A REPORT

ON

DEVELOPING SOFTWARE TO ASSESS THE STATUS OF MINE FIRES AND THE EXPLOSIBILITY OF GAS MIXTURES IN SEALED OFF AREAS IN MINES

BY

Varghese Mannampalli 2016B4A70176G

AT

CSIR - CENTRAL INSTITUTE OF MINING AND FUEL RESEARCH, DHANBAD

A PRACTICE SCHOOL–I STATION OF

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI, K.K. BIRLA GOA CAMPUS

(July, 2018)

A REPORT

ON

DEVELOPING SOFTWARE TO ASSESS THE STATUS OF MINE FIRES AND THE EXPLOSIBILITY OF GAS MIXTURES IN SEALED OFF AREAS IN MINES

BY

Varghese Mannampalli	2016B4A70176G	MSc. (Hons) Mathematics B.E. (Hons) Computer Science

Prepared in partial fulfilment of the
Practice School-1 Course

AT

CSIR - CENTRAL INSTITUTE OF MINING AND FUEL RESEARCH, DHANBAD
A PRACTICE SCHOOL–I STATION OF

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI, K.K. BIRLA GOA CAMPUS

(July, 2018)

Acknowledgements

First, I would like to express gratitude towards Dr. Pradeep KR Singh, the Director of CIMFR Dhanbad for allowing me to use the resources of this institution to complete this project.

I would also like to thank Dr. Ishtiaque Ahmed who was in charge of the Practice School programme at CIMFR for his assistance and guidance.

This project could not have been undertaken without the guidance of Dr S.K. Ray, Scientist of the Mine Ventilation Department of CIMFR Dhanbad and his assistance in choosing an area to work on.

Finally, I would like to thank Dr. Debasis Das of BITS Pilani K.K Birla Goa Campus for his mentorship during the project.

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI, K.K. BIRLA GOA CAMPUS
Practice School Division

Station: CSIR-Central Institute of Mining and Fuel Research **Centre:** Dhanbad

Duration: 50 days **Date of Start:** 22 May 2018

Date of Submission: 10 July 2018

Title of the Project: Developing Software to Assess the Status of Mine Fires and the Explosibility of Gas Mixtures in Sealed Off Areas in Mines

Name: Varghese Mannampalli **ID Number:** 2016B4A70176G

Disciplines: Msc.(Hons) Mathematics and B.E.(Hons) Computer Science

Name of the Guide: Dr. S.K. Ray

Designation of the Guide: Principal Scientist

Name of the PS Faculty: Dr. Debasis Das

Key Words: Mine Fires, Explosibility, Sealed Off Area, Ellicott's Extension, Linear Regression

Project Areas: Mining Environment Engineering, Safety Engineering, Website Design

Abstract: Mine fires and explosions take many lives and cause much damage to property every year. As such a proper understanding of the mine atmosphere during and after a fire is essential. However, current methods of analysing sealed off areas in mines involve slow, cumbersome manual processes and are prone to error. The author of this report has developed software to automate both the data collection and data analysis stages for analysing mine fires and explosions. The process of collection of gas concentration data from underground areas and conversion of this data to a suitable format has been automated. A graphical interface has also been provided by which the user can see the the status of the fire and the explosibility of the gas mixture in the sealed off region. Other tools provided include a database for long term data analysis, graphical representations of explosibility trends and a prediction tool for gauging explosibility of the area for the near future. This software has a simple interface and is capable of use by a layman.

Signature of Student
Date

Signature of PS Faculty
Date

Table of Contents

1. Introduction

Mine fires and mine gas explosions are some of the most dangerous hazards that miners face. These regularly take lives and cause widespread damage to property. Hence, when a fire is encountered in a section of an underground mine, the area is sealed off and monitored for a period of time. An accurate assessment of the strength of a mine fire can help mine operators take the best steps to put out the fire and can help them decide when to remove the seal on the mine section.

Mine fires also often lead to gas explosions and coal dust explosions. These explosions are very dangerous as they can directly threaten the life of miners and can cave in sections of the mine and hamper rescue work. Therefore, it is essential that the explosibility of the gas mixture in the sealed off area and explosibility trends of the gas mixture over time are monitored.

The current system for monitoring underground fires requires professionals who are well versed in mining engineering to be present at the site for an extended period of time. The current process is for an operator to measure gas levels on site and then manually transfer the data to an external skilled person. Various fire ratios are calculated using the data. The data is then used to graph a Coward's explosibility diagram and then the data points are plotted onto the graph manually to check for explosibility. Both the data transfer process and the graphing process do not have sufficient accuracy for a process that lives depend upon.

The author of this report has created software that will automate this process from the data collection to the data analysis stage. At the data collection level, a program has been written for a microcontroller which collects data from gas sensors placed in a sensor box. A script has been written to collect the data outputted from the microcontroller and store it in a file. The program then calculates various fire ratios and outputs the status of the fire. It also indicates the explosibility of the gas mixture. At the data analysis level, several tools have been provided to the end user of the application. The first is an implementation of Ellicott's Extension of the Coward Explosibility diagram which allows the user to see a graphical trend of the explosibility of the gas mixture over time. The software also provides an analytical tool which uses linear regression algorithms to predict the explosibility of the gas mixture in the future by analysing previous gas content patterns. Finally, a database of gas levels and explosibility has been provided which the user can download and analyse using graphing software.

Since the objective of the software was to present an interface that a layman could use, a graphical interface has been created in two forms, a standalone application and a website. This will allow the checking of the status of a mine fire to be speedy and error free. It will also require very little prior knowledge on the part of the user. In its final stage the software allows the end user to simply place gas sensors in the sealed off area of the mine and then analyse the data that the sensors collect.

2. Literature Review

Since the objective of this paper is to give ways of estimating fire strength and explosibility in sealed off areas it is essential to understand the underlying terms and mathematical models.

2.1 Sealed off Areas

Underground mines contain areas which are sealed off after mining operations have been concluded in the area. These areas sometime have high combustible gas content due to the leaking of methane and other gases from the coal seam. Hence, these areas need to be periodically checked for gas levels to determine whether fires have started. When a mine fire breaks out these areas are maintained at a high pressure by the injection of inert gases such as nitrogen and carbon dioxide. Hence, gas only leaks out from these regions and never into them.

Combustible gases such as carbon monoxide, hydrogen, methane and hydrogen sulphide are often present in underground mines. These, along with coal dust, are often the fuel for mine fires. Mine fires also evolve these gases and others such as carbon dioxide. Several algorithms, referred to as gas ratios, are used to determine how strong fires in sealed off areas in mines are. The software developed by this author uses five of these gas ratios and the logic behind these ratios has been detailed.

2.2 Graham’s Ratio

The strongest fires involve rapid evolution of carbon monoxide gas due to the excess availability of combustibles. Strong fires also display a rapid decrease in the concentration of oxygen gas which is consumed by combustion. The numerator of Graham’s Ratio gauges the carbon monoxide concentration in the area. The denominator measures the oxygen deficiency by calculating the difference between initial and final oxygen levels.

$$GR = \frac{100 \times CO_f}{0.265 \times N_{2f} - O_{2f}}$$

Different values of Graham’s ratio indicate various states of a mine fire:

- 0.4 or less indicates that the situation is normal and that no heating can be observed

- 0.5 is slightly greater than normal and indicates that a check-up is necessary
- 1 indicates that the heating exists in the area
- 2 indicates that extensive heating is occuring which can lead to a fire
- 3 and above indicates that a blazing fire is present

2.3 Young's Ratio

As a fire progresses from a smoldering one to a stronger flame the carbon dioxide percentage in the fire increases. Hence, in a similar way to Graham's Ratio which uses carbon monoxide, Young's ratio uses carbon dioxide in the numerator to check fire strength.

$$Young's\ Ratio\ =\ \frac{\triangle CO_2\%}{\triangle O_2\%}$$

- When the ratio is less than 25 this indicates superficial heating
- When it is over 50 this is representative of a higher intensity fire

2.4 Oxides of Carbon Ratio

This is the ratio of the carbon monoxide content to the carbon dioxide content. It is high for fuel rich fires and during flaming combustion. When its value is greater than 2 this indicates an active fire.

$$Oxides\ of\ Carbon\ Ratio\ =\ \frac{CO\%}{CO_2\%}$$

This ratio is unaffected by inflows of methane gas, air or inert nitrogen. As such it is a very good indicator when the gas mixture is being inertized with nitrogen and carbon dioxide.

2.5 Jones and Trickett Ratio

The Jones and Trickett ratio is based on the principle that different types of fuel produce different amounts of each gas. This ratio helps determine what exactly is burning and helps distinguish between a methane and a coal dust explosion.

$$JTR\ =\ \frac{CO_2\% + 0.75 \times CO\% - 0.25 \times H_2\%}{0.265 \times N_2\% - O_2\%}$$

- A value of less than 0.4 indicates extinguishment of the fire or greatly reduced coal temperatures

- A value from 0.4-0.5 indicates a methane fire
- 0.5-1 indicates a coal, oil or a conveyor belt fire
- 1-1.6 indicates a timber fire

2.6 C/H Ratio

The C/H ratio operates on the idea that temperature can be used to identify how much carbon and hydrogen have been part of the fuel of the mine fire. Hydrogen combusts in its entirety at a reduced temperature while carbon which has not combusted deposits as soot which results in a decrease in the ratio of carbon to hydrogen in the final resultant mixture.

$$C/H = \frac{3 \times (CO_2 + CO + CH_4 + 2C_2H_4)}{(0.2468N_2 - O_2 - CO_2 - 0.5H_2 + CH_4 + C_2H_4) + H_2 - CO}$$

- C/H ratios of up to 3 indicate superficial heating
- Values greater than 5 indicate active fire
- Values greater than 20 indicate blazing fire

2.7 Explosibility

It is important to predict whether an explosion is likely to happen in a sealed off area as explosions can cause a direct threat to life and can cause cave-ins which hamper rescue missions. This is usually done by measuring gas concentrations as certain mixtures of combustible gases and oxygen are combustible. In general an explosion requires 5 parameters to be satisfied: the presence of fuel, oxygen, ignition, confinement and dispersion of the gases involved.

Several methods are used in the mining industry in order to determine the explosibility of gas mixtures. The method used in the software created is the Coward's Revised Explosibility diagram.

The Coward explosive triangle considers the three most common explosive gases; methane, carbon monoxide and hydrogen. When the three combustible gases mix with normal air, the explosibility of the mixture depends on the percentage of the combustible gases and oxygen. In this method, a Coward triangle is graphed based on the flammability limits of the gases. Then a point is located on the diagram based on the gas concentrations in the mine. The location of the point on the graph determines the explosibility of the gas mixture.

The procedure to generate the Coward Triangle is given below:

Determine total combustible percentage

If the volume percentages of the three combustible gases are P_1, P_2 and P_3 then the total combustible percentage is

$$P_T = P_1 + P_2 + P_3$$

Determine gas flammability

An explosibility triangle similar to the Coward's explosibility triangle for methane has to be constructed for the three combustible gases. In order to do so the lower, upper and nose limits of the mixture have to be computed. The Le Chatelier principle is used to do this. In order to obtain the lower explosibility of the mixture(L_{mix}), the equation below can be used where L_1, L_2 and L_3 are the lower limits for the individual combustible gases. Similar computations can be performed for the upper and nose limits.

$$\frac{P_T}{L_{mix}} = \frac{P_1}{L_1} + \frac{P_2}{L_2} + \frac{P_3}{L_3}$$

Determine the required excess nitrogen

An infusion of a certain volume of nitrogen gas(N_{ex}) can be injected into a combustible gas-air mixture in order to render it non-explosive. The following equation is used to compute N_{ex}. Here, L_n is the nose flammability of the mixed gases and $N+$ is the volume of excess nitrogen to be added in order to make flammable gases extinctive(inert ratio)

$$N_{ex} = \frac{L_n}{P_T}\{N_1^+ P_1 + N_2^+ P_2 + N_3^+ P_3\}$$

Determine the oxygen percentage at the nose limit(O_n)

$$O_n = 0.2093\,(100 - N_{ex} - L_n)$$

These equations give the points necessary in order to plot Coward's explosibility triangle for the mixture of air and the combustible gases. The state point(for which the x-value is the concentration of oxygen and the y value is the sum of the combustible gases) can also be plotted on this graph. Depending upon which section of the explosibility triangle graph the state point is in, the explosibility of the mixture can be computed. See Appendix D for the algorithm.

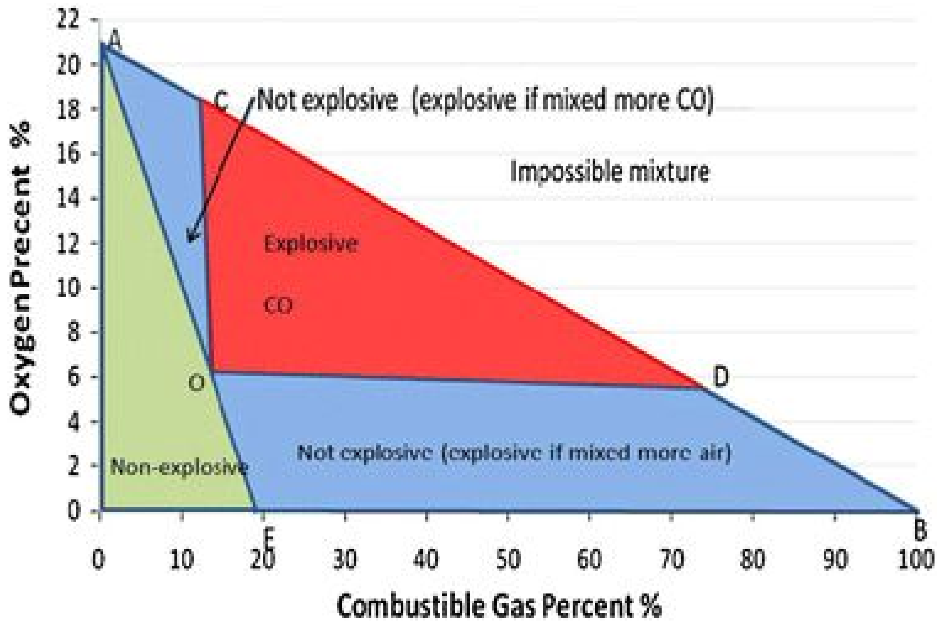

Figure 1: Coward's Explosibility Diagram

3. Method and Materials

This section details the software and hardware the author used to collect and analyse the data on fire strength and explosibility. The first step in monitoring for mine fires and explosions is the collection of data on the concentrations of various gases. A prototype sensor box was created for this purpose, containing a programmed microcontroller. A script was also created in order to convert the analog output from the microcontroller into a digital output and upload it to the analysing software. Finally, a website and a standalone PC application were also created to provide a graphical interface for data observation and analysis.

3.1 Sensor Box and Microcontroller

Gas sensors detect mine gases using a variety of mechanisms such as electrochemical reactions or thermal conductivity. The majority of sensors used in this project are infrared sensors. These sensors work by irradiating a gas sample. Depending on the natural frequency of the sample, the radiation excites the gas molecules to different levels. The excitation of the gas molecules increases their temperature which is observed by the sensor. Infrared sensors are expensive but have higher accuracy than their counterparts.

These sensors are then placed in an enclosure which has a gas chamber which collects a gas sample which then analyses for gas concentrations. For the purpose of this project a prototype sensor box was created to display the concentrations of certain gases.

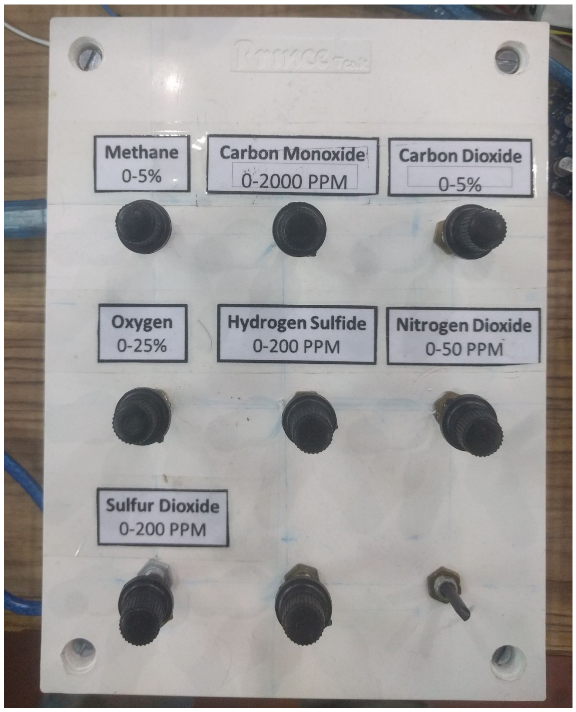

Figure 2: Prototype Sensor Box

The microcontroller used in this project is an Arduino Mega. The gas sensors detect concentrations in a certain range. For instance the carbon monoxide sensor detects gases in a 0-2000 ppm range. The sensors then output a proportional voltage to the analog input pins of the Arduino. The Arduino has been programmed to take the analog values and scale them to the range of the gas sensor. The program then prints a string with the gas values to the serial port of a connected computer. See Appendix-A for the Arduino program.

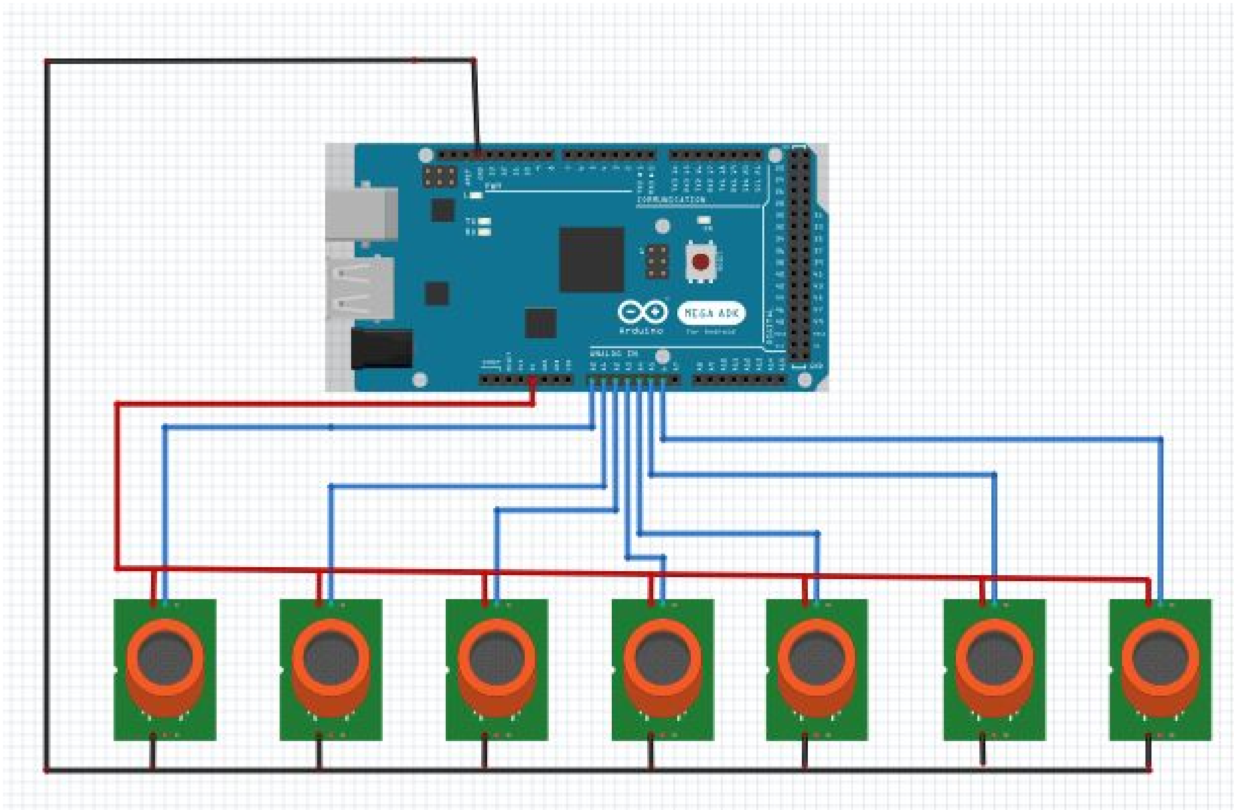

Figure 3: Microcontroller Circuit Diagram

3.2 Data Upload

A Python script has been written to accept the serial input from the microcontroller and edit it. The script makes use of the pyserial library for Python. The gas value string is accepted by the script and the string is edited so that it conforms to the comma separated values(csv) format.

The csv format is a format for storing values in text files, separating each value by a space and a comma. This format is used to store tabular data in .txt files. A comma indicates that one column should be traversed along the row and an end-of-line(EOL) character describes a new row in the csv format. Many tabulating softwares such as Microsoft Excel or LibreOffice Calc directly convert text files in the csv format to tabular data which can be graphed and analysed.
See Appendix-B for the script to convert the data to csv format.

3.3 Windows Forms Applications

A Windows Forms Application was created in order to present a graphical interface by which the user could enter values of gas levels and receive various fire ratios, explosibility and instructions to proceed. The application was created using the C# programming language and the .NET framework. The Visual Studio IDE was also used.

C# is a strongly typed and object oriented programming language. It was created by Microsoft for Windows application development under the .NET framework. The .NET framework provides user interface, database connectivity and other features. A Windows Forms Application is a graphical interface based application that functions across most versions of Windows operating systems and a few others as well.

The created application opens up with a page which displays several textboxes in which the user enters the gas levels. Upon clicking a ‘Calculate’ button, the values of various gas ratios are shown. Based on the values of these ratios, appropriate actions to be undertaken are also displayed alongside in a textbox.

An explosibility calculator plots the gas mixture onto a Coward’s extended explosivity diagram. This calculator then displays the explosibility of the mixture in one of four categories: Non-Explosive, Potentially Explosive(if air is added to the mixture), Potentially Explosive(if combustible gas is added to the mixture) and Explosive.

Furthermore, the software presents a graph underneath which shows how the explosibility of the mixture varies with time. It takes the data for the previous twenty dates for explosibility and graphs as the explosibility category versus time.

3.4 Website Module

A team from CSIR-CIMFR Dhanbad is undertaking the creation of a website under a project known as ‘Digital Mining using Internet of Things’. The objective of this project is to create a comprehensive consultation website for mining safety. The author of this report was tasked with creating a module for this website that deals with mine fires and mine gas explosibility.

The website was created using the Python programming language and the Django framework. HTML was used to render the webpages. SQLite was the database used to store the gas values.

The opening page of the website displays several textboxes into which the user enters gas levels. The user has a ‘Show’ button to display fire ratios and explosibility and a ‘Reset’ button in order to reset the form. There is also the option to add the values to a database and a ‘Display Database’ button to redirect the user to a page which displays the database values. Finally, there is an option to input data directly to the website database if a microcontroller is connected to the server hosting the webpage.

The database displays the concentrations of the various gases along with the dates on which these values were collected. In order to edit the database the user has to visit the page domain_name/admin where they will be prompted for a username and password. Once this has been done database access is granted and the database fields can be edited or deleted.

The Data Analysis section has tools for trend analysis and explosibility prediction. These tools allows the user to observe explosibility trends over time using methods such as Ellicott's extension and linear regression which will be discussed in further sections. See Appendix D to see the code used to calculate explosibility.

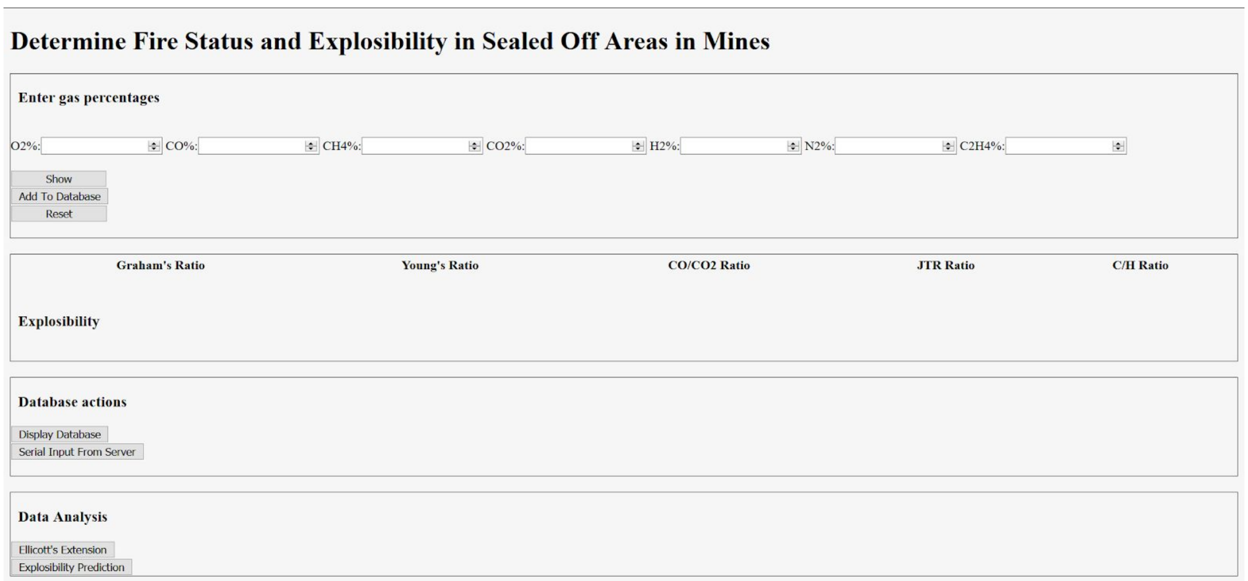

Figure 4: Screenshot of Website Interface

4. Results

The end result of this project is the creation of a multi-purpose software that can be used in many different situations and environments. In its final form the software will be in two forms, an application that can be run on a microcontroller with gas sensors attached and a website.

When a mine fire is detected the first procedure is the evacuation of workers and the sealing of the region which contains the fire. Over the next few days various operations such as the inertization of the enclosed area with nitrogen and carbon dioxide will also occur. During this time period is when the status of the fire and explosibility is checked. The author envisions the gas sensors (which will be encased in an enclosure) to be inserted into the sealed area through a bore hole.The gas sensors will then collect the gas sample into a collection chamber and will analyse the gas levels. The levels will be processed and converted from analog to digital values using a microcontroller such as an Arduino board. The observed gas levels will then be sent to the website and stored in a database (or stored locally using the independent software in case connectivity is poor).

Once the data is sent to the database, the software user can then use either the website or the independent software to calculate the status of the fire and explosibility of the gas mixture. Several gas ratios have been provided for the end user to draw conclusions from. Graham's ratio is the most commonly used ratio for calculating the fire strength. However it is inaccurate in several conditions such as when oxygen levels are very low. In such cases the user can use other ratios such as the CO/CO_2 ratio to determine the strength of the fire.

The translations of the various ratio values have also been provided in the software so that an operator who is not well versed in the underlying science of mining safety can still understand the situation in the mine.

Finally, a set of graphs and other analytical tools have been provided to the end user so that they can analyse the situation in the mine over time. The first is a simple graph which shows whether the gas mixture observed for a particular date is explosive or not. This can help in rescue work by allowing the software operator to order rescue operations when the gas mixture has been shown to be non-explosive for a long period of time.

The second graph which is provided is a more involved representation based on Ellicott's extension of Coward's explosivity diagram. This graph gives information about how close each gas mixture is to explosibility by providing a scatter plot.

An explosibility prediction tool which uses linear regression to analyse patterns and predicts the future state of the sealed off area has also been provided. The margin of error of this prediction is sufficiently low that a mine operator can check approximately how long the sealed off area will be non-explosive to plan extended rescue operations. The operator will also be able to check if the mixture is expected to be non-explosive for the foreseeable future and, hence, will be able to decide whether to remove the seal on the mine section.

Ultimately, the software provided and the sensor box will automate the process of collecting and processing mine gas data. This will eliminate any errors that usually develop in previous methods of graphing Coward's diagram and plotting points manually. The speed and efficiency of the process will also be improved. The analytics provided will make it easier for the operator to understand the data trends and will allow them to direct the rescue process efficiently and in the process save many lives.

5. Discussion

Other software of a similar nature has been created by researchers in China, South Africa and the United States. Software of this nature usually deals with three facets: fire status and explosibility determination, explosibility trend analysis and explosibility risk analysis.

5.1 Fire Status and Explosibility Determination

The software created by the author handles determining the strength of a fire in multiple ways by utilising several different fire ratios in order to describe the mine fire. A few ratios such as Litton's ratio have been omitted due to their being superfluous as the currently utilised ratios paint a complete picture. The ratios being used are valid in most environmental conditions and attempt give the strength of the fire as well as describe the fuel that the fire is consuming.

For explosibility calculation there are several documented methods to make the calculation such as the Kakuscka method and Maximum Allowable Oxygen method. However, the Coward's Explosibility diagram has been used since it is a universally accepted and popular method. The method can be used to quite accurately predict the explosibility of a mixture. It can also be extended to allow discussion of time trends of the explosibility. The Coward's diagram can, however, be improved by taking into account the extra effectiveness of carbon dioxide as an inertizing agent (in a method called Coward's revised explosibility diagram)

In most modern day mine fires N_2 is not the only gas used to inertize the sealed off area. Carbon dioxide is also used since it is not combustible. The presence of CO_2 as an inertizing gas requires the Coward's explosibility diagram to be changed since there will be a change in the shape of the explosive area. The improved algorithm is given below:

Separate the gas mixture into different groups
Most gaseous mixtures in mine fire environments are composed of four major groups of gases: oxygen, carbon dioxide, combustibles and nitrogen. The gas mixture must first be divided into two sections, one containing the combustibles, nitrogen and oxygen and the other containing combustibles, carbon dioxide and oxygen.

Calculate the explosive triangle for group 1:

1. Calculate total combustibles' volumetric concentration: Suppose there are six combustible gases with volumetric concentrations C_1, C_2, …, and ,C_6 respectively. The volumetric concentration of the combustible sum is $C_T = \sum_{n=1}^{6} C_n$
2. Determine the gas flammability: An explosibility triangle similar to the Coward's explosibility triangle for methane has to be constructed for the six combustible gases. In order to do so the lower, upper and nose limits of the mixture have to be computed. The Le Chatelier principle is used to do this. In order to obtain the lower explosibility of the mixture(L_{mix}), the equation below can be used where L_1, L_2 … L_6 are the lower limits for the individual combustible gases. Similar computations can be performed for the upper and nose limits. $\frac{C_T}{L_{mix}} = \sum_{n=1}^{6} \frac{C_n}{L_n}$
3. Determine the required excess nitrogen: An infusion of a certain volume of nitrogen gas(N_{ex}) can be injected into a combustible gas-air mixture in order to render it non-explosive. The following equation is used to compute N_{ex}. Here, L_n is the nose flammability of the mixed gases and $N+$ is the volume of excess nitrogen to be added in order to make flammable gases extinctive(inert ratio) $N_{ex} = \frac{L_n}{C_T} \times \sum_{n=1}^{6} N_n^+ C_n$
4. Determine the oxygen volumetric concentration in the nose limit $O_{n(N2)} = 0.2093(100 - N_{ex} - L_{n(N2)})$

Calculate the explosive triangle for group 2:

1. Calculate total combustibles volumetric concentration: Repeat step 1 performed for group 1
2. Determine gas flammability: Repeat step 2 for group 2 replacing N_2 with CO_2 as the inert gas
3. Determine the required carbon dioxide: $L_{n(CO2)}$ is the nose flammability limit of group 2. CO^+ is the volume of carbon dioxide to be added in order to make flammable gases extinctive $C_{CO2} = \frac{L_n}{C_T} \times \sum_{n=1}^{6} CO_n^+ C_n$
4. Determine the oxygen volumetric concentration at the nose limit: $O_{n(CO2)} = 0.2093(100 - CO_{CO2} - L_{n(CO2)})$

Generate the resultant explosive triangle

Two explosive triangles and two sets of nose limits have been calculated so far. The resultant nose limit can be calculated using Le Chatelier's principle again.

$$\frac{C_{N2} + C_{CO2}}{L_n} = \frac{C_{N2}}{L_{n(N2)}} + \frac{C_{CO2}}{L_{n(CO2)}}$$

$$\frac{C_{N2}+C_{CO2}}{O_n} = \frac{C_{N2}}{O_{n(N2)}} + \frac{C_{CO2}}{O_{n(CO2)}}$$

The state point(for which the x-value is the concentration of oxygen and the y value is the sum of the combustible gases) can also be plotted on this graph. Depending upon which section of the explosibility triangle graph the state point is in, the explosibility of the mixture can be computed.

5.2 Explosibility Trend Analysis

The software tools provided handle explosibility trend analysis for previous dates to allow the user to analyse trends and also for future dates using explosibility prediction.

5.2.1 Ellicott's Extension

Explosibility trend analysis in the software is implemented by graphing Ellicott's Extension of the Coward method. Ellicott's Extension takes the points plotted by the Coward method and uses an algorithm to place them in four different quadrants signifying:

State	x_m y_m
Explosive	$+ \ +$
Potentially Explosive	$+ \ -$
Non Explosive	$- \ +$
	$- \ -$

Ellicott's method first takes the nose point from Coward's diagram as the origin of a new graph. It then converts all the data points into polar coordinates. Then an algorithm places the points into different coordinates based on explosibility. The points are then converted back into Cartesian coordinates and plotted.

Directional information in the Coward diagram with respect to air or inert dilution or increasing combustible content is retained in the modified diagram. In general, air dilution causes a shift of the sample point towards the top left of the diagram and increase in combustible content causes a shift towards the bottom right.

Further advancements have also been made by other softwares developed by contemporary researchers. These softwares deals with all the facets of mining explosion safety. For trend analysis the software provides three different metrics:

- The quantity of fresh air that needs to be ventilated for suppression of an explosive atmosphere

- The quantity of inert gas injected for suppression of an explosive atmosphere
- The self inertisation time needed for an explosive atmosphere

See Appendix C for the Ellicott's Extension algorithm.

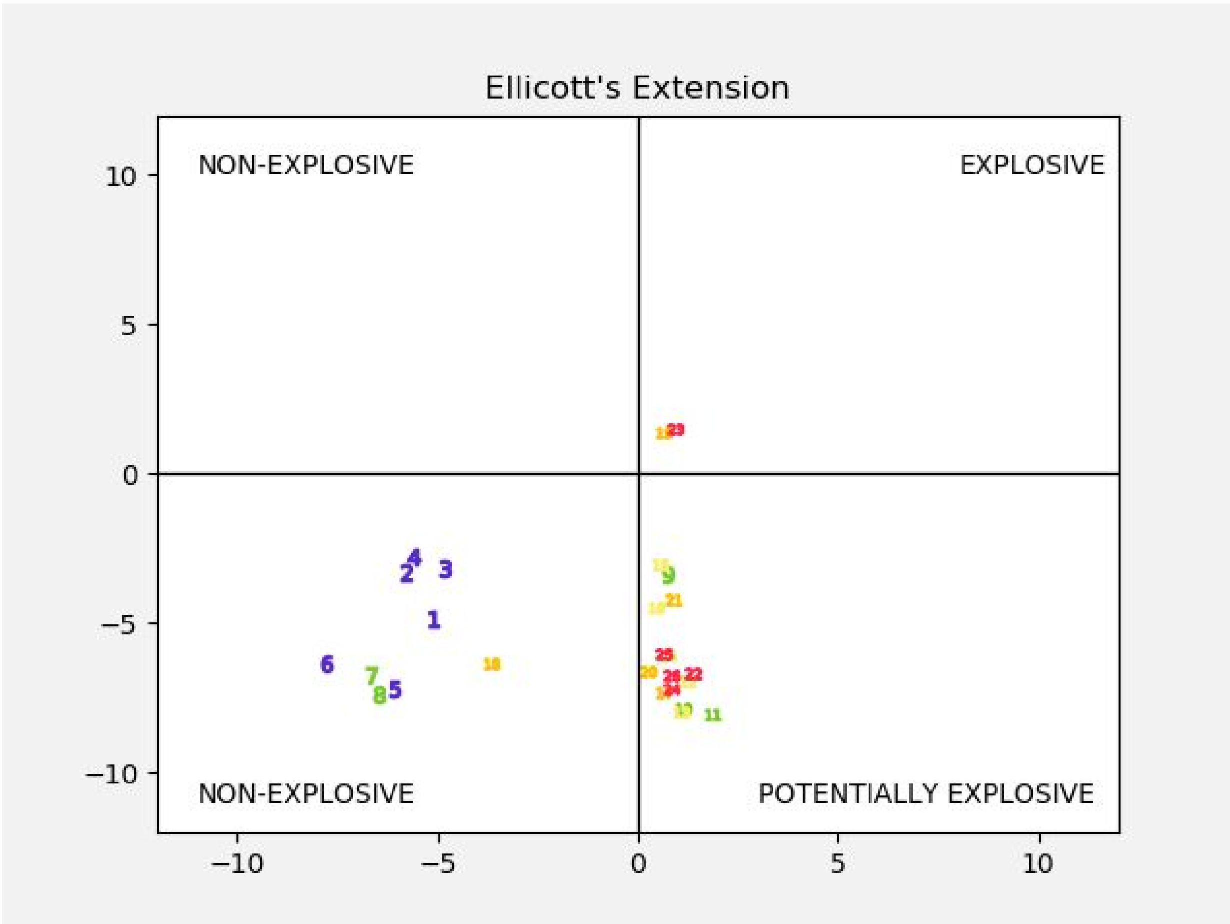

Figure 5: Ellicott's Extension

5.2.2 Explosibility Prediction

The prediction tool gives the user an estimate of the explosibility of the gas mixture between a given starting and ending date. The tool uses linear regression to extrapolate gas concentration values from the existing values given in the database. It then predicts the future gas concentration and the explosibility on the future date.

Linear regression is an algorithm which plots a best fit line for each gas using the values from the database. The value of gas concentration is plotted along the y-axis and the date is the independent variable. Once a line has been plotted, the line can be extrapolated and used to estimate gas values for future dates.

In order to plot a best fit line the algorithm first plots a random line:

$$y = \theta_1 x + \theta_0$$

where θ_0 and θ_1 are arbitrarily assigned

The method used to ensure that this line is the best fit line is the least squares approach which checks to see whether the sum of squares of the residuals is minimised.

Hence a function, called the cost function, needs to be minimised. This function is given by:

$$Cost\ Function = \frac{1}{N} \times \sum_{i=1}^{n} (y_i - (\theta_1 x_i + \theta_0))^2$$

where N is the number of data values and y_i *and* x_i are the y and x values of the data points.

This function is minimised using an algorithm called the gradient descent algorithm which returns the final values of θ_0 and θ_1

Gradient descent algorithm

repeat until convergence {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

$$(\text{for } j = 1 \text{ and } j = 0)$$

}

5.3 Explosibility Risk Analysis

Very little research has been done in India towards explosibility risk analysis. However, J.Cheng et al. from China University of Mining and Technology have developed a method to calculate a safety score by extending Coward's explosibility diagram that quantifies exactly how close a gas mixture is to exploding. The method uses an algorithm which calculates the weighted distance of a plotted point from each zone of the Coward explosibility diagram.

6. Conclusion

Ultimately, the objective of the project has been satisfied as the author has developed a system of software that is capable of automating the process of collection of gas data and its analysis. The system has been designed in a manner such that even laymen mine operators can use the graphical user interface provided without having a background in mining science. The software also contains analytical tools to analyse time trends of explosibility. However, more features such as a risk analysis calculator can be added to bring it on par with similar software developed elsewhere.

7. References

1. McPherson M.J. (1993) Subsurface fires and explosions. In: Subsurface Ventilation and Environmental Engineering. Springer, Dordrecht

2. Chaulya S., Prasad G.M. (2016) Sensing and Monitoring Technologies for Mines and Hazardous Areas. Elsevier

3. Cheng, J., Luo, Y. & Zhou, F. Fire Technol (2015) 51: 309. https://doi.org/10.1007/s10694-013-0324-y

4. Cheng, J., Luo, Y, ‘Methods to determine the mine gas explosibility - An overview’, Journal of Loss Prevention in the Process Industries (2013) 26: 714-722

5. Ray, S.K., Singh, R.P., Sahay, N., Varma, N.K., ‘Assessing the status of sealed fire in underground coal mines’, Journal of Scientific & Industrial Research (2004) 63: 579-591

8. Appendix A

The program that runs on the microcontroller which is connected to the sensors.

ser_inp_ard | Arduino 1.8.5 (Windows Store 1.8.10.0)

File Edit Sketch Tools Help

ser_inp_ard

```
int counter = 0;
float avCH4 = 0.0;
float sumCH4 = 0.0;
float avCO = 0.0;
float sumCO = 0.0;
float avCO2 = 0.0;
float sumCO2 = 0.0;
float avO2 = 0.0;
float sumO2 = 0.0;

void setup() {
  Serial.begin(9600);
}

void loop() {
  int methane = analogRead(A0);
  int carbonMonoxide = analogRead(A1);
  int carbonDioxide = analogRead(A2);
  int oxygen = analogRead(A3);
  int hydrogenSulphide = analogRead(A4);
  int nitrogenDioxide = analogRead(A5);
  int sulfurDioxide = analogRead(A6);

  float CH4 = 0.004887585532*methane;
  float CO = 1.95503421309*carbonMonoxide;
  float CO2 = 0.004887585532*carbonDioxide;
  float O2 = 0.0244379276637*oxygen;
```

```
  counter++;
  sumCH4 += CH4;
  avCH4 = sumCH4/counter;
  sumCO += CO;
  avCO = sumCO/counter;
  sumCO2 += CO2;
  avCO2 = sumCO2/counter;
  sumO2 += O2;
  avO2 = sumO2/counter;

  if (counter==100)
  {
    Serial.print(avO2);
    Serial.print(",");
    Serial.print(avCO);
    Serial.print(",");
    Serial.print(avCH4);
    Serial.print(",");
    Serial.println(avCO2);
  }

//  Serial.write(CH4);
//  Serial.write(",");


  delay(10);
}
```

9. Appendix B

The script which is used to take the serial input from the microcontroller and convert it to csv format.

ser_inp.py - C:\Users\Varghese\Documents\Python\DjangoStuff\FireExplosModule\ser_inp.py (3.6.5)

File Edit Format Run Options Window Help

```
import serial
from time import sleep
flag = 0

try:
    ser = serial.Serial("COM1", 9600)
except serial.serialutil.SerialException:
    try:
        ser = serial.Serial("COM2", 9600)
    except serial.serialutil.SerialException:
        try:
            ser = serial.Serial("COM3", 9600)
        except serial.serialutil.SerialException:
            try:
                ser = serial.Serial("COM4", 9600)
            except serial.serialutil.SerialException:
                try:
                    ser = serial.Serial("COM5", 9600)
                except serial.serialutil.SerialException:
                    try:
                        ser = serial.Serial("COM6", 9600)
                    except serial.serialutil.SerialException:
                        try:
                            ser = serial.Serial("COM7", 9600)
                        except serial.serialutil.SerialException:
                            print ('Microcontroller not connected')
                            flag = 1

if (flag==0):
    print("Wait for 2 seconds")
    sleep(2) # Delay for 2 seconds
    datastring = str(ser.readline()) # Read the newest output from the Arduino
    datafin = ''
    i = 0
    for i in range (2, len(datastring)-5):
        datafin += datastring[i]
    print (datafin)
    ##now need to edit the string and save it to a text file in csv format (one
    f = open("datasetfile.txt","a")
    f.write(datafin)
    f.write('\n')
    f.close()
    fs = open("datafile.txt","w")
    fs.write(datafin)
    fs.write('\n')
    fs.close()
    print("Data collected")
```

10. Appendix C

Algorithm to convert a point in Coward's Triangle to Ellicott's Extension.

```
##calculating new x,y coordinates after origin shift
xx = pt-Lnose
yx = x.o2 - Oxnose
xp = Llow-Lnose
yp = Ob-Oxnose
xq = Lhigh-Lnose
yq = Oc-Oxnose
xs = Le-Lnose
ys = -Oxnose

#calculating polar coordinates
def properarctan(valuex, valuey):
    if valuex>=0:
        if (np.degrees(np.arctan(valuey/valuex)<0)):
            return (360+np.degrees(np.arctan(valuey/valuex)))
        else:
            return np.degrees(np.arctan(valuey/valuex))
    else:
        return (np.degrees(np.arctan(valuey/valuex))+180.0)


rx = np.sqrt(xx*xx+yx*yx)
thx = properarctan(xx,yx)
rp = np.sqrt(xp*xp+yp*yp)
thp = properarctan(xp,yp)
rq = np.sqrt(xq*xq+yq*yq)
thq = properarctan(xq,yq)
rs = np.sqrt(xs*xs+ys*ys)
ths = properarctan(xs,ys)


##calculating r,theta values based on explosibiility
if x.explos == 3:
    rm = rx
    thm = 90*((thx-thq)/(thx-thq+thp-thx))
elif (x.explos ==1 or x.explos ==2):
    rm = rx
    thm = 270 + (90*((thx-ths)/(thx-ths+thx-thq))) ##HERE MADE A
elif x.explos == 0:
    rm = rx
    thm = 90 + (180*((thx-thp)/(thx-thp+ths-thx)))
else:
    rm=0
    thm=0

x.elx = rm * np.cos(np.radians(thm))
x.ely = rm * np.sin(np.radians(thm))
```

11. Appendix D

The algorithm used to calculate the explosibility of a point.

```
##explosibility
explos = 5
pt = ch4 + co + h2

ch4low = 5
colow = 12.5
h2low = 4
ch4high = 14
cohigh = 74.2
h2high = 74.2
ch4nose = 5.9
conose = 13.8
h2nose = 4.3
ch4np = 6.07
conp = 4.13
h2np = 16.59


Llow = pt / (ch4 / ch4low + co / colow + h2 / h2low)
Lhigh = pt / (ch4 / ch4high + co / cohigh + h2 / h2high)
Lnose = pt / (ch4 / ch4nose + co / conose + h2 / h2nose)
Nex = Lnose / pt * (ch4np * ch4 + conp * co + h2np * h2)
Oxnose = 0.2093 * (100 - Nex - Lnose)

##total combustible at extinctive point
Le = 20.93 * Lnose / (20.93 - Oxnose)
##oxygen at lower limit
Ob = -20.93 * Llow / 100 + 20.93
##oxygen at upper limit
Oc = -20.93 * Lhigh / 100 + 20.93

if ((o2 >= 0) and (pt >= 0)):
    if (100 * o2 + 20.93 * pt >= 2093):
        explos = 4
    if (Le * o2 + 20.93 * pt <= Le * 20.93):
        explos = 0
    if ((100 * o2 + 20.93 * pt <= 2093) and (Le * o2 + 20.93 * pt >= Le * 20.93) and
        ((Lnose-Llow)*o2+(Ob-Oxnose)*pt<=Ob*Lnose-Ob*Llow-Oxnose*Llow+Ob*Llow)):
        explos = 2
    if ((100 * o2 + 20.93 * pt <= 2093) and (Le * o2 + 20.93 * pt >= Le * 20.93) and
        ((Lnose - Llow) * o2 + (Ob - Oxnose) * pt >= Ob * Lnose - Ob * Llow - Oxnose * Llow + Ob * Llow) and
        ((Lnose-Lhigh)*o2+(Oc-Oxnose)*pt<=Oc*Lnose-Oc*Lhigh-Oxnose*Lhigh+Oc*Lhigh)):
        explos = 3
    if ((100 * o2 + 20.93 * pt <= 2093) and (Le * o2 + 20.93 * pt >= Le * 20.93) and
        ((Lnose - Llow) * o2 + (Ob - Oxnose) * pt >= Ob * Lnose - Ob * Llow - Oxnose * Llow + Ob * Llow) and
        ((Lnose - Lhigh) * o2 + (Oc - Oxnose) * pt >= Oc * Lnose - Oc * Lhigh - Oxnose * Lhigh + Oc * Lhigh)):
        explos = 1
```

```
##explosibility message
if(explos==0):
    explosm = "Not explosive"
elif(explos==1):
    explosm = "Potentially explosive(if air is added)"
elif(explos==2):
    explosm = "Potentially explosive(if combustible gas is added)"
elif(explos==3):
    explosm = "Explosive"
elif(explos==4):
    explosm = "Impossible mixture"
else:
    explosm = "Unidentified"
```