ENVIRONMENTAL MONITORING SYSTEM AND ANALYSIS

CAPSTONE PROJECT REPORT

Submitted by

V JAI PRASHANTH (14BCE1019)

CH JAYADEEP SHANKAR (14BCE1170)

G SIDDARDHA VARMA (14BCE1174)

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

COMPUTER SCIENCE AND ENGINEERING

School of Computing Science and Engineering

Vandalur - Kelambakkam Road, Chennai - 600 127

APRIL 2018

School of Computing Science and Engineering

DECLARATION

We hereby declare that the project entitled "**Environmental Monitoring and Analysis**" submitted by me to the School of Computing Science and Engineering, VIT Chennai, 600127 in partial fulfillment of the requirements of the award of the degree of B. Tech CSE is a bona-fide record of the work carried out by me under the supervision of **Prof. Dr. Hepsiba Mabel V**. We further declare that the work reported in this project, has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma of this institute or of any other institute or University.

Place: Chennai

Date: Signature of Candidate(s)

V Jai Prashanth	**14BCE1019**
CH Jayadeep Shankar	**14BCE1170**
G Siddardha Varma	**14BCE1174**

School of Computing Science and Engineering

CERTIFICATE

This is to certify that the report entitled **“Environmental Monitoring and Analysis”** is prepared and sub-mitted by **V Jai Prashanth** (Reg. No. 14BCE1019), **CH Jayadeep Shankar Reddy** (Reg. No. 14BCE1170), & **G Siddardha Varma** (Reg. No. 14BCE1174), to VIT Chennai, in partial fulfillment of the requirement for the award of the degree of B.Tech CSE programme is a bona-fide record carried out under my guidance. The project fulfills the requirements as per the regulations of this University and in my opinion meets the necessary standards for submission. The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma and the same is certified.

Guide

Name: Prof. **Dr. Hepsiba Mabel V**

Date:

External Examiner	Internal Examiner	Internal Examiner
Signature:	Signature:	Signature:
Name:	Name:	Name:
Date:	Date:	Date:

(Seal of SCSE)

Acknowledgement

We wish to express our sincere thanks to a number of people without whom we could not have completed the thesis successfully.

We would like to place on record our deep sense of gratitude and thanks to our guide Prof. Dr. Hepsiba Mabel V, School of Computer Science and Engineering (SCSE), VIT, Chennai campus whose esteemed support and immense guidance encour-aged us to complete the project successfully.

We would like to thank our Program Chair Dr. B Rajesh Kanna & Co-Chair Dr. C Sweetlin Hemalatha, and B.Tech. Computer Science and Engineering and Project Coordinator Dr. B V A N S S Prabhakar Rao, VIT Chennai campus, for their valuable support and encouragement to take up and complete the thesis.

Special mention to our Dean to Dr. Vaidehi Vijayakumar, Associate Deans Dr. Vijayakumar V and Dr. Subbulakshmi T, School of Computing Science and Engineering (SCSE), VIT Chennai campus, for spending their valuable time and efforts in sharing their knowledge and for helping us in every minute aspect of software engineering.

We thank our management of VIT, Chennai campus for permitting us to use the library resources. We also thank all the faculty members for giving us the courage and the strength that we needed to complete our goal. This acknowledgment would be incomplete without expressing the whole hearted thanks to our family and friends who motivated us during the course of our work.

V Jai Prashanth 14BCE1019

CH Jayadeep Shankar 14BCE1170

G Siddardha Varma 14BCE1174

Executive Summary

Here we are proposing a solution which is advanced in the field of monitoring the environmental conditions. Internet of Things (IoT), is the technology behind our project which is a modern and efficient solution to connect various IoT devices to the internet.

Our system deals with analyzing and monitoring of environmental conditions which are usually temperature, level of gas, humidity content present in the air. These are sensed with the help of sensors and uploads data to website through IoT module.

Now the data which is obtained from our system is accessible from anywhere through internet. This is done with the help of customized IoT module which have internet connectivity.

Now, the data collected is analyzed using RapidMiner tool which performs an algorithm called Linear Regression. This is used to predict the defective sensor values in case of malfunction with the help of other sensor values and the previous data collected. This will make the system more resilient.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. Introduction

The system proposed here deals with the monitoring of the environmental conditions with the help of various sensors which senses various parameters which are used in environmental monitoring. We are sensing parameters like temperature, gas and humidity using the sensors which can sense these parameters. Here we have used IoT (Internet of Things) to process the functionalities like sensing the values, uploading them to the website. Here the reason for using IoT is, it is modern and the efficient method to work with the internet connected devices. Our system deals with analyzing and monitoring of environmental conditions which are usually temperature, level of gas, humidity content present in the air. These are sensed with the help of sensors and uploads data to website through IoT module. Further the analysis of the sensor data will make the system more resilient.

1.1 Objective

- To remotely sense and gather the data about the environment through various sensors on wireless sensor networks / IoT.
- To monitor and analyse the data from the implemented system.

1.2 Motivation

The current changes in atmosphere have expanded the significance of environmental monitoring, making it a topical and profoundly dynamic research territory. This field depends on remote detecting and on remote sensor systems for gathering environment information of a particular place like temperature, humidity and gas. Latest headways in the Internet of Things (IoT) offer help for the transmission and administration of measures of information in regards to the patterns observed in the environmental changes.

2. Project Description and Goals

The main objective of our system is collecting the environmental data and uploading it into a website at the same time plot the data in an excel sheet. In Order to do this the system should have temperature, humidity and gas sensors to sense the data. The project is seperated into three main subjectives to accomplish the target usefulness:

- Information Sensing
- Uploading the sensed data through IoT module
- Analysis of sensor data

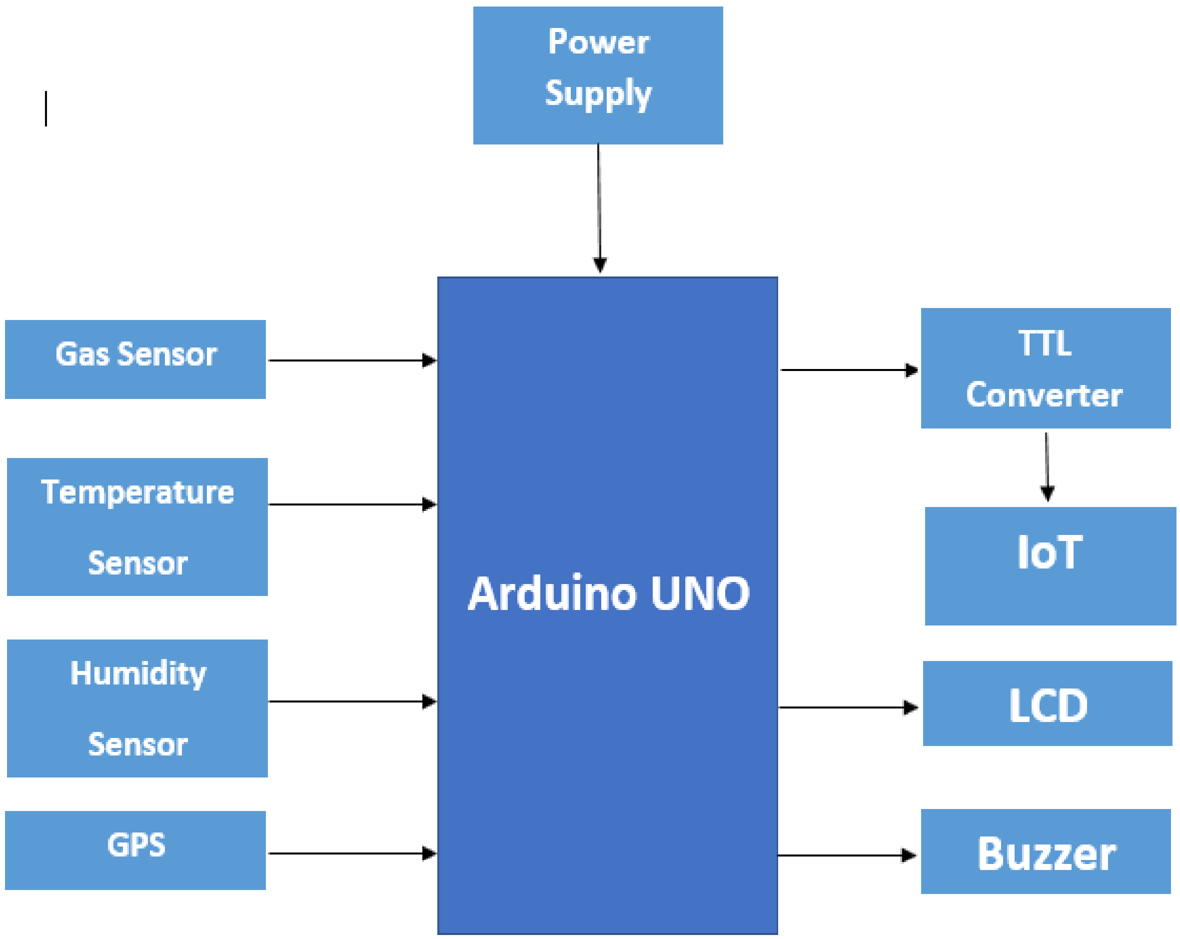

Figure 1 Block Diagram

With the above-mentioned goal in mind the environmental monitoring and analyzing system has the following functions:

I. Sensors in our system sense the different types of parameters like temperature, gas and humidity values.

II. The sensors which are connected to the Arduino UNO will send their respective values to the Arduino UNO.

III. All these sensor values will be displayed on the LCD screen

IV. In case of any abnormality in the sensor values, like if any of the sensors exceeds the threshold limit, the buzzer alerts.

V. The data which is sensed from various sensors will be uploaded to the website using an IoT Module

VI. The data is updated to the excel sheet.

VII. Analysis of the data to be done with the sensor values obtained.

3. Hardware, Software & Technical Specifications

While the general objectives, techniques and destinations have been expressed, the determinations of the segments will be resolved as they are distinguished for their appropriateness in our project.

3.1 Hardware Requirements

- Arduino Uno
- LM 35 – Temperature Sensor
- DHT-11 – Humidity Sensor
- MQ-135 – Air Quality Gas Sensor
- LCD
- Buzzer
- IoT Module with adapter
- GPS Module with adapter
- TTL to Serial Converter

3.2 Software Specification

Embedded C

- Embedded C is a set of language extensions for the C Programming language by the C.

- Standards advisory group to address shared characteristic issues that exist between C expansions for various installed frameworks.

- Historically, embedded C programming language needs nonstandard extensions to the C programming language in order to help with exotic functionalities. For example, fixed-point arithmetic, multiple distinct memory banks, and essential I/O activities.

RapidMiner

- RapidMiner is an information science programming tool created by the organization of a similar name that gives an incorporated domain to information planning, machine adapting, profound learning, content mining, and prescient examination.

- It is utilized for commerical business applications and also to investigate, instruction, preparing, quick prototyping, and application improvement and backings all means of the machine learning process including information readiness, results visualization, optimization and model validation.

- RapidMiner tool was developed on an open center model.

3.3 Technical Specifications

3.3.1 Arduino Uno

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (6 provide PWMoutput)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32KB(ATmega328P)0.5 KB Btldr
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
LED_BUILTIN	13
Length	68.6 mm
Width	53.4 mm
Weight	25 g

Table 1 Arduino UNO Specifications

3.3.2 Temperature Sensor - The LM35

The LM35 sensor is an integrated circuit sensor that can be utilized to quantify temperature with an electrical yield relative to the temperature (in Celsius)

LM35 Temperature Sensor

Accuracy (Max) (+/- Deg C)	0.5
Temperature Range (Deg C)	-40 to 110, -55 to 150, 0 to 100, 0 to 70
Voltage (Minimum) (Volts)	4
Voltage (Maximum) (Volts)	30
Current (Maximum) (u A)	114
Sensor Gain (mv/ C)	10
Features	UL Recognized
Rating	Catalog
Impedance Output (Ohms)	0.4
Interface	Output in Analog

Table 2 LM 35 Specifications

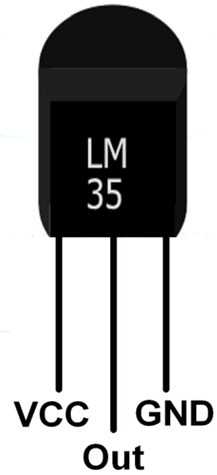

Figure 2 LM 35 Diagram

- What is the necessity of using LM35 – Sensor?
  - You can gauge temperature more precisely than utilizing a thermistor.
  - The sensor hardware is fixed and not subject to oxidation.
  - The LM35 Sensor creates a higher yield voltage than thermo-couples and might not require the yield voltage to be amplified.

- How does the LM35 Sensor Work ?
  - It has a yield voltage that is corresponding to the temperature in Celsius.
  - The scaling factor for LM35 is 0.01 Volts / Deg C.
  - The LM35 does not require any outside alignment or trimming and keeps up a precision of +/ - 0.4 Deg C at the room temperature and +/ - 0.8 Deg C over a scope of 0 Deg C to +100 Deg C.
  - Another crucial functionality for the LM35 is that it will draw just 60 smaller scale amps from its supply and has a low self-warming ability. The sensor self-warming causes under 0.1 Deg C temperature ascend in still air.

The LM35 comes in a wide range of bundles, including the below types.

- TO-92 is a plastic transistor type bundle,
- TO-202 is a bundle. (shown above)
- 8-lead surface shaft SO-8 tiny outline package
- T0-46 is of metal can transistor type bundle

- How to use an LM35 Sensor?
  - This is normally utilized circuit. For electrical connections check the photo above.
  - In this circuit, parameter esteems normally utilized are:
    - V_c = 4 volts to 30 volts
    - 5 volts or 12 volts are common esteems utilized.
    - $R_a = V_c / 10^{-6}$
    - Actually, it can range from 80,000 to 6,00,000 yet most simply utilize 80,000.

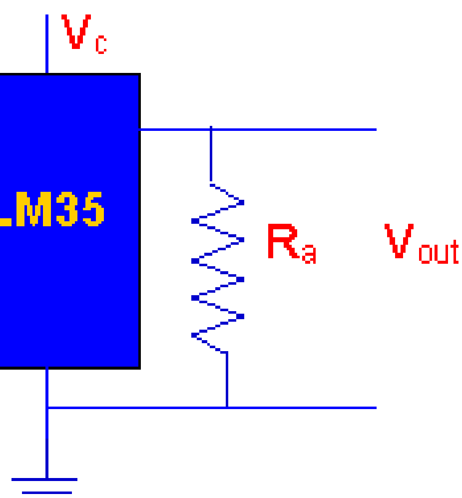

Figure 3 LM 35 Circuit Diagram

3.3.3 Gas sensor

Figure 4 Gas Sensor Diagram

3.3.3.1 Construction

The Gas sensor comprises of two electrodes which are in touch with an electrolyte. The electrodes are commonly manufactured by settling a high surface region valuable metal on to the permeable hydrophobic film. The working electrodes contacts both the electrolyte and the encompassing air to be checked typically by means of a permeable film. The electrolyte most ordinarily utilized is a mineral corrosive to the anodes and lodging are generally as a rule in a plastic lodging which contains a gas passage gap for gas and electrical contacts.

3.3.3.2 Theory of Operation

The gas will be diffused into gas sensor, through the backside of the permeable film to the working terminal where oxidization or reduction will happen. This electrochemical response brings about an electric current that goes through the outside circuit. Including the estimating, opening up and performing other flag handling capacities, the outer circuit keeps up the voltage over the sensor between the working and counter anodes for a two-cathode sensor or between the working and reference terminals for a three-terminal cell. At the counter terminal an equivalent and inverse response happens, to such an extent that if the working cathode is an oxidation, at that point the counter anode is reduced.

3.3.3.3 Specifications

Work Conditions

Variable name	Denoted as	Technical condition	Comment
Voltage of the Circuit	Vc	5 Volts ± 0.1	Alternating / Direct Currents
Heating voltage	V_H	5 Volts ± 0.1	Alternating / Direct Currents
Resistance of Load	R_L	adjustable	
Resistance of the Heater	R_H	33 Ω ± 5%	Optimum Temperature
Heat Uitilization	P_H	< than 800 mw	

Table 3 Gas Sensor Work Conditions

3.3.4 Humidity Sensor

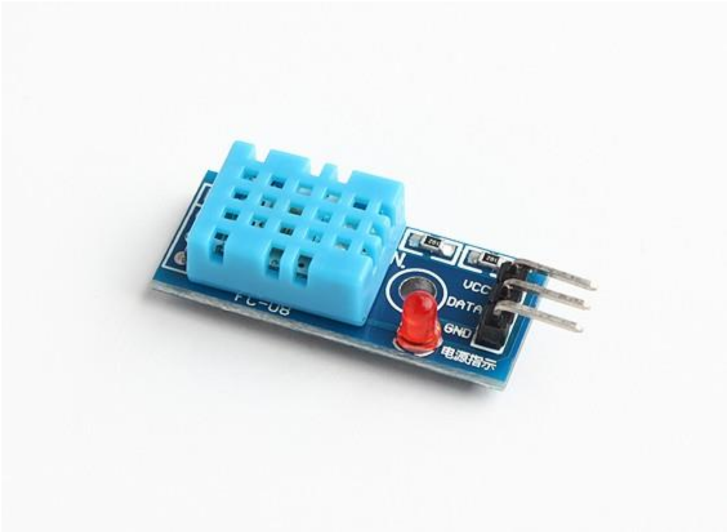

Figure 5 Humidity Sensor Diagram

Work Conditions

Size of the PCB	22.0mm X 20.5mm X 1.6mm
Operating voltage	3.3 or 5V Direct Current
Range of Measurement	20-95% RH 0-50℃
Resolution	8bit（Hum），8bit（Temp）

Table 4 Humidity Sensor Standard Work Conditions

Electrical Functionalities

Variables	Maximum	Minimum	Optimal	Units
Working voltage	5.5	3	5	Volts (Direct Current)
Working current（T=25℃, VCC=5 Volts）	2.5	0.5	-	mA
Interval for sampling	-	1	-	s

Table 5 Humidity Sensor Electrical Characteristics

Humidity Conditions

Parameter	Minimum	Optimal	Maximum	Units
Accuracy (25℃)	-	±4	-	%RH
Accuracy (0-50℃)	-	-	±5	%RH
Measurement range (25℃)	20	-	95	%RH
Responding Time: 2 Deg C, 1/e(63%, 1m/s air	6	10	15	s

Table 6 Humidity Sensor Humidity Conditions

3.3.5 GPS Module

A GPS route gadget, GPS collector, or essentially GPS is a gadget that is fit for getting data from GPS satellites and afterward to figure the gadget's geological position.

Through this GPS Module, location of our system is found in the form of latitude and longitude values.

Parametric Specifications

Parameter	Specification			
Receiver type	50 Channels GPS L1 frequency, C/A Code SBAS: WAAS, EGNOS, MSAS			
Time-To-First-Fix[1]		NEO-6G/Q/T	NEO-6M/V	NEO-6P
	Cold Start[2]	26 s	27 s	32 s
	Warm Start[2]	26 s	27 s	32 s
	Hot Start[2]	1 s	1 s	1 s
	Aided Starts[3]	1 s	<3 s	<3 s
Sensitivity[4]		NEO-6G/Q/T	NEO-6M/V	NEO-6P
	Tracking & Navigation	-162 dBm	-161 dBm	-160 dBm
	Reacquisition[5]	-160 dBm	-160 dBm	-160 dBm
	Cold Start (without aiding)	-148 dBm	-147 dBm	-146 dBm
	Hot Start	-157 dBm	-156 dBm	-155 dBm
Maximum Navigation update rate		NEO-6G/Q/M/T	NEO-6P/V	
		5Hz	1 Hz	
Horizontal position accuracy[6]	GPS	2.5 m		
	SBAS	2.0 m		
	SBAS + PPP[7]	< 1 m (2D, R50)[8]		
	SBAS + PPP[7]	< 2 m (3D, R50)[8]		
Configurable Timepulse frequency range		NEO-6G/Q/M/P/V	NEO-6T	
		0.25 Hz to 1 kHz	0.25 Hz to 10 MHz	
Accuracy for Timepulse signal	RMS	30 ns		
	99%	<60 ns		
	Granularity	21 ns		
	Compensated[9]	15 ns		
Velocity accuracy[6]		0.1m/s		
Heading accuracy[6]		0.5 degrees		
Operational Limits	Dynamics	≤ 4 g		
	Altitude[10]	50,000 m		
	Velocity[10]	500 m/s		

Figure 6 GPS Parametric Specifications

Pin Diagram

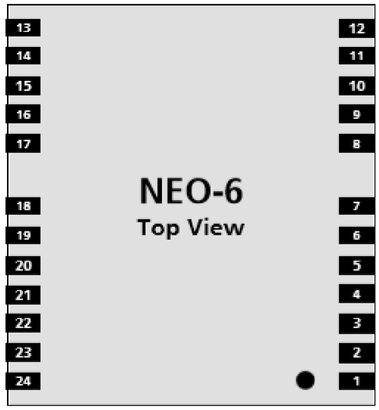

Figure 7 GPS Pin Diagram

Electrical Characteristics

Parameter	Symbol	Module	Min	Max	Units	Condition
Power supply voltage	VCC	NEO-6G	-0.5	2.0	V	
		NEO-6Q, 6M, 6P, 6V, 6T	-0.5	3.6	V	
Backup battery voltage	V_BCKP	All	-0.5	3.6	V	
USB supply voltage	VDDUSB	All	-0.5	3.6	V	
Input pin voltage	Vin	All	-0.5	3.6	V	
	Vin_usb	All	-0.5	VDDUSB	V	
DC current trough any digital I/O pin (except supplies)	Ipin			10	mA	
VCC_RF output current	ICC_RF	All		100	mA	
Input power at RF_IN	Prfin	NEO-6Q, 6M, 6G, 6V, 6T		15	dBm	source impedance = 50Ω, continuous wave
		NEO-6P		-5	dBm	
Storage temperature	Tstg	All	-40	85	°C	

Figure 8 GPS Electrical Characteristics

3.3.6 GSM Module

SIM 800C Module is an entire Quad-band GSM/GPRS arrangement in a SMT compose, which can be implanted in the client applications. These modules are sub-system of the Internet-of-everything hardware. SIM800C underpins Quad-band 850/900/1800/1900MHz, it can transmit Voice, SMS and information data with low power utilization.

SIM800C is a quad-band GSM/GPRS module that takes a shot at frequencies GSM-850MHz, DCS1-800MHz, EGSM-900MHz, and PCS1900MHz. SIM800C highlights GPRS multi-space class 12/class10 (discretionary) and underpins the GPRS coding plans CS-1, CS-2, CS-3 and CS-4.

With a modest design of 17.6 x 15.7 x 2.3 mm, SIM800C can meet all the space necessities in client applications, for example, advanced cell, PDA and other cell phones.

SIM800C is a SMT bundle with 42 cushions and gives all equipment interfaces between the module and custom sheets. One full modem serial ports One USB and One serial port and the USB interface can troubleshoot, download virtual platform, One sound channel which incorporate a speaker yields and receiver input, Programmable normal characteristic I/O useful information and yields, interface of a SIM card.

SIM800C is composed with control sparing strategy so the present utilization is as low as 0.7mA in rest mode.

Pin Diagram

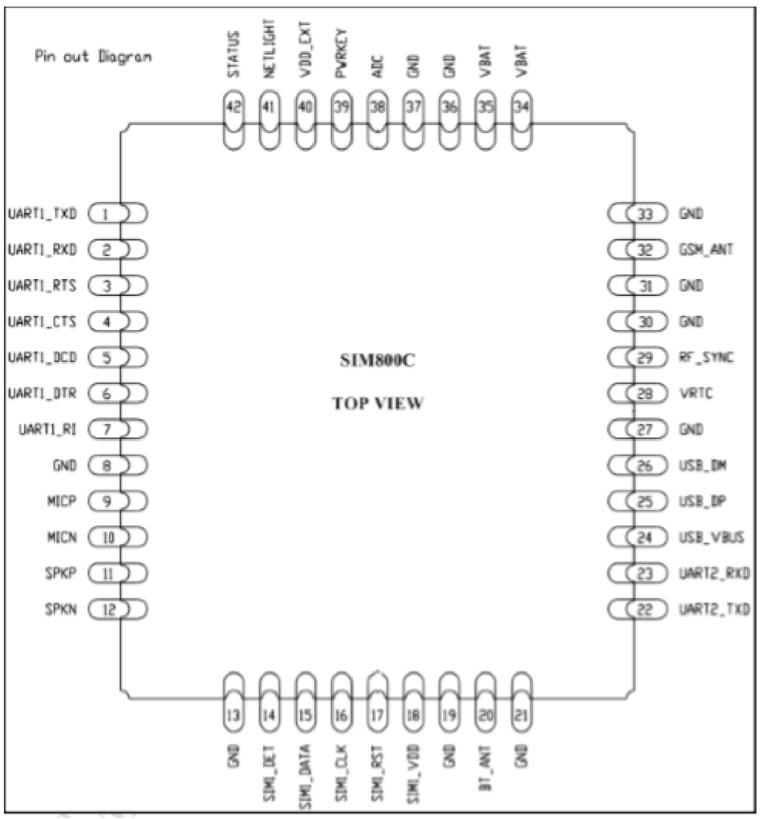

Figure 9 GSM Pin Diagram

Functional Diagram

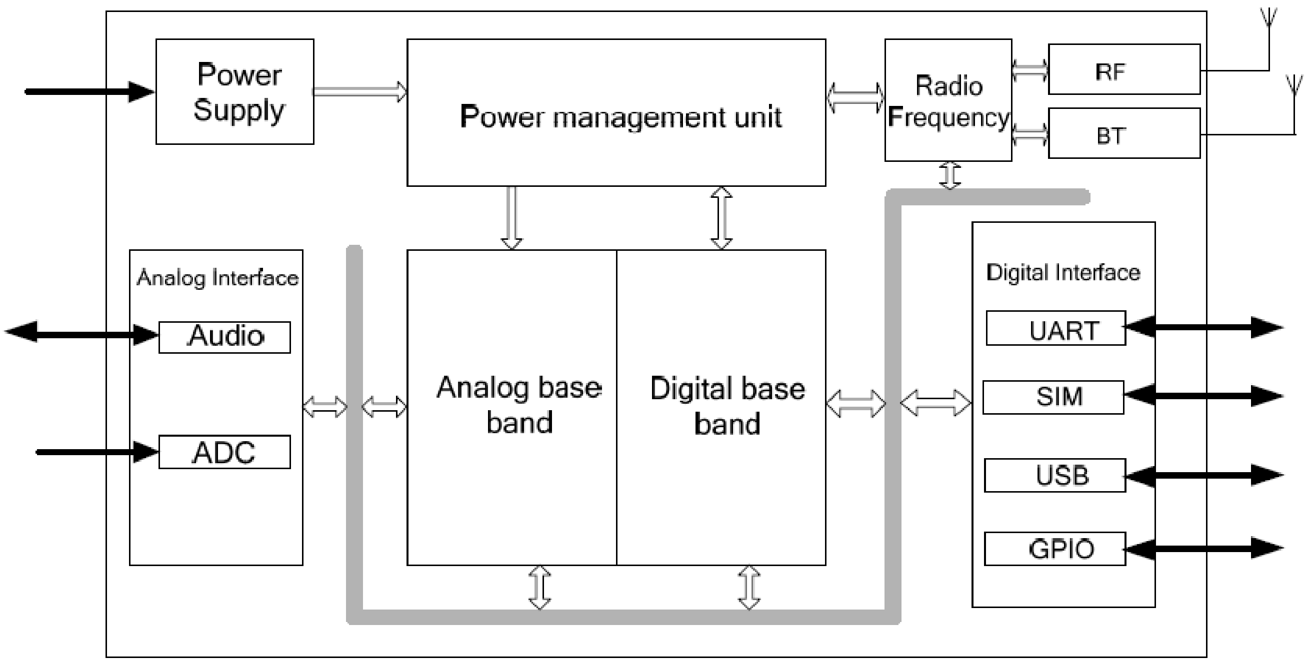

Figure 10 GSM Functional Diagram

3.3.7 LCD

Liquid Crystal Display (LCD) screen is an electronic module and has an extensive variety of uses. A 16x2 LCD screen is exceptionally fundamental module and is regularly utilized as a part of different gadgets and circuits. These modules are favored more than seven fragments and other multi portion LEDs.

The reasons being:

- LCDs are sparing.
- Effectively programmable.
- Have no impediment of showing extraordinary and even custom characters (not at all like in seven portions), livelinesss etc,.

A **16x2** LCD implies it can show 16 characters for every line and there are 2 such lines. In this LCD each character is shown in 5x7 pixel grid. This LCD has two registers, in particular, Command and Data.

The command register stores the command guidelines given to the LCD screen. A command is the direction provided to LCD to complete a predefined instruction like introducing it, clearing its screen, setting the cursor position, controlling showcase and so forth. The information enlist stores the information to be shown on the LCD. The information is the ASCII estimation of the character to be shown on the LCD.

Pin Description

Pin No	Name of the Pin	Characteristic
1	Ground	Ground Voltage (0 Volts)
2	Vcc	Input Voltage; 5 Volts (4.7Volts – 5.3Volts)
3	V_{EE}	Adjustment of the contrast through a variable resistor
4	Register Select	Selects data register when high, Selects command register when low
5	Read/write	High for reading from the register, Low for writing to the register
6	Enable	Sends data to data pins when a high to low pulse is inputed
7	DB0	Data pins (8-bit)
8	DB1	
9	DB2	
10	DB3	
11	DB4	
12	DB5	
13	DB6	
14	DB7	
15	Led+	Backlight V_{CC} (5 Volts)
16	Led-	Backlight Ground (0 Volts)

Table 7 LCD Pin Description

Electrical Characteristics

ITEM	SYMBOL	CONDITION		STANDARD VALUE			UNIT
				MIN.	TYP.	MAX.	
Input Voltage	VDD	VDD = + 5V		4.7	5.0	5.3	V
		VDD = + 3V		2.7	3.0	5.3	V
Supply Current	IDD	VDD = 5V		–	1.2	3.0	mA
Recommended LC Driving Voltage for Normal Temp. Version Module	VDD - V0	- 20 °C		–	–	–	V
		0°C		4.2	4.8	5.1	
		25°C		3.8	4.2	4.6	
		50°C		3.6	4.0	4.4	
		70°C		–	–	–	
LED Forward Voltage	VF	25°C		–	4.2	4.6	V
LED Forward Current	IF	25°C	Array	–	130	260	mA
			Edge	–	20	40	
EL Power Supply Current	IEL	Vel = 110VAC:400Hz		–	–	5.0	mA

Figure 11 LCD Electrical Characteristics

Characteristics

- + 5 Volts power supply (Available for + 3 Volts)
- 1/16 obligation cycle.
- Built-in controller (KS 0066 or Equivalent)
- B/L to be operated by pin 1, pin 2 or pin 15, pin 16 or A.K (LED)
- N.V. optional for + 3 Volts power supply. 5 x 8 dots with cursor

Max. Rating

ITEM	SYMBOL	STANDARD VALUE			UNIT
		MIN.	TYP.	MAX.	
Power Supply	VDD-VSS	- 0.3	–	7.0	V
Input Voltage	VI	- 0.3	–	VDD	V

Figure 12 LCD Absolute Rating

3.8 TTL to serial Converter

Figure 13 TTL to Serial Converter Diagram

This is a MAX3232-based level shifter for changing over TTL-level serial to standard RS-232 serial. This will permit a microcontroller to speak with a PC and broadly utilized as a part of radio adjustment, telephone streak, XBOX360 streak, GPS, vehicle location, DVD streak, hard plate repair set-top box redesign.

Features

- Pre-assembled compact size board
- Suitable for use with most microcontrollers
- Has operating range up to 250 kilobit / second
- Low input current 300 µA normally
- Working Voltage: 3.3V - 5V
- Standard Header Connector (TTL)
- Communication chip: MAX3232
- DB9 Female Connector (RS-232)
- Interface: 4 pin connectors – GND
- Logic output (STX), +5v, Logic Input (SRX)
- Package size: 33mm x 32mm x 14mm

4. Design Approach and Details

4.1 Design Approach

Location Sensor

Our Environmental Monitoring System uses a GPS (Global Positioning System) Sensor which is used to determine its location in the form of Latitude and Longitude values. This latitude and longitude values are transmitted to the website through which we can monitor the particular location of that system. This can also be useful if there are multiple monitoring systems.

Environmental Sensing

Here we are using different sensors like LM-35 temperature sensor which has the range of -55° C to 150° C to sense the surrounding temperature, MQ-135 sensor which is also called as air quality sensor which can detect gases like NH_3, NO_2, smoke, Carbon Dioxide and other harmful gases and DHT Humidity sensor is used to sense the humidity in the air.

Data Transmission

The data transmission is done with the help of IoT module which is integrated with GSM Module and serial port. As the arduino has data pins and the IoT module has a serial port, so in order to communicate with each other arduino pins are connected to the TTL to serial converter andthe output of the serial converter is connected to the serial port of the IoT module. Here we need aworking SIM with internet connectivity in order to transmit the data from the IoT module to the website.

Interfacing

Here Arduino UNO is used which is the central processing unit which has all the sensors are Connected to it and the threshold is set in the arduino programming according to the requirement. Here, the arduino programming is done in Arduino IDE compiler. The IoT Module here used consists an Integrated GSM Module which uses a working SIM with internet connectivity. Here the sensors which are connected to the arduino senses the surroundings and the data which is sensed is transmitted from arduino to the GSM of the IoT module through a serial to TTL converter, here the data received to the arduino is given from its TTL pins of the conveter which is connected to serial port of the IoT module. Now the data is transmitted to the IoT module in which the integrated GSM module transmits the data to the website with the help of SIM connected to it.

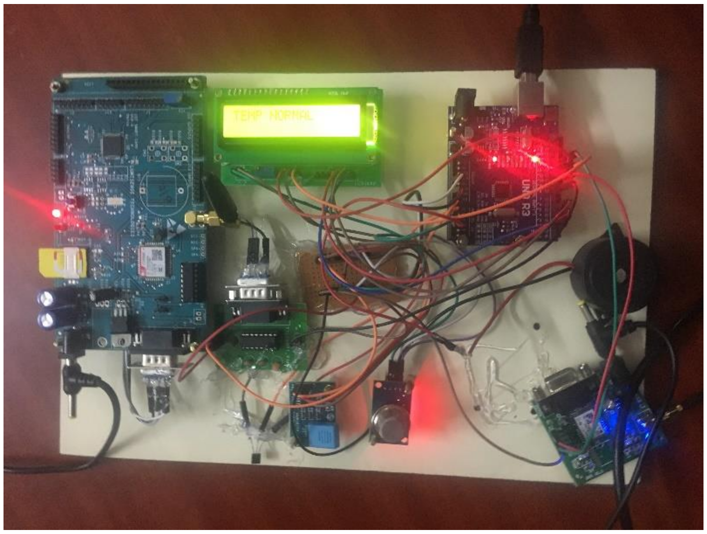

Figure 14 Project Interfacing 1

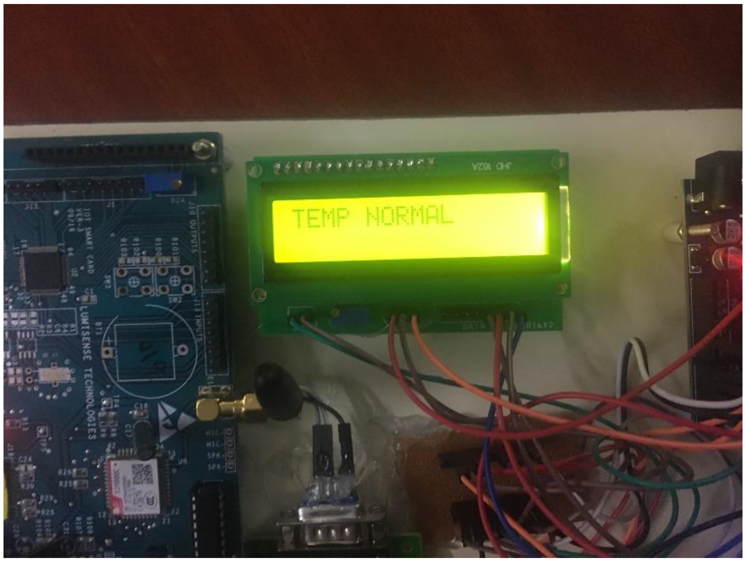

Figure 15 Project Interfacing 2

Analysis

With the help of this analysis we can understand the patterns behind the sensor data of the environment through which we can predict the defective sensor values in case of malfunction with the help of other sensor values and the previous data collected. This will make the system more resilient.

Analysis Technique

Here the technique used for the analysis of the data is Regression Technique. This Regression Technique is basically a predicition technique normally used to demonstrate the dependency between one or more independent or predictor variables and a dependent or response variable. Regression analysis used here is linear and multiple linear regression.

- **Linear regression**

- The Linear regression technique endeavors to show the connection between two factors by fitting a linear equation to the gathered information. One variable is thought to be an informative variable, and the other is thought to be a dependent variable. For instance, a modeler should need to relate the weights of people to their statures utilizing a straight regression model.

- Before we try to fit a linear model to the gathered data, the modeler shall first determine whether or not there is a relationship between the variables of interest. This does not necessarily imply that one variable causes the other (for example, higher SAT scores do not cause higher college grades), but that there is some significant association between the two variables. A scatterplot can be a helpful tool in determining the strength of the relationship between two variables. If there appears to be no association between the proposed informative and dependent variables (i.e., the scatterplot does not indicate any increasing or decreasing trends), then fitting a linear regression model to the data probably will not provide a useful model. A valuable numerical measure of association between two variables is the correlation coefficient, which is a value between -1 and 1 indicating the strength of the association of the observed data for the two variables.

- The linear regression line consists an equation in the form $Y = a + bX$, where X is the informative variable and Y is the dependent variable. The slope of the line is b, and a is the intercept (the value of y when $x = 0$).

➢ **Method of least squares**

- The most basic strategy for fitting a regression line is the technique for least squares. This strategy figures the best-fitting line for the gathered information by limiting the total of the squares of the vertical deviations from every data point to the line (if a point lies on the fitted line precisely, at that point its vertical deviation is 0). Since the deviations are first squared, at that point then added, there are no cancelations amongst positive and negative values.

- Estimation of the Best-Fitting Line:

$$w_1 = \frac{\sum_{i=1}^{|D|}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{|D|}(x_i - \bar{x})^2}$$

Eqn 1 Regression 1

$$w_0 = \bar{y} - w_1 \bar{x}$$

Eqn 2 Regression 2

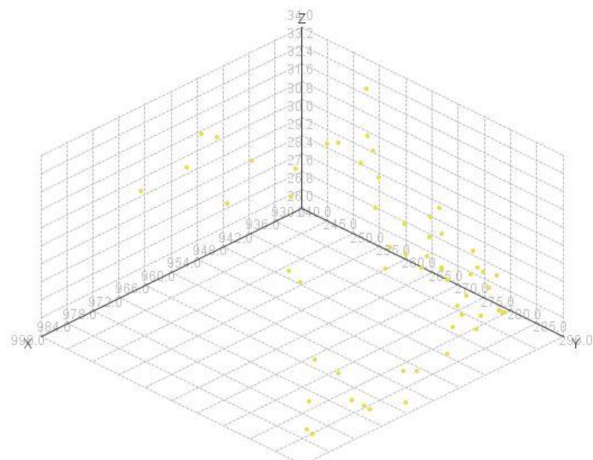

Figure 16 Data before prediction

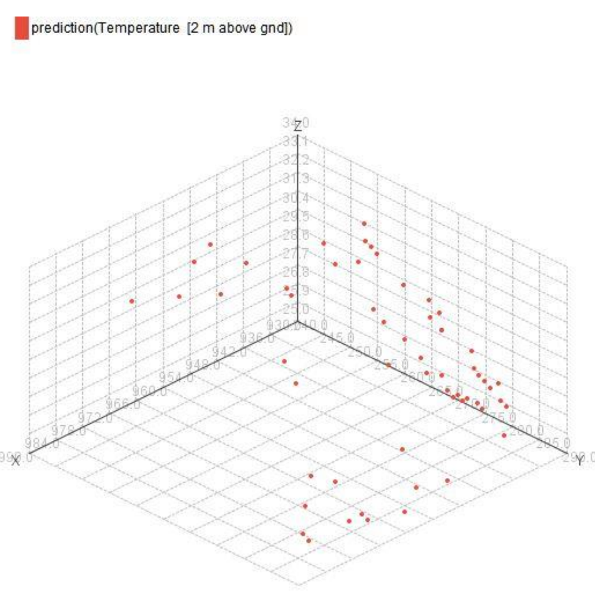

Figure 17 Data after prediction

4.2 Codes and Standards

IEEE 802.15.4 (IoT Module)

IEEE 802.15.4 is a specialized standard which characterizes the activity of low-rate remote personal area systems (LR-WPANs). It indicates the physical layer and media get to control for LR-WPAN's and is kept up by the IEEE 802.15 working gathering, which characterized the standard in 2003. It is the reason for the ZigBee WirelessHART, MiWi, SNAP, and Thread particulars, every one of which additionally expands the standard by building up the upper layers which are not characterized in IEEE 802.15.4. On the other hand, it can be utilized with 6LoWPAN, the innovation used to convey the IPv6 adaptation of the Internet Protocol (IP) over WPANs, to characterize the upper layers.

4.3 Constraints, Alternatives and Tradeoffs

- Thermistor can also be used to measure the temperature but temperature can be measured more accurately using LM 35 than using a thermistor. LM-35 sensor hardware is fixed and not subjected to oxidation or any common phenomenons. The LM35 produces a higher yield voltage than thermocouples and may not require any amplification of the yielded voltage. LM35 is operated from a 5V supply.

- The MQ-135 sensor is air quality sensor for the detection of harmful gases that are present in the air. The MQ-135 gas sensor layer of the sensor unit is made up of tin dioxide (SnO_2); it has lower conductivity compare to clean air and due to air pollution, the conductivity increases. This gas sensor is more suitable for our system than other gas sensors like MQ 2 gas sensor which is for detecting smoke and gas.

- Arduino is used as the processor for the project. It comparatively inexpensive compared to other microcontrollers. It allows you to connect complex circuitry like RF or Bluetooth (Transceivers) in terms of simple shields which derive their power from the Arduino board. Furthermore, the coding is much easier as you get to code in C instead of Assembly and also use existing libraries to easily interface between the shields. For example, there are libraries for I2C Serial communication, LCD interfacing, etc.

- For building the prototype products Arduino is best option. For prototyping, the Arduino platform gives you a lot of pre-wiring and free code libraries that will let you concentrate on testing your idea instead of spending your time building supporting circuitry or writing tons of low level code.

5. Schedule, Tasks and Milestones

There are five noteworthy milestones and also a few littler undertakings that must be accomplished with a specific end goal to achieve the milestones.

These five major milestones are:

- Design & Simulation
- Components interfacing
- Coding
- Implementation and uploading data to website
- Applying Data Analytics & result

Tasks were part up among individuals as indicated by every individual's level of skill or expertise. Each task has a pioneer who is in charge of fulfillment of that particular task. Nonetheless, other gathering individuals are relied upon to give help if necessary. Along these lines, a designer can focus on a task yet at the same time get help if necessary. Likewise, each task pioneer is considered responsible if the task comes up short. The group members communicate and explain topics to each other and complete the project.

Tasks Scheduling

Tasks	Starting date	Completion Date	Description
Project Brainstorming	28th November, 2017	3rd December,	Research on selective Project Ideas
Project title submission (Review 1)	4th December, 2017	-	Title Submission
Project Approach	5th December, 2017	13th December, 2017	Research on the methodology of the project.
Module Separation and Assignment	14th December, 2017	21st December, 2017	Module Separation based on the individual's interest
Component selection based on the project requirement	22nd December,2017	-	Finalizing the components
Coding	3rd January, 2018	7th January, 2018	Developing arduino code
Rough Simulation of the project using Proteus	8th January , 2018	12th January, 2018	Feasibility check of the project .
Interfacing sensors to Arduino	13th January , 2018	20th January, 2018	Sensor connection to the arduino
Review Preparation	21nd January, 2018	22nd January 2018	PPT preparation
Review 2	24th January, 2018	24th January, 2018	
Reviewing Feedback	25th January, 2018	31st January, 2018	Working on given feedback and making sufficient changes

Interfacing Arduino and IoT Module	1st February, 2018	5th February, 2018	Interfacing with TTL to serial Converter
Understanding about location information	6th February, 2018	8th February, 2018	Working with GPS Module
LCD Module Connection	9th February, 2018	13th February, 2018	Interfacing LCD Module to arduino
Working on the Research on how to store sensor data in cloud	14th February, 2018	20th February, 2018	Interfacing arduino with IoT and uploading data to cloud
Literature Survey on Data Mining	21st February, 2018	25th February,2018	Refering Research Papers
Searching for suitable Data Mining Technique	26th February, 2018	4th March, 2018	Studying Algorithms
Review 3	6th March, 2018	-	
Understanding Regression Technique	7th March, 2018	9th March, 2018	Understanding Algorithm
Reviewing Feedback	10th March, 2018	14th March, 2018	Working on sufficient changes
Interfacing All Section	15th March, 2018	17th March, 2018	Checking all hardware connections and gathering data
Understanding and implementing the Data Analysis	18th March, 2018	24th March, 2018	Analyzing sensor values
Making Final Report	25th March, 2018	28th March, 2018	-

Table 8 Task Scheduling

6. Project Demonstration

- The important specification of the project is monitoring environmental factors remotely from any place where you get internet. The initial part demonstration of the project is hardware. We will insert a working sim card with internet connectivity in the sim slot in the IOT module and then connect the adapters and once the power is supplied the GPS will be on and also all the sensor will be active and will sense the values of temperature, gas and humidity which will be stored as well as displayed on LCD screen and also transmitted to the website simultaneously. For each sensor we are setting threshold values through our Arduino coding and buzzer will be on in case that sensor values are above threshold.

- The secondary part of demonstration of project is showing the sensor values which are updated in our website which can be accessed from anywhere with internet connection.

- The last part of demonstration is the output of analysis of sensor values which we stored in excel sheet parallel while sending to IoT. Few techniques are applied on it and the analysis result is obtained.

7. Cost Analysis

S.NO	COMPONENTS	COST (Rs/-)
1	Arduino UNO	550
2	LM35 Temperature sensor	70
3	MQ135 gas sensor	165
4	Humidity sensor	125
5	LCD	150
6	IOT Module with adapter	2500
7	GPS with adapter	930
8	Buzzer	15
9	TTL to Serial Converter	250
Total		**4755/-**

Table 9 Cost Analysis

8. Summary

The Environmental monitoring system has been experimentally proven to work satisfactorily by connecting all the components and was successfully monitored from a wireless device. Project was successfully tested on different environments, thus proving its portability and wide compatibility. Thus, a low-cost Environmental monitoring system was successfully designed, implemented and tested.

9. References

[1]. Krishnamurthi, K., Thapa, S., Kothari, L., & Prakash, A. (2015). Arduino based weather monitoring system. *International Journal of Engineering Research and General Science*, *3*(2), 452-458.

[2]. Poonam, Y. M., & Mulge, Y. (2013). Remote temperature monitoring using LM35 sensor and intimate android user via C2DM service. *International Journal of Computer Science and Mobile Computing*, *2*(6), 32-36.

[3]. David, N., Chima, A., Ugochukwu, A., & Obinna, E. (2015). Design of a home automation system using arduino. *International Journal of Scientific & Engineering Research*, *6*(6), 795-801.

[4]. Chaudhryl, V. (2013). Arduair: air quality monitoring. *International Journal of Environmental Engineering and Management*, *4*(6), 639-646.

[5]. Pham, H. D., Drieberg, M., & Nguyen, C. C. (2013, December). Development of vehicle tracking system using GPS and GSM modem. In *Open Systems (ICOS), 2013 IEEE Conference on* (pp. 89-94). IEEE.

[6]. Costanzo, A. (2013, October). An arduino based system provided with GPS/GPRS shield for real time monitoring of traffic flows. In *Application of Information and Communication Technologies (AICT), 2013 7th International Conference on* (pp. 1-5). IEEE.

[7]. Ganchev, I., Ji, Z., & O'Droma, M. (2016, October). Designing a low-cost data transfer unit for use in IoT applications. In *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2016 8th International Congress on* (pp. 85-88). IEEE.

[8]. Hong-De, D., Yang, L., & Liang, W. (2012, December). The data transmission system design between intelligent multi-serial port and ethernet based on arm and fpga. In *Instrumentation, Measurement, Computer, Communication and Control (IMCCC), 2012 Second International Conference on* (pp. 172-175). IEEE.

[9]. Kui, Z., Jianhua, W., Jianguo, J., & Huaping, L. Based on serial port communication meteorology information receiving system. *Dmitriy Zhukov*, 409.

10. Appendix

Code

```
#include <SoftwareSerial.h>
#define temp A0
#define hum A1
#define gas A2
#define buz A3

char str[70];
String gpsString = "";

char *test = "$GPGGA";
String latitude = "No Range      ";
String longitude = "No Range     ";
int i;
boolean gps_status = 0;
```

```
#include <LiquidCrystal.h>
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
SoftwareSerial gps(10, 13); // RX, TX
SoftwareSerial iot(7, 8); // RX, TX
void setup() {
  Serial.begin(9600);
  iot.begin(9600);
  gps.begin(9600);
  lcd.begin(16, 2);
  Serial.println("CLEAR DATA");
  Serial.println("LABEL,Computer Time,TEMP,HUM,GAS");
  pinMode(temp, INPUT);
  pinMode(hum, INPUT);
  pinMode(gas, INPUT);
  pinMode(buz, OUTPUT);
  lcd.clear();
  lcd.setCursor(1, 0);
  lcd.print("ENVIRONMENT");
  lcd.setCursor(3, 1);
  lcd.print("MONITOR");
  delay(1000);
  lcd.clear();
  digitalWrite(buz, LOW);
}
```

```
void loop() {
  float voltage = analogRead(temp);
  voltage = voltage * 5.0;
  voltage / 1024.0;
  float temperatureC = (voltage * 50 / 1024)  ;

  //sensorValue = (sensorValue * 500) / 1023;
  float volt = temperatureC + 10;

  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("TEMP:");
  lcd.setCursor(6, 0);
  lcd.print(volt);
  delay(2000);



  int humval = analogRead(hum);
  float humvalue = humval;
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("HUM:");
  lcd.setCursor(6, 0);
  lcd.print(humvalue);
  delay(2000);
```

```
int gasval = analogRead(gas);
int gasvalue = gasval;

lcd.clear();
lcd.setCursor(0, 0);
lcd.print("GAS");
lcd.setCursor(6, 0);
lcd.print(gasvalue);
delay(1000);

get_gps();

lcd.clear();
lcd.setCursor(0, 0);
lcd.print("LATI ");
lcd.print(latitude);
lcd.setCursor(0, 1);
lcd.print("LONG ");
lcd.print(longitude);
delay(1000);
Serial.print("DATA,TIME,");
Serial.print(volt);
Serial.print(",");
Serial.print(humvalue);
Serial.print(",");
Serial.println(gasvalue);
```

```
  delay(2000);



iot.print("*");delay(1000);
iot.print("L");delay(1000);
iot.print("T");delay(1000);
iot.print(" ");delay(1000);
iot.print("L");delay(1000);
iot.print("G");delay(1000);
iot.print(latitude);delay(1000);
iot.print("#");delay(1000);

iot.print("*");delay(1000);
//iot.print("L");delay(1000);
//iot.print("G");delay(1000);
//iot.print("N");delay(1000);
//iot.print(" ");delay(1000);
iot.print(longitude);delay(1000);
iot.print("#");delay(1000);




  if (volt > 40)
  {
```

```
  digitalWrite(buz, HIGH);
  delay(2000);
  digitalWrite(buz, LOW);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("TEMP HIGH"); delay(1000);
  iot.print("*"); delay(1000);
  iot.print("T"); delay(1000);
  iot.print("E"); delay(1000);
  iot.print("M"); delay(1000);
  iot.print("P"); delay(1000);
  iot.print(" "); delay(1000);
  iot.print("H"); delay(1000);
  iot.print("I"); delay(1000);
  iot.print("G"); delay(1000);
  iot.print("H"); delay(1000);
  iot.print(" "); delay(1000);
  iot.print(volt);delay(1000);
  iot.print("#"); delay(1000);
}
else if (volt < 40)
{
  digitalWrite(buz, LOW);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("TEMP NORMAL"); delay(1000);
```

```
  iot.print("*"); delay(1000);
  iot.print("T"); delay(1000);
  iot.print("E"); delay(1000);
  iot.print("M"); delay(1000);
  iot.print("P"); delay(1000);
  iot.print(" "); delay(1000);
  iot.print("N"); delay(1000);
  iot.print("R"); delay(1000);
  iot.print("M"); delay(1000);
  iot.print("L"); delay(1000);
  iot.print(" "); delay(1000);
  iot.print(volt);delay(1000);
  iot.print("#"); delay(1000);

}

if (humvalue > 200)
{
  digitalWrite(buz, HIGH);
  delay(2000);
  digitalWrite(buz, LOW);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("HUM LOW"); delay(1000);
   iot.print("*"); delay(1000);
  iot.print("H"); delay(1000);
```

```
  iot.print("U"); delay(1000);
  iot.print("M"); delay(1000);
  iot.print(" "); delay(1000);
  iot.print("L"); delay(1000);
  iot.print("O"); delay(1000);
  iot.print("W"); delay(1000);
  iot.print(" "); delay(1000);
  iot.print(humvalue); delay(1000);
  iot.print("#"); delay(1000);
 }

else if (humvalue <200)
 {
  digitalWrite(buz, LOW);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("HUM NORMAL"); delay(1000);

  iot.print("*"); delay(1000);
  iot.print("H"); delay(1000);
  iot.print("U"); delay(1000);
  iot.print("M"); delay(1000);
  iot.print(" "); delay(1000);
  iot.print("N"); delay(1000);
  iot.print("R"); delay(1000);
  iot.print("M"); delay(1000);
```

```
  iot.print("L"); delay(1000);
  iot.print(" "); delay(1000);
  iot.print(humvalue); delay(1000);
  iot.print("#"); delay(1000);

}
if (gasvalue < 900)
{
  digitalWrite(buz, HIGH);
  delay(2000);
  digitalWrite(buz, LOW);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("GAS DETECTED"); delay(1000);
  iot.print("*"); delay(1000);
  iot.print("G"); delay(1000);
  iot.print("A"); delay(1000);
  iot.print("S"); delay(1000);
  iot.print(" "); delay(1000);
  iot.print("D"); delay(1000);
  iot.print("E"); delay(1000);
  iot.print("T"); delay(1000);
  iot.print("E"); delay(1000);
  iot.print("C"); delay(1000);
  iot.print("T"); delay(1000);
  iot.print(" "); delay(1000);
```

```
    iot.print(gasvalue); delay(1000);
    iot.print("#"); delay(1000);

  }

if (gasvalue >900)
  {
     digitalWrite(buz, LOW);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("NO GAS"); delay(1000);
    iot.print("*"); delay(1000);
    iot.print("N"); delay(1000);
    iot.print("O"); delay(1000);
    iot.print(""); delay(1000);
    iot.print("G"); delay(1000);
    iot.print("A"); delay(1000);
    iot.print("S"); delay(1000);
    iot.print(" "); delay(1000);
    iot.print(gasvalue); delay(1000);
    iot.print("#"); delay(1000);

  }
}
```

```
void gpsEvent()
{
  gpsString = "";
  while (1)
  {
    while (gps.available() > 0)
    {
      char inChar = (char)gps.read();
      gpsString += inChar;
      i++;
      if (i < 7)
      {
        if (gpsString[i - 1] != test[i - 1])
        {
          i = 0;
          gpsString = "";
        }
      }
      if (inChar == '\r')
      {
        if (i > 65)
        {
          gps_status = 1;
          break;
        }
```

```
        else
        {
          i = 0;
        }
      }
    }
    if (gps_status)
      break;
  }
}


void get_gps()
{
  gps_status = 0;
  int x = 0;
  while (gps_status == 0)
  {
    gpsEvent();
    int str_lenth = i;
    latitude = "";
    longitude = "";
    int comma = 0;
    while (x < str_lenth)
    {
      if (gpsString[x] == ',')
```

```
        comma++;
      if (comma == 2)
        latitude += gpsString[x + 1];
      else if (comma == 4)
        longitude += gpsString[x + 1];
      x++;
    }
    int l1 = latitude.length();
    latitude[l1 - 1] = ' ';
    l1 = longitude.length();
    longitude[l1 - 1] = ' ';
    i = 0; x = 0;
    str_lenth = 0;
    delay(2000);
  }
}
```