGH);
          delay(6000);
      }
      if(ResVal>200)
      {
          lcd.setCursor(0,0);
          lcd.print("Air Quality value is");
          lcd.setCursor(0,1) ;
          lcd.print(ResVal);
          lcd.setCursor(0,2);
          lcd.print("Air very Poor here");
          lcd.setCursor(0,3);
          lcd.print("Please Leave Qyickly");
          digitalWrite(40, LOW);
          digitalWrite(41, LOW);
          digitalWrite(42, LOW);
          digitalWrite(43, HIGH);
          tone(buzzer, 1000);
          delay(6000);
          noTone(buzzer);
          delay(500);
      }

}
void drawGraph()    //for drawing the axis
{
  // draw title
  tft.setCursor(10, 10); // set the cursor
  tft.setTextColor(BLUE); // set the colour of the text
  tft.setTextSize(2); // set the size of the text
  tft.println(graphName);
  // draw outline
  tft.drawLine(originX, originY, (originX + sizeX), originY, graphColor);
  tft.drawLine(originX, originY, originX, (originY - sizeY), graphColor);
  // draw lables
  for( i = 0; i < numberOfMarks; i++)
  {
    tft.drawLine(mark[i], originY, mark[i], minorSizeY, graphColor);
  }
```

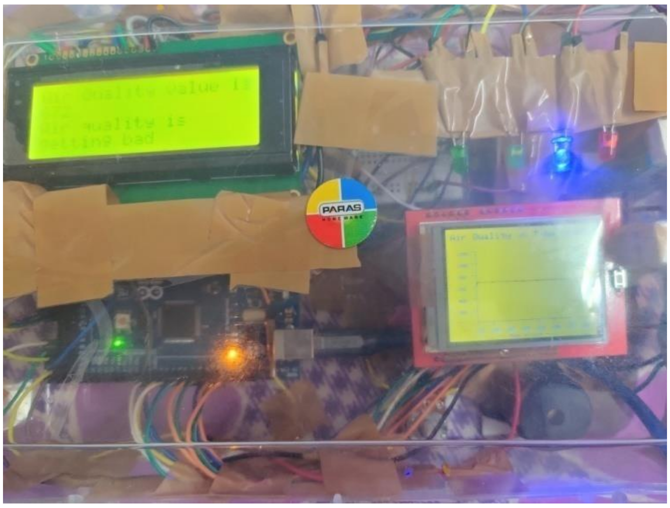
```
// draw numbers
  for( i = 0; i < 6; i++)
  {
    tft.drawLine(originX, (originY - number[i]), minorSizeX, (originY - number[i]), graphColor);
  }
  // draw number values
  for( i = 0; i < 6; i++)
      {
        tft.setCursor((minorSizeX - 30), (number[i] + numberSize));
        tft.setTextColor(graphColor);
        tft.setTextSize(1);
        tft.println(val[i]);
       }
  tft.setCursor(130, 230); // set the cursor
  tft.setTextColor(BLUE); // set the colour of the text
  tft.setTextSize(1); // set the size of the text
  tft.println("time in unit --->");
}
void graph()
    {
    AQI=(((CO2ppm1+NH3ppm1+COppm1+LPGppm1+CH4ppm1)/5)+cf);
    timeBlock[valuePos] = ((millis() - 4500) / 1000);
     valueBlock[valuePos] = AQI;
     if(proDebug)
         {
         Serial.println(timeBlock[valuePos]);
         }
        if(blockPos < 8)
         {
         // print the time
         tft.setCursor((mark[valuePos] - 5), (originY + 16));
         tft.setTextColor(graphColor, WHITE);
         tft.setTextSize(1);
         tft.println(timeBlock[valuePos]);
         // map the value
          locationBlock[valuePos] = map(AQI, 0, graphRange, originY, (originY - sizeY));
         // draw point
         tft.fillRect((mark[valuePos] - 1), (locationBlock[valuePos] - 1), markSize, markSize, pointColor);
         // try connecting to previous point
        if(valuePos != 0)
        {
        tft.drawLine(mark[valuePos], locationBlock[valuePos], mark[(valuePos - 1)], locationBlock[(valuePos
- 1)],      pointColor);
        }
blockPos++;
        }
       else
         {
        // clear the graph's canvas
         tft.fillRect((originX + 2), (originY - sizeY), sizeX, sizeY, YELLOW);

         // map the value - current point
         locationBlock[valuePos] = map(AQI, 0, graphRange, originY, (originY - sizeY));
         // draw point - current point
           tft.fillRect((mark[7]), (locationBlock[valuePos] - 1), markSize, markSize, pointColor);
           // draw all points
        for( i = 0; i < 8; i++)
           {
      tft.fillRect((mark[(blockPos - (i + 1))] - 1), (locationBlock[(valuePos - i)] - 1), markSize, markSize,
pointColor);
           }
    // draw all the lines
        for(int i = 0; i < 8; i++)
    {
      tft.drawLine(mark[blockPos - (i + 1)], locationBlock[valuePos - i], mark[blockPos - (i + 2)],
locationBlock[valuePos - (i + 1)], pointColor);
    }
    // change time lables
           for(i = 0; i <8; i++)
         {
          tft.setCursor((mark[(7 - i)] - 5), (originY + 16));  //7
          tft.setTextColor(graphColor, WHITE);
          tft.setTextSize(1);
          tft.println(timeBlock[valuePos - i]);
         }
  }
  valuePos++;
}
void setup()
{
                Serial.begin(9600);
                tft.reset();
                uint16_t identifier = tft.readID();
                Serial.print("ID = 0x");
                Serial.println(identifier, HEX);
                if (identifier == 0xEFEF)
                identifier = 0x9325;
```

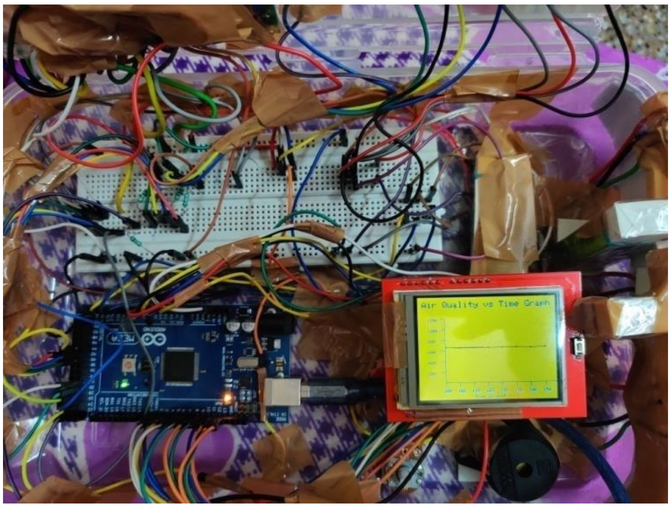
```
                 tft.begin(identifier);
                ledlights();
                LCD20into4home();
                tft.setRotation(aspect);
                drawHome();
                drawGraph();
}
void loop()
{
        int t;
        uint16_t x = 00, y = 00;
        tft.setRotation(aspect);
        sensorValue2 = analogRead(SENSOR_GAS2);  //mq135
        VrL11= (sensorValue2*5)/1023;
        VrL1=VrL11;
        rS=(((Vc*rL)/VrL1)-rL);
        CO2ppm1=114.3544*(pow((rS/r0),(-2.9355)));
        NH3ppm1=102.694*(pow((rS/r0),(-2.48818)));
        sensorValue1 = analogRead(SENSOR_GAS1);   //mq-7
        VrL22=(sensorValue1*3.3)/4095;
        COppm1=3.027*(exp(1.0698*VrL22));
        sensorValue0 = analogRead(SENSOR_GAS0); //mq-2
        VrL33=(sensorValue0*3.3)/4095;
        LPGppm1=26.572*(exp(1.2894*VrL22));
        CH4ppm1=10.938*(exp(1.7742*VrL22));
        ResVal=(((CO2ppm1+NH3ppm1+COppm1+LPGppm1+CH4ppm1)/5)+cf);
        LCD20into4();
        graph();
}
```

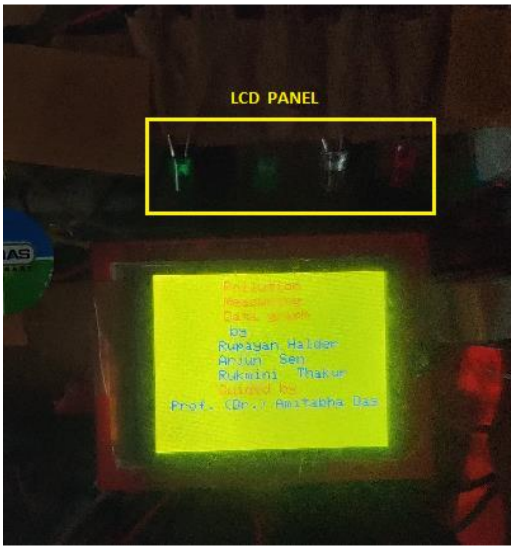
Chapter-12
Project Pictures

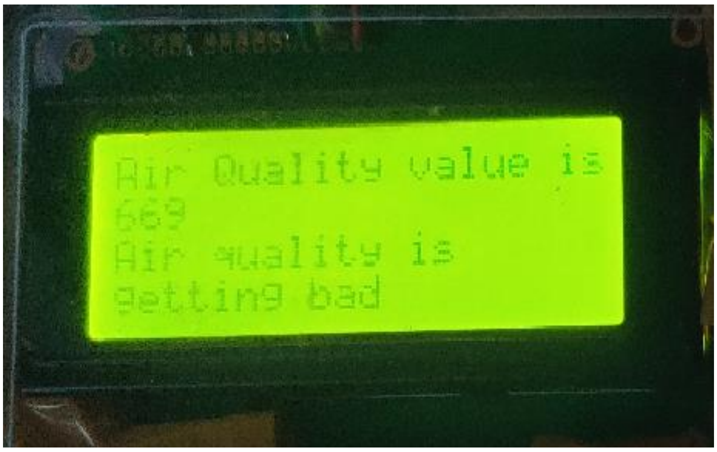












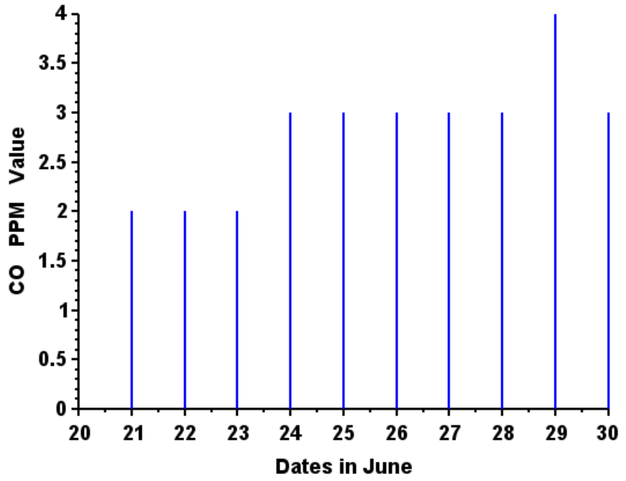
Chapter13

Results of The Project

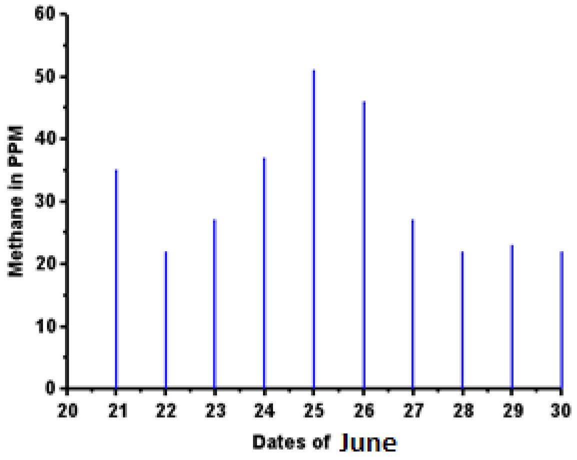
We have carried out an experiment from 21st June to 30Th June with this project. We have taken one sample data each day from 21st June to 30th June , Here is our result :

	21st June	22nd June	23rd June	24th June	25th June	26th June	27th June	28th June	29th June	30th June
CO	2	2	2	3	3	3	3	3	4	3
Methane	35	22	27	37	51	46	27	22	23	22
LPG	20	22	23	21	25	23	22	21	20	29
NH3	256	255	256	255	256	255	256	257	256	257
CO2	423	421	424	421	423	422	423	425	424	425
Air Quality Index(AQI)	72.2	69.4	71.4	72.4	76.6	74.8	71.2	70.6	70.4	72.2

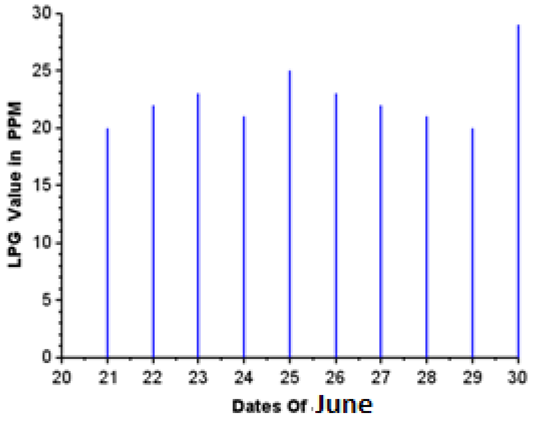
Graphical Analysis Of data:



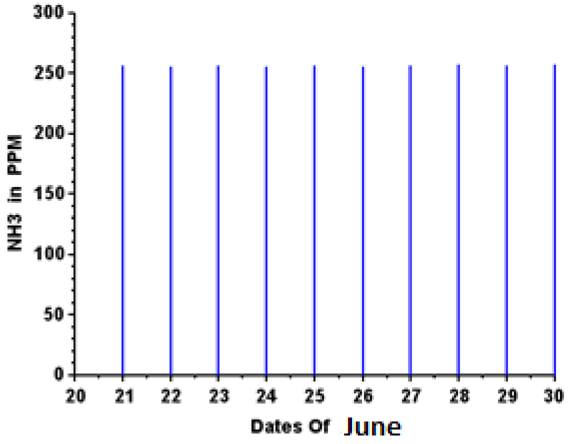
Graphical Representation of CO in different days of JUNE



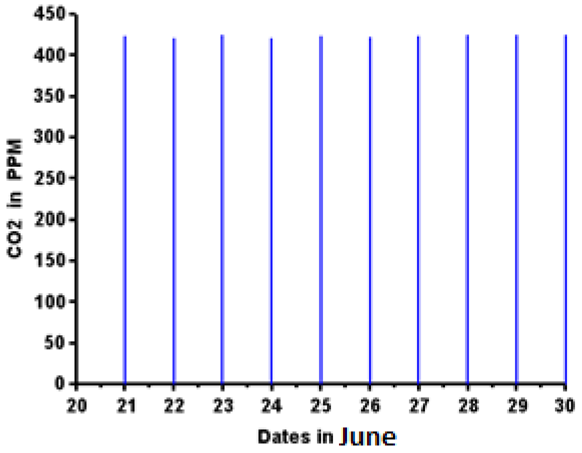
Graphical Representation of Methane in different days of JUNE



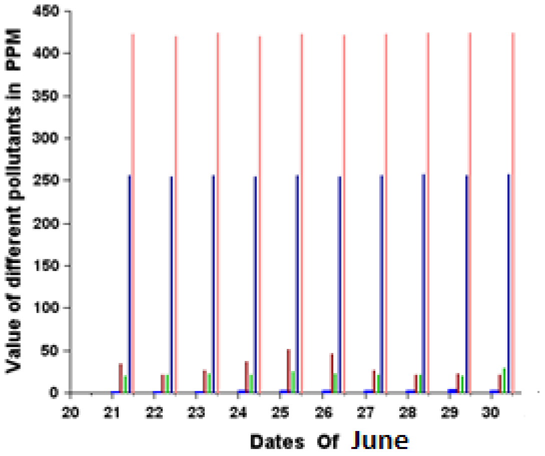
Graphical Representation of LPG in different days of JULY



Graphical Representation of NH3 in different days of JUNE

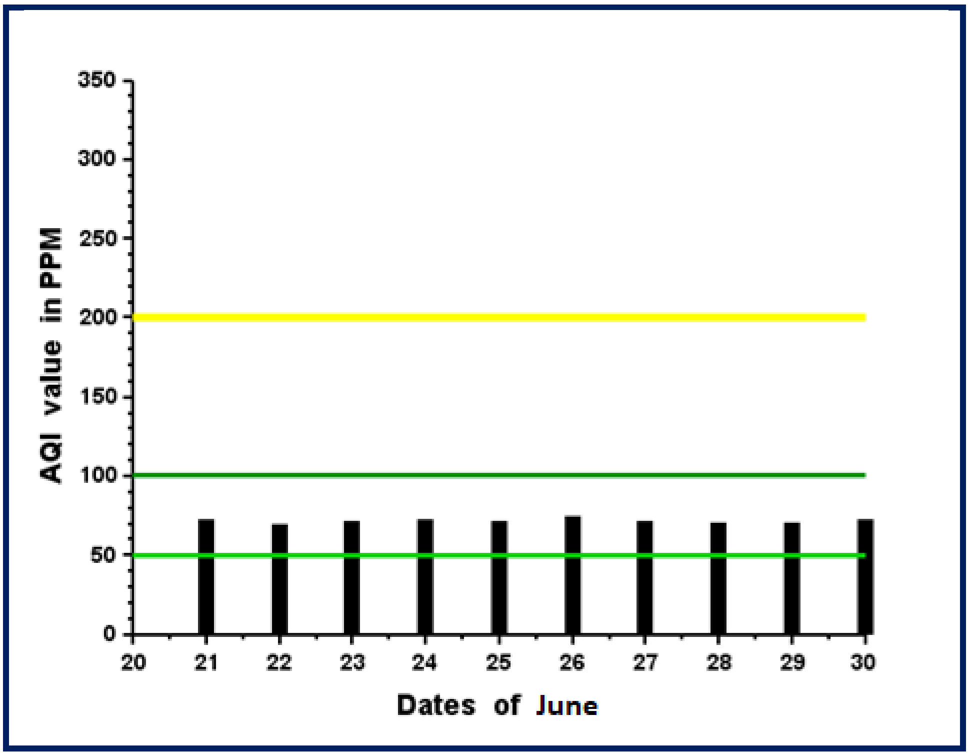


Graphical Representation of CO2 in different days of JUNE



Graphical Representation and comparison of different pollutants in different days of JUNE

Graphical Representation Of Air Quality Index:



Discussion of the Result:

So in our Area the quality remains in the satisfactory level. Each and every day the quality almost remained same and satisfactory. The average Air Quality Index is 72.1 PPM which in beyond 50PPM but under 100 PPM . So And also the quality of air doesn’t change here frequently which is pretty good sign.

Chapter 14

ADVANTAGES & APPLICATION

(i)ADVANTAGES:

- Sensors are easily available.
- Simple, compact, easy to handle.
- Sensors have long life and less cost.
- Quality of air can be checked indoor as well as outdoor.
- Detecting a wide range of physical parameters including Carbon mono-oxide, Methane, LPG,NH3 and carbon dioxide.

(ii)APPLICATIONS:

- Indoor air quality monitoring.
- Industrial place’s Air quality monitoring.
- Roadside pollution monitoring.
- To make this data available to common man.

Chapter 15

Conclusion and Future Scope

(i)Conclusion: The smart way to monitor air quality with a low cost but efficient and embedded system is presented in this paper. In the proposed architecture functions of different sensors and their working procedure were discussed. How they work, their functionality, their optimal uses and their data taking procedures and comparison with standard base data’s are also discussed here. The air pollution monitoring system can be tested for monitoring the gas levels on different parts of the country. Our project device showed that it is effective and cheap and with some highly working sensors it can really be a reliable one to everybody and its data’s will be a key to take some necessary steps for the betterment of the society as it will help to identify the affected area so that we can take early steps to reduce damages for the next generation.

(ii)Future Work: Our work can demonstrate vast opportunities to work on the device, It also sent the sensor parameters to the data server using WIFI module. On the app and also on the field using the device that we have worked with. The device can be used any time efficiently in different locations of a city and then research with the achieved data for that particular area in that city. The device can be updated with additional sensors that can sense data from the existence of other gases such as O2 and H2. These gases will provide the condition of the atmosphere and authority can take into further decisions accordingly. The sensors that we have been worked with can also be reset according to most recent time update. An android app can be developed for turning on and off the device can be updated with newer features by implementing necessary codes. In future time, our device can be kept testing for checking whether the sensors still runs properly and give real time data. A webpage can be designed, where we will have more opportunities to add options like related tables, pie chart, diagram that will be implemented by back-end programming(server side) so that those options can be visible to the administrator and user as well. We can monitor Through-out the year on which time the amount of gases are in what level and also the increase and decrease level and rates of the gases.

Chapter-16
References

1. Jamila,M.S., Jamilb,M.A., Mazharc,A., Ikrama,A., Ahmeda, A., Munawa, U. (2015). Smart Environment Monitoring System by employing Wireless Sensor Networks on Vehicles For Pollution Free Smart Cities.

2. Leman,A.M., Omar, A.R., Jung,W., Yusof, M.Z.M. (2010). The development of an industrial air pollution monitoring system for safety and health enhancement and a sustainable work environment using QFD approach .

3. Kularatna,N. (2008). An Environmental Air Pollution Monitoring System Based on the IEEE 1451 Standard for Low Cost Requirements.

4. Al-Ali, A.R. (2010). A Mobile GPRS-Sensors Array for Air Pollution Monitoring.

5. A SMART AIR POLLUTION MONITORING SYSTEM Kennedy Okokpujie, Etinosa Noma-Osaghae, Odusami Modupe, Samuel John and Oluga Oluwatosin Department of Electrical and Information Engineering, Covenant University, Ogun State, Nigeria

6. IoT based Real-time Air Quality Monitoring and Control System to Improve the Health and Safety of Industrial Workers by T.Veeramanikandasamy, Gokul Raj.S, A.Balamurugan, A.P.Ramesh, Y.A.Syed Khadar.

7. https://circuitdigest.com/microcontroller-projects/arduino-mq137-ammonia-sensor

8.Arun Raj V., Priya R.M.P., and Meenakshi, V., "Air Pollution Monitoring In Urban Area," International Journal of Electronics and Communication Engineering , 2017 .

9. Oscar Alvear,Nicola Roberto Zema, Enrico Natalizio,, "Using UAV-Based Systems to Monitor Air Pollution in Areas with Poor Accessibility," Journal of Advanced Transportation, p. 14, 2017.

10.https://www.circuito.io/app?components=512,9590,9591,9591,10168,11061,11372,341099,398782,398787,398790,956215