**IOT based gas leakage detector System using Arduino and GSM module**

**INTRODUCTION**

LPG is the abbreviation; it’s a non-renewable supply of energy. It is taken out from rock oil and gas. LPG is very burnable and should thus behold on-off from sources of a solenoid and during a blowy space so that any run will safely. LPG vapors are more steam than air thus care to be taken all over storage so that any run won't sink to the bottom and find accumulated in a district that is low untruthful and tough to disperse. LPG gas is an alkane and it's scentless in its state of nature. The stink that we tend to observe once there's a run is really of a wholly different agent. This material is added to the gas at one time it leaves the most storage terminals. This project aims to detect Gas leakage in houses, restaurants, schools, and other places, and gives messages to the nearby people. These days Gas sensors are being used nationally in the field like safety, health, appliances, etc. This project is an implementation using an MQ sensor. The MQ sensor is used for detecting gas leakage for different implementations. The device also keeps displaying the leakage in the LCD. The MQ sensor searches the concentration of gas and outputs an analog value that can be converted to a digital signal using an inbuilt A to D Convertor. This project permitsthe user to set the low, medium, and dangerous levels for leakage based on the same digital measure. The strength values are differentiated with two thresholds and based on that, it classifies into three different classes. Liquified petroleum gas (LPG) is used in every sector. It is also used for industries-based purposes. The main advantages of LPG Gas leakage many accidents happen and their result shows both material, product loss, and human injuries. The principal motivation behind our frame is to differentiate the gas in houses and other homegrown with the help of a gas sensor. After identifying the message will be conveyed to the person.

**PROPOSED SYSTEM**

The sensors are powered by microcontrollers or relays and LCDs and a buzzer. This voltage rule sector is accountable for converting alternate power to direct current as well as lowering the transmitted signal. The sensors can detect a gas leak. The sensor MQ working here to detect LPG levels in the air. The gasses on the scale between 200 and 10000 ppm maybe identify as well as the reaction time is completely speedy. The result of the sensors would be an analog power. A sequential communication circuit makes over the change from an analog resistor to voltage. The microcontrollers report that voltage. This analog voltage is digitally converted using a 10-bit Analog to a digital converter. In the advanced system of a gas detection system, the implementation quells both the monitoring and detection of the gases which are very dangerous to the surrounding. In the observation of the gas, the sensor which is used to hear many gases is MQ sensor. After the detection of leakage in the gas, the sensor sends the signal to the Arduino UNO for further operation where other hardware components are connected. Through Arduino UNO, it sends the signal to the LCD for displaying the alert message as LPG Detected, suitably, the buzzer be on so that the backdrop people will the warn, as well as the main power supply, will be cut off. Using the relay of 5V, the power supply is given to the expend fan to detach the harmful gas from the surrounding. Even the container of the application will accept the message through the GSM module.

**LITERATURE SURVEY**

1. IOT BASED GAS LEAKAGE SYSTEM USING ARDUINO

**AUTHORS** - Prof. A.P. Linge, Aakanksha A. Gulhane, Pragati P. Bacahte, Aditya P. Pethkar, Kalyani S. Nakhate, Amruta V. Mehare.

**ABSTRACT**

The Internet of Things (IoT) aims to automate the lives of the world by giving the path with or without human interference which will automate the tasks which may be bigger or smaller than we encounter. Because the Internet of Things (IoT) intends to simplify working, it is also practical to use well-being to reinforce present security standards. The essential goal of every project has not gone ignored by IoT. In open or closed situations, gas leakage may be savage. While traditional gas detection systems are noiseless and accurate, they are unaware of a few key aspects in the area of warning people of a leak. As a result, we have built the implementation for both industry and the society which will detect the leakage of gas and also monitor the gas availability. Alerting techniques that include sending messages to the applicable command as well as the ability to analyze sensor reading data. These days, gas leakage and detection are major concerns in our daily lives. LPG gas is very burnable, posing a risk to both people and property. To avoid such accidents, a notable amount of try has gone into developing reliable systems for detecting gas leaks. Our significant objective is to recommend a gas detection that includes gas leakage detecting hardware to households in the area. This can monitor dangerous chemicals in the air at workplaces and it may also be used in households by alerting through an LCD and sending a message to a recorded phone number.

1. Gas Leakage Detection Based on IOT, Proceedings of the Third International Conference on Electronics Communication and Aerospace Technology [ICECA 2019] IEEE Conference Record # 45616; pp.no 1312 1315.

**AUTHORS** - Suma V, Ramya R Shekar, Akshay Kumar A

**ABSTRACT**

Gas leakages results a serious problem in household and other areas where household gas is used, therefore the proposed gas leakage detection and monitoring system is developed. There are many methods available for booking a Gas Refill, methods include online booking, telephonic booking etc. It will be difficult situation for the

one who uses LPG gas for cooking regularly. The aim of this paper is to present a new system automatically books a cylinder when the gas is about to empty is by sending a notification to the gas agency using wifi using Internet of Things approach. In addition to that sensor is used to detect gas leakage at home. If the gas leakage is sensed automatically it will send SMS to the user. wifi is one of the most used networks across the world. Hence, load cell has been used to monitor the weight of the LPG gas regularly. The values are next fed to the microcontroller. If the gas in the cylinder indicates a value where the remaining percentage level is crossed below the threshold level set for gas to be indicated as getting emptied, then a notification will be delivered to gas enterprise automatically to book the new cylinder. Subsequently, reply notification will be sent to the customer about the booking status. At the same time, application software is developed in the gas enterprise to inform and record the booking. This, work this helps the society to specifically indicate gas leakage and also helps both customers and the agency to get the gas booking made automatically using the IOT technique.

1. IOT Based LPG Gas Leakage Detection System

**AUTHORS** - Kirti Nigam, Manuraj Singh Thakur, Jatin Yadav, Juhi Saxena, Assistant Professor Dr. Prashant

**ABSTRACT**

Liquefied Petroleum Gas (LPG) is a main source of fuel, especially in urban areas because it is clean compared to firewood and charcoal. Gas leakage is a major problem in the industrial sector, residential premises, etc. Nowadays, home security has become a major issue because of increasing gas leakage. Gas leakage is a source of great anxiety with ateliers, residential areas and vehicles like Compressed Natural Gas (CNG), buses, and cars which are run on gas power. One of the preventive methods to stop accidents associated with the gas leakage is to install a gas leakage detection kit at vulnerable places. The aim of this paper is to propose and discuss a design of a gas leakage detection system that can automatically detect, alert and control gas leakage. This proposed system also includes an alerting system for the users. The system is based on a sensor that easily detects a gas leakage.

1. Gas Leakage Detection and Alert System using IoT

**AUTHORS -** Sayali Joshi, Shital Munjal, Prof. Uma B. Karanje

**ABSTRACT**

The Internet of things (IoT) is the system of gadgets, vehicles, and home machines that contain hardware, programming, actuators, and network which enables these things to interface, collaborate and trade

information. IoT includes broadening Internet network past standard device, for example, work areas, workstations, cell phones and tablets, to any scope of generally stupid or non-web empowered physical

device and ordinary articles. Installed with innovation, these gadgets can convey and connect over the Internet, and they can be remotely observed and controlled. The meaning of the Internet of things has advanced because of union of numerous innovations, ongoing examination, AI, ware sensors, and implanted frameworks. Conventional fields of installed frameworks, remote sensor systems, control frameworks computerization (counting home and building mechanization), and others all add to empowering the Internet of things. A gas spill alludes to a hole of petroleum gas or different vaporous item from a pipeline or other regulation into any territory where the gas ought not to be available. Since a little hole may steadily develop a hazardous convergence of gas, spills are perilous. Notwithstanding causing flame and blast dangers, holes can slaughter vegetation, including huge trees, and may discharge amazing ozone harming substances to the environment. Keywords: IOT, MQ5 sensor, Arduino module, GSM networks.

1. SENSOR-BASED GAS LEAKAGE DETECTOR SYSTEM

**AUTHORS** - Machavrapu Lalitha Sree, Iswarya .S, Pooja .A, Poovarasi .G

**ABSTRACT**

Liquefied Petroleum Gas (LPG) is a main source of fuel, especially in urban areas because it is clean compared to firewood and charcoal. Gas leakage is a major problem in the industrial sector, residential premises, etc. Nowadays, home security has become a major issue because of increasing gas leakage. Gas leakage is a source of great anxiety with ateliers, residential areas and vehicles like Compressed Natural Gas (CNG), buses, and cars which are run on gas power. One of the preventive methods to stop accidents associated with the gas leakage is to install a gas leakage detection kit at vulnerable places. The aim of this paper is to propose and discuss a design of a gas leakage detection system that can automatically detect, alert and control gas leakage. This proposed system also includes an alerting system for the users. The system is based on a sensor that easily detects a gas leakage.

**BLOCK DIAGRAM**

POWER SUPPLY

ARDUINO UNO

BUZZER

16\*2 LCD

DC FAN

RELAY

GSM MODULE

MQ SENSOR

Figure 3.1.1

**HARDWARE COMPONENTS**

* ARDUINO UNO
* POWER SUPPLY
* LCD
* BUZZER
* RELAY
* FAN
* MQ SENSOR
* GSM MODULE

**SOFTWARE COMPONENTS**

* EMBEDDED C
* ARDUINO IDE

**HARDWARE EXPLANATION**

* **POWER SUPPLY**

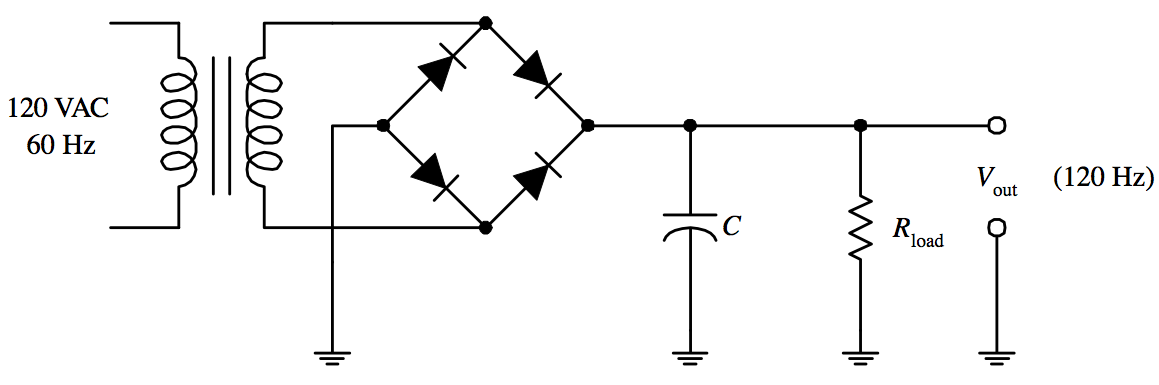


Figure 3.1.2

A power supply is an electrical device that supplies electric power to an electrical load. The primary function of a power supply is to convert electric current from a source to the correct voltage, current, and frequency to power the load. As a result, power supplies are sometimes referred to as electric power converters. Some power supplies are separate standalone pieces of equipment, while others are built into the load appliances that they power.

All power supplies have a power input connection, which receives energy in the form of electric current from a source, and one or more power output connections that deliver current to the load. The source power may come from the electric power grid, such as an electrical outlet, energy storage devices such as batteries or fuel cells. The power supply which we are using consists of transformers which reduces alternating current to required voltage coming with the flow of power supply after reduction of voltage then we use a full wave bridge rectifier to convert AC to DC and then the DC current obtained is fed to the filters and then to the required voltage regulator and the to a load, the output of this power supply is obtained from the load and given to the components and the microcontroller. The function of a linear voltage regulator is to convert a varying DC voltage to a constant, often specific, lower DC voltage. In addition, they often provide a current limiting function to protect

the power supply and load from overcurrent (excessive, potentially destructive current). A constant output voltage is required in many power supply applications, but the voltage provided by many energy sources will vary with changes in load impedance. Furthermore, when an unregulated DC power supply is the energy source, its output voltage will also vary with changing input voltage. To circumvent this, some power supplies use a linear voltage regulator to maintain the output voltage at a steady value, independent of fluctuations in input voltage and load impedance. Linear regulators can also reduce the magnitude of ripple and noise on the output voltage.

**ARDUINO UNO**

The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc. 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 I/O pins (six capable of PWM output), 6 analog I/O pins, and is programmable with the Arduino IDE (Integrated Development Environment), via a type B USB cable. It can be powered by the USB cable or by an external 9-volt battery, though it accepts voltages between 7 and 20 volts. It is similar to the Arduino Nano and Leonardo. The hardware reference design is distributed under a Creative Commons Attribution Share-Alike 2.5 license and is available on the Arduino website. Layout and production files for some versions of the hardware are also available.

The word "uno" means "one" in Italian and was chosen to mark the initial release of Arduino Software. The Uno board is the first in a series of USB-based Arduino boards; it and version 1.0 of the Arduino IDE were the reference versions of Arduino, which have now evolved to newer releases. The ATmega328 on the board comes preprogrammed with a bootloader that allows

uploading new code to it without the use of an external hardware programmer.

 Figure 3.2.1

While the Uno communicates using the original STK500 protocol, it 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.

**TECHNICAL SPECIFICATIONS**

* Microcontroller: Microchip ATmega328P.
* Operating Voltage: 5 Volts.
* Input Voltage: 7 to 20 Volts.
* Digital I/O Pins: 14 (of which 6 can provide PWM output).
* UART: 1.
* I2C: 1.
* SPPI: 1.
* Analog Input Pins: 6.
* DC Current per I/O Pin: 20 Ma.
* DC Current for 3.3V Pin: 50 mA.
* Flash Memory: 32 KB of which 0.5 KB used by bootloader.
* SRAM: 2 KB.
* EEPROM: 1 KB.
* Clock Speed: 16 MHz
* Length: 68.6 mm.
* Width: 53.4 mm.
* Weight: 25 g.

**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 is off.
* VIN: The input voltage to the Arduino/Genuino board when it is 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 that 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, under software control (using pinMode(), digitalWrite(), and digitalRead() functions). They operate at 5 volts. Each pin can provide or receive 20 mA as the recommended operating condition and has an internal pull-up resistor (disconnected by default) of 20-50K ohm. A maximum of 40mA must not be exceeded on any I/O pin to avoid permanent damage to the microcontroller. The Uno has 6 analog inputs, labeled A0 through A5; each provides 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 the range using the AREF pin and the analogReference() function.

In addition, some pins have specialized functions:

* Serial / 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 (pulse-width modulation): pins 3, 5, 6, 9, 10, and 11. Can provide 8-bit PWM output with the analogWrite() function.
* SPI (Serial Peripheral Interface): pins 10 (SS), 11 (MOSI), 12 (MISO), and 13 (SCK). These pins support SPI communication using the SPI library.
* TWI (two-wire interface) / I²C: pin SDA (A4) and pin SCL (A5). Support TWI communication using the Wire library.
* AREF (analog reference): Reference voltage for the analog inputs.[7]

**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. 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 1). A SoftwareSerial library allows serial communication on any of the Uno's digital pins.

**Automatic (software) reset**

Rather than requiring a physical press of the reset button before an upload, the Arduino/Genuino Uno board is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2/16U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip.

This setup has other implications. When the Uno is connected to a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader

is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened.

**GSM MODULE**

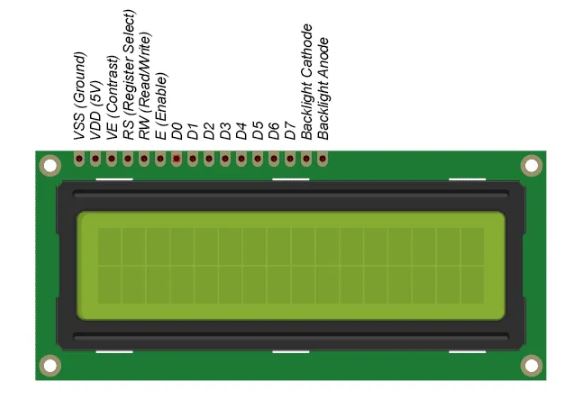
The Global System for Mobile Communications (GSM) is a standard developed by the European Telecommunications Standards Institute (ETSI) to describe the protocols for second-generation (2G) digital cellular networks used by mobile devices such as mobile phones and tablets. It was first deployed in Finland in December 1991. By the mid-2010s, it became a global standard for mobile communications achieving over 90% market share, and operating in over 193 countries and territories. 2G networks developed as a replacement for first generation (1G) analog cellular networks. The GSM standard originally described a digital, circuit-switched network optimized for full duplex voice telephony. This expanded over time to include data communications, first by circuit-switched transport, then by packet data transport via General Packet Radio Service (GPRS), and Enhanced Data Rates for GSM Evolution (EDGE). Subsequently, the 3GPP developed third-generation (3G) UMTS standards, followed by fourth-generation (4G) LTE Advanced standards, which do not form part of the

ETSI GSM standard. "GSM" is a trade mark owned by the GSM Association. It may also refer to the (initially) most common voice codec used, Full Rate.

Figure 3.2.2

**GENERAL FEATURES OF SIM 800C**

* Quad-band 850/900/1800/1900MHz
* GPRS multi-slot class 12/10
* GPRS mobile station class B
* Compliant to GSM phase 2/2+
  + Class 4 (2 W @ 850/900MHz)
  + Class 1 (1 W @ 1800/1900MHz)
* Dimensions: 17.6\*15.7\*2.3mm
* Weight: 1.3g
* Control via AT commands (3GPP TS 27.007, 27.005 and SIMCom enhanced AT Commands)
* Supply voltage range 3.4 ~ 4.4V
* Low power consumption
* Operation temperature:-40℃ ~85℃
* **16\*2 LCD**

 Figure 3.3.1

LCD modules are very commonly used in most embedded projects, the reason being its cheap price, availability and programmer friendly. Most of us would have come across these displays in our day to day life, either at PCO’s or calculators. The appearance and the pinouts have already been visualized above now let us get a bit technical.

16×2 LCD is named so because; it has 16 Columns and 2 Rows. There are a lot of combinations available like, 8×1, 8×2, 10×2, 16×1, etc. but the most used one is the 16×2 LCD. So, it will have (16×2=32) 32 characters in total and each character will be made of 5×8 Pixel Dots.

Now, we know that each character has (5×8=40) 40 Pixels and for 32 Characters we will have (32×40) 1280 Pixels. Further, the LCD should also be instructed about the Position of the Pixels. Hence it will be a hectic task to handle everything with the help of MCU, hence an Interface IC like HD44780is used, which is mounted on the backside of the LCD Module itself. The function of this IC is to get the Commands and Data from the MCU and process them to display meaningful information onto our LCD Screen. You can learn how to interface an LCD using the above mentioned links. If you are an advanced programmer and would like to create your own library for interfacing your Microcontroller with this LCD module then you have to understand the HD44780 IC is working and commands which can be found its datasheet.

* **RELAY**



Relays are the switches which aim at closing and opening the circuits electronically as well as electromechanically. It controls the opening and closing of the circuit contacts of an electronic circuit. When the relay contact is open (NO), the relay isn’t energize with the open contact. However, if it is closed (NC), the relay isn’t energize given the closed contact. However, when energy (electricity or charge) is supplied, the states are prone to change. Relays are normally used in the control panels, manufacturing and building automation to control the power along with switching the smaller current values in a control circuit. However, the supply of amplifying effect can help control the large amperes and voltages because if low voltage is applied to the relay coil, a large voltage can be switched by the contacts. If preventive relays are being used, it can detect overcurrent, overload, undercurrent, and reverse current to ensure the protection of electronic equipment. Last but not the least; it is used to heat the elements, switch on audible alarms, switch the starting coils, and pilots the lights.

**Relay Types**

In addition to the electromechanical and electromagnetic relay, there is a wide variety of relays with different working principles; principles of operation and polarity.

* Electro-thermal Relay – When two different material gets in contact, bimetallic strip is formed, and when it is energized, it bends. This bending allows the users to make contact connections
* Electromechanical Relay – When different mechanical parts are connected on the basis of the electromagnet, contact connection is established
* Solid State Relay –This relay uses semiconductor devices to make a connection to ensure the effectiveness, efficiency, and easiness of the switching speed. This is commonly used for two reasons; faster-switching process and durability
* Hybrid Relay – It is the name given to the solid-state and electromechanical relays

**BUZZER**

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

Figure 3.4.1 

**TYPES**

* Electromechanical

Early devices were based on an electromechanical system identical to an electric bell without the metal gong. Similarly, a relay may be connected to interrupt its own actuating current, causing the contacts to buzz (the contacts buzz at line frequency if powered by alternating current) Often these units were anchored to a wall or ceiling to use it as a sounding board. The word "buzzer" comes from the rasping noise that electromechanical buzzers made.

* Mechanical

A joy buzzer is an example of a purely mechanical buzzer and they require drivers. Other examples of them are doorbells.

* Piezoelectric

A piezoelectric element may be driven by an oscillating electronic circuit or other audio signal source, driven with a piezoelectric audio amplifier. Sounds commonly used to indicate that a button has been pressed are a click, a ring or a beep.

Interior of a readymade loudspeaker, showing a piezoelectric-disk-beeper (With 3 electrodes ... including 1 feedback-electrode ( the central, small electrode joined with red wire in this photo), and an oscillator to self-drive the buzzer.

A piezoelectric buzzer/beeper also depends on acoustic cavity resonance or Helmholtz resonance to produce an audible beep.

**Modern applications**

While technological advancements have caused buzzers to be impractical and undesirable, there are still instances in which buzzers and similar circuits may be used. Present day applications include:

* Novelty uses
* Judging panels
* Educational purposes
* Annunciator panels
* Electronic metronomes
* Game show lock-out device
* Microwave ovens and other household appliances
* Sporting events such as basketball games
* Electrical alarms
* Joy buzzer (mechanical buzzer used for pranks)

**MQ 3 SENSOR**



Figure 3.4.2

**PIN DISCRIPTION**

|  |  |
| --- | --- |
| **Pin Name** | **Description** |
| VCC | This pin powers the module, typically the operating voltage is +5V |
| GND | Used to connect the module to system ground |
| Digital Out (DO) | You can also use this sensor to get digital output from this pin, by setting a threshold value using the potentiometer |
| Analog Out (AO) | This pin outputs 0-5V analog voltage based on the intensity of the gas |

MQ-3 module is suitable for detecting Alcohol, Benzine, CH4, Hexane, LPG, CO. Sensitive material of MQ-3 gas sensor is SnO2, which with lower conductivity in clean air. When the target alcohol gas exist, the sensor’s conductivity is more higher along with the gas concentration rising. MQ-3 gas sensor has high sensitity to Alcohol, and has good resistance to disturb of gasoline, smoke and vapor.

This sensor provides an analog resistive output based on alcohol concentration. When the alcohol gas exist, the sensor’s conductivity gets higher along with the gas concentration rising.

There is a resistance across an A and B inside the sensor which varies on detection of alcohol. More the alcohol, the lower the resistance. The alcohol is measured by measuring this resistance. The sensor and load resistor form a voltage divider, and the lower the sensor resistance, the higher the voltage reading will be. The heater provides

necessary work conditions for work of sensitive components. The enveloped MQ-3 have 6 pin, 4 of them are used to fetch signals, and other 2 are used for providing heating current.

**FEATURES OF ALCOHOL SENSOR**

1. Sensor Type - Semiconductor
2. Easy SIP header interface
3. Compatible with most of the microcontrollers
4. Low-power standby mode
5. Requires heater voltage
6. Good sensitivity to alcohol gas
7. Fast response and High sensitivity
8. Long life and low cost
9. Requires simple Drive circuit

**Applications of MQ Gas Sensor**

1. Gas level over-limit alarm
2. Breathalyser
3. Portable alcohol detector
4. Stand-alone/background sensing device
5. Environmental monitoring equipment

**SOFTWARE EXPLANATION**

* **EMBEDDED C**

Embedded C is a set of language extensions for the C programming language by the C Standards Committee to address commonality issues that exist between C extensions for different embedded systems.

Embedded C programming typically requires nonstandard extensions to the C language in order to support enhanced microprocessor features such as fixed-point arithmetic, multiple distinct memory banks, and basic I/O

operations. In 2008, the C Standards Committee extended the C language to address such capabilities by providing a common standard for all implementations to adhere to. It includes a number of features not available in normal C, such as fixed-point arithmetic, named address spaces and basic I/O hardware addressing. Embedded C uses most of the syntax and semantics of standard C, e.g., main() function, variable definition, datatype declaration, conditional statements (if, switch case), loops (while, for), functions, arrays and strings, structures and union, bit operations, macros, etc.

Embedded software is computer software, written to control machines or devices that are not typically thought of as computers, commonly known as embedded systems. It is typically specialized for the particular hardware that it runs on and has time and memory constraints. This term is sometimes used interchangeably with firmware.

A precise and stable characteristic feature is that no or not all functions of embedded software are initiated/controlled via a human interface, but through machine-interfaces instead.

Manufacturers build embedded software into the electronics of cars, telephones, modems, robots, appliances, toys, security systems, pacemakers, televisions and set-top boxes, and digital watches, for example. This software can be very simple, such as lighting controls running on an 8-bit microcontroller with a few kilobytes of memory with the suitable level of processing complexity determined with a Probably Approximately Correct Computation framework (a methodology based on randomized algorithms), or can become very sophisticated in applications such as airplanes, missiles, and process control systems.

**OPERATING SYSTEMS**

Unlike standard computers that generally use an operating systems such as OS X, Windows or GNU/Linux, embedded software may use no operating system, or when they do use on, a wide variety of operating systems can be chosen from, typically a real-time operating system. Code is typically written in C or C++, but various high-level programming languages, such as Python and JavaScript, are now also in common use to target microcontrollers and embedded systems. Ada is used in some military and aviation projects.

**DIFFERENCES FROM APPLICATION SOFTWARE**

Most consumers are familiar with application software that provide functionality on a computer. However embedded software is often less visible, but no less complicated. Unlike application software, embedded software has fixed hardware requirements and capabilities, and addition of third-party hardware or software is strictly controlled.

Embedded software needs to include all needed device drivers at manufacturing time, and the device drivers are written for the specific hardware. The software is highly dependent on the CPU and specific chips chosen. Most embedded software engineers have at least a passing knowledge of reading schematics, and reading data sheets

for components to determine usage of registers and communication system. Conversion between decimal, hexadecimal and binary is useful as well as using bit manipulation.

Web applications are rarely used, although XML files and other output may be passed to a computer for display. File systems with folders are typically absent as are SQL databases.

Software development requires use of a cross compiler, which runs on a computer but produces executable code for the target device. Debugging requires use of an in-circuit emulator, JTAG or SWD. Software developers often have access to the complete kernel (OS) source code.

Size of the storage memory and RAM can vary significantly. Some systems run in 16 KB of Flash and 4 KB of RAM with a CPU operating at 8 MHz, other systems can rival contemporary computers. These space requirements lead to more work being done in C or embedded C++, instead of C++. Interpreted languages like BASIC (while e.g. Parallax Propeller can use compiled BASIC) and Java (Java ME Embedded 8.3 is available for e.g. ARM Cortex-M4, Cortex-M7 microcontrollers and older ARM11 used in Raspberry Pi and Intel Galileo Gen. 2) are not commonly used; while an implementation of the interpreted Python 3 language – MicroPython – is however available expressly for microcontroller use, e.g. 32-bit ARM-based (such as BBC micro:bit) and 16-bit PIC microcontrollers.

**COMMUNICATION PROTOCOLS**

Communications between processors and between one processor and other components are essential. Besides direct memory addressing, common protocols include I²C, SPI, serial ports, and USB.

**ARDUINO IDE**

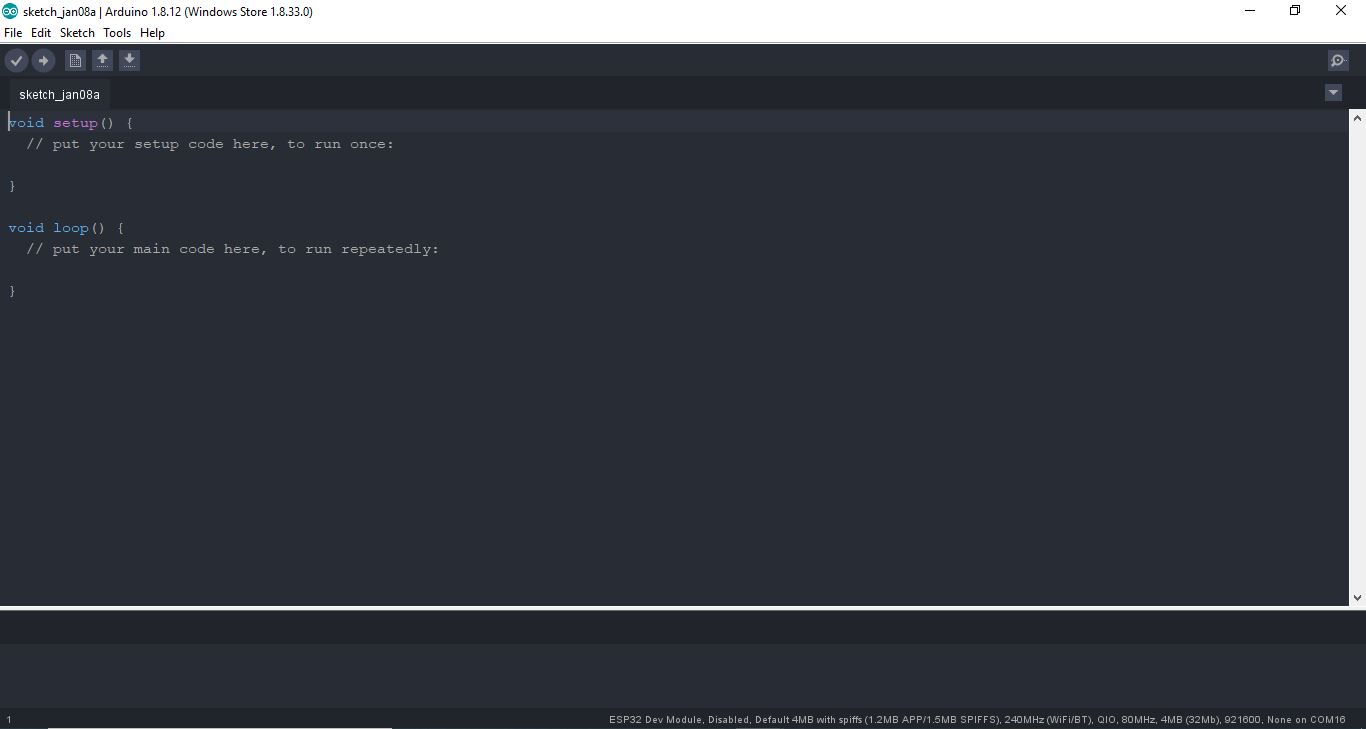
****

Figure 3.5.1

The Arduino Integrated Development Environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in functions from C and C++. It is used to write and upload programs to Arduino compatible boards, but also, with the help of third-party cores, other vendor development boards.

The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution. The Arduino IDE employs the program avrdude to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware. By default, avrdude is used as the uploading tool to flash the user code onto official Arduino boards.

With the rising popularity of Arduino as a software platform, other vendors started to implement custom open source compilers and tools (cores) that can build and upload sketches to other microcontrollers that are not supported by Arduino's official line of microcontrollers.

In October 2019 the Arduino organization began providing early access to a new Arduino Pro IDE with debugging and other advanced features.

Arduino (/ɑːrˈdwiːnoʊ/) is an open-source hardware and software company, project and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices. It’s hardware products are licensed under a CC-BY-SA license, while software is licensed under the GNU Lesser

General Public License (LGPL) or the GNU General Public License (GPL), permitting the manufacture of Arduino boards and software distribution by anyone. Arduino boards are available commercially from the official website or through authorized distributors.

Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards ('shields') or breadboards (for prototyping) and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs from personal computers. The microcontrollers can be programmed using the C and C++ programming languages, using a standard API which is also known as the "Arduino language". In addition to using traditional compiler toolchains, the Arduino project provides an integrated development environment (IDE) and a command line tool (arduino-cli) developed in Go.

The Arduino project began in 2005 as a tool for students at the Interaction Design Institute Ivrea in Ivrea, Italy, aiming to provide a low-cost and easy way for novices and professionals to create devices that interact with their environment using sensors and actuators. Common examples of such devices intended for beginner hobbyists include simple robots, thermostats and motion detectors.

The name Arduino comes from a bar in Ivrea, Italy, where some of the founders of the project used to meet. The bar was named after Arduin of Ivrea, who was the margrave of the March of Ivrea and King of Italy from 1002 to 1014.

A program for Arduino hardware may be written in any programming language with compilers that produce binary machine code for the target processor. Atmel provides a development environment for their 8-bit AVR and 32-bit ARM Cortex-M based microcontrollers: AVR Studio (older) and Atmel Studio (newer).

**IDE**

The Arduino integrated development environment (IDE) is a cross-platform application (for Windows, macOS, and Linux) that is written in the programming language Java. It originated from the IDE for the languages Processing and Wiring. It includes a code editor with features such as text cutting and pasting, searching and

replacing text, automatic indenting, brace matching, and syntax highlighting, and provides simple one-click mechanisms to compile and upload programs to an Arduino board. It also contains a message area, a text console, a toolbar with buttons for common functions and a hierarchy of operation menus. The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project,

which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution. The Arduino IDE employs the program avrdude to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

**Pro IDE**

On October 18, 2019, Arduino Pro IDE (alpha preview) was released. The system still uses Arduino CLI (Command Line Interface), but improvements include a more professional development environment, autocompletion support, and Git integration. The application frontend is based on the Eclipse This is a Open Source IDE. The main features available in the alpha release are -

* Modern, fully featured development environment
* Dual Mode, Classic Mode (identical to the Classic Arduino IDE) and Pro Mode (File System view)
* New Board Manager
* New Library Manager
* Board List
* Basic Auto-Completion (Arm targets only)
* Git Integration
* Serial Monitor
* Dark Mode

**Sketch**

A sketch is a program written with the Arduino IDE. Sketches are saved on the development computer as text files with the file extension .ino. Arduino Software (IDE) pre-1.0 saved sketches with the extension .pde.

A minimal Arduino C/C++ program consists of only two functions:

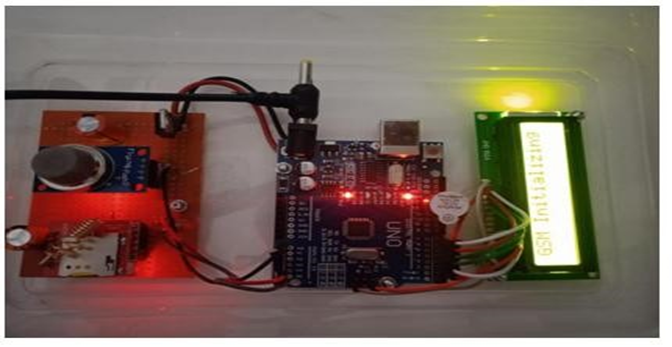
* setup(): This function is called once when a sketch starts after power-up or reset. It is us to initialize variables, input and output pin modes, and other libraries needed in the sketch. It is analogous to the function main().
* loop(): After setup() function exits (ends), the loop() function is executed repeatedly in the main program. It controls the board until the board is powered off or is reset. It is analogous to the function while(1).

**ADVANTAGES AND APPLICATIONS**

* The advantage of the Arduino Uno-based LPG detector system project is that it gives remote indications to the user about the LPG leakage with the help of SMS sing. This project has applications in our home. We can also use this gauge in industries, offices, and colleges where the LPG gas cylinder is used in the canteen. This project also has use in hotels and restaurants. To refine this project, we can add a GPS modem to this system. It is used in dangerous Gas detection. It is used in Fire Hazard Prevention. It is also used in Oxygen level Measurement. The sensor has exquisite sensitivity combined with a precipitate response time. The system is highly authentic, tamper-proof, and fixed. in the long run, the preservation cost is very less when compared to the present systems.

**Applications & Features:**

* The module is low-powered, and portable, hence, it is used in other applications such as Smoke Detector. ii. They are used in a household where the owner has to regulate and detect the gas leakage in the absence of the owner. iii. Gas detectors can be used to detect gases that catch fire easily, that are flammable, and which exhaust the oxygen (oxygen depletion). iv. This module can be seen at various Oil Plant, Manufacturing units to monitor the various process and where there is the constant use of oil takes place. v. This system can be used in Firefighting in the Fire Extinguishing Departm ent. Vi. Ensure worker’s health. Get an immediate gas alert. Prevent fire hazards about leakage.
* Arduboy, a handheld game console based on Arduino
* Arduinome, a MIDI controller device that mimics the Monome
* Ardupilot, drone software and hardware
* ArduSat, a cubesat based on Arduino.
* C-STEM Studio, a platform for hands-on integrated learning of computing, science, technology, engineering, and mathematics (C-STEM) with robotics.
* Data loggers for scientific research.
* OBDuino, a trip computer that uses the on-board diagnostics interface found in most modern cars
* OpenEVSE an open-source electric vehicle charger
* XOD, a visual programming language for Arduino

**RESULT**  Figure 3.5.2

**CONCULSION**

This system provides a fast and cost-effective solution to avert the gas leak effect by reducing the risk to human life. The statistics of the application of gas clam on to the application can be useful to own the faulty valves and regulators prior and do the necessary replacement. Apart from detecting the leakage, a two-level prevention apparatus makes the system more valid. The cost involved in developing the system is crucially low. In recent brood, the use of LPG is taking a big giant. From the use of cylinders up to the use of petroleum lines. The biggest warning in using this technology is security. Our project will prove to be resonance for households and industries.

**FUTURE SCOPE**

This monitoring system can be further increased by using Bluetooth in place of GSM to send the alert messages to the user, which abetment another real-time application. For the industrial sector, the data collected by the mobile application is beneficiary and used for data analytics. The combination of other sensors like temperature, pressure sensors, etc. makes the system a home computerization project. IoT turns drones into gas observation sensors. Another very interesting and extraordinary improvement would be to board reoccurring receiver MODEMS at different positions in the geographical area carrying duplicate SIM cards. The display can be another added variant in the project. Audio output can be settling to make it user-friendly.

**REFERENCES**

1. Suma V, Ramya R Shekar, Akshay Kumar A, Gas Leakage Detection Based on IOT, Proceedings of the Third International Conference on Electronics Communication and Aerospace Technology [ICECA 2019] IEEE Conference Record # 45616; pp.no 1312 1315.
2. M Athish Subramanian, Naveen Selvam, Rajkumar S, R Mahalakshmi, J Ramprabhakar , Gas Leakage Detection System using IoT with integrated notifications using Pushbullet-A Review, Proceedings of the Fourth International Conference on Inventive Systems and Control (ICISC 2020) IEEE Xplore Part Number: CFP20J06-ARTI; pp.no 359363.
3. Ravi Kishore Kodali, Greeshma, R.N.V, Kusuma Priya Nimmanapalli, Yatish Krishna Yogi Borra, IOT Based Industrial Plant Safety Gas Leakage Detection System, International Conference on Computing Communication and Automation (ICCCA), pp.no, 1-5, published in 2018.
4. Nagib Mahfuz, Shawan Karmokar, Md. Ismail Hossain Rana, A Smart Approach of LPG Monitoring and Detection System Using IoT, 11th International Conference on Computing Communication and Networking Technologies(ICCCNT), published in 2020.
5. Asmita Varma, Prabhakar S, Kayalvizhi Jayavel, Gas Leakage Detection and Smart Alerting and Prediction Using IoT, 2017 Second International Conference On Computing and Communications Technologies (ICCCT17) 2017 IEEE, pp.no, 327-333.
6. T.Soundarya, J.V.Anchitaalagammai, G. Deepa Priya, S.S. Karthick kumar, "Cylinder LPG Gas Leakage Detection for Home Safety", Journal of Electronics and Communication Engineering (IOSR-JECE), published in 2014.
7. Legg, SW and Wang, C and Benavides-Serrano, AJ and Laird, CD Optimal gas detector placement under uncertainty considering ConditionalValue-at-Risk, Journal of Loss Prevention in the Process Industries Volume 26 Elsevier Publisher 2013.
8. Amsaveni, M and Anurupa, A and Preetha, RS Anu and Malarvizhi, C and Gunasekaran, Mr Gsm based LPG leakage detection and controlling system, The International Journal Of Engineering And Science (IJES) ISSN (e) 2015.