**MINING WORKERS SENSORY SAFETY**

**HELMET USING IOT.**

**ABSTRACT:**

This project is based on IOT related ideas. Coal mines provides that the essential energy for supporting high speed development of society and it also very important for every nation’s economy. Now a days thousands of miners lose their lives in mining areas, each year mainly from underground coal mining.

Mines are too dark in the underground if any miners fall by unknowingly, pressing the panic switch which is placed in helmet automatically an alert signal passed through the mining management, then they take a sudden actions to save the life of a miners. Here the gas sensors can be placed at mining areas where the hazardous gases which causes explosions like carbon monoxide, methane and highest risks are occurrence of and soil erosion may other disasters. To overcome all these accidents an better technology can be introduced in mining workers sensory safety helmet.

“The deeper the mine, greater the risk”

**1.INTRODUCTION**

India is a voluminous country with rich coals. However, the current safe production level of coal mine is still low, especially in recent years, disasters in coal mine occur frequently, which lead to great loss of possession and life. The safety problems of coal mine has gradually become to the focus that the nation and society concern on. The disasters happening in coal mine are due to the complexity of mine environment and the variety of work condition of coal mine, so it is very necessary to monitor mine working environment. Traditional coal mine monitoring systems tend to be wired network systems, which play an important role in coal mine safe production. With continuous enlarging of exploiting areas and extension of depth in coal mine, many laneways become blind areas, where there are lots of hidden dangers.

Moreover, it is inconvenient to lay cables which are expensive and consume time. In order to solve the problems, IOT design a coal mine safety monitoring system based on wireless sensor network, which can improve the level of monitoring production safety and reduce accidents in the coal mines. Wireless sensor networks is composed of a large number of micro-sensor nodes which have small volume and low cost. It possesses self-organized capability by wireless Communication. In recent years, it is widely used in the fields of our lives, scientific research, military, intelligent traffic, environmental monitoring, intelligent weapon, and so on.

Underground mines are usually extensive labyrinths, of which the tunnels are generally long and narrow with a few kilometers in length and a few meters in width. Thousands of mining personnel are needed to work under extreme conditions according to the construction requirements, and hundreds of miners die from mining accidents every year. It is now widely approved that the underground mining operations are of high risk. In view of this, a monitoring and control system needs to be deployed as one important infrastructure in order to ensure the mining safety and coordinate various tasks.

However, underground coal mines mainly consist of random passages and branch tunnels, and this disorganized structure makes it very difficult to deploy any networking skeleton. In such a case, the utilization of a wireless sensor network (WSN) and other sensing devices may have special advantages for realizing the automation of underground monitoring and control due to the rapid and flexible deployment.

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

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.

**1.1 INTERNET OF THINGS**

**HOW IOT WORKS**

An IoT ecosystem consists of web-enabled smart devices that use embedded processors, sensors and communication hardware to collect, send and act on data they acquire from their environments. [IoT devices](https://internetofthingsagenda.techtarget.com/definition/IoT-device) share the sensor data they collect by connecting to an IoT gateway or other edge device where data is either sent to the cloud to be analyzed or analyzed locally. Sometimes, these devices communicate with other related devices and act on the information they get from one another. The devices do most of the work without human intervention, although people can interact with the devices -- for instance, to set them up, give them instructions or access the data. The connectivity, networking and communication protocols used with these web-enabled devices largely depend on the specific IoT applications deployed.

**INTRODUCTION OF EMBEDDED SYSTEMS**

An embedded system is a computer system(combination of both hardware and software) with a dedicated function within a larger mechanical or electrical system, often with real-time computing constraints. It is embedded as part of a complete device often including hardware and mechanical parts. Embedded systems control many devices in common use today.

Examples of properties of typically embedded computers when compared with general-purpose counterparts are low power consumption, small size, operating ranges, and low per-unit cost. This comes at the price of limited processing resources, which make them significantly more difficult to program and to interact with. Modern embedded systems are often based on microcontrollers (i.e. CPU's with integrated memory or peripheral interfaces), but ordinary microprocessors (using external chips for memory and peripheral interface circuits) are also common, especially in more-complex systems.

A common standard class of dedicated processors is the digital signal processor (DSP). Embedded systems range from portable devices such as digital watches and MP3 players, to large stationary installations like traffic lights, factory controllers, and largely complex systems like hybrid vehicles, MRI, and avionics.

**User interface**

Embedded system text user interface using MicroVGA Embedded systems range from no user interface at all, in systems dedicated only to one task, to complex graphical user interfaces that resemble modern computer desktop operating systems. Simple embedded devices use buttons, LEDs, graphic or character LCDs (HD44780 LCD for example) with a simple menu system. More sophisticated devices which use a graphical screen with touch sensing or screen-edge buttons provide flexibility while minimizing space used: the meaning of the buttons can change with the screen, and selection involves the natural behavior of pointing at what is desired. Handheld systems often have a screen with a "joystick button" for a pointing device.

Some systems provide user interface remotely with the help of a serial (e.g. RS-232, USB, I²C, etc.) or network (e.g. Ethernet) connection. This approach gives several advantages: extends the capabilities of embedded system, avoids the cost of a display, simplifies BSP and allows one to build a rich user interface on the PC. A good example of this is the combination of an embedded web server running on an embedded device (such as an IP camera) or a network router. The user interface is displayed in a web browser on a PC connected to the device, therefore needing no software to be installed.

**Processors in embedded systems**

Embedded processors can be broken into two broad categories. Ordinary microprocessors (μP) use separate integrated circuits for memory and peripherals. Microcontrollers (μC) have on-chip peripherals, thus reducing power consumption, size and cost. In contrast to the personal computer market, many different basic CPU architectures are used, since software is custom-developed for an application and is not a commodity product installed by the end user. Both Von Neumann as well as various degrees of Harvard architectures are used. RISC as well as non-RISC processors are found. Word lengths vary from 4-bit to 64-bits and beyond, although the most typical remain 8/16-bit.

Most architectures come in a large number of different variants and shapes, many of which are also manufactured by several different companies. Numerous microcontrollers have been developed for embedded systems use. General-purpose microprocessors are also used in embedded systems, but generally require more support circuitry than microcontrollers.

**Peripherals**

* Serial Communication Interfaces (SCI): RS-232, RS-422, RS-485, etc.
* Synchronous Serial Communication Interface: I2C, SPI, SSC and ESSI (Enhanced Synchronous Serial Interface) Universal Serial Bus (USB)
* Multi Media Cards (SD cards, Compact Flash, etc.)
* Networks: Ethernet, LonWorks, etc. Fieldbuses: CAN-Bus, LIN-Bus, PROFIBUS, etc.
* Timers: PLL(s), Capture/Compare and Time Processing Units
* Discrete IO: aka General Purpose Input/ Output (GPIO)
* Analog to Digital/Digital to Analog (ADC/DAC)
* Debugging: JTAG, ISP, ICSP, BDM Port, BITP, and DB9 ports.

**EMBEDDED COMPUTER SYSTEMS :**

* An Embedded System is one that has computer hardware with software embedded in it as one of its important components.
* An embedded computer is frequently a computer that is implemented for a particular purpose. In contrast, an average PC computer usually serves a number of purposes: checking email, surfing the internet, listening to music, word processing, etc...
* However, embedded systems usually only have a single task, or a very small number of related tasks that they are programmed to perform.
* An embedded computer system is an electronic system, which includes a microcomputer .It is configured to perform a specific dedicated application. Software is programmed into ROM .
* This software is not accessible to the user of the device , and software solves only a limited range of problems .Here the microcomputer is embedded or hidden inside the system.
* Each embedded microcomputersystem, accepts inputs, performs calculations, and generates outputs and runs in “real time.”

For Example a typical automobile now a days contains an average of ten microcontrollers. In fact, modern houses may contain as many as 150 microcontrollers and on average a consumer now interacts with microcontrollers up to 300 times a day. General areas that employ embedded microcomputers encompass every field of engineering namely: Communications, automotive, military, medical, consumer, machine control etc...

**CHARACTERISTICS OF AN EMBEDDED SYSTEMS**

* Speed (bytes/sec)
* Power (watts)
* Size (cm3) and weight (g)
* Accuracy (% error)
* Adaptability

So, an embedded system must perform the operations at a high speed so that it can be readily used for real time applications and its power consumption must be very low and the size of the system should be as for as possible small and the readings must be accurate with minimum error .The system must be easily adaptable for different situations.

**2. SYSTEM ANALYSIS**

**2.1 EXISTING SYSTEM**

An existing mining safety helmet is very difficult to monitor the miners working conditions and checking the workers environments continuously. A further challenge was to modify the helmet without changing its physical structure. The system must determine whether or not a miner has sustained a life-threatening injury. These two events are defined as hazardous events. Thirdly, dangerous gases need to be detected and announced.

**2.2 PROPOSED SYSTEM**

Security basedcoal miners monitoring system is proposed. The task was extended to designing the system small enough to fit into the safety helmet and last long enough while running on battery power. The proposed system monitors the coal mine workers conditions in every situation. The combinations of all those components are made as complete module which is embed in the worker’s helmet. Each worker has to wear helmet while performing digging process. Temperature level and gas level is always monitored in the miners environment.

If the oxygen level is reduce below the threshold level then the information and location has been detected. Hazardous gases like carbon monoxide, carbon dioxide contents are present more in underground. If the gases are detected more than the threshold value, It will monitored in the above monitoring centre. In both condition, information will send to the monitoring centre through IOT. In this emergency situation buzzer produce alert signal to alert the people around them.

Along with these temperature and Humidity level also monitored. If it increased and decreased from the threshold value, then the alert sound is emits from buzzer which is interfaced with the controller. Buzzer can be used at any critical situations. If buzzer is pressed an alert message will send to mining management an sudden actions to be taken to save the life of the miner.

**3.SYSTEM ANALYSIS**

**3.1 HARDWARE REQUIREMENTS**

* Arduino Uno(Atmega 238)Board
* IOT Kit
* Communication
* Temperature Sensor
* Humidity Sensor
* Gas Sensor
* Ultrasonic Sensor
* LDR Sensor
* Power Supply
* Light (LED),Server
* LCD Display
* Buzzer

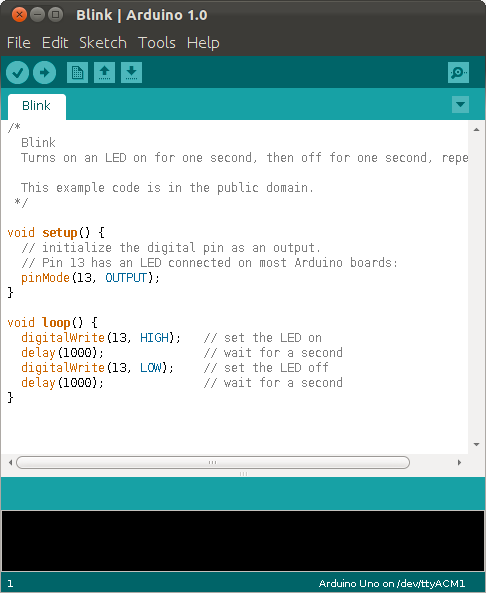
**3.2 SOFTWARE REQUIREMENTS**

* **Coding -** Arduino IDE
* **MC Programming Language -**Embedded C
* **Software -** Proteus
* **Simulation -** Proteus ISIS7

**3.3 SOFTWARE DESCRIPTIONS**

**3.3.1 Arduino IDE for coding ;**

Arduino is an open source, computer tackle and software company, ground plan, and stoner neighbourhood that designs and manufactures microcontroller outfit for constructing digital bias and interactive objects that can smell and hold objects in the physical world. Arduino board designs use a variety of microprocessors and regulators. The boards are trained with sets of digital and analog input/affair( I/O) legs that may be associated to colourful expansion boards( securities) and other circuits. The boards feature serialized dispatches interfaces, including Universal periodical machine( USB) on some models, which are also used for payload programs from peculiar computers. The microcontrollers are usually programmed using a language of features from the programming languages C.



**CONNECTING THE ARDUINO**

Connecting an Arduino board to your PC is quite simple. On Windows:

1. Plug in the USB cable - one end to the PC, and one end to the Arduino board.

2. When prompted, select "Browse my computer for driver" and then select the folder to which you extracted your original Arduino IDE download.

3. You may receive an error that the board is not a Microsoft certified device - select “Install anyway.”

4. Your board should now be ready for programming.

When programming your Arduino board it is important to know what COM port the Arduino is using on your PC. On Windows, navigate to Start->Devices and Printers, and look for the Arduino. The COM port will be displayed underneath.

Alternatively, the message telling you that the Arduino has been connected successfully in the lower-left hand corner of your screen usually specifies the COM port is it using.

**PREPARING THE BOARD**

Before loading any code to your Arduino board, you must first open the IDE. Double click the Arduino .exe file that you downloaded earlier. A blank program, or "sketch," should open.

The Blink example is the easiest way to test any Arduino board. Within the Arduino window, it can be found under File->Examples->Basics->Blink.

Before the code can be uploaded to your board, two important steps are required.

1. Select your Arduino from the list under Tools->Board. The standard board used in RBE 1001, 2001, and 2002 is the Arduino Mega 2560, so select the "Arduino Mega 2560 or Mega ADK" option in the dropdown.

2. Select the communication port, or COM port, by going to Tools->Serial Port.

If you noted the COM port your Arduino board is using, it should be listed in the dropdown menu. If not, your board has not finished installing or needs to be reconnected.

**LOADING CODE**

The upper left of the Arduino window has two buttons: A checkmark to Verify your code, and a right-facing arrow to upload it. Press the right arrow button to compile and upload the Blink example to your Arduino board.

The black bar at the bottom of the Arduino window is reserved for messages indicating the success or failure of code uploading. A "Completed Successfully" message should appear once the code is done uploading to your board. If an error message appears instead, check that you selected the correct board and COM port in the Tools menu, and check your physical connections.

If uploaded successfully, the LED on your board should blink on/off once every second. Most Arduino boards have an LED prewired to pin 13.

It is very important that you do not use pins 0 or 1 while loading code. It is recommended that you do not use those pins ever.

Arduino code is loaded over a serial port to the controller. Older models use an [FTDI](http://www.ftdichip.com/) chip which deals with all the USB specifics. Newer models have either a small AVR that mimics the FTDI chip or a built-in USB-to-serial port on the AVR micro-controller itself.

**3.3.2 MC Programming Language : Embedded C**

Making around, we detect ourselves to be encircled by chromatic types of embedded system. Be it a digital camera or a mobile phone or a washing machine, all of them has some kind of processor performing inside it. Associated with each processor is the embedded software. However, embedded processor acts as the brain, and embedded software forms its soul, If tackle forms the body of an embedded system. It's the embedded software which primarily governs the functioning of embedded systems secure as well.

**Embedded systems are programmed using different type of languages**:

1. Machine Code
2. Low level language, i.e., assembly
3. High Position level language like C, C++, Java, Ada, etc.
4. Application-level language like Visual Basic, scripts, Access, etc.

**Advantage**

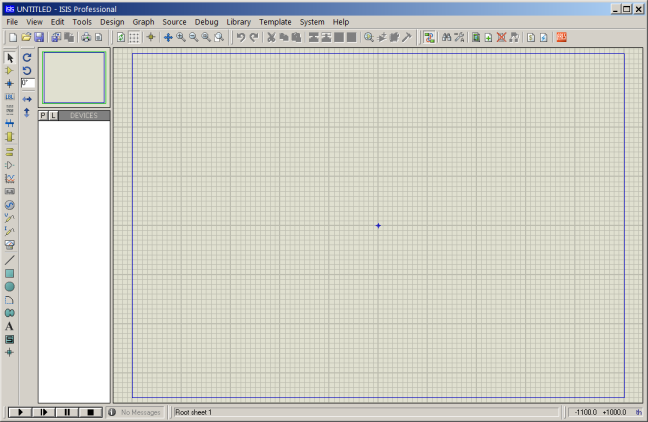
* It is small and considerately simpler to pick up, understand, program and debug.
* C Compilers are available for nearly all embedded devices in use moment, and there is a large pool of educated C programmers.
* Unlike assembly, C has advantage of processor-independence and is not specific to any particular microprocessor/ microcontroller or any system.

**3.3.3 Software : Proteus**

The microcontroller can gather a program written in assembly language, it must be collected into a language of bottoms and bones. Assembly language and Assembler don't have the same meaning. The foremost one refers to the set of rules used for writing program for the microcontroller, while the after refers to a program on a peculiar computer used to restate assembly language statements into the language of bottoms and bones. A collected program is also called Machine Code. In machine law, the same command is portrayed by a 14- bit array of bottoms and bones coherent by the microcontroller. All assembly language commands are also collected into the corresponding array of bottoms and bones.

* + 1. **Simulation : Proteus ISIS7**

Many of the components in Proteus can be simulated. There are two options for simulating. Run simulator and advance frame by frame. The "Run simulator" option simulates the circuit in a normal speed (If the circuit is not heavy).



This can be useful for debugging digital circuits.  
You can also simulate microcontrollers. The microcontrollers which can be simulated include PIC24, dsPIC33, 8051, Arduino, ARM7 based microcontrollers. You can download the compilers for Proteus or use different compiler and dump the hex files in the microcontroller in Proteus. You can even interact in real-time with the simulation using switches, resistors, LDRs, etc. There are even virtual voltmeter, ammeter, oscilloscope, logic analyser.

****

**About Proteus:**It is a software suite containing [schematic](http://www.circuitstoday.com/pcb-design-and-layout-software), [simulation](http://www.circuitstoday.com/circuit-design-and-simulation-softwares) as well as [PCB designing](http://www.circuitstoday.com/how-to-build-pcb-online-using-web-based-eda-tools).

* [ISIS](http://www.labcenter.com/products/pcb/schematic_intro.cfm) is the software used to draw schematics and simulate the circuits in real time. The simulation allows human access during run time, thus providing real time simulation.
* [ARES](http://www.labcenter.com/products/pcb/pcb_intro.cfm)  is used for PCB designing. It has the feature of viewing output in 3D view of the designed PCB along with components.
* The designer can also develop 2D drawings for the product.

**Features**

ISIS has wide range of components in its library. It has sources, signal generators, measurement  and analysis tools like [oscilloscope](http://www.circuitstoday.com/best-analog-oscilloscope-guide), voltmeter, ammeter etc., probes for real time monitoring of the parameters of the circuit, [switches](http://www.circuitstoday.com/proteus-tutorial-switches-and-relays), [displays](http://www.circuitstoday.com/proteus-tutorial-led-and-bar-graph), loads like motors and lamps, discrete components like resistors, capacitors, inductors, transformers, digital and analog Integrated circuits, semi-conductor switches, relays, microcontrollers, processors, sensors etc.

**4. PROJECT DESCRIPTIONS**

**4.1 Arduino Uno(Atmega 238)Board**

The Arduino UNO is an open-source microcontroller board based on the  [Microchip](https://en.wikipedia.org/wiki/Microchip_Technology) [Atmega328P](https://en.wikipedia.org/wiki/ATmega328P) microcontroller and developed by [Arduino.cc](https://en.wikipedia.org/wiki/Arduino). The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The board has 14 Digital pins, 6 Analog pins, and programmable with the [Arduino IDE](https://en.wikipedia.org/wiki/Arduino#Software) (Integrated Development Environment) via a type B USB cable. It can be powered by a USB cable or by an external 9 volt battery, though it accepts voltages between 7 and 20 volts. The Atmega328 on the Arduino Uno comes pre-programmed with a boot loader that allows uploading new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol. The Uno also differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it uses the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

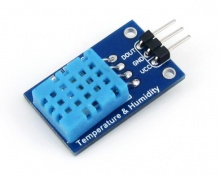
* 1. **Communication**

The Arduino/Genuino Uno has a number of facilities for communicating with a computer, another Arduino/Genuino board, 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 (IDE) includes a serial monitor which allows simple textual data to be sent to and from the 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 A Software Serial library allows serial communication on any of the Uno's digital pins.

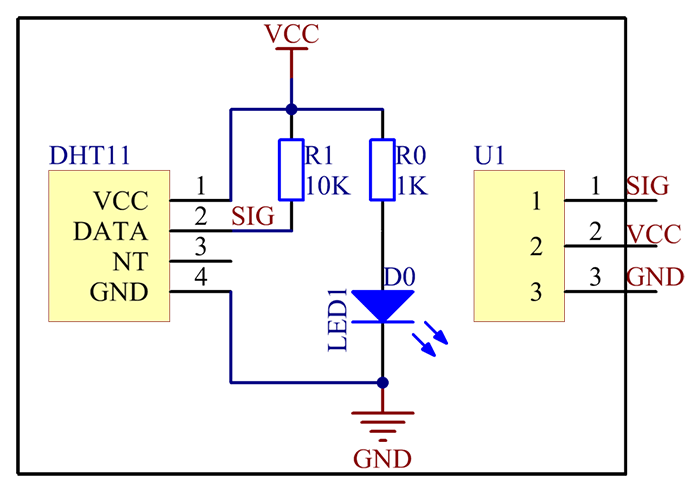


**4.3 TEMPERATURE SENSOR**

The digital temperature and humidity sensor DHT11 is a composite sensor that contains a calibrated digital signal output of temperature and humidity. The technology of a dedicated digital modules collection and the temperature and humidity sensing technology are applied to ensure that the product has high reliability and excellent long-term stability.



The sensor includes a resistive sense of wet component and an NTC temperature measurement device, and is connected with a high-performance 8-bit microcontroller. The schematic diagram of the Humiture Sensor Module is as shown following:



Only three pins are available for use: VCC, GND, and DATA. The communication process begins with the DATA line sending start signals to DHT11, and DHT11 receives the signals and returns an answer signal. Then the host receives the answer signal and begins to receive 40-bit humiture data (8-bit humidity integer + 8-bit humidity decimal + 8-bit temperature integer + 8-bit temperature decimal + 8-bit checksum).

**4.4 HUMIDITY SENSOR**

Humidity is the presence of water in air. The amount of water vapor in air can affect human comfort as well as many manufacturing processes in industries. The presence of water vapor also influences various physical, chemical, and biological processes. Humidity measurement in industries is critical because it may affect the business cost of the product and the health and safety of the personnel. Hence, humidity sensing is very important, especially in the control systems for industrial processes and human comfort.

Controlling or monitoring humidity is of paramount importance in many industrial & domestic applications. Higher humidity reduces the effectiveness of sweating in cooling the body by reducing the rate of evaporation of moisture from the skin. This effect is calculated in a heat index table or humidex. There are three main measurements of humidity: absolute, relative and specific. Absolute humidity is the water content of air at a given temperature expressed in gram per cubic metre. Relative humidity, expressed as a percent, measures the current absolute humidity relative to the maximum (highest point) for that temperature. Specific humidity is a ratio of the water vapor content of the mixture to the total air content on a mass basis.

****

A humistor is a type of variable resistor whose resistance varies based on humidity**.**

**4.5 GAS SENSOR**

A gas detector is a device that detects the presence of gases in an area, often as part of a safety system. This type of equipment is used to detect a gas leak and interface with a control system so a process can be automatically shut down. A gas detector can sound an alarm to operators in the area where the leak is occurring, giving them the opportunity to leave. This type of device is important because there are many gases that can be harmful to organic life, such as humans or animals. Gas detectors can be used to detect combustible, flammable and toxic gases, and oxygen depletion. They may be used in firefighting. These sensors usually employ an audible alarm to alert people when a dangerous gas has been detected.



**FEATURES:**

* 5V DC or AC circuit
* Requires heater voltage
* Operation Temperature: -10 to 70 degrees C
* Heater consumption: less than 750Mw

### PINS General Pin functions

* **LED**: There is a built-in LED driven by digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
* **VIN**: The input voltage to the Arduino/Genuino 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 - 20V), the USB connector (5V), or the VIN pin of the board (7-20V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage the board.
* **3V3**: A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
* **GND**: Ground pins.
* **IOREF**: This pin on the Arduino/Genuino 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 to work with the 5V or 3.3V.
* **Reset**: Typically used to add a reset button to shields which block the one on the board.

**Special Pin Functions**

Each of the 14 digital pins and 6 Analog pins on the Uno can be used as an input or output, using pinMode (), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive 20 mA as recommended operating condition and has an internal pull-up resistor (disconnected by default) of 20-50k ohm. A maximum of 40mA is the value that must not be exceeded on any I/O pin to avoid permanent damage to the microcontroller. The Uno has six analog inputs, labeled 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 is it possible to change the upper end of their range using the AREF pin and the analogReference() function.

In addition, some pins have specialized functions:

* **Serial** / [UART](https://en.wikipedia.org/wiki/UART): pins 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**: pins 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.
* **PWM** (**P**ulse **W**idth **M**odulation): 3, 5, 6, 9, 10, and 11 Can provide 8-bit PWM output with the analogWrite () function.
* **SPI** (**S**erial **P**eripheral **I**nterface): 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.
* **TWI** (**T**wo **W**ire **I**nterface) / [I²C](https://en.wikipedia.org/wiki/I%C2%B2C): A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library.
* **AREF** (**A**nalog **REF**erence): Reference voltage for the analog inputs.

|  |  |  |
| --- | --- | --- |
| **Pin No** | **Function** | **Name** |
| 1 | Ground (0V) | Ground |
| 2 | Supply voltage; 5V (4.7V – 5.3V) | Vcc |
| 3 | Contrast adjustment; through a variable resistor | VEE |
| 4 | Selects command register when low; and data register when high | Register Select |
| 5 | Low to write to the register; High to read from the register | Read/write |
| 6 | Sends data to data pins when a high to low pulse is given | Enable |
| 7 | 8-bit data pins | DB0 |
| 8 | DB1 |
| 9 | DB2 |
| 10 | DB3 |
| 11 | DB4 |
| 12 | DB5 |
| 13 | DB6 |
| 14 | DB7 |
| 15 | Backlight VCC (5V) | Led+ |
| 16 | Backlight Ground (0V) | Led- |
|  |  |  |
|  |  |  |

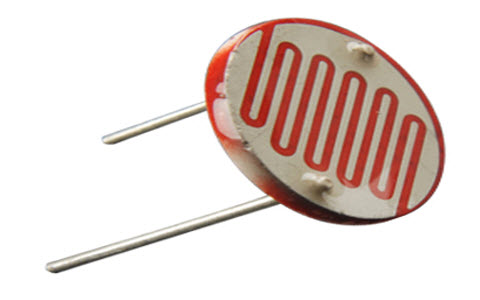
**4.6ULTRASONIC SENSOR**

Ultrasonic transducers are transducers that convert ultrasound waves to electrical signals or vice versa. Those that both transmit and receive may also be called ultrasound transceivers; many ultrasound sensors besides being sensors are indeed transceivers because they can both sense and transmit. These devices work on a principle similar to that of transducers used in radar and sonar systems, which evaluate attributes of a target by interpreting the echoes from radio or sound waves, respectively. Active ultrasonic sensors generate high-frequency sound waves and evaluate the echo which is received back by the sensor, measuring the time interval between sending the signal and receiving the echo to determine the distance to an object.



**4.7 LDR SENSOR**

A Light Sensor generates an output signal indicating the intensity of light by measuring the radiant energy that exists in a very narrow range of frequencies basically called “light”, and which ranges in frequency from “Infra-red” to “Visible” up to “Ultraviolet” light spectrum. The [Light Sensor](http://www.amazon.in/s/?field-keywords=Transducers%2C%20sensors%20%26%20detectors) is a passive devices that convert this “light energy” whether visible or in the infra-red parts of the spectrum into an electrical signal output. Light sensors are more commonly known as “Photoelectric Devices” or “Photo Sensors” because the convert light energy (photons) into electricity (electrons).



**4.7 POWER SUPPLY CIRCUIT**

Power supply is a reference to a source of electrical power. A device or system that supplies electrical or other types of energy to an output load or group of loads is called a power supply unit or PSU. The term is most commonly applied to electrical energy supplies, less often to mechanical ones, and rarely to others.

Power supplies for electronic devices can be broadly divided into linear and switching power supplies. The linear supply is a relatively simple design that becomes increasingly bulky and heavy for high current devices; voltage regulation in a linear supply can result in low efficiency. A switched-mode supply of the same rating as a linear supply will be smaller, is usually more efficient, but will be more complex.

**Linear Power supply**

An AC powered linear power supply usually uses a transformer to convert the voltage from the wall outlet (mains) to a different, usually a lower voltage. If it is used to produce DC, a rectifier is used. A capacitor is used to smooth the pulsating current from the rectifier. Some small periodic deviations from smooth direct current will remain, which is known as ripple. These pulsations occur at a frequency related to the AC power frequency (for example, a multiple of 50 or 60 Hz).

### Transformer:

### transformer symbol

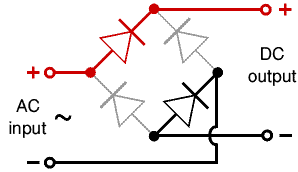
**Transformer**

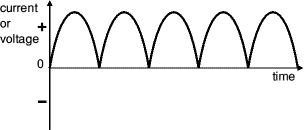
### Transformers convert AC electricity from one voltage to another with little loss of power. Transformers work only with AC and this is one of the reasons why mains electricity is AC.

### Step-up transformers increase voltage, step-down transformers reduce voltage. Most power supplies use a step-down transformer to reduce the dangerously high mains voltage (230V in UK) to a safer low voltage.

The input coil is called the primary and the output coil is called the secondary. There is no electrical connection between the two coils; instead they are linked by an alternating magnetic field created in the soft-iron core of the transformer. The two lines in the middle of the circuit symbol represent the core.

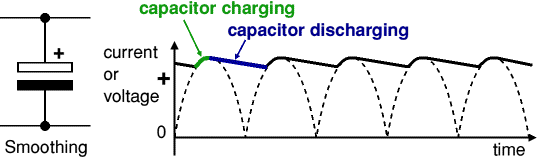
**Bridge rectifier**

A bridge rectifier can be made using four individual diodes, but it is also available in special packages containing the four diodes required. It is called a full-wave rectifier because it uses the entire AC wave (both positive and negative sections). 1.4V is used up in the bridge rectifier because each diode uses 0.7V when conducting and there are always two diodes conducting, as shown in the diagram below.



**Smoothing:**

Smoothing is performed by a large value electrolytic capacitor connected across the DC supply to act as a reservoir, supplying current to the output when the varying DC voltage from the rectifier is falling. The diagram shows the unsmoothed varying DC (dotted line) and the smoothed DC (solid line). The capacitor charges quickly near the peak of the varying DC, and then discharges as it supplies current to the output.

 Smoothing is not perfect due to the capacitor voltage falling a little as it discharges, giving a small ripple voltage. For many circuits a ripple which is 10% of the supply voltage is satisfactory and the equation below gives the required value for the smoothing capacitor. A larger capacitor will give fewer ripples. The capacitor value must be doubled when smoothing half-wave DC.

Smoothing Capacitor for 10% ripple, C=5\*10/vs.\*f

C = smoothing capacitance in farads (F)

Io = output current from the supply in amps (A)

Vs = supply voltage in volts (V), this is the peak value of the unsmoothed

f    = frequency of the AC supply in hertz (Hz), 50Hz in the UK.

### Smooth DC power supply, transformer + rectifier + smoothing power supply circuit

### Regulator:

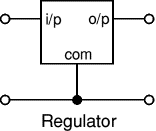
### Voltage regulator ICs are available with fixed (typically 5, 12 and 15V) or variable output voltages. They are also rated by the maximum current they can pass. Negative voltage regulators are available, mainly for use in dual supplies. Most regulators include some automatic protection from excessive current ('overload protection') and overheating ('thermal protection').

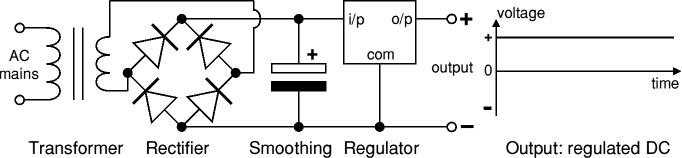
1. **Positive regulator**
   1. input pin
   2. ground pin
   3. output pin

It regulates the positive voltage

1. **Negative regulator**
   1. ground pin
   2. input pin
   3. output pin

It regulate the negative voltage.





**4.8 LIQUID CRYSTAL DISPLAY**

A liquid crystal display (LCD) is a flat panel display, electronic visual display, or video display that uses the light modulating properties of liquid crystals. Liquid crystals do not emit light directly. LCDs are available to display arbitrary images (as in a general-purpose computer display) or fixed images which can be displayed or hidden, such as preset words, digits, and 7-segment displays as in a digital clock. They use the same basic technology, except that arbitrary images are made up of a large number of small pixels, while other displays have larger elements. An LCD is a small low cost display. It is easy to interface with a micro-controller because of an embedded controller (the black blob on the back of the board). This controller is standard across many displays (HD 44780) which means many micro-controllers (including the Arduino) have libraries that make displaying messages as easy as a single line of code.

** LCD display unit**

**4.9 NODE MCU**

NodeMCU is an open source [Lua](https://www.lua.org/) based firmware for the [ESP8266 Wi-Fi SOC from Express if](http://espressif.com/en/products/esp8266/) and uses an on-module flash-based [SPIFFS](https://github.com/pellepl/spiffs) file system. NodeMCU is implemented in C and is layered on the [Express if NON-OS SDK](https://github.com/espressif/ESP8266_NONOS_SDK).

The firmware was initially developed as is a companion project to the popular ESP8266-based [NodeMCU development modules](https://nodemcu.readthedocs.io/en/master/(https:/github.com/nodemcu/nodemcu-devkit-v1.0)), but the project is now community-supported, and the firmware can now be run on any ESP module.



**Features:**

• Open-source

• Interactive

• Programmable

• Low cost

• Simple

• Smart

• WI-FI enabled

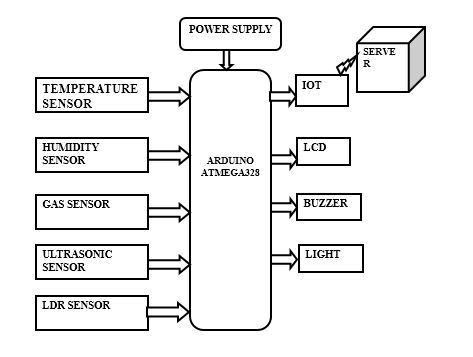
**4.10 BUZZER**

A buzzer or beeper is a signalling device, usually electronic, typically used in automobiles, household appliances such as a microwave oven, or game shows. It most commonly consists of a number of switches or sensors connected to a control unit that determines if and which button was pushed or a present time has lapsed, and usually illuminates a light on the appropriate button or control panel, and sounds a warning in the form of a continuous or intermittent buzzing or beeping sound. Initially this device was based on an electromechanical system which was identical to an electric bell without the metal gong (which makes the ringing noise). Often these units were anchored to a wall or ceiling and used the ceiling or wall as a sounding board.



**5 .SYSTEM DESIGN**

**5.1 ARCHITECTURAL DIAGRAM**



**5.2 DIAGRAM DESCRIPTION**

This proposed system is classified as transmitter and receiver block. The transmitter block is consisting of gas sensor, Temperature Sensor, Humidity sensor, Buzzer, Ultrasonic, sensor, LDR sensor and IOT which are interfaced with the Arduino controller. Power supply is providing by 230V supply through step down transformer and rectifier circuit. Gas sensor detects the gas level in environment. It detects the gas as analog value and the value is converted as digital in the Arduino controller. The safety value is limited in the controller. If the value is exceeded it can be detected by the controller. Ultrasonic Sensor is used when the miner hits with stone or any other objects before buzzer it will alerts the miner inside the working area. LDR is used to detect the presence of light and turn on(when its darker in the area) and turned off (in presence of light) accordingly.

Similar to the gas sensor, Temperature sensor is analog sensor. It is interfaced with the controller in analog input pins. The value is limited according to the value required. If value from sensor beyond the limited value it can detected. Humidity sensor is a digital sensor which gives either 0 or 1 in output hence it connected to the digital output pins.

**6.IMPLEMENTATION**

IOT is used to transmit that information to the monitoring centre. In case emergency buzzer emits the alert sound and hence buzzer is interfaced with the controller. Ultrasonic Sensor is used when the miner hits with stone or any other objects before buzzer it will alerts the miner inside the working area. LDR is used to detect the presence of light and turn on(when its darker in the area) and turned off (in presence of light) accordingly. All operations of the system are show on LCD which is interfaced with the controller. From this proposed system, all miners are under the control of monitoring centre. In case of emergency, can able to give protection system and immediate treatment for them.

**7.CONCLUSION**

This system will enhances the security system for underground coal miners. A Larger area and more depth inside hazardous underground mines are now can be covered and secured and then the accidents can be controlled effectively. Then sensor and the micro-controller module can be preferably installed over the helmet for mine workers.

By the proper monitoring can help to take appropriate actions more rapidly and smartly if any abnormal situation occurs. Thus a smart helmet for hazardous event detection, monitoring the surrounding environmental conditions and upgrading all the information through the IOT technology to save the life of a miners at any dangerous conditions. To finally conclude a smart helmet has been designed in order to help the workers from any casualty citations. This system provides a clear and point to point perspective of underground mining system and provides reliable communication using IOT. Hence it is beneficial to the miners present inside the underground and ensures the safety of the miners.

**8.FUTURE SCOPE**

The system can be improved by adding more measuring devices to check the miners health conditions by using HEALTH MONITORING SENSORS, and also to monitor natural disasters like Landslides, Soil Erosions by using the RADAR SENSORS.

**9. REFERENCES**

[1] M. A. Hermanus, “Occupational health and safety in mining—status, new developments, and concerns,” The Journal of The Southern African Institute of Mining and Metallurgy, vol. 107, pp. 531-538, Aug. 2020.

[2] A.P. Squelch, “Virtual reality for mine safety training in South Africa,” The Journal of The South African Institute of Mining and Metallurgy, pp. 209-216, July 2021.

[3] C. Qiang, S. Ji-ping, Z. Zhe and Z. Fan, “ZigBee Based Intelligent Helmet for Coal Miners,” IEEE World Congress on Computer Science and Information Engineering (WRI 2009), 31 Mar. -2 April 2009, vol. 3, pp. 433–435, 2018.

[4] H. Hongjiang and W. Shuangyou, “The application of ARM and ZigBee technology wireless networks in monitoring mine safety system,” IEEE International Colloquium on Computing, Communication, Control, and Management (ISECS 2008), 3-4 Aug. 2008, Guangzhou, pp. 430–433, 2018.

[5] X. Liu, J. S. Huang and Z. Chen, “The research of ranging with timing over packet network for the mine safety application,” Journal of Networks, vol. 7, no. 7, pp. 1054–1062, Jul. 2020.

[6] R. S. Nutter, “Hazard evaluation methodology for computer-controlled mine monitoring/control systems,” IEEE Trans. on Industry Applications, vol. IA-19, no. 3, pp. 445-449, May/June 1983.

[7] R. S. Nutter, “A distributed microprocessor monitoring and control system for coal mines,” in Proc. 4th WVU Conf. on Coal Mine Electro technology, Aug. 2-4, 2005.

[8] R. S. Nutter and M. D. Aldridge, “Status of mine monitoring and communications,” IEEE Trans. on Industry Applications, vol. 24, no. 5, pp. 820-826, Sep./Oct. 2008.

[9] A. D. Kock and J. W. Oberholzer, “The development and application of electronic technology to increase health, safety, and productivity in the South African coal mining industry,” IEEE Trans. on Industry Applications, vol. 33, no. 1, pp. 100-105, Jan/Feb. 2015.

[10] P. Misra, S. Kanhere, D. Ostry and S. Jha, “Safety assurance and rescue communication systems in high-stress environments: a mining case study,” IEEE Communications Magazine, vol. 48, no. 4, pp. 66-73, April 2018.

[11] A. E. Forooshani, S. Bashir, D. G. Michelson and S. Noghanian, “A survey of wireless communications and propagation modelling in underground mines,” IEEE Communications Surveys and Tutorials, vol. 15, no. 4, pp. 1524-1545, Nov. 2021.

[12] C. P. Kruger and G. P. Hancke, “Implementing the Internet of Things vision in industrial wireless sensor networks,” IEEE Int. Conf. on Industrial Informatics, pp. 627-632, July 2014.

[13] A. Kumar and G. P. Hancke, “Energy efficient environment monitoring system based on the IEEE 802.15.4 standard for low cost requirements”, IEEE Sensors Journal, vol. 14, no. 8, pp. 2557-2566, Aug. 2014.

[14] D. M. Han and J. H. Lim, “Smart home energy management system using IEEE 802.15.4 and ZigBee,” IEEE Trans. on Consumer Electronics, vol. 56, no. 3, pp. 1403-1410, Aug. 2010.

[15] C. P. Kruger and G. P. Hancke, “Benchmarking Internet Of Things devices,” IEEE Int. Conf. on Industrial Informatics, pp. 611-616, July 2014.

[16] K. Gill, S. H. Yang, F. Yao and X. Lu, “A ZigBee-based home automation system,” IEEE Trans. on Consumer Electronics, vol. 55, no. 2, pp. 422-430, May 2009.

[17] J. Byun and S. Park, “Development of a self-adapting intelligent system for building energy-saving and context-aware smart services,” IEEE Trans. on Consumer Electronics, vol. 57, no. 1, pp. 90-98, 2011.

[18] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: a survey,” Computer Networks, vol. 38, pp. 393-422, 2002.

[19] V. C. Gungor and G. P. Hancke, Industrial Wireless Sensor Networks: Applications, Protocols, and Standards, Published by CRC Press, 2013.

[20] ZigBee Alliance, "Zigbee wireless standards”. www.didgi.com/ technology/rf-articles/wireless-zigbee

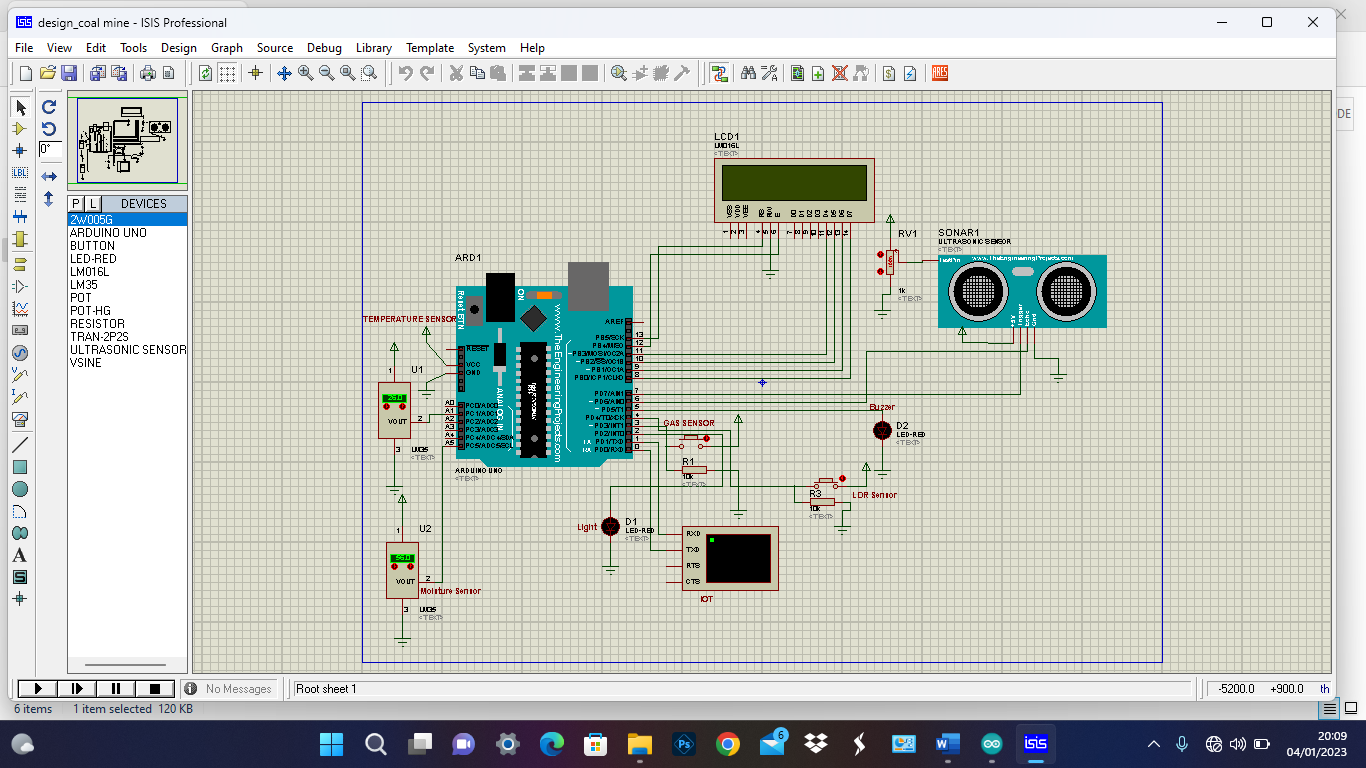
[21] Contiki os. Available: <http://www.contiki-os.org/>.

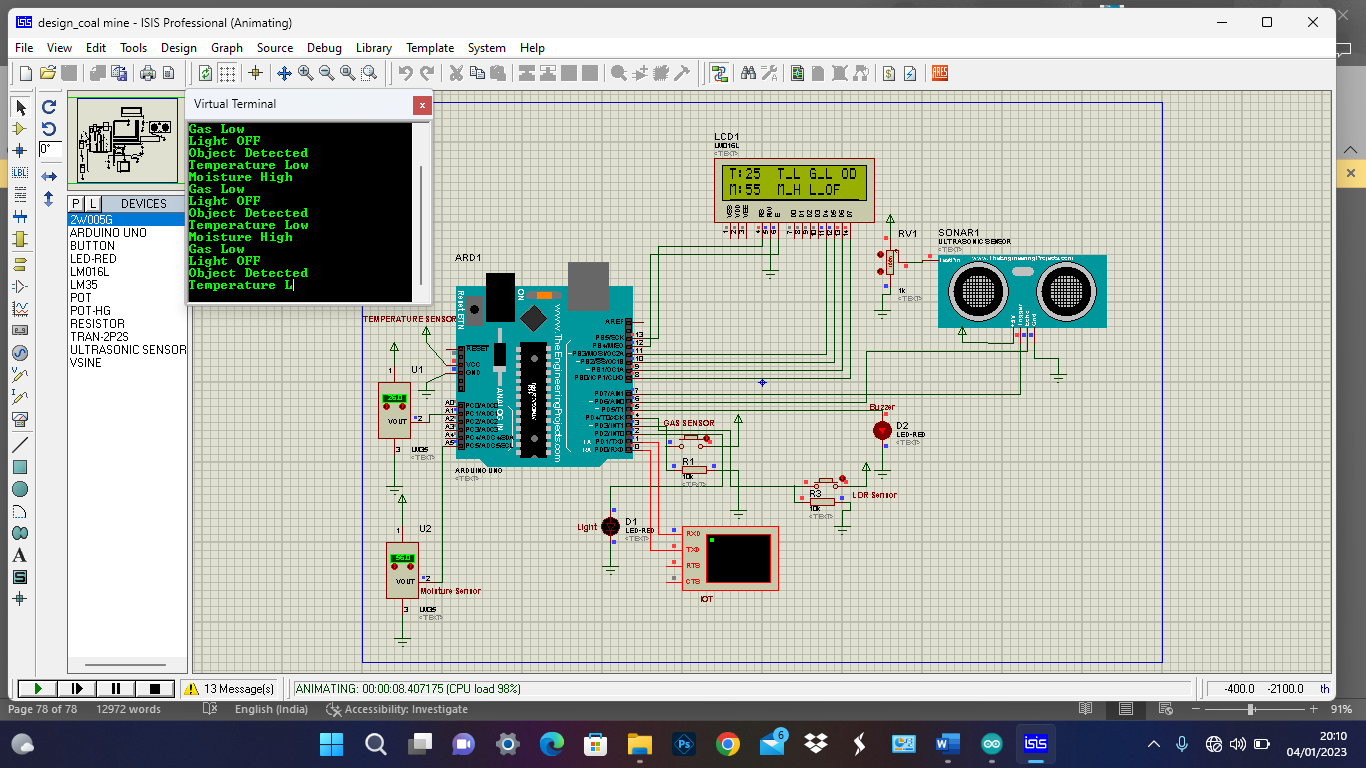
[22] B. Silva, A. Kumar and G. P. Hancke, “Experimental Link Quality Characterization of Wireless Sensor Networks for Underground Monitoring,” IEEE Trans. on Industrial Informatics, vol. 11, no. 5, pp. 1099-1110, Oct. 2015.

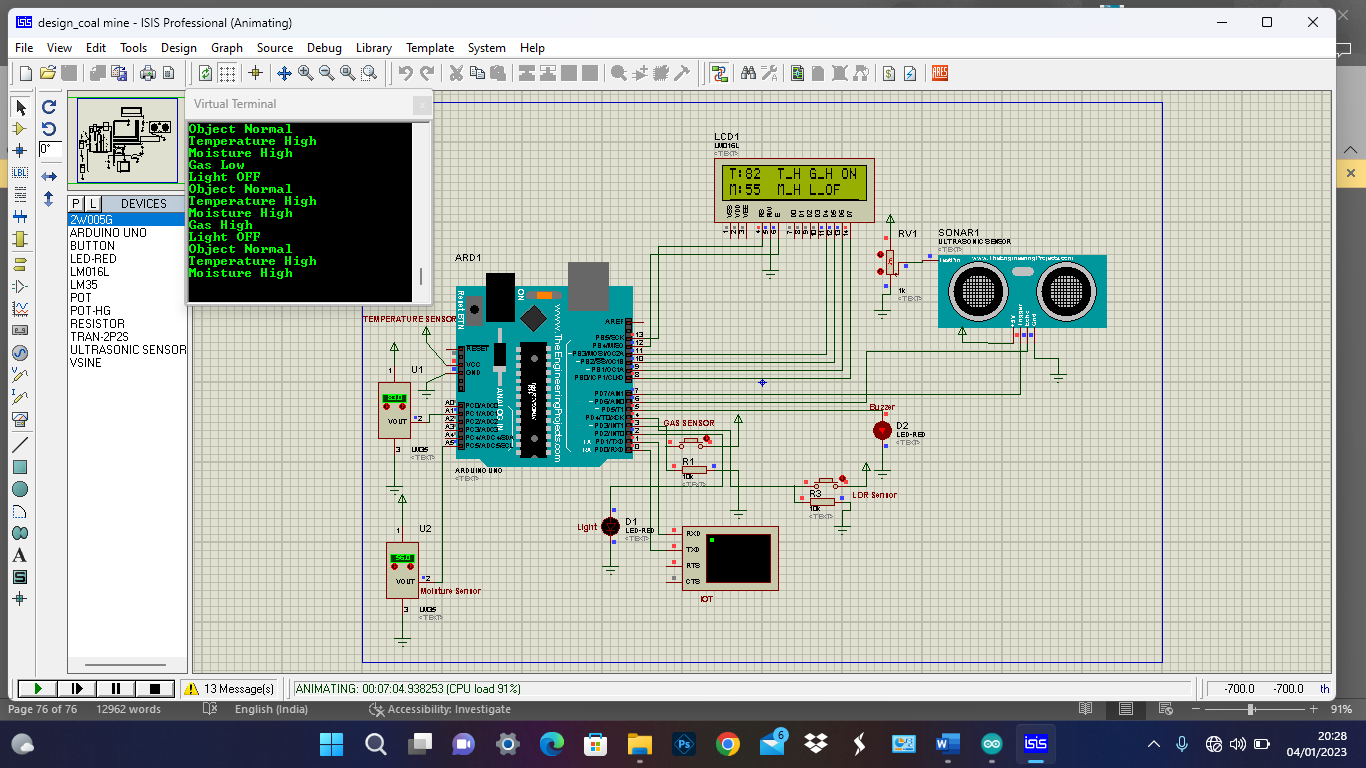
[23] R. Fisher, L. Ledwaba, G. P. Hancke and C. Kruger. “Open Hardware: A Role to Play in Wireless Sensor Networks?,” Sensors, vol.15, no. 3, pp. 6818-6844, 2015.

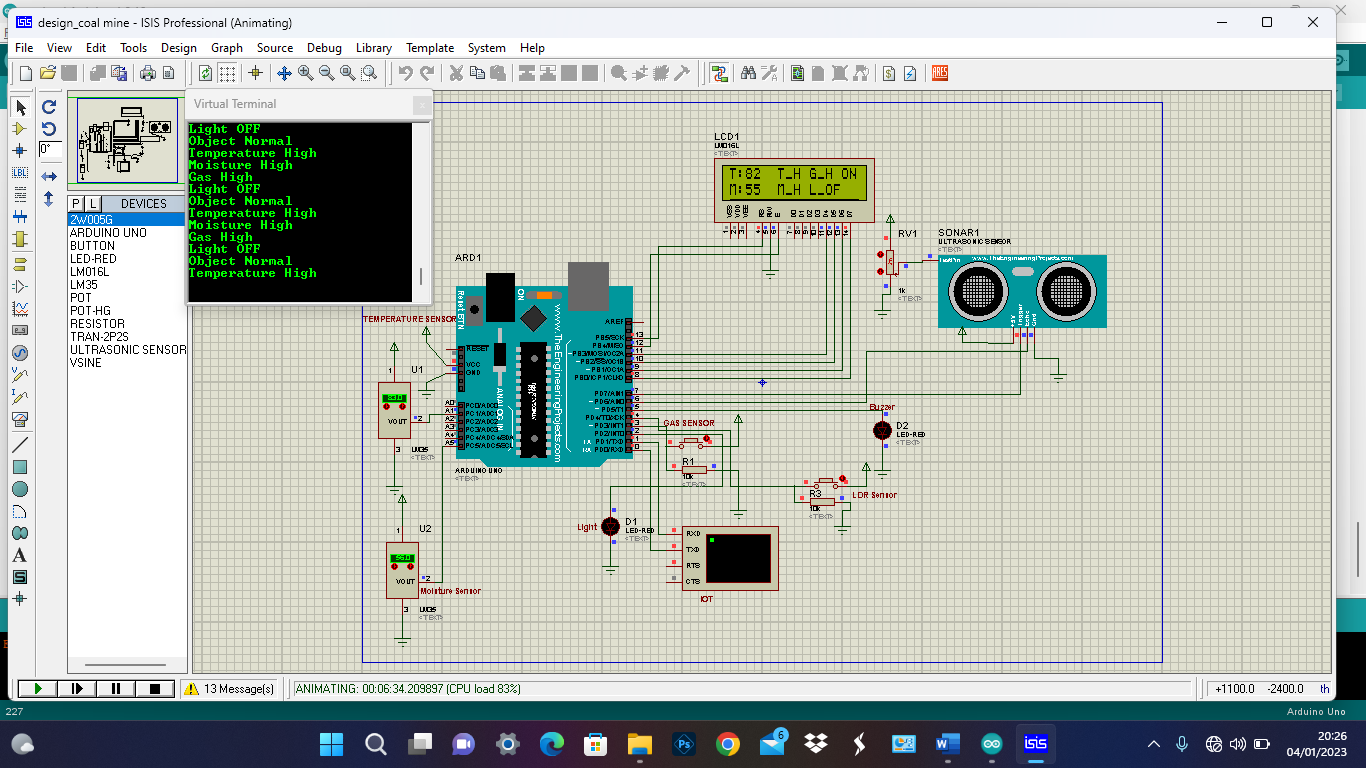
**Appendices:**

**a) Samples Screenshots:**









**b) Sample Coding :**

**#include<LiquidCrystal.h>**

**LiquidCrystal lcd(13,12,11,10,9,8);**

**int moist,t,gas,c,ldr;**

**int t1;**

**int m1;**

**int m;**

**const int trigPin = 7;**

**const int echoPin = 6;**

**long duration;**

**int distanceCm, distanceInch;**

**void setup() {**

**// put your setup code here, to run once:**

**Serial.begin(9600);**

**pinMode(3,INPUT); //gas**

**pinMode(4,INPUT); // ldr**

**pinMode(5,OUTPUT); // buzzer**

**pinMode(2,OUTPUT); // light**

**pinMode(A1,INPUT); //temperature**

**pinMode(A5,INPUT); //moisture**

**pinMode(trigPin, OUTPUT);**

**pinMode(echoPin, INPUT);**

**lcd.begin(16,2);**

**}**

**void loop()**

**{**

**ultrasonic();**

**gas=digitalRead(3);**

**ldr=digitalRead(4);**

**t=analogRead(A1);**

**t1=t\*0.48;**

**m=analogRead(A5);**

**m1=m\*0.48;**

**lcd.setCursor(0,0);**

**lcd.print("T:");**

**lcd.setCursor(2,0);**

**lcd.print(t1);**

**lcd.print(" ");**

**lcd.setCursor(0,1);**

**lcd.print("M:");**

**lcd.setCursor(2,1);**

**lcd.print(m1);**

**lcd.print(" ");**

**if (t1>40)**

**{**

**lcd.setCursor(6,0);**

**lcd.print("T\_H");**

**Serial.println("Temperature High");**

**delay(500);**

**digitalWrite(5,HIGH); ////buzzzer**

**}**

**else**

**{**

**lcd.setCursor(6,0);**

**lcd.print("T\_L");**

**Serial.println("Temperature Low");**

**delay(500);**

**digitalWrite(5,LOW); ////buzzzer**

**}**

**if (m1>50)**

**{**

**lcd.setCursor(6,1);**

**lcd.print("M\_H");**

**Serial.println("Moisture High");**

**delay(500);**

**digitalWrite(5,HIGH); ////buzzzer**

**}**

**else**

**{**

**lcd.setCursor(6,1);**

**lcd.print("M\_L");**

**Serial.println("Moisture Low");**

**delay(500);**

**digitalWrite(5,LOW); ////buzzzer**

**}**

**if(gas==HIGH )**

**{**

**lcd.setCursor(10,0);**

**lcd.print("G\_H");**

**Serial.println("Gas High");**

**delay(500);**

**digitalWrite(5,HIGH); ////buzzzer**

**}**

**else**

**{**

**lcd.setCursor(10,0);**

**lcd.print("G\_L");**

**Serial.println("Gas Low");**

**delay(500);**

**digitalWrite(5,LOW); ////buzzzer**

**}**

**if(ldr==LOW)**

**{**

**lcd.setCursor(10,1);**

**lcd.print("L\_ON");**

**Serial.println("Light ON");**

**delay(500);**

**digitalWrite(2,HIGH); ////buzzzer**

**}**

**else**

**{**

**lcd.setCursor(10,1);**

**lcd.print("L\_OF");**

**Serial.println("Light OFF");**

**delay(500);**

**digitalWrite(2,LOW); ////buzzzer**

**}**

**}**

**void ultrasonic()**

**{**

**digitalWrite(trigPin, LOW);**

**delayMicroseconds(2);**

**digitalWrite(trigPin, HIGH);**

**delayMicroseconds(10);**

**digitalWrite(trigPin, LOW);**

**duration = pulseIn(echoPin, HIGH);**

**distanceCm= duration\*0.034/2,**

**lcd.setCursor(8,1);**

**lcd.print("D:");**

**lcd.print(distanceCm);**

**lcd.print(" ");**

**if(distanceCm >500)**

**{**

**lcd.setCursor(14,0);**

**lcd.print("OD");**

**Serial.println("Object Detected");**

**delay(500);**

**digitalWrite(5,HIGH); ////buzzzer**

**}**

**else**

**{**

**lcd.setCursor(14,0);**

**lcd.print("ON");**

**Serial.println("Object Normal");**

**delay(500);**

**digitalWrite(5,LOW); ////buzzer**

**}**