

SAFETY SYSTEM FOR COAL MINERS

Submitted in partial fulfilment of the requirements for the award of
Bachelor of Engineering / Technology degree in
ELECTRONICS AND COMMUNICATION ENGINEERING

By

RAJIV REDDY SONTAM (38130178)

MERRIL M (38130133)



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

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

MAY - 2022



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC I 128 Status by UGC I Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **RAJIV REDDY SONATM (38130178)**, **MERRIL M (38130133)** who carried out the project entitled **SAFETY SYSTEM FOR COAL MINERS** under my / our supervision from

to

Internal Guide

(Name in Capital letters with Signature)

External Guide (if applicable)

(Name in Capital letters with Signature)

HEAD OF THE DEPARTMENT

(Name in Capital letters with Seal & Signature)

Submitted for Viva voice Examination held on

Internal Examiner

(Name in Capital letters with Signature)

External Examiner

(Name in Capital letters with Signature)

DECLARATION

We, RAJIV REDDY SONTAM (38130178)MERRIL M (38130133)hereby declare that the Project Report entitled “SAFETY SYSTEM FOR COAL MINERS” done by us under the guidance of **DR. R.Narmadha** (ECE) is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in **ELECTRONICS AND COMMUNICATION ENGINEERING.**

SIGNATURE OF CANDIDATES

RAJIV REDDY SONTAM (38130178)

MERRIL M (38130133)

DATE:

PLACE: CHENNAI

ACKNOWLEDGEMENT

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

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

We would like to express my sincere and deep sense of gratitude to my Project Guide **DR. R.NARMADHA, ECE.,**for her valuable guidance, suggestions and constant encouragement paved the way for the successful completion of our project work.

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

ABSTRACT:

Mining is the first step in the dirty life cycle of coal. When coal mines move in, whole communities are forced off their land by expanding mines, coal fires, subsidence, and overused and contaminated water supplies. Mines are quick to dig up and destroy forests and soils. But once the coal is gone, the problems they leave behind, like acid mine drainage, can persist for decades. Around the world, Greenpeace campaigns to help communities stop coal mines, and speed up the shift to 100 percent clean, safe renewable energy. In this we are going to use Arduino, temperature sensor, humidity sensor, and heartbeat sensor and smoke sensor. The sensor information is collected by the Arduino and they are stored in cloud. And the alert message is passed to the Gmail and mobile phone using IOT. With continuous enlarging of exploiting areas and extension of depth during a coal pit, several laneways become blind areas, whereby there are unit uncountable hidden dangers. Moreover, it's inconvenient to put cables that area unit costly and consume time. so as to resolve the issues, we'll style a coal pit safety watching system supported wireless detector network, which may improve the extent of watching production safety and scale back accident within the coal mines Wireless detector networks area unit composed of an oversized variety of micro-sensor nodes that have tiny volume and low price.

OBJECTIVE:

Mining environment often has hidden dangers within such as toxic gases, which may present severe health exposures to the people working within mining. These gases need to be detected at times and informed the dangerous situation in right time for the safety of miners. Wired network monitoring systems have assisted the mine safety significantly, but it is not idea for all types of mining environment.

A real-time monitoring system may assist in monitoring and control over the mining environment. Zigbee technology offers its most of the advantages ideal for the real-time monitoring system. Thus, the primary objective of this project is decided to design an efficient real-time monitoring system so that various leaked mine gases could be identified at times and preventive measures could be devised accordingly. The research investigations to be carried out with the following objectives:

- Detection of different toxic gases within mining environment
- Communication establishment between sensors and Zigbee
- Establishment of Wireless Sensor Network
- Design of a real-time monitoring system

ORGANIZATION OF THE THESIS:

The present work is an attempt to analyse the safety scenario of a mine. The idea is an improvisation on previous related works where WSN is used to detect the toxic gases present in the mine. The real time monitoring which is the requirement now a days is designed for the purpose. The thesis presents all the related technologies such as ZigBee and embedded designing etc. the work can be summarized as follows:

- Introduction
- Literature Review
- Mine gases and their Impacts
- System Design
- Experiments and Results
- Result

TABLE OF CONTENTS:

CHAPTER NUMBER	TITLE	PAGE NO
1	INTRODUCTION	1
2	LITERATURE REVIEW	4
3	MINE GASES AND THEIR IMPACT	9
4	SYSTEM DESIGN	19
	4.1 EXISTING SYSTEM	
	4.2 PROPOSED SYSTEM	20
	4.3 BLOCK DIAGRAM	21
5	HARDWARE REQUIREMENTS	22
6	SOFTWARE REQUIREMENTS	43
7	CONCLUSION	64
8	REFERENCE	65

LIST OF FIGURES

FIGURE NUMBER	FIGURE NAME	PAGE NO
1.1	WIRELESS SENSOR NETWORK	3
3.1	CONCENTRATION AND EXPOSURE TIME	15
5.10	ARDUINO	23
5.2	NODE MCU	32
5.21	NODE MCU	32
5.3	GAS SENSOR	36
5.31	GAS SENSOR	36
5.4	MEMS	38
5.5	TEMPERATURE SENSOR	41
5.6	HEARTBEAT SENSOR	42
5.61	HEARTBEAT SENSOR	43
6.10	CONNECTION	44
6.11	CODE	45
6.12	CODE	45
6.13	CODE	46
6.20	PROTEUS	54
6.21	SUMMARY	55
6.30	UBIDOTS	58
6.31	SYNTHETIC VARIABLE ENGINE	59
6.32	SYNTHETIC VARIABLE ANALYTICS	60
6.33	DEFAULT	61
6.34	EVENT TYPES	62
6.35	EVENT TYPES	62
6.36	EVENT TYPES	63

LIST OF TABLES

TABLE NUMBER	TABLE NAME	PAGE NO
3.1	HEALTH EFFECTS & POLLUTANT CONCENTRATION BREAKPOINTS CAUSED BY NO ₂	12
3.2	HEALTH EFFECTS & POLLUTANT CONCENTRATION BREAKPOINTS CAUSED BY SO ₂	13
3.3	HEALTH EFFECTS & POLLUTANT CONCENTRATION BREAKPOINTS CAUSED BY CO	15
3.4	CONCENTRATION OF CO & ITS EXPOSURE TIME	16
3.5	CLASSIFICATION OF TOXIC GASES & THEIR HAZARDOUS LIMIT	18

CHAPTER 1

INTRODUCTION:

Underground mining operations proves to be a risky venture as far as the safety and health of workers are concerned. These risks are due to different techniques used for extracting different minerals. The deeper the mine, the greater is the risk. These safety issues are of grave concern especially in case of coal industries. Thus, safety of workers should always be of major consideration in any form of mining, whether it is coal or any other minerals.

Underground coal mining involves a higher risk than open pit mining due to the problems of ventilation and potential for collapse. However, the utilization of heavy machinery and the methods performed during excavations result into safety risks in all types of mining.

Modern mines often implement several safety procedures, education and training for workers, health and safety standards, which lead to substantial changes and improvements and safety level both in opencast and underground mining.

Coal has always been the primary resource of energy in India, which has significantly contributed to the rapid industrial development of the country. About 70% of the power generation is dependent on it thus, the importance of coal in energy sector is indispensable. But the production brings with it the other by-products, which proves to be a potential threat to the environment and the people associated with it. In lieu of that the present work is a sincere attempt in analysing the graveness and designing a real time monitoring system of detection.

BACKGROUND AND MOTIVATION:

In underground mine, ventilation systems are critical to supply adequate oxygen, keeping up non-dangerous and non-lethal environments and an effective working

mine. To monitor an underground mine, can help killing high hazard environments. Primitive procedures of monitoring a mine's air can be followed back to the utilization of canaries and different creatures to ready diggers when the climate gets to be lethal. Incorporating ventilation monitoring systems empowers a mine to insightfully roll out ventilation improvements in view of the far-reaching information given by the monitoring systems. Sudden changes in the ventilation system are identified by the monitoring system, permitting quick move to be made. New and creating correspondence and following systems can be used to monitor mines more proficiently and transfer the information to the surface.

The progression of technology has allowed mine monitoring techniques to become more sophisticated, yet explosions in underground coal mines still occur. The safety issues of coal mines have gradually turned into a major concern for the society and nation. The occurrence of disasters in coal mines is mainly due to the harsh environment and variability of working conditions. So, it makes the implementation of mine monitoring systems essential for the safety purpose. Wired network systems used to be a trend for traditional coal mines, which have really played a significant role in safely production in coal mines. With the continuous enlargement of exploiting areas and depth expansion, laneways have become blind zones, where numerous unseen dangers are hidden out. Moreover, it is not possible there to lay expensive cables, which is also time consuming. So, it is essential to have a wireless sensor network mine monitoring system, which can be disposed in such mines in order to have a safe production within.

Wireless sensor networks (WSNs) have earned a significant worldwide attention in current scenario. A WSN is a special ad-hoc, multi-hop and self-organizing network that consists of a large number of nodes arranged in a wide area in order to monitor the phenomena of interest. It can be useful for medical, environmental, scientific and military applications. Wireless sensor networks mainly consist of sensor nodes or motes responsible for sensing a phenomenon and base nodes, which are responsible for managing the network and collecting data from remote nodes.

The design of the sensor network is influenced by many factors, including scalability, operation system, fault tolerance, sensor network topology, hardware constraints, transmission media and power-consumption.

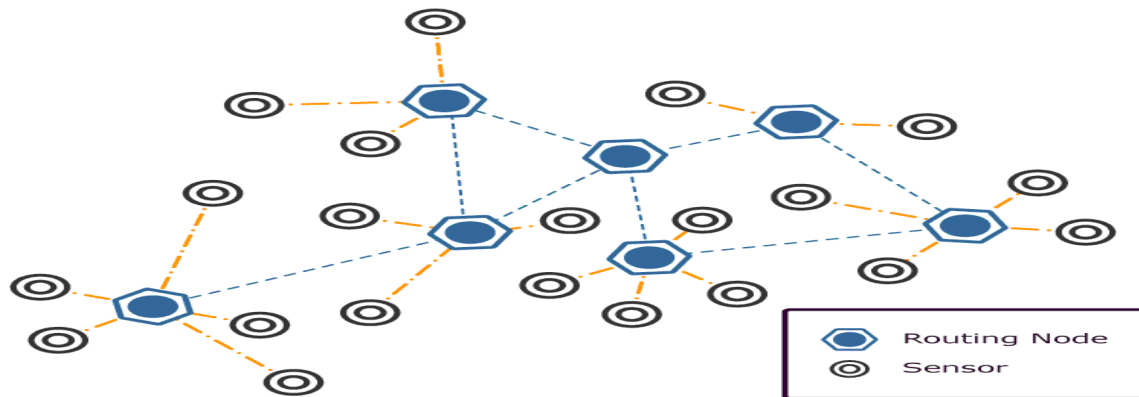


Figure 1.1 Wireless Sensor Network

(<http://www.purelink.ca/en/technologies/related-technologies.php>)

These small sized sensors are quite inexpensive compared to traditional sensors and also require limited computing and processing resources. These sensor nodes can detect, measure and collect information from the environment and based on some local statistical decision process, they can convey the collected data to the control room. It has three major advantages over wired monitoring network systems:

- There is no need of cables to lay and easy installation in blind areas, reducing cost of the monitoring system. The number of nodes can be increased to eliminate blind areas. Also, it offers a general communication and allocation of the goal.
- The dense nodes ensure the data acquisition with high accuracy and optimum data transmission, and further realization of real-time monitoring system for mine environment.
- A little computing ability, storage capacity with data fusion of sensor nodes makes them suitable for the remote monitoring system.

The above-mentioned advantages make wireless sensor network ideal for monitoring of safely production of coal mine.

CHAPTER 2

LITERATURE REVIEW:

In underground mining, ventilation systems are crucial to supply sufficient oxygen, maintaining non-explosive and non-toxic atmospheres and operating an efficient mine. Mine ventilation system can help in eliminating high risk atmosphere. Primitive techniques to monitor the mining atmosphere can be traced back to the use of canaries and other animals to alert miners, when the atmosphere becomes toxic. Integrating ventilation monitoring system enables mine to intelligently make ventilation changes based on the extensive data, the monitoring system provides. Unexpected changes in the ventilation system are noticed by the monitoring arrangement, allowing prompt action to be considered. New and developing communication and tracking systems can be utilized to monitor mines more efficiently and relay the data to the surface.

PREVIOUS WORK:

These are the previous research work on different systems using different technologies for the safety of the environment.

Yu et al. (2005) proposed a real-time forest fire detection system based on wireless sensor network. The system collects the data and processes it in the WSN for detecting the forest fire. They designed the monitoring and detecting sensor networks using neural network.

Joseph et al. (2007) focused on the problems and hazards of fire in libraries or archives and described the necessary preventive steps to be adopted. They identified the diverse parts which are applied for fire detection and alert system and also provided necessary strategies for the selection and installation of an ideal fire alarm system.

Fischer (2007) considered the simulation technique and applied this technique to design a fire detection system. This system detects the fire as well as differentiates fire and non-fire spot to decrease the false alarm rate in the non-fire event.

Tan et al. (2007) designed a system, which is applied for mine safety monitoring. They called the system WSN based Mine Safety System. This system is capable of real time monitoring of the mine environment and provide the pre-warning for the fire or explosion.

Niu Xiaoguang et al. (2007) presented a distributed heterogeneous hierarchical mine safety monitoring prototype system (HHMSM) which is based on features of the underground mine gallery and necessities of mine safety. This system monitors the methane concentration and the location of miner. They proposed an overhearing-based adaptive data collecting system, which makes use of the redundancy and the correlation of the sampling readings in both time and space to ease the traffic and control.

He Hongjiang et al. (2008) designed a system using low power ARM (Advanced RISC Machines) processor chip S3C2410 as the control of core and Zigbee as a communication platform of WSN. This system composed of network mode, communication network of CAN BUS as well as monitoring sensors.

Zheng Sun et al. (2008) analyzed the problems of mine safety monitoring and an improved TinyOS Beaconing-based WSN. This protocol can not only aware Energy and repair route automatically, but also can prevent the number of child nodes and that of system levels. The features are small routing Table, high stabilization, high self-repairing and long lifetime. It may be suitable for coal mine data acquisition and applied to mine safety monitoring.

Lin-Song Weng et al. (2009) planned a framework, which is viably observing all circumstances in mine, particularly for the wellbeing of mineworkers. They named the system the real-time mine auxiliary monitoring system (RMAMS), which is

embraced for a real-time mine-monitoring system. Mine auxiliary sensor system (MASS) consists of an intelligent activity sensor and repeater and arrives at decision to resolve the procedure of processing.

Hua Fu *et al.* (2009) studied the fuzzy theory and neural network technology and by using this information they designed an intelligent fuzzy neural network sensor system for coal mine. This technology can make accurate detection of different parameters.

Shi Wei *et al.* (2009) designed a multi parameter monitoring system for coal mine tunnel, which is based on WSN network. This system uses the RS-485 communication protocol and hardware modular. It automatically sends warning signals to the main control room and accomplishes corresponding control.

Wenge Li *et al.* (2009) designed a system for remote monitoring and analysis of mines using virtual instrument technology, network technology and database technology. This organization consists of sensors, remote clients, the ground monitoring center and the underground substation. The remote client, through internet explorer, can browse the real time monitoring data of mine safety such as temperature, gasoline, wind velocity, carbon monoxide and so on. The system stored the data using ADO in LabVIEW.

Shao Chang'an *et al.* (2009) studied the coal mine safety information and based on this information such as dynamic changes, activity, closely related to the space, they used special data mining and GIS technology for designing a coal mine safety monitoring.

Hongmei Wu *et al.* (2010) proposed a remote monitoring system for mine vehicle based on wireless sensor technology. This scheme uses the sensor nodes,

deployed on the vehicles to collect speed, mileage, pressure, oil level value, and data to the ARM based information processing terminal.

Li-Chien Huang *et al.* (2011) designed a system for building electrical safety. No-fuse breakers (NFBs) and electrical wall plugs are the main components of traditional distribution, which is used for power transmission and overload protection. NFBs have the utilization of over-burden security and are not completely compelling in forestalling electrical flames created by poor contact or dust pollution. This plan is built with assurance instruments so as to upgrade the parts of customary circulation frameworks. The effects on other equipment in the same branch circuit can be avoided by the threshold limit of the system when the outlet disconnects the power.

Ge Bin *et al.* (2011) suggested a method for monitoring coal mine using Zigbee technology. This system measures the various safety factor of production such as gas, temperature, humidity and other environmental indicators.

Cheng Bo *et al.* (2012) proposed a restful web services mashup improved coalmine safety monitoring and control automation using WSN network. This system can collect the values of methane, temperature, humidity and personal position information inside the mine.

Rajkumar Boddu *et al.* (2012) designed a coal mine monitoring system using Zigbee based on GSM technology. The degree of monitoring safety can be improved using this scheme and reduce misfortune in the coal mine. They purposed a solution suitable for mine wireless communication, and safety monitoring using this scheme.

Isaac O. Osunmakinde (2012) studied the different types of toxic fumes in dangerous regions and their conditions and trends in the air for preventing miners from contracting diseases. They developed an autonomous remote monitoring system of WSNs which combines Ohm's law and mobile sensing coupled with

ambient intelligence governing decision-making for mine workers. The system has been monitored the indoor scenarios which is successfully deployed in underground mines. The system provides pre warning for safety purpose.

Mohit Kumar *et al.* (2013) proposes a wireless control and monitoring system for an induction motor based on Zigbee communication protocol for safe and economic data communication in industrial fields, where the wired communication is more expensive or impossible due to physical conditions. This system monitors the parameters of induction machine and transmit the data. A microcontroller-based system is used for collecting and storing data and accordingly generating a control signal to stop or start the induction machine wireless through a computer interface developed with Zigbee.

Mr. Kumarsagar *et al.* (2013) designed a wireless sensor network with the help of MSP430xx controller, which is monitor the smoke, gas, temperature and humidity in an underground mine. This system also controls the ventilation demand to miners depending using upon the monitoring data from the mine. This system utilizes a wireless Zigbee transceiver for remote logging of data at a central location to control the environmental state with the assistance of a motor and valve control circuitry.

Berardo Naticchia *et al.* (2013) proposed the infrastructure less real-time monitoring system to provide prompt support for inspecting the health and safety management on construction sites. They tested the specific applications of monitoring, interference between teams working on large construction sites. The system is capable of alert in the occurrence of interference and to log any unexpected behaviour.

Zhang Xiaodong *et al.* (2014) presented the problems and faultiness of current coal mine monitoring system. They examined the plan and implementation of a platform to remotely monitor and control coal mine production processes over Industrial Ethernet based on the embedded engineering. Integrated with each lower

computer terminal are S3C2410 microprocessors that can be utilized for linking up to the monitoring network effectively.

CONCLUSION

The chapter explains all the previous work related to the monitoring of mine safety. The different researchers performed their work in this regard and came out with their own systems of surveillance for the mine gases and fires. Based on their study the following work is an improvisation on the real time monitoring system with wireless technology of data transfer.

CHAPTER 3

MINE GASES AND THEIR IMPACT:

The air of the atmosphere that we breathe is a mixture of several gases and its composition is practically constant over the whole surface of the earth. Because air is mixture and not a chemical substance, the components can be separated.

- **MINE GASES**

In mine, gases are released during mining operations. It will be observed that return air is depleted in oxygen content and contaminated by mine gases. Impurities come from exhalation by men, blasting, and underground fires, burning of lights, bacterial action and gases given off from strata. It also contains moisture and dust of coal and rock.

When referring to noxious and poisonous gases met with in a mine the commonly used names are as follows:

- **Blackdamp:** It is a mechanical mixture of the extinctive gases, carbon monoxide and excess nitrogen; sometimes it is referred to as chokedamp or scythe.

- **Firedamp:** It is used either as synonymous with methane or referring to the mechanical mixture of the gases, chiefly inflammable, given off naturally from coal and consisting for the most part of methane.
- **Whitedamp:** It is synonymous with carbon monoxide.
- **Stinkdamp:** It is synonymous with sulphureted hydrogen (H₂S).
- **Afterdamp:** This is a mechanical mixture of gases existing in a mine after an explosion of firedamp or coal dust. Its composition is extremely variable, but usually includes carbon monoxide, carbon dioxide, nitrogen and sometimes H₂S and SO₂ with a very small percentage of oxygen.

The necessities for gas distinguishing proof can move massively, on the other hand, there are five fundamental sources of hazardous gas in mining applications.

- ***Gasses from Blasting:***

Blasting generates toxic and harmful gases. These harmful gasses include carbon monoxide and nitrogen dioxide. As a result of the utilization of oxygen in any such impact, oxygen deficiency might likewise be an outcome.

- ***Methane from Coal Beds:***

Profoundly flammable methane (CH₄) or firedamp, as it is brought in numerous coal- fields, is framed in the last phases of coal arrangement, and due to the profundities and weights, it gets to be imbedded in the coal. As unearthing are made, methane gas is freed into the air. Gas is transmitted from the purpose of unearthing, as well as from the coal being transported to the surface.

- ***Vehicle Exhaust:***

Vehicles are also generated various toxic and poisonous gases. These poisonous gases are an aftereffect of the operation of burning motors.

- ***Underground Explosions and Fires:***

- ***Penetrating into Stagnant Water:***

Pockets of stagnant water can contain a lot of hydrogen sulfide coming about essentially from the breakdown of pyrites.

These are some harmful gases and their effects:

- **Nitrogen Dioxide (NO₂)**

NO₂ is a reddish-brown gas with a sharp and chafing scent. It changes noticeable all around to shape vaporous nitric corrosive and harmful natural nitrates. NO₂ additionally assumes a noteworthy part in climatic responses that create ground-level ozone, a noteworthy segment of brown haze. It is additionally an antecedent to nitrates, which add to expanded respirable molecule levels in the climate.

Sources of NO₂

All burning in air produces oxides of nitrogen (NO_x) such as NO, NO₂, N₂O₃ and have choking smell. These oxides are easily dissolved by moisture in the mine air. NO₂ are formed during the blasting of explosives containing nitro glycerine as one of the constituents if the explosives are not detonated completely.

Impacts of NO₂

NO₂ can chafe the lungs and lower impervious to respiratory contamination. Affectability increments for individuals with asthma and bronchitis. NO₂ artificially changes into nitric corrosive and, when stored, adds to Lake Acidification. NO₂, when artificially changed to nitric corrosive, can consume metals, blur fabrics and corrupt elastic. It can harm trees and products, bringing about considerable misfortunes.

Table 3.1 Health effects and Pollutant concentration breakpoints caused by NO₂

Category	Pollutant Concentration Breakpoints (ppb)	Health Effects
Very Good	0 -50	No health impacts
Good	51 -110	Slight smell.
Moderate	111 - 200	Smell.
Poor	201 - 524	Air smells and looks brown. Some increment in bronchial hyperactivity in asthmatics people.
Very Poor	525 or over	Expanding affectability for asthmatics and individuals with bronchitis.

- **Sulfur Dioxide (SO₂)**

SO₂ is a colourless gas with a strong sulphureous smell, neither combustible nor a supporter of combustion. It is 2.21 times heavier than air. It can be oxidized to sulfur trioxide, which in the region of water vapor is instantly changed to sulphuric corrosive fog. SO₂ can be oxidized to shape corrosive vaporizers. SO₂ is a forerunner to sulfates, which are one of the principal segments of respirable particles in the air.

Sources of SO₂

It may be produced in small quantities during blasting in mines, and after a fire or coal dust explosion.

Impacts of SO₂

This gas is very poisonous and extremely irritating to the eyes and respiratory passages. Health impacts brought about by the presentation to abnormal amounts

of SO₂ incorporate breathing issues, respiratory sickness, changes in the lung's safeguards, and intensifying respiratory and cardiovascular ailment. Individuals with asthma or perpetual lung or coronary illness are the most delicate to SO₂. It also harms trees and harvests. SO₂, alongside nitrogen oxides, is the principal antecedents of corrosive downpour. This adds to the fermentation of lakes and streams, quickened consumption of structures and diminished deceivability. SO₂ additionally causes development of minute corrosive mist concentrates, which have genuine wellbeing ramifications and adding to environmental change.

Table 3.2 Health effects and Pollutant concentration breakpoints caused by SO₂

Category	Pollutant Concentration Breakpoints (ppb)	Health Effects
Very Good	0 - 79	No health impacts
Good	80 - 169	Damages some vegetation in combination with ozone.
Moderate	170 - 250	Damages some vegetation.
Poor	251 - 1999	Smell; increasing vegetation damage.
Very Poor	2000 or over	Increasing vulnerability for asthmatics and individual with bronchitis.

- **Carbon Monoxide (CO)**

Carbon monoxide gas is colourless, odourless, tasteless and non-irritating. It is only slightly higher than air. It is combustible but does not support combustion. It is soluble in water. In air it burns with a light blue flame to CO₂.

Sources of CO

The production of the CO in a mine may be due to any one or more of the following cases:

- **Oxidation of coal and other carbonaceous matter:** Incomplete oxidation may result in its formation and under normal mining condition, the percentage found is negligible and harmless in return of a coal mine.
- **Explosives:** Explosives contain the amount of oxygen required for complete chemical reaction, but the chemical reaction when the explosive is blasted is seldom perfect and this results in the formation of CO.
- **Spontaneous Combustion:** This is a main source of production of dangerous percentage of CO in a coal mine. Active fire in an underground mine also forms CO in dangerous percentage.
- **Methane or Coal dust Explosion:** Gases produced by the explosion of methane coal dust invariably contain a large percentage of CO.
- **Underground Machinery:** Air compressor, run faultily, and exhaust gas of internal combustion engines like diesel locomotives, are common sources of production of CO. In fact, every machine some CO if proper lubricants are not used.

Impacts of CO

CO is a very poisonous gas and its affinity for the haemoglobin of the blood is nearly 300 times that of oxygen. If CO is present even in small quantities in the inhaled air, it is difficult for blood to absorb proper quantities of oxygen to support life. CO enters the circulation system and lessens oxygen conveyance to the organs and tissues. Individuals with coronary illness are especially touchy. Introduction to abnormal states is connected with weakness of vision, work limit, learning capacity and execution of troublesome undertakings.

Table 3.3 Health effects and Pollutant concentration breakpoints caused by CO

Category	Pollutant Concentration Breakpoints (ppb)	Health Effects
Very Good	0 - 12	No health Impacts
Good	13 - 22	No health impacts.
Moderate	23 - 30	Blood chemistry changes, but no noticeable damage.
Poor	31 - 49	Increased warning sign in smokers with heart disease.
Very Poor	50 or over	Increasing warning sign in non-smokers with heart disease; blurred vision; some clumsiness.

Typical sickness symptoms due to the high concentration of the CO are mild headache, fatigue, nausea and dizziness. A CO concentration of 12-13000 ppm is dead after 1-3 minutes. A CO concentration of 1600 ppm is deadly after one hour.

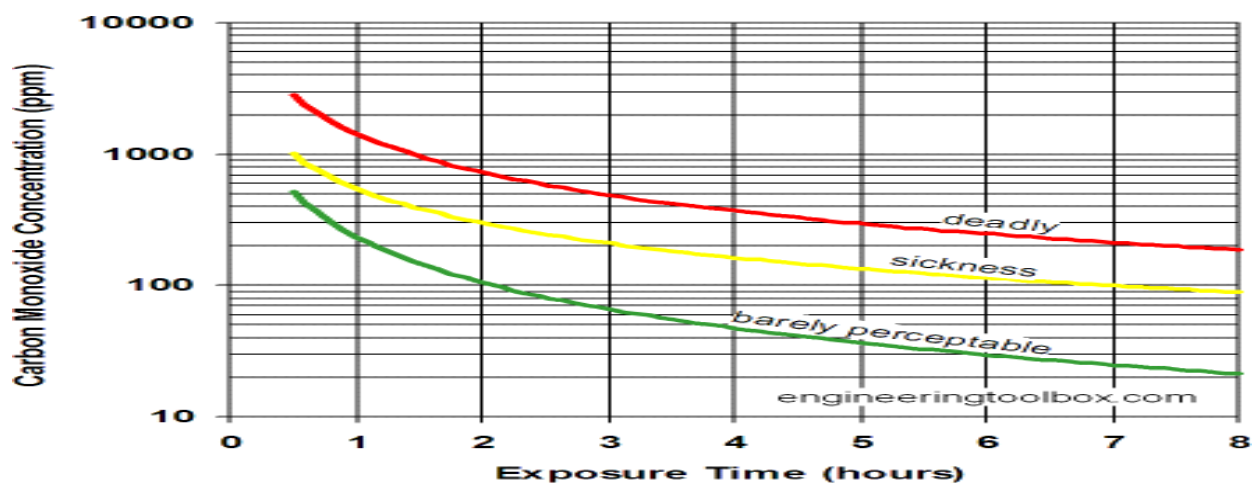


Figure 3.1 Concentration and Its exposure time

Table 3.4 Concentration of CO and Its exposure time

Conc. of CO in the air (ppm)	Breathing Time	Toxic Symptoms
9	Short term exposure	ASHRAE recommended maximum allowable concentration in living area.
35	8 hours	The maximum exposure allowed by OSHA in the workplace over an eight-hour period.
200	2-3 hours	Slight headache, tiredness, fatigue, nausea and dizziness.
400	1-2 hours	Serious headache-other symptoms intensify. Life threatening after 3 hours.
800	45 minutes	Dizziness, nausea and convulsions. Unconscious within 2 hours. Death after 2-3 hours.
1,600	20 minutes	Headache, dizziness and nausea. Death within 1 hour.
3,200	5-10 minutes	Headache, dizziness, nausea. Death within 1 hour.
6,400	1-2 minutes	Headache, dizziness, nausea. Death within 25-30 minutes.
12,800	1-3 minutes	Death within 1-3 minutes

- **Methane (CH₄)**

Methane is a colourless, odourless, tasteless, flammable gas and lighter than air. Because of the largest component of fire damp, it is commonly known as firedamp. Firedamp refers to the mixture of gases. Such mixture consists of practically methane with small traces of ethane (C₂H₆), and other hydrocarbons, such as propane (C₃H₈) and butane (C₄H₁₀).

Sources of CH₄

Methane in mine is mainly released from five sources:

- To recover methane in advance of mining from gob or goaf wells.
- From ventilation air in underground mines (dilute concentrations of methane).
- From an abandoned or closed mines, from which methane may leak out through the vent holes or through fissures or crevices in the earth.
- Extremely flammable methane (CH₄) or firedamp, as it is brought in numerous coal- fields, is framed in the last phases of coal arrangement, and due to the profundities and weights, it gets to be imbedded in the coal. As unearthing are made, methane gas is freed into the air.

Fugitive emissions from post-mining operations, in which coal keeps on give off methane as it is stacked away in pores and transported

Impacts of CH₄

Methane is a very poisonous gas. Methane gas causes headaches, reduces the oxygen level in the physical structure. If the oxygen level reduces to less than 12%, the individual can get to be unconscious and turn out to be dead in some cases. This gas symptoms are Nausea and vomiting, heart palpitations (which causes a painful sensation of the heart beating), memory loss, poor judgment, dizziness and blurred vision. Some patients also display flu-like symptoms. Methane gas is extremely inflammable. When it is burnt, carbon monoxide will be brought forth.

All above toxic gases summarized in Table 3.5.

Table 3.5 Classification of Toxic gases and their hazardous limit

Name	Primary sources in mines	Hazards	Guidelines TLVs	Flammability limits in air (%)
Methane (CH ₄)	Strata	Explosive, Breathing problem	1% isolate electrical power 2%remote personnel	5 to 15
Carbon dioxide (CO ₂)	Oxidation of carbon, fires, explosions	Increased heart rate and breathing	TWA=0.5% STEL=3.0%	
Carbon monoxide (CO)	Fires, Explosions, blasting, incomplete combustion of carbon compounds	Highly toxic, Explosive	TWA=0.05% STEL=0.04%	12.5 to 74.2
Sulphur dioxide (SO ₂)	Oxidation of Sulphides, acid water on sulphide ores	Toxic, irritant to eyes, Throat and lungs	TWA=2 ppm STEL=5 ppm	
Nitrogen dioxide (NO ₂)	IC engines, blasting, fumes, welding	Toxic, Throat and lung infections	TWA=3 ppm Ceiling: 5ppm	
Hydrogen Sulphide (H ₂ S)	Acid water on sulphides, Strata decomposition of organic materials	Highly Toxic, irritant to eyes and explosive	TWA= 10ppm STEL= 15ppm	4.3 to 45.5

TWA-- Time-weighted average (8 h shift and a 40-h work week)

STEL-- Short-term exposure limit (TWA concentration occurring more than 15 min).

Ceiling limit is the concentration that should not be exceeded at any time. This is relevant for the most toxic substances or those that produce in an immediate irritant effect.

- **Conclusion**

The chapter deals with generation and the effects of mine gases. Different hazardous gases such as NOX, SO₂, CO, CH₄ etc. has been discussed at length. The health impacts and the maximum exposure limit of each of them is described. In lieu of that a suitable monitoring system for these gases has been designed, which is mentioned in the next chapter.

CHAPTER 4

SYSTEM DESIGN

4.1 EXISTING SYSTEM

- This helmet is equipped with methane and carbon monoxide gas sensor. This sensor senses the gas and the data is transmitted to the control room wirelessly, through a wireless module called x-bee connected with the helmet.
- When the methane or carbon-monoxide gas concentration is beyond the critical level, controller in the control room triggers an alarm and keeps the plant and the workers safe by preventing an upcoming accident.
- In this paper, a coal mine safety system is implemented using a Thinger IoT platform as a medium to transmit the data. The system is implemented to monitor and control various parameters in the coal mines such as light detection, leakage of gas, temperature and humidity conditions, Fire detection in the coal mine. These all sensors are together considered as one unit and are placed in the coal

mines. All the esteems of the sensors are continuously uploaded to the thinger for analysis. Here the gas is continuously monitored if any uncertainties in the level of gas arise, then buzzer is used to alert the workers. In this system LDR sensor is utilized to detect the presence of light. Automatically light gets on and can be controlled using the LED button. In case if any fire occurs in the coal mine, then an alert notification is sent to the mail of the authorized person. Temperature and humidity values are also continuously monitored and displayed on the serial monitor and also in the thinger platform. The developed system is mainly implemented to improve the working condition inside the coal mines and also to ensure workers safety.

DRAWBACKS

- The Zigbee transmit the data from place to another place it is having some drawbacks
- it transmits the data around 300 meters only

4.2 PROPOSED SYSTEM:

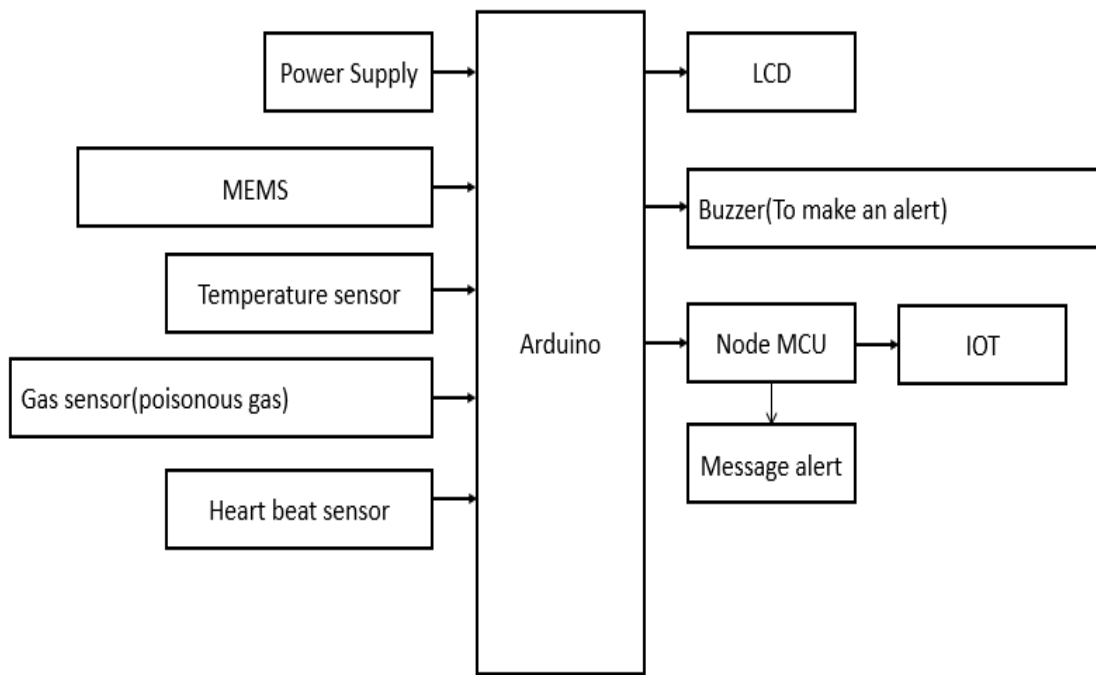
- In the proposed project we are going to implement the temperature sensor to sense the temperature of the workers, Temperature sensor to check the temperature.
- Gas sensor to check whether any poisonous gas came out during coal mining and heartbeat sensor to monitor the condition of the workers.
- The MEMS, Heart beat sensor and respiratory sensor are used to find the health parameters of the workers.
- When the information is gathered, they are monitored by using Arduino and send to the cloud
- If the critical level is happening alert message and Gmail alert is sent to an authenticated person
- The Node MCU is used to transfer the data from the board to the web page
- To monitor the concentration level of harmful gases, semiconductor gas sensors are used. Due to any reason miner's falls down and lose consciousness also

proper treatment is not provided them at that time, so number of miners are died. To overcome this problem the system provide emergency alert to the supervisor if person fall down by any reason. Some workers are not aware for safety and they are not wear helmet. A Limit switch was then used to successfully determine whether a miner has removed his helmet or not. This system also provides an early warning, which will be helpful to all miners present inside the mine to save their life before any casualty occurs. The system uses Zigbee technology and GSM for transmission of data. There is alert switch at receiver and transmitter side for emergency purpose.

ADVANTAGES:

- It can be monitored continuously.
- If any critical situation occurred an immediate message is passed to the mail or message.

4.3 BLOCK DIAGRAM:



HARDWARE REQUIREMENTS:

- Arduino
- Temperature sensor
- Gas sensor
- MEMS
- Humidity sensor
- Node MCU
- Buzzer

SOFTWARE REQUIREMENTS:

- Arduino IDE
- Embedded c Programming language

APPLICATION:

- It is used in underground works.
- It is used for coal mining

CHAPTER 5**HARDWARE REQUIREMENTS****ARDUINO UNO AND ITS PROGRAMMING:**

Arduino is a tool for making computers that can sense and control more of the physical world than your desktop computer. It's an open-source physical computing platform based on a simple microcontroller board, and a development environment for writing software for the board.

Arduino can be used to develop interactive objects, taking inputs from a variety of switches or sensors, and controlling a variety of lights, motors, and other physical outputs. Arduino projects can be stand-alone, or they can be communicating with

software running on your computer. The boards can be assembled by hand or purchased preassembled; the open-source IDE can be downloaded for free.

The Arduino programming language is an implementation of Wiring, a similar physical computing platform, which is based on the Processing multimedia programming environment.

5.1 Overview

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

This is what the Arduino board looks like.

Fig 5.10



The Duemilanove board features an Atmel ATmega328 microcontroller operating at 5 V with 2 Kb of RAM, 32 Kb of flash memory for storing programs and 1 Kb of EEPROM for storing parameters. The clock speed is 16 MHz, which translates to about executing about 300,000 lines of C source code per second. The board has 14 digital I/O pins and 6 analog input pins. There is a USB connector for talking to the host computer and a DC power jack for connecting an external 6-20 V power source, for example a 9 V battery, when running a program while not connected to the host computer. Headers are provided for interfacing to the I/O pins using 22 g solid wire or header connectors.

The Arduino programming language is a simplified version of C/C++. If you know C, programming the Arduino will be familiar. If you do not know C, no need to worry as only a few commands are needed to perform useful functions.

An important feature of the Arduino is that you can create a control program on the host PC, download it to the Arduino and it will run automatically. Remove the USB cable connection to the PC, and the program will still run from the top each time you push the reset button. Remove the battery and put the Arduino board in a closet for six months. When you reconnect the battery, the last program you stored will run. This means that you connect the board to the host PC to develop and debug your program, but once that is done, you no longer need the PC to run the program.

The Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

5.1.2 Summary:

Microcontroller ATmega328

Operating Voltage 5V

Input Voltage (recommended) 7-12V

Input Voltage (limits) 6-20V

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

Analog Input Pins 6

DC Current per I/O Pin 40 mA

DC Current for 3.3V Pin 50 mA

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

SRAM 2 KB (ATmega328)

EEPROM 1 KB (ATmega328)

Clock Speed 16 MHz

5.1.3 Pin Configuration

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall- wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

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

IOREF. This pin on the Arduino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs for working with the 5V or 3.3V.

Memory

The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM.

Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using pin Mode (), digital Write(), and digital Read() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull- up resistor (disconnected by default) of 20-50 k ohms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and Transmit (TX) TTL serial data.

These pins are connected to the corresponding pins of the ATmega8U2 USB-to- TTL Serial chip.

- **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attach Interrupt () function for details.

- **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the analog Write () function.

- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication using the SPI library.

- **LED: 13.** There is a built- in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, labelled A0 through A5, each of which provide 10 bits of resolution (i.e., 1024 different values). By default, they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and the `analogReference()` function. Additionally, some pins have specialized functionality:

- **TWI: A4 or SDA pin and A5 or SCL pin.** Support TWI communication using the `Wire` library.

There are a couple of other pins on the board:

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

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

5.1.4 Communication

Microcontrollers depend on a host computer for developing and compiling programs. The software used on the host computer is known as an integrated development environment, or IDE. For the Arduino, the development environment is based on the open-source Processing platform (www.processing.org) which is described by its creators as a “programming language and environment for people who want to program images, animation, and interactions. “The Arduino programming language leverages an open-source project known as Wiring (wiring.org.co). The Arduino language is based on good old- fashioned C. If you are unfamiliar with this language, don’t worry; it’s not hard to learn, and the Arduino IDE provides some feedback when you make mistakes in your programs. The Arduino Uno has a number of facilities for communicating with a

computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a inf file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A Software Serial library allows for serial communication on any of the Uno's digital pins. The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation for details. For SPI communication, use the SPI library. As you go through the list of programming statements available in the Arduino IDE (choose Help->Reference), you might think there isn't much power for doing things like running servos, operating stepper motors, reading potentiometers, or displaying text on an LCD. Like most any language based on C, the Arduino supports the notion of "libraries" code

Repositories that extend core programming functionality. Libraries let you re- use code without having to physically copy and paste it into all your programs. The standard Arduino software installation comes with several libraries you may use, and you can download others from the Arduino support pages and from third-party websites that publish Arduino library code. A good example of a library you'll use with the Robot and likely many other robot projects is

NODE MCU

NodeMCU is an open-source Lua based firmware and **development board** specially targeted for IoT based Applications. It includes firmware that runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module.

Programming NodeMCU ESP8266 with Arduino IDE

The NodeMCU Development Board can be easily programmed with Arduino IDE since it is easy to use.

Programming NodeMCU with the Arduino IDE will hardly take 5-10 minutes. All you need is the Arduino IDE, a USB cable and the NodeMCU board itself. You can check this [Getting Started Tutorial for NodeMCU](#) to prepare your Arduino IDE for NodeMCU.

NodeMCU is an open-source firmware for which open source [prototyping](#) board designs are available. The name "NodeMCU" combines "[node](#)" and "MCU" ([micro-controller](#) unit). The term "NodeMCU" strictly speaking refers to the firmware rather than the associated [development kits](#).

Both the firmware and prototyping board designs are [open source](#).

The firmware uses the [Lua](#) scripting language. The firmware is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. It uses many open-source projects, such as lua-cjson and [SPIFFS](#). Due to resource constraints, users need to select the modules relevant for their project and build a firmware tailored to their needs. Support for the 32-bit [ESP32](#) has also been implemented.

The prototyping hardware typically used is a circuit board functioning as a [dual in-line package](#) (DIP) which integrates a USB controller with a smaller surface-mounted board containing the MCU and antenna. The choice of the DIP format allows for easy prototyping on [breadboards](#). The design was initially based on the ESP-12 module of the [ESP8266](#), which is a Wi-Fi SoC integrated with a [Tensilica](#) Xtensa LX106 core, widely used in IoT applications (see [related projects](#)).

History

NodeMCU was created shortly after the [ESP8266](#) came out. On December 30, 2013, [Espressif Systems](#) began production of the ESP8266. NodeMCU started on 13 Oct 2015, when Hong committed the first file of nodemcu-firmware to GitHub. Two months later, the project expanded to include an open-hardware platform when developer Huang R committed the Gerber file of an ESP8266 board, named devkit

v0.9. Later that month, Tuan PM ported MQTT client library from Contiki to the ESP8266 SoC platform, and committed to NodeMCU project, then NodeMCU was able to support the MQTT IoT protocol, using Lua to access the MQTT broker. Another important update was made on 30 Jan 2015, when Devsaurus ported the u8glib to the NodeMCU project, enabling NodeMCU to easily drive LCD, Screen, OLED, even VGA displays.

In the summer of 2015, the original creators abandoned the firmware project and a group of independent contributors took over. By the summer of 2016 the NodeMCU included more than 40 different modules.

ESP8266 Arduino Core

As Arduino.cc began developing new MCU boards based on non-AVR processors like the ARM/SAM MCU and used in the Arduino Due, they needed to modify the Arduino IDE so that it would be relatively easy to change the IDE to support alternate toolchains to allow Arduino C/C++ to be compiled for these new processors. They did this with the introduction of the Board Manager and the SAM Core. A "core" is the collection of software components required by the Board Manager and the Arduino IDE to compile an Arduino C/C++ source file for the target MCU's machine language. Some ESP8266 enthusiasts developed an Arduino core for the ESP8266 Wi-Fi SoC, popularly called the "ESP8266 Core for the Arduino IDE".^[17] This has become a leading software development platform for the various ESP8266-based modules and development boards, including NodeMCUs.

NodeMCU is an open-source LUA based firmware developed for the ESP8266 Wi-Fi chip. By exploring functionality with the ESP8266 chip, NodeMCU firmware comes with the ESP8266 Development board/kit i.e., NodeMCU Development board.

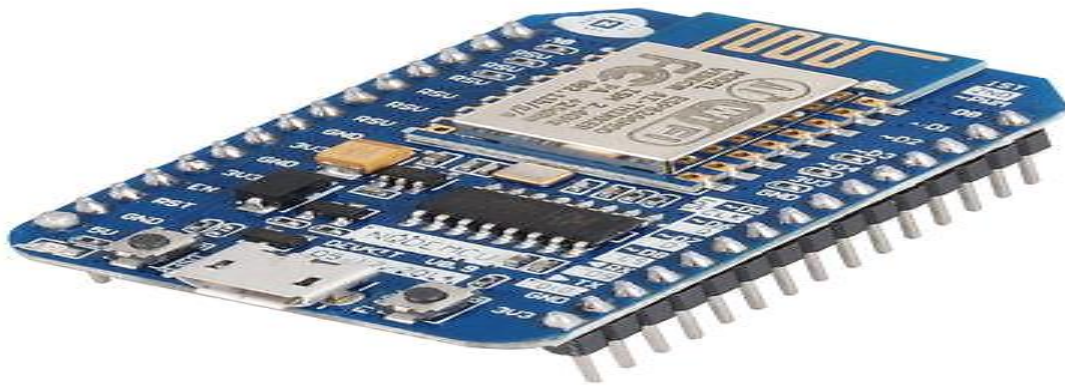


Fig 5.2

NodeMCU Development Board/kit v0.9 (Version1)

Since NodeMCU is an open-source platform, its hardware design is open for edit/modify/build.

NodeMCU Dev Kit/board consist of ESP8266 wifi enabled chip. The **ESP8266** is a low-cost Wi-Fi chip developed by Espressif Systems with TCP/IP protocol. For more information about ESP8266, you can refer to the ESP8266 HYPERLINK "<https://www.electronicwings.com/sensors-modules/esp8266-wifi-module>" WiFi HYPERLINK "<https://www.electronicwings.com/sensors-modules/esp8266-wifi-module>" Module.

There is Version2 (V2) available for NodeMCU Dev Kit i.e., **NodeMCU Development Board v1.0 (Version2)**, which usually comes in black coloured PCB.

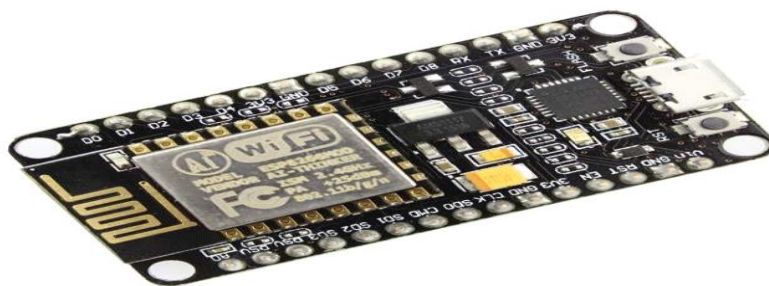


Fig 5.21

NodeMCU Development Board/kit v1.0 (Version2)

NodeMCU Dev Kit has **Arduino like** Analog (i.e., A0) and Digital (D0-D8) pins on its board.

It supports serial communication protocols i.e., UART, SPI, I2C, etc.

Using such serial protocols, we can connect it with serial devices like I2C enabled LCD display, Magnetometer HMC5883, MPU-6050 Gyro meter + Accelerometer, RTC chips, GPS modules, touch screen displays, SD cards, etc.

How to start with NodeMCU?

NodeMCU Development board is featured with wifi capability, analog pin, digital pins, and serial communication protocols.

To get started with using NodeMCU for IoT applications first we need to know about how to write/download NodeMCU firmware in NodeMCU Development Boards. And before that where this NodeMCU firmware will get as per our requirement.

There are online NodeMCU custom builds available using which we can easily get our custom NodeMCU firmware as per our requirement.

How to write codes for NodeMCU?

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

NodeMCU with ESPlorer IDE

Lua scripts are generally used to code the NodeMCU. Lua is an open-source, lightweight, embeddable scripting language built on top of C programming language.

For more information about how to write Lua script for NodeMCU refer to [Getting started with NodeMCU using ESPlorerIDE](#)

NodeMCU with Arduino IDE

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

The difference in using ESPlorer and Arduino IDE

Well, there is a programming language difference we can say while developing an application for NodeMCU using ESPlorer IDE and Arduino IDE. We need to code in C\C++ programming language if we are using Arduino IDE for developing NodeMCU applications and Lua language if we are using ESPlorer IDE.

Basically, NodeMCU is Lua Interpreter, so it can understand Lua script easily. When we write Lua scripts for NodeMCU and send/upload it to NodeMCU, then they will get executed sequentially. It will not build a binary firmware file of code for NodeMCU to write. It will send Lua script as it is to NodeMCU to get executed.

In Arduino IDE when we write and compile code, the ESP8266 toolchain in the background creates a binary firmware file of code we wrote. And when we upload it to NodeMCU then it will flash all NodeMCU firmware with newly generated binary firmware code. In fact, it writes the complete firmware.

That's the reason why NodeMCU not accept further Lua scripts/code after it is getting flashed by Arduino IDE. After getting flashed by Arduino sketch/code it will be no more Lua interpreter and we got an error if we try to upload Lua scripts. To again start with Lua script, we need to flash it with NodeMCU firmware.

Since Arduino IDE compiles and upload/writes complete firmware, it takes more time than ESPlorer IDE.

NodeMCU ESP8266 Specifications & Features

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

Gas sensor

In current technology scenario, monitoring of gases produced is very important. From home appliances such as air conditioners to electric chimneys and safety systems at industries monitoring of gases is very crucial. Gas sensors are very important part of such systems. Small like a nose, gas sensors spontaneously react to the gas present, thus keeping the system updated about any alterations that occur in the concentration of molecules at gaseous state.

Gas sensors are available in wide specifications depending on the sensitivity levels, type of gas to be sensed, physical dimensions and numerous other factors. This Insight covers a methane gas sensor that can sense gases such as ammonia which might get produced from methane. When a gas interacts with this sensor, it is first ionized into its constituents and is then adsorbed by the sensing element. This adsorption creates a potential difference on the element which is conveyed to the processor unit through output pins in form of current. What is this sensing element? Is it kept in some chamber or is kept exposed? How does it get current and how it is taken out? Let's find out in this Insight!!!



Fig 5.3

The gas sensor module consists of a steel exoskeleton under which a sensing element is housed. This sensing element is subjected to current through connecting leads. This current is known as heating current through it, the gases coming close to the sensing element get ionized and are absorbed by the sensing element. This changes the resistance of the sensing element which alters the value of the current going out of it.



Fig 5.31

Image shows externals of a standard gas sensor module: a steel mesh, copper clamping ring and connecting leads. The top part is a stainless-steel mesh which takes care of the following:

- Protecting the insides of the sensor.
- Exhibits an anti-explosion network that keeps the sensor module intact at high temperatures and gas pressures.

In order to manage above listed functions efficiently, the steel mesh is made into two layers. The mesh is bound to rest of the body via a copper plated clamping ring.

The connecting leads of the sensor are thick so that sensor can be connected firmly to the circuit and sufficient amount of heat gets conducted to the inside part. They are casted from copper and have tin plating over them. Four of the six leads (A, B, C, D) are for signal fetching while two (1, 2) are used to provide sufficient heat to the sensing element.

The pins are placed on a Bakelite base which is a good insulator and provides firm gripping to the connecting leads of the sensor. The top of the gas sensor is removed off to see the internals parts of the sensor: sensing element and connection wiring. The hexapod structure is constituted by the sensing element and six connecting legs that extend beyond the Bakelite base.

Image shows the hollow sensing element which is made up from Aluminium Oxide based ceramic and has a coating of tin oxide. Using a ceramic substrate increases the heating efficiency and tin oxide, being sensitive towards adsorbing desired gas' components (in this case methane and its products) suffices as sensing coating.

The leads responsible for heating the sensing element are connected through Nickel-Chromium, well known conductive alloy. Leads responsible for output signals are connected using platinum wires which convey small changes in the current that passes through the sensing element. The platinum wires are connected to the body of the sensing element while Nickel-Chromium wires pass through its hollow structure. While

other wires are attached to the outer body of the element, Nickel-Chromium wires are placed inside the element in a spring shaped. Image shows coiled part of the wire which is placed on the inside of the hollow ceramic.

Image shows the ceramic with tin dioxide on the top coating that has good adsorbing property. Any gas to be monitored has specific temperature at which it ionizes. The task of the sensor is to work at the desired temperature so that gas molecules get ionized. Through Nickel-chromium wire, the ceramic region of the sensing element is subjected to heating current. The heat is radiated by the element in the nearby region where gases interact with it and get ionized. Once, ionized, they are absorbed by the tin dioxide. Adsorbed molecules change the resistance of the tin dioxide layer. This changes the current flowing through the sensing element and is conveyed through the output leads to the unit that controls the working of the gas sensor.

Micro-electromechanical systems (MEMS)

Micro-electromechanical systems (MEMS) is a technology that combines computers with tiny mechanical devices such as sensors, valves, gears, mirrors, and actuators embedded in semiconductor chips. MEMS or what he calls analogy computing will be "the foundational technology of the next decade." MEMS is also sometimes called smart matter.

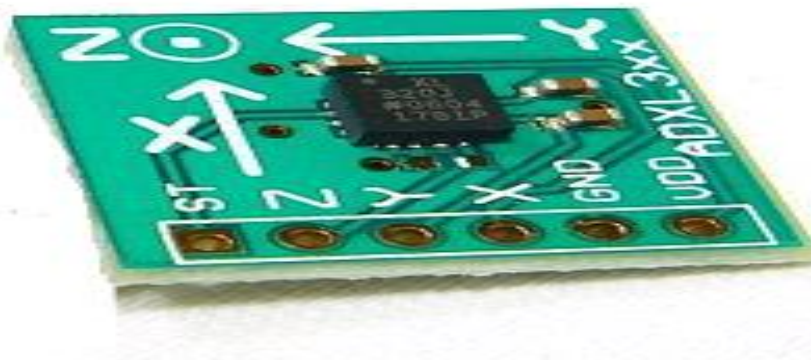


Fig 5.4

MEMS are already used as accelerometers in automobile air-bags. They've replaced a less reliable device at lower cost and show promise of being able to inflate a bag not only on the basis of sensed deceleration but also on the basis of the size of the person they are protecting. Basically, a MEMS device contains micro-circuitry on a tiny silicon chip into which some mechanical device such as a mirror or a sensor has been manufactured. Potentially, such chips can be built in large quantities at low cost, making them cost-effective for many uses.

Among the presently available uses of MEMS or those under study are:

- Global position system sensors that can be included with courier parcels for constant tracking and that can also sense parcel treatment en route
- Sensors built into the fabric of an airplane wing so that it can sense and react to air flow by changing the wing surface resistance; effectively creating a myriad of tiny wing flaps
- Optical switching devices that can switch light signals over different paths at 20-nanosecond switching speeds
- Sensor-driven heating and cooling systems that dramatically improve energy savings
- Building supports with imbedded sensors that can alter the flexibility properties of a material based on atmospheric stress sensing

Applications:

- Vibration measurement and Tilt inclination (to adjust rotation speed, circuit breaker security, fault prediction),
- Professional and domestic Alarms
- Maintenance management and Asset Tracking for shocks monitoring
- Alarm and Anti Tamper function

TEMPERATURE SENSOR (LM35)

FEATURES DESCRIPTION

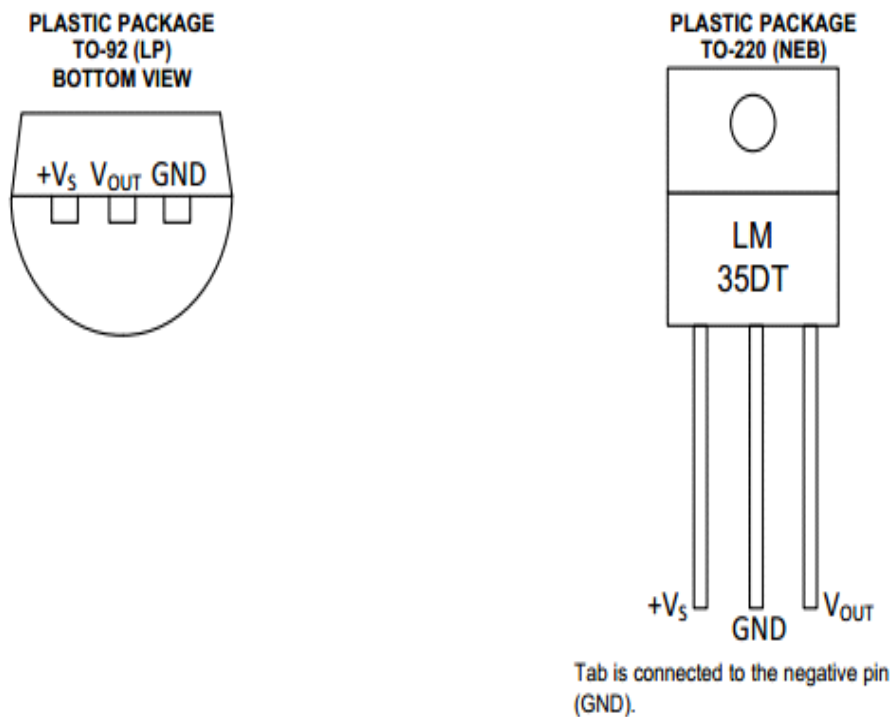
- Calibrated Directly in ° Celsius (Centigrade)
- Linear + 10 mV/°C Scale Factor
- 0.5°C Ensured Accuracy (at +25°C) LM35 has an advantage over linear temperature
- Rated for Full –55°C to +150°C Range
- Suitable for Remote Applications
- Low Cost Due to Wafer-Level Trimming
- Less than 60-μA Current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

DESCRIPTION

The LM35 series are precision integrated-circuit temperature sensors, with an output voltage linearly proportional to the Centigrade temperature. Thus, the LM35 has an advantage over linear temperature sensors calibrated in ° Kelvin, as the user is not required to subtract a large constant voltage from the output to obtain convenient Centigrade scaling. The LM35 does not require any external calibration or trimming to provide typical accuracies of $\pm\frac{1}{4}^{\circ}\text{C}$ at room temperature and $\pm\frac{3}{4}^{\circ}\text{C}$ over a full –55°C to +150°C temperature range. Low cost is assured by trimming and calibration at the wafer level. The low output impedance, linear output, and precise inherent calibration of the LM35 make interfacing to readout or control circuitry especially easy. The device is used with single power supplies, or with plus and minus supplies. As the LM35 draws only 60 μA from the supply, it has very low self-heating of less than 0.1°C in still air. The LM35 is rated to operate over a –55°C to +150°C temperature range, while the LM35C

is rated for a -40°C to $+110^{\circ}\text{C}$ range (-10° with improved accuracy). The LM35 series is available packaged in hermetic TO transistor packages, while the LM35C, LM35CA, and LM35D are also available in the plastic TO-92 transistor package. The LM35D is also available in an 8-lead surface-mount small outline package and a plastic TO-220 package.

Fig 5.5



HEARTBEAT SENSOR

A pulse wave is the change in the volume of a blood vessel that occurs when the heart pumps blood, and a detector that monitors this volume change is called a pulse sensor. First, there are four main ways to measure heart rate: electrocardiogram, photoelectric pulse wave, blood pressure measurement, and phonocardiography. Pulse sensors use the photoelectric method.

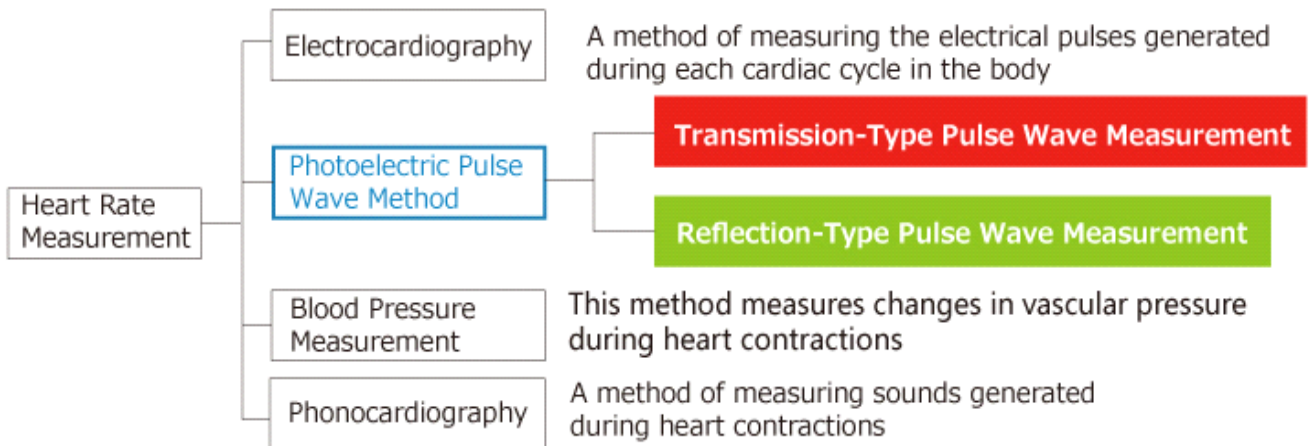


fig 5.6

Pulse sensors using the photoelectric pulse wave method are classified into 2 types depending on the measurement method: transmission and reflection. Transmission types measure pulse waves by emitting red or infrared light from the body surface and detecting the change in blood flow during heart beats as a change in the amount of light transmitted through the body. This method is limited to areas where light can easily penetrate, such as the fingertip or earlobe. ROHM is currently developing a reflection-type pulse sensor (Optical Sensor for Heart Rate Monitor). The reflection-type pulse sensor (Optical Sensor for Heart Rate Monitor) is explained below.

Reflection-Type Pulse Sensor (Optical Sensor for Heart Rate Monitor)

Reflection-type pulse sensors (Optical Sensors for Heart Rate Monitor) emit infrared, red, or green light (~550nm) towards the body and measure the amount of light reflected using a photodiode or phototransistor. Oxygenated haemoglobin present in the blood of the arteries has the characteristic of absorbing incident light, so by sensing the blood flow rate (change in blood vessel volume) that changes following heart contractions over time we are able to measure the pulse wave signal. Also, since reflected light is measured, the range of suitable areas is not limited as with transmission-type pulse sensors.

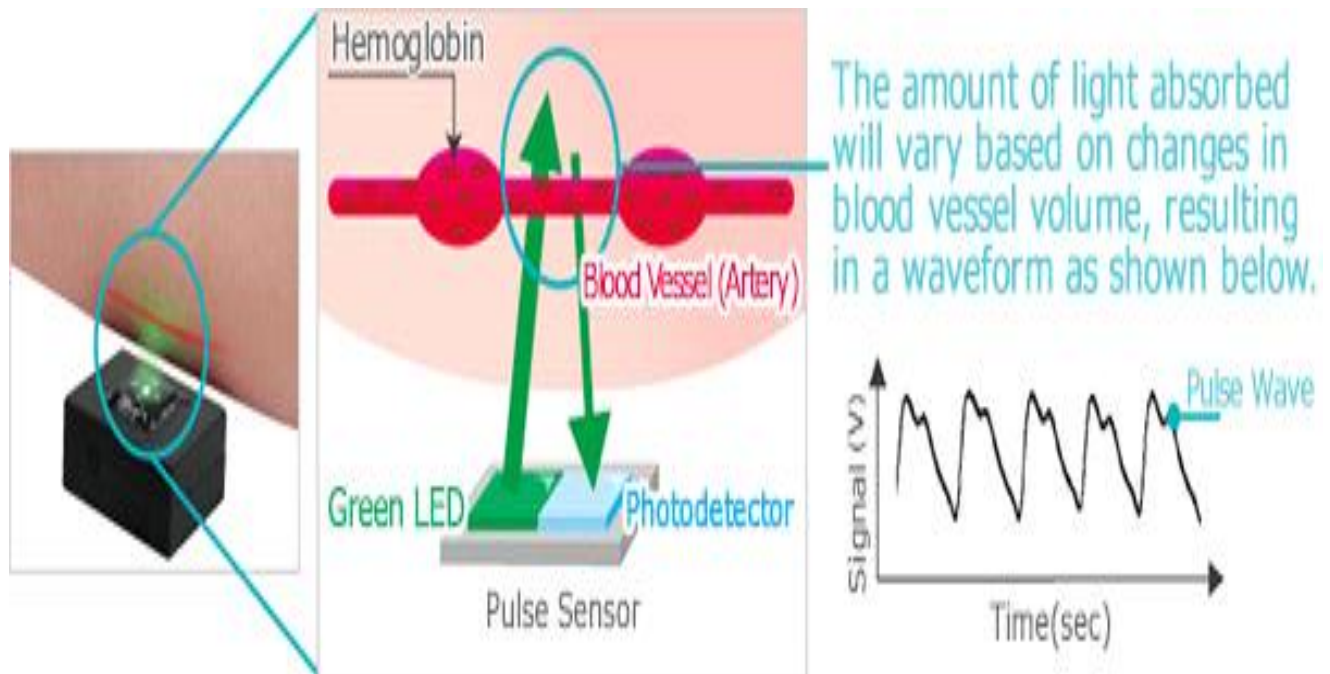


fig 5.61

Pulse wave measurement using red or infrared light can be affected by infrared rays contained in sunlight (i.e., outdoors), preventing stable operation. For this reason, usage indoors or semi-indoors is recommended. For pulse wave measurement outdoors (i.e., by smart watches), a green light source which has a high absorption rate in haemoglobin and less susceptibility to ambient light is preferred, so ROHM utilizes green LEDs as transmission light sources.

CHAPTER 6

SOFTWARE REQUIREMENTS:

Arduino IDE:

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

Features:

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

Working:

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

Fig 6.10



Step 2. Writing the code in the preferred language.

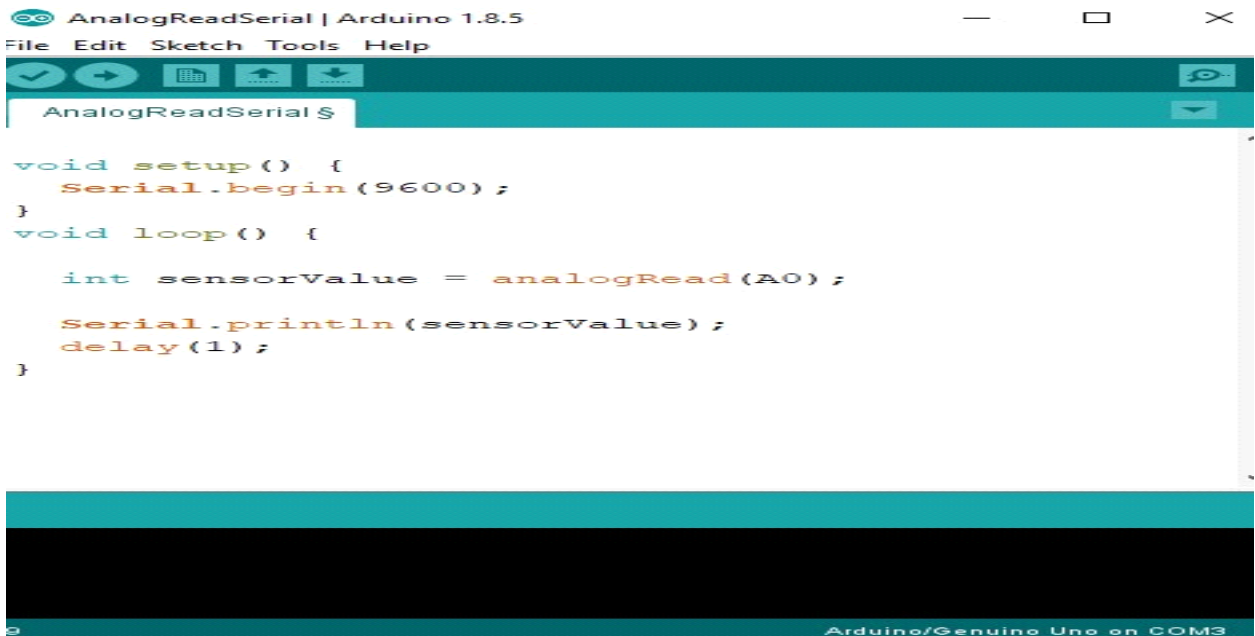


fig 6.11

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

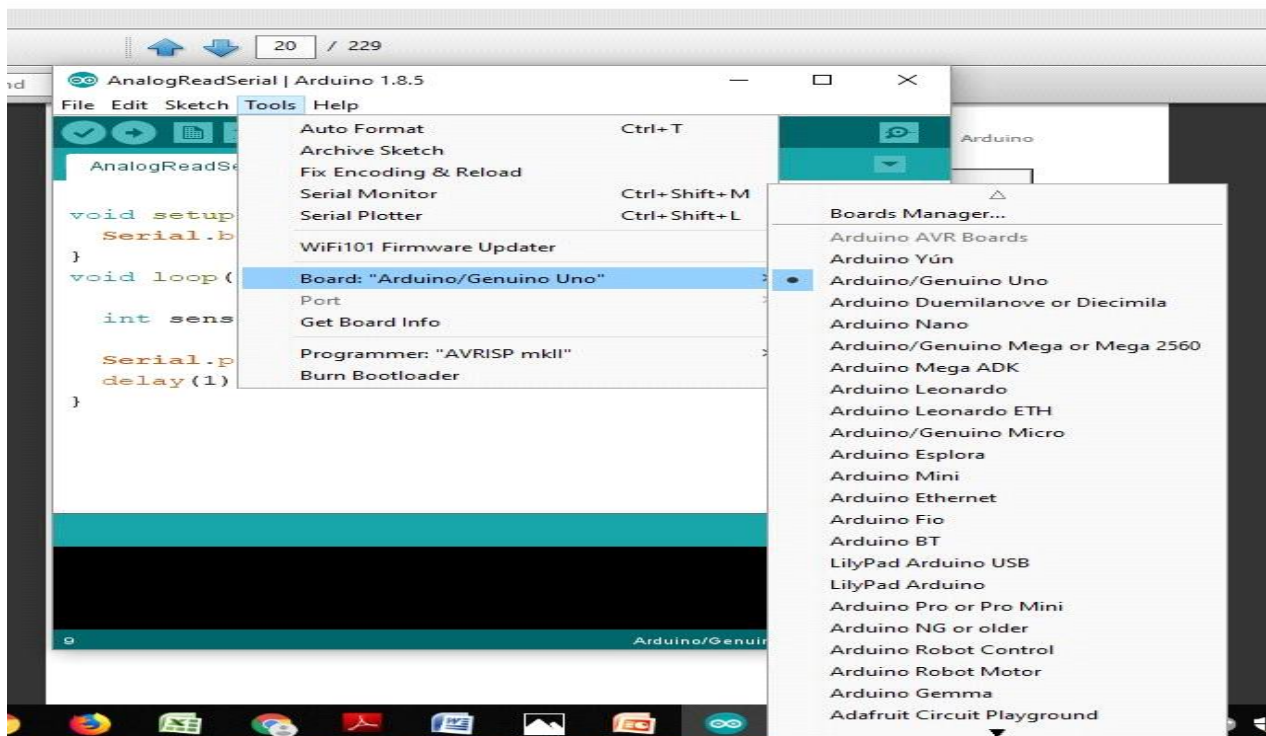


Fig 6.12

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

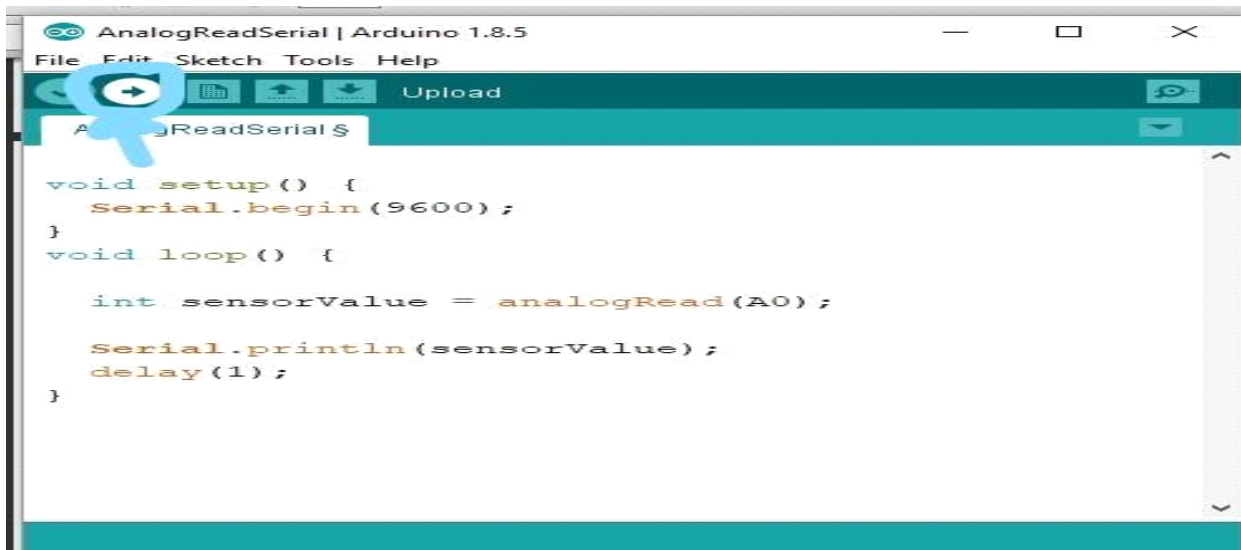


fig 6.13

IOT

The structure of this chapter is the following: Section 1.1 defines the motivation for this work. Section 1.2 shows the research objectives and questions. Section 1.3 presents the steps required to achieve our objectives. Section 1.4 shows the structure of the report

- ***Background***

The Future Internet goal is to provide an infrastructure to have an immediate access to information about the physical world and its objects. Physical objects can be applicable to different application domains, such as e-health, warehouse management, etc. Each application domain may have different types of physical devices. Each physical device can have its own specifications, which is required to use in order to interact with it. To achieve the future Internet goal, a layered vision is required that can facilitate data access. Internet of Things (IoT) is a vision that aims to integrate the virtual world of information to the real world of devices through a layered architecture.

The term „Internet of Things“ consists of two words, namely *Internet* and *Things*. *Internet* refers to the global network infrastructure with scalable, configurable

capabilities based on interoperable and standard communication protocols. *Things* are physical objects or devices, or virtual objects, devices or information, which have identities, physical attributes and virtual personalities, and use intelligent interfaces. For instance, a virtual object can represent an abstract unit of sensor nodes that contains metadata to identify and discover its corresponding sensor nodes. Therefore, IoT refers to the *things* that can provide information from the physical environment through the Internet.

Middleware is as an interface between the hardware layer and the application layer, which is responsible for interacting with devices and information management. The role of a middleware is to present a unified programming model to interact with devices. A middleware is in charge of masking the heterogeneity and distribution problems that we face when interacting with devices.

Motivation

IoT-based system is in charge of providing knowledge from an environment to a non-expert user. IoT-based system can be used in different environments, so it needs to be able to address many heterogeneous devices. Thus, a major concern within developing an IoT-based system is how to handle the interaction with the heterogeneous devices for non-expert users. This concern can be addressed by a middleware layer between devices and non-expert users. This layer is responsible to hide the diversity of devices from the user perspective, and provides access transparency to the devices for the end users.

The idea of creating abstractions of devices been addressed in the literature. The middleware we found in the literature can provide satisfaction by facilitating the interaction with devices, but they do not support low-level device configuration.

IoT definition

In this section, we explain some of the IoT definitions. Also, we explain the layered architecture for IoT.

Internet of Things (IoT) has increasingly gained attention in industry to interact with different types of devices. IoT can have influence on industry and society by integrating physical devices into information networks. IoT impacts can be on different perspectives, namely for private and business users. From the perspective of a private user, IoT has effect on both working and personal fields, such as smart homes and offices, e-health and assisted living. From the aspect of a business user the impacts would be in fields such as automation and industrial manufacturing, logistics, business process management, intelligent transportation of people and goods.

IoT integrates physical things into information networks. IoT covers the overall infrastructure, including software, hardware and services, which is used to support these information networks. The integrated physical things can exchange data about the physical properties and information that they sense in their environment. To identify devices, we can use identification technologies like for example RFID, which allow each device be uniquely identified.

International Telecommunication Union (ITU)¹ defines IoT as “*A global infrastructure for the Information Society, enabling advanced services by interconnecting (physical and virtual) things based on, existing and evolving, interoperable information and communication technologies*”.

IoT has a layered architecture designed to answer the demands of various industries, enterprises and society. Fig. 2.1 shows a generic layered architecture for IoT that consist of five layers, which are discussed, in the following:

- Edge Technology layer

This is a hardware layer that consists of embedded systems, RFID tags, sensor networks and all of the other sensors in different forms. This hardware layer can perform several functions, such as collecting information from a system or an environment, processing information and supporting communication.

- Access Gateway layer

This layer is concerned with data handling, and is responsible for publishing and subscribing the services that are provided by the *Things*, message routing, and hovelling the communication between platforms.

- Middleware layer

This layer has some critical functionalities, such as aggregating and filtering the received data from the hardware devices, performing information discovery and providing access control to the devices for applications.

- Application layer

This layer is responsible for delivering various application services. These services are provided through the middleware layer to different applications and users in IoT-based systems. The application services can be used in different industries such as, logistics, retail, healthcare, etc.

PROTEUS

PROTEUS SOFTWARE:

Proteus 8 is a best simulation software for various designs with microcontroller. It is mainly popular because of availability of almost all microcontrollers in it. So, it is a handy tool to test programs and embedded designs for electronics hobbyist. You can simulate your programming of microcontroller in Proteus 8 Simulation Software. After simulating your circuit in Proteus 8 Software you can directly make PCB design with it so it could be all in one package for students and hobbyists. So, I think now you have a little bit idea about what is proteus software.

Proteus is a Virtual System Modelling (VSM) that combines circuit simulation, animated components and microprocessor models to co-simulate the complete microcontroller-based designs. This is the perfect tool for engineers to test their microcontroller designs before constructing a physical prototype in real time. This program allows users to

interact with the design using on-screen indicators and LED and LCD displays and, if attached to the PC, switches and buttons.

Proteus comes with extensive debugging features, single stepping and variable display for a neat design prior to hardware prototyping. Proteus is the program to use when you want to simulate the interaction between software running on a microcontroller and any analog or digital electronic device connected to it.

Proteus was initially created as a multiplatform (DOS, Windows, Unix) system utility, to manipulate text and binary files and to create CGI scripts. The language was later focused on Windows, by adding hundreds of specialized functions for: network and serial communication, database interrogation, system service creation, console applications, keyboard emulation, ISAPI scripting (for IIS). Most of these additional functions are only available in the Windows flavor of the interpreter, even though a Linux version is still available.

Proteus was designed to be practical (easy to use, efficient, complete), readable and consistent.

Its strongest points are:

- powerful string manipulation;
- comprehensibility of Proteus scripts;
- Availability of advanced data structures: arrays, queues (single or double), stacks, bit maps, sets, AVL trees.
- The language can be extended by adding user functions written in Proteus or DLLs created in C/C++.

GET PROTEUS 8 SOFTWARE?

After getting introduction about what is proteus software if you wish to download and give a try to proteus 8 software then lab center is giving latest version of proteus, proteus 8 demo version for free of cost. The latest Proteus Software version is [Proteus 8](#). You can download it from [here](#).

EXPLANATION:

If you have clear idea about what is proteus 8 simulation software then you can start proteus 8 simulations at this point of time. I am sure that you have your proteus 8. If you don't have then download from above link. To learn Proteus 8 You can start with a series of [video tutorials](#) or you can download a [Proteus 8 Software Tutorial PDF](#). But main and most important thing with this software is hands on experiments. the more you work the better you become There is no shortcut.

After making your Schematic you have to do PCB designing so here is the video tutorial which describes [Single layer auto routing](#).

At first sight, Proteus may appear similar to Basic because of its straight syntax, but similarities are limited to the surface:

LANGUAGE FEATURES:

Proteus has a fully functional, procedural approach; variables are untyped, do not need to be declared, can be local or public and can be passed by value or by reference; all the typical control structures are available (if-then-else; for-next; while-loop; repeat-until; switch-case); new functions can be defined and used as native functions.

Data types supported by Proteus are only three: integer numbers, floating point numbers and strings. Access to advanced data structures (files, arrays, queues, stacks, AVL trees, sets and so on) takes place by using handles, i.e., integer numbers returned by item creation functions.

Type declaration is unnecessary: variable type is determined by the function applied – Proteus converts on the fly every variable when needed and holds previous data renderings, to avoid performance degradation caused by repeated conversions.

There is no need to add parenthesis in expressions to determine the evaluation order, because the language is fully functional (there are no operators).

Proteus includes hundreds of functions for:

- accessing file system;
- sorting data;
- manipulating dates and strings;
- interacting with the user (console functions)
- Calculating logical and mathematical expressions.

Proteus supports associative arrays (called sets) and AVL trees, which are very useful and powerful to quickly sort and lookup values.

The functional approach and the extensive library of built-in functions allow to write very short but powerful scripts; to keep them comprehensible, medium-length keywords were adopted. The user, besides writing new high-level functions in Proteus, can add new functions in C/C++ by following the guidelines and using the templates available in the software development kit; the new functions can be invoked exactly the same way as the predefined ones, passing expressions by value or variables by reference. Proteus is an interpreted language: programs are loaded into memory, pre-compiled and run; since the number of built-in functions is large, execution speed is usually very good and often comparable to that of compiled programs. One of the most interesting features of Proteus is the possibility of running scripts as services or ISAPI scripts. Running a Proteus script as a service, started as soon as the operating system has finished loading, gives many advantages: no user needs to login to start the script; a service can be run with different privileges so that it cannot be stopped by a user. This is very useful

to protect critical processes in industrial environments (data collection, device monitoring), or to avoid that the operator inadvertently closes a utility (keyboard emulation). The ISAPI version of Proteus can be used to create scripts run through Internet Information Services and is equipped with specific functions to cooperate with the web server.

For intellectual property protection Proteus provides:

digital signature of the scripts, by using the development key (which is unique); the option to enable or disable the execution of a script (or part of it) by using the key of the customer. Proteus is appreciated because it is relatively easy to write short, powerful and comprehensible scripts; the large number of built-in functions, together with the examples in the manual, keep low the learning curve.

The development environment includes a source code editor with syntax highlighting and a context-sensitive guide. Proteus does not need to be installed: the interpreter is a single executable (below 400 Kb) that does not require additional DLLs to be run on recent Windows systems.

HOW TO SETUP PROTEUS?

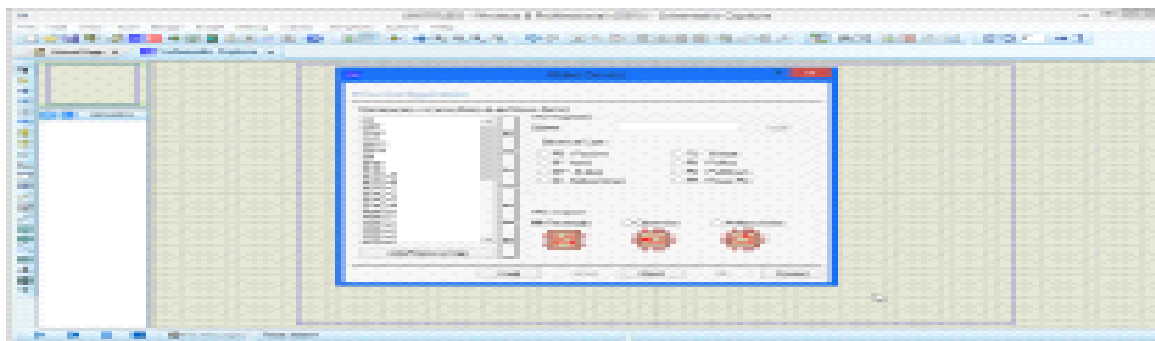
1. Start-up the Microsoft Windows.
2. Place the ECOM 4315 CD into CD ROM drive.
3. Double click on the software tools folder and then choose Proteus Simulator after that double click Proteus 7.1 folder and run the setup.
4. Press next until you reach the window which ask for the key.
5. From browse for key; browse until you reach the same folder where the setup exists, and then open the second folder and chose the MAXIM_LICENCE folder.
6. Then click on the top icon which is MAXIM and then press install.
7. Then choose yes and then close the window, after that browse and install the program.

8. After that run the patch which exist in the same folder were the MAXIM_LICENCE exists.
9. Then browse for c:\Program files\Lab centre electronics \Proteus.
10. Finally chose next, and then finish.

GET STARTED TO PROTEUS

1. From start menu chose the PROTEUS, and then chose ISIS (blue).
2. To get a part click on Devices (P).
3. Write the name of the PIC16F84A.
4. To get LED write led then chose green led.
5. To get a resistor write 1k and chose the first part. Device Libraries

Proteus capture comes with pre-supplied libraries for TTL, CMOS, ECL, Microprocessor, Memory and Analogue ICs plus libraries with hundreds of named Bipolar, FET and Diode discrete semiconductors. Also included are manufacturer specific libraries from National Semiconductor, Philips, Motorols, Teccor, Texas Instruments, Dallas and Zetex amongst others - an expanding total, currently over 10,000 parts. Fig 6.20



Importing Components from BSDL

Direct import of BSDL (Boundary Scan Description Library) files provides a semi-automatic way to quickly create and package larger components.

Libraries provided with Proteus also contain default properties for PCB packaging and/or simulator models. Both homogenous and heterogeneous multi-element devices are supported so you can place a relay's coil and contacts on separate parts of the drawing. Connector pins can also be distributed. Device editing is performed directly on the drawing using the general editing tools. Devices may be constructed from lines, boxes, circles, arcs, text and special pin objects. Non-electrical symbols may also be created for use in small mechanical or block diagrams.

A summary listing of included embedded peripheral libraries can be found [here](#).

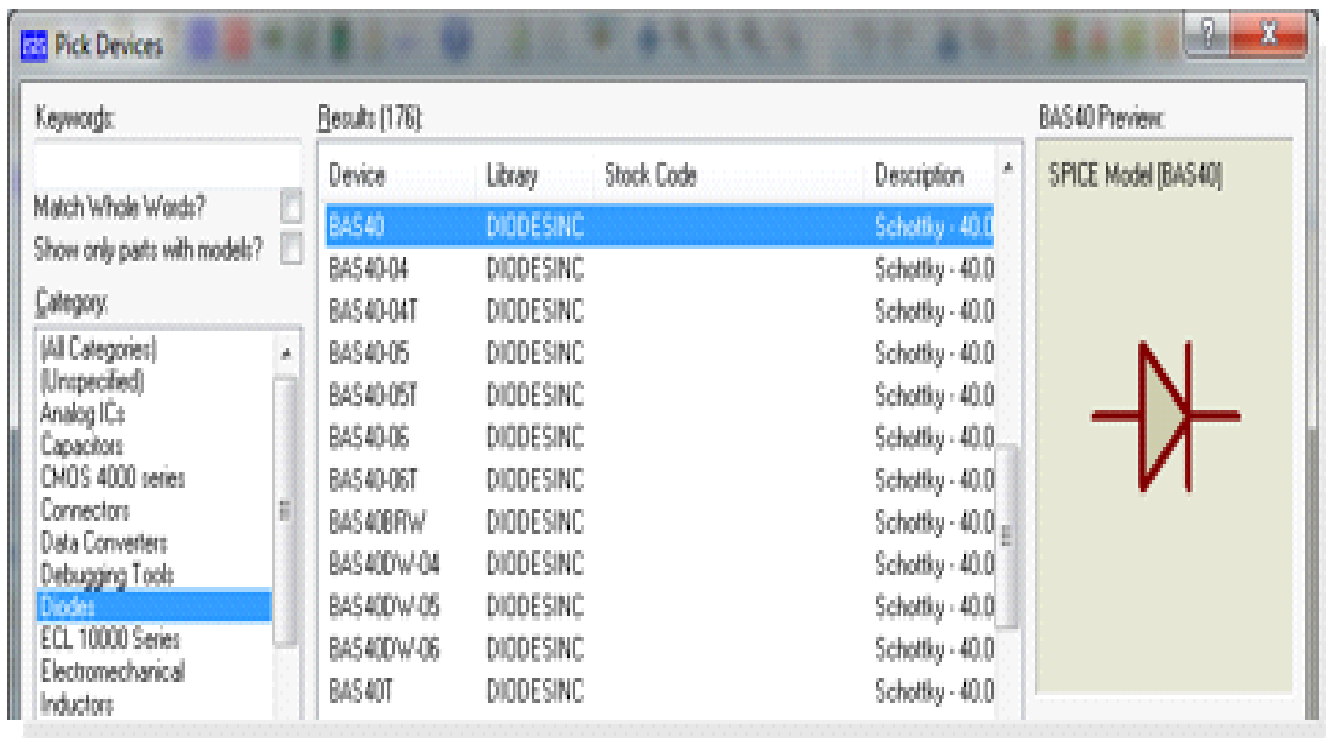


fig 6.21

UBIDOTS:

Ubidots is an IoT Platform empowering innovators and industries to prototype and scale IoT projects to production. Use the Ubidots platform to send data to the cloud from any Internet-enabled device. You can then configure actions and alerts based on your real-time data and unlock the value of your data through visual tools. Ubidots offers a REST API that allows you to read and write data to the resources available: data sources,

variables, values, events and insights. The API supports both HTTP and HTTPS and an API Key is required.

Your data will be protected with two more replication, encrypted storage and optional TLS/SSL data support. You can also customize permission groups to each module of the platform, making sure the right information is shown to the right user.

Born as an engineering services firm in 2012, Ubidots delivered end-to-end IoT solutions in tandem with its partner and co-founding company [Netux](#), to remotely monitor, control, and automate processes for healthcare clients as well funded start-ups and Fortune 1,000s in the American Southeast and across Latin America.

Between 2012 and 2014, Ubidots accomplished countless internet connected projects across Healthcare, Energy / Utilities, Manufacturing, Transportation, and Retail – learning the many small characteristics of IoT and Cloud enablement that digital transformation experts cannot speak to unless they have gotten their hands dirty in the field.

With a strong backbone and a steadfast determination to become a product-led company, Ubidots joined the Boston Mass Challenge Accelerator in 2013, pivoting into a global tech start up, gaining US venture and support, and leaving behind the local services business.

Since Ubidots pivot in 2014, Ubidots has become known within hardware, software, embedded engineering, and maker circles as the affordable, reliable, and most usable platform in the IoT Application Enablement ecosystem.

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

Ubidots Basics: Devices, Variables, Dashboards, and Alerts

The basics components of any Internet of Things application powered by Ubidots are: **Devices**, **Variables**, **Synthetic Variables Engine**, **Dashboards**, and **Events**. Within this article we will address each of these concepts as they relate to Ubidots IoT Development and Deployment Platform and how you can better organize your Ubidots Apps to best connect with the users.

Once your devices, variables, and general Application is assembled, give your App some layers with [Ubidots Device Management](#) to learn more about Ubidots internal architecture and how you can use Apps, Organizations, and Users to efficiently connect your data with those who should be using it.

- Devices
- Variables
- Synthetic Variables engine
- Dashboard and Visualisation
- Events

- ***Devices***

A Ubidots' device is a virtual representation of a data-source or simply, an asset taking sensor data and transmitting said data through a connection protocol to Ubidots' cloud. Click [here](#) for current firmware examples and tutorials for connecting your device to Ubidots.

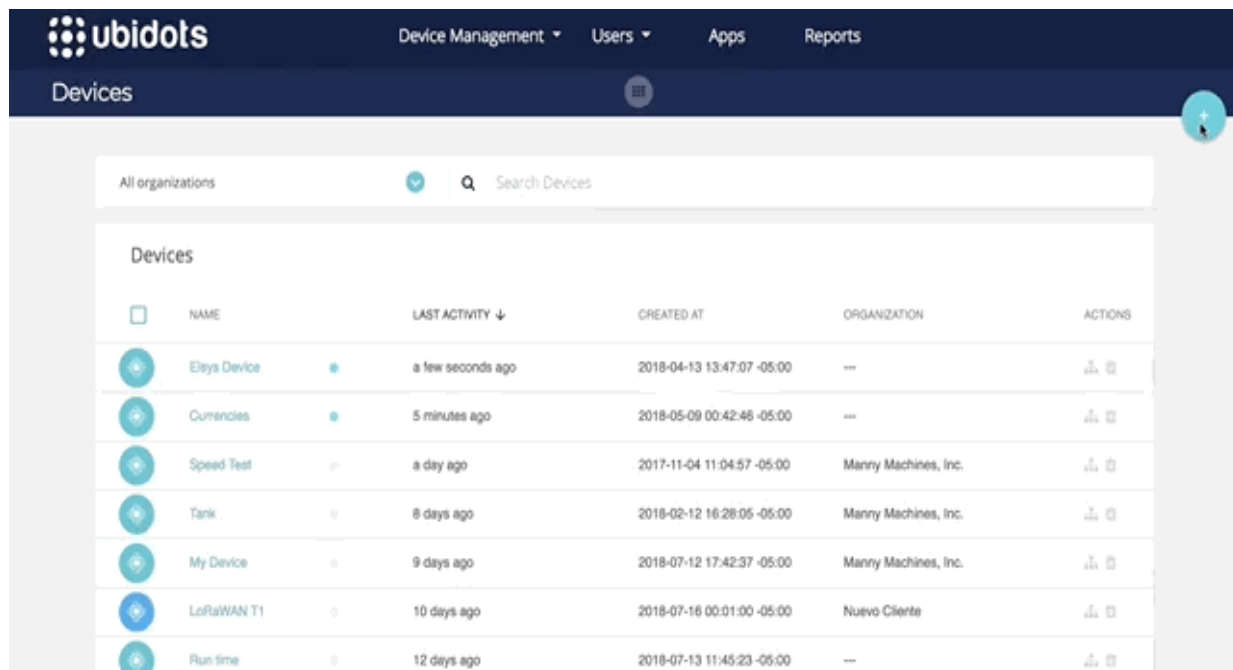
All devices are different, but the standard setup on any device entails:

- A library that should be installed in the device's IDE. (not required)
- Filling parameters for authentication and connection such as a [TOKEN](#) (i.e. the unique ID for each account or user), device and variable labels (i.e. devices and variables unique identifiers in Ubidots), Wi-Fi [SSID](#) and Password, depending on the device and the requirements.
- Making API [request](#) (i.e. a call from a device to the web server). Ubidots' libraries spare the need to manually make these requests. However, if you're working with a

device not currently in the list of supported hardware, you can [choose a connectivity protocol](#) such as [MQTT](#) or [HTTP](#) and make a request using the API accordingly.

Creating devices: there are three ways to create devices:

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



The screenshot shows the Ubidots web interface. At the top is a dark blue navigation bar with the Ubidots logo and links for Device Management, Users, Apps, and Reports. Below this is a 'Devices' section with a search bar and a table of devices. The table has columns for NAME, LAST ACTIVITY, CREATED AT, ORGANIZATION, and ACTIONS. The devices listed are: Elys Device, Currencies, Speed Test, Tank, My Device, LoRaWAN T1, and Run time.

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

Fig 6.30

- **Variables**

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

Types of Variables:

– **Default** - raw data coming from devices (people counted).

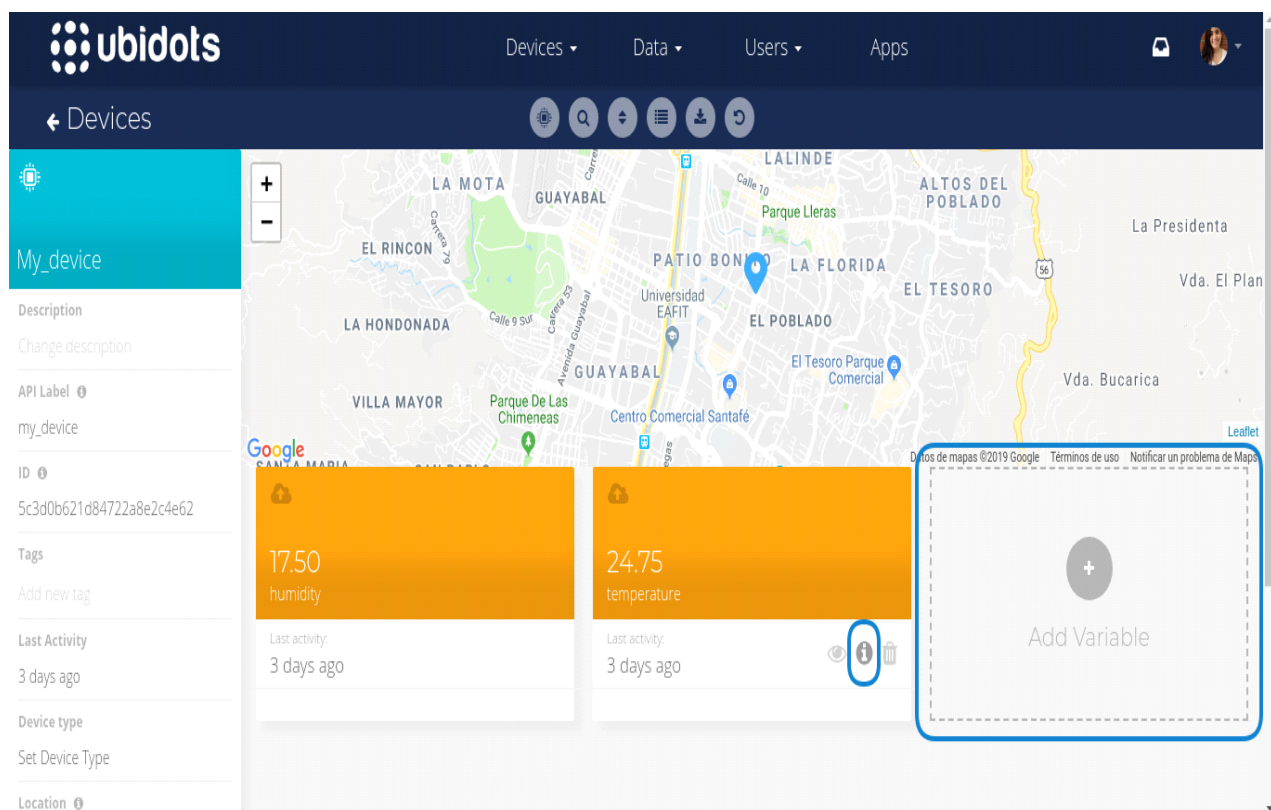
– **Synthetic** - correspond to statistical or arithmetical operations of default variables in a determined time-frame (e.g. average daily traffic this month).

Below you can read more about synthetic variable setups.

Creating variables: there are two ways to create variables.

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

Fig 6.31



- **Synthetic Variables Engine**

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

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

[Synthetic Variables Analytics](#)

Energy Synthetic Variable

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

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

Timezone: America/Bogota

Click on a variable to add it to your expression:

- Current
- Voltage

Cancel Accept

fig 6.32

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

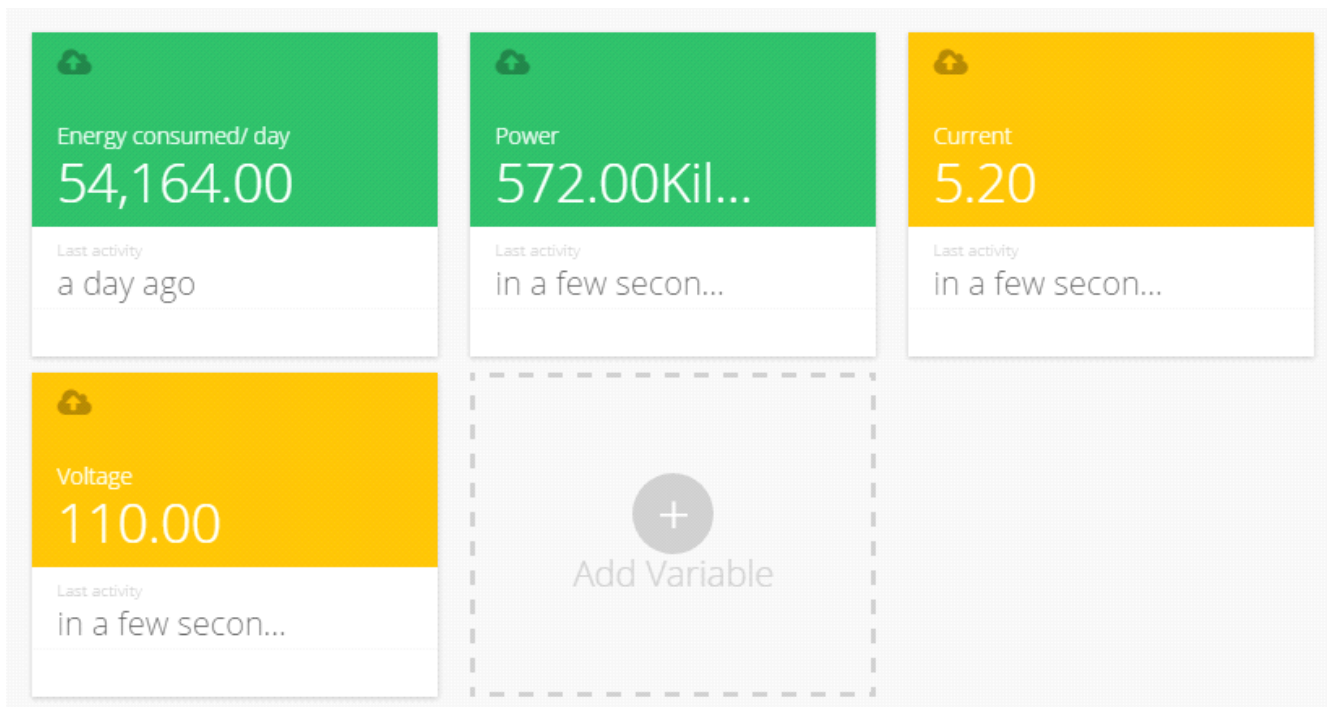


fig 6.33

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

- ***Dashboards and Visualizations***

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

- ***Events***

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

Three Event types:

- When sensor readings reach a specific value:

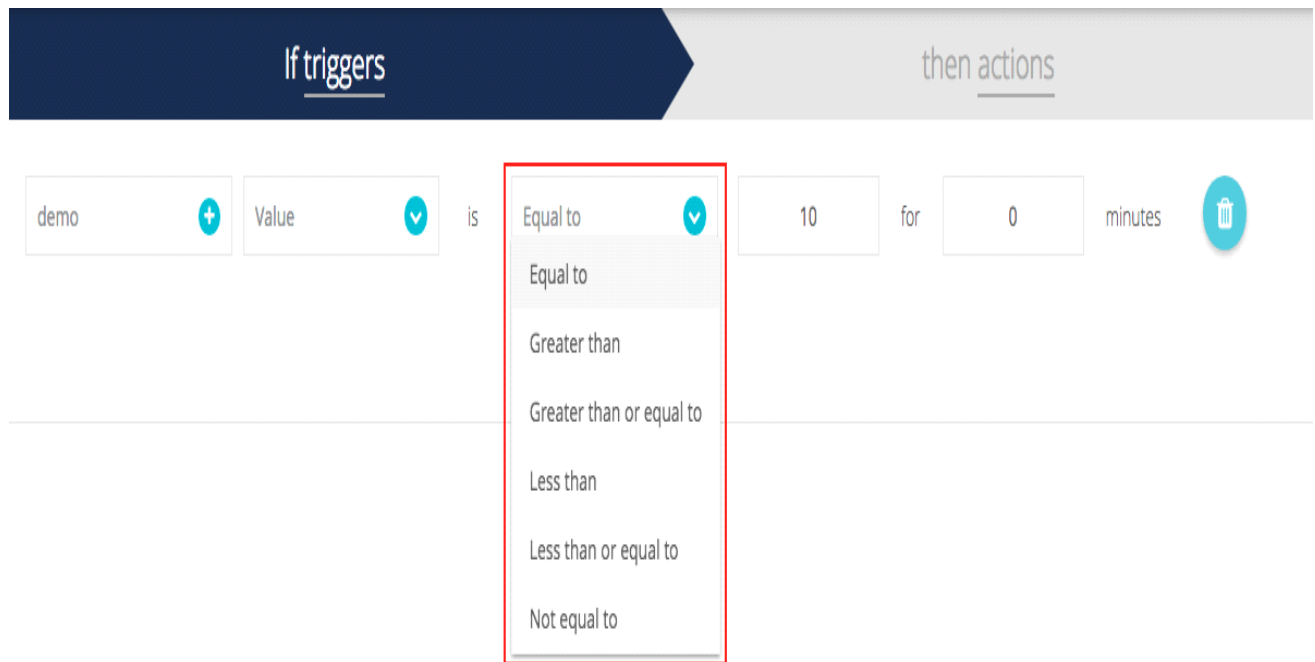


fig 6.34

- [Variable is inactive](#) during a specific amount of time:

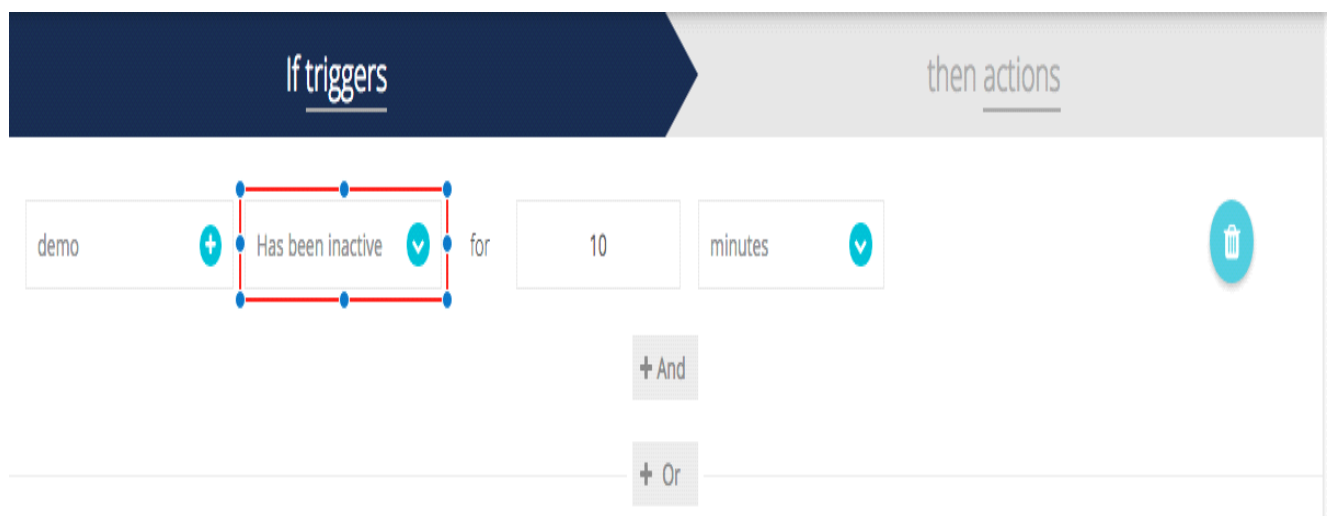


Fig 6.35

- When a device [enters or exits a specific area \(geofence\)](#):

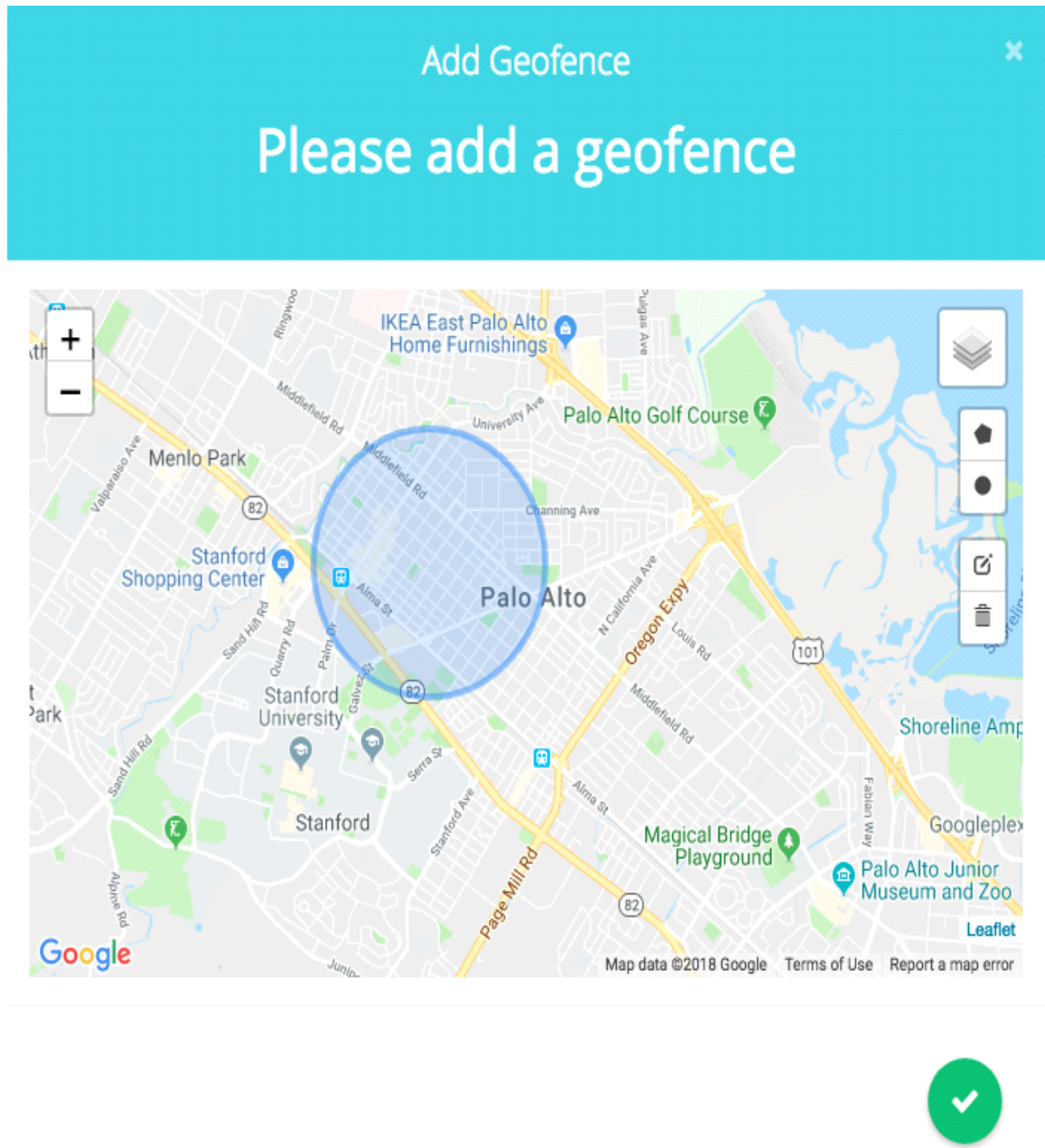


fig 6.36

You can also determine the format of alerts by choosing between [emails](#), [SMS](#), [Telegrams](#) or [Webhook](#)

CHAPTER 7

CONCLUSION:

The study on real time monitoring of toxic gases and other parameters present in underground mine has analysed using wireless sensor network. A real time monitoring system is developed to provide clearer and more point-to-point perspective of the underground mine. This system is displaying the parameters on the IOT at the underground section where sensor unit is installed as well as on the monitoring unit; it will be helpful to all miners present inside the mine to save their life before any casualty occurs. Alarm triggers when sensor values cross the threshold level. This system also stores all the data in the computer for future inspection. From the experiments and observations, the following conclusion can be drawn:

- Each node in a particular framework function as the pioneer robot when all its parameters are configured properly.
- Sensor nodes can reconfigure remotely over a wireless network and most of the processing done in software on computer side.
- The calibration equations of gas sensors may have affected the accuracy of the ppm results

This is a low cost and lifelong system. The overall cost of this system is around 320-380 \$ when using 2 sensor nodes and 250\$ extra for each additional sensor node.

The study on real-time monitoring of toxic gases and other parameters present in underground mine has analysed using wireless sensor networks. A real-time monitoring system is developed to provide clearer and more point-to-point perspective of the underground mine. This system is displaying the parameters on the serial monitor at the underground section where sensor unit is installed as well as on the monitoring unit; it will be helpful to all miners present inside the mine to save their life before any casualty occurs. Alarm triggers when sensor values cross the threshold level. This system also stores all the data in the IOT cloud.

Future Scope

- Using additional sensors all possible safety issues could be monitored such as gases, dust, vibrations, fire etc.
- Zigbee can also be used for the surveillance of mining operations such as subsidence, water leakage etc.
- The other important data can be communicated through this system making it feasible where wired communication is a hindrance.

CHAPTER 8

References

Bin, G., Huizong, L. (2011), "The research on ZigBee-based Mine Safety Monitoring System".

Bo, C., Xiuqan, Q., Budan, W., Xiaokun, W. *et al.* (2012), "Restful Web Service Mashup Based Coal Mine Safety Monitoring and Control Automation with Wireless Sensor Network".

Boddu, R., Balanagu, P., Babu, N.S. (2012), "Zigbee based mine safety monitoring system with GSM".

Borkar, C., "Development of wireless sensor network system for indoor air quality monitoring"

Dange, K.M., Patil, R.T. (2013), "Design of Monitoring System for Coal Mine Safety Based on MSP430".

Dubaniewicz, T.H., Chilton, T.H., Doboroski (1993), "Fiber optic for atmospheric mine monitoring. IEEE Transactions on Industry Applications".

Fu, H., Wang, T., Yuang, C. (2009), "Intelligent Fuzzy Sensing System in Coal Mine Safety Monitoring".

Gottuk, D.T., PeAtross, M.J., Roby, R.J., Beyler, C.L. (2002), "Advanced fire detection using multi-signature alarm algorithms".

Giglio, L., Discloitres, J., Justice, C.O., Kaufman, Y.J. (2003), "An enhanced contextual fire detection algorithm for MODIS, Remote Sensing of Environment".

Hongjiang, H., Shuangyou, W. (2008), "The Application of ARM and ZigBee Technology Wireless Networks in Monitoring Mine Safety System".

Huang, L.C., Chang, H.C., Chen, C.C., Kuo, C.C. (2011), "A ZigBee-based monitoring and protection system for building electrical safety".

Kumar, M., Sharma, M., Narayan, R., Joshi, S., Kumar, S. (2013), "Zigbee based parameter

Kumarsagar, Dange, M., Patil, R.T. (2013), "Design of Monitoring System for Coal Mine Safety Based on MSP430".

Li, W., Li, Z., Liu, W., Wei, P. (2009), "The remote monitoring and analysis system of mine safety based on virtual instrument".

Liu, T., Wei, Y., Song, G., Li, Y. (2013), "Advances of optical fiber sensors for coal mine safety monitoring applications".

McPherson, M.J., "Subsurface Ventilation and Environmental Engineering".

Naticchia, B., Vacarani, M., Carbonari, A. (2012), "A monitoring system for real-time interference control on large construction sites".

Osunmakinde, I.O. (2013), "Towards Safety from Toxic Gases in Underground Mines Using Wireless Sensor Networks and Ambient Intelligence".

Sun, Z., Zhang, X., Li, H., Li, A. (2008), "The Application of TinyOS Beaconing WSN Routing Protocol in Mine Safety Monitoring".

Tan, W., Wang, Q., Huang, H., Guo, Y., Zhang, G. (2007), "Mine Fire Detection System Based on Wireless Sensor Network".

Wu, H., Liu, L., Yuan, X. (2010), "Remote Monitoring System of Mine Vehicle Based on Wireless Sensor Network".

Weng, L.S., Lin, C.L., Chang, H.H. (2009), "Development of a Real-time Mine Auxiliary Monitoring System Using RF Wireless Sensor Networks".

Wei, S, Li, L.L. (2009), "Multi-parameter Monitoring System for Coal Mine based on Wireless Sensor Network Technology".

Xianli, Q., Mingchao, F., Bin, S. (2011), “Coal Mine Gas Wireless Monitoring System Based on WSNs”.

Xiaodong, Z., Yuegang, T., Yan, H. (2014), “Research and Application of Embedded Technology in Remote Network Monitoring System of Coal Mine”.

Xiaoguang, N., Huang, X., Zhao, Z., Zhang, Y. *et al.* (2007), “The Design and Evaluation of a Wireless Sensor Network for Mine Safety Monitoring”.

Yang, W., Huang, Y. (2007), “Wireless Sensor Network Based Coal Mine Wireless and Integrated Security Monitoring Information System”.

Yick, J., Mukherjee, B., Ghosal, D. “Wireless sensor network survey”.

<http://www.arduino.cc>

<http://www.digi.com>

<http://www.purelink.ca/en/technologies/related-technologies.php>

http://ftp1.digi.com/support/documentation/90000976_V.pdf

<http://datasheet.octopart.com/A000066-Arduino-datasheet-38879526.pdf>

<https://www.sparkfun.com/datasheets/Sensors/Biometric/MQ-7.pdf>

<https://www.sparkfun.com/datasheets/Sensors/Biometric/MQ-4.pdf>

<http://www.micropik.com/PDF/dht11.pdf>

<http://www.dfrobot.com/image/data/SEN0159/CO2b%20MG811%20datasheet.pdf>

<http://www.epa.gov/coalbed/basic.html>

<http://www.buzzle.com/articles/methane-gas-exposure-symptoms.html>

<http://www.worldcoal.org/coal-society/safety-issues/>