**GAS LEAKAGE MONITORING AND ALERTING SYSTEM FOR INDUSTRIES**

|  |  |
| --- | --- |
| Project name | GAS LEAKAGE MONITORING AND ALERTING SYSTEM FOR INDUSTRIES |
| Team ID | PNT2022TMID18930 |
| Team Members | Hari Vignesh S, Ezhilarasan M, Abishiek A, Gokul Chandru S |

**Abstract**

The objective of this project is to design a gas leakage detection system. In this project we are going to monitor the gas leakage in the air conditioner. Commonly a small leakage of gas may gradually build up an explosive concentration of gas. This project monitors the gas leakage in the air conditioner by using the gas sensor. Gas sensor is used to detect the gas and smoke present inside a room. This project used Arduino UNO as the Microcontroller where it process the input from the sensor. If gas leakage is detected, then the system gives alert alarm by using a buzzer sound and sent SMS through GSM module to communicate with the user.

## KEYWORDS:

Use of Arduino, Gas Sensor, GSM (Global System for Mobile Communication).

**Introduction**

Many has installed the air conditioner in their houses which makes them comfortable sleep-in night with the temperature-regulated room. Air conditioning units use R22 or Freon coolants. The latest models use a refrigerant called R410-A. All of these coolants are chlorofluorocarbons that pose several dangers to human life. Therefore, they must be disposed of very carefully. If your air conditioner isn’t cooling your room as well as it used to or if your evaporator coils are suddenly covered in ice, it’s most likely because your air conditioner gas is leaking.

The problem of air conditioner leakage can cause irreversible damage to your machine’s compressor. Moreover, it will most certainly significantly increase your overall electricity consumption. However, the most significant problem associated with gas leakage is that the gas released poses a substantial danger to human health. The coolant leaking from an air conditioner rapidly evaporates into a gas. This gas is lethal and can cause nausea, and in some cases, even asphyxiation. The leaked gas can also lead to skin irritation and dryness as well as increased heart rate. Continued exposure to this gas can lead to life-threatening medical conditions. The leakage of gas may due to some factors like corrosion of condenser coil which made up of copper present inside the air conditioner. The problem of corrosion has failed the HVAC industry for years now. It is difficult, though not impossible, to safeguard homes and industrial appliances against corrosion. Pitting corrosion is the most common kind of corrosion that affects aircon units and leads to such leaks. Gas sensor are now used to detect the presence of gas in room. This can help to detect the gas leakage and gives an alert and by this many lives were saved.

**SYSTEM REQUIREMENTS**

**HARDWARE REQUIREMENTS:**

* Arduino Uno
* LCD Display
* Gas Sensor
* Buzzer
* GSM

**SOFTWARE REQUIREMENTS:**

* Arduino IDE

**PROPOSED METHOD**

**GAS SENSOR**

**ARDUINO UNO**

**LCD**

**BUZZER**

**GSM**

**Figure 4.1. Gas leakage detection system**

The above-mentioned block figure 4.1 shows a block diagram for gas leakage detection system. The gas leakage detection system helps to detect the leakage of gas from air conditioner. This project used Arduino UNO as the Microcontroller where it processes the input from the sensor. If gas leakage is detected, then the system gives alert alarm by using the buzzer sound and sent SMS through GSM module to communicate with the user.

**ARDUINO UNO:**

Arduino Uno is a microcontroller board based on the ATmega328P. Figure. 2 has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller, simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started. You can tinker with your UNO without worrying too much about doing something wrong, worst-case scenario you can replace the chip for a few dollars and start over again.



**Arduino Uno**

“UNO” means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0. The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases. The Uno board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform; for an extensive list of current, past or outdated boards see the Arduino index of boards.

Power

The Arduino Uno board can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm centre-positive plug into the board's power jack. Leads from a battery can be inserted in the GND and Vin pin headers of the POWER connector.

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

The power pins are as follows:

Vin - The input voltage to the Arduino board when it’s using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin. 5V this pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 12V), the USB connector (SV), or the VIN pin of the board (712V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don’t advise it. 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 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.

Memory - The ATmega328 has 32 KB (with 0.5 KB occupied by the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output - See the mapping between Arduino pins and ATmega328P ports. The mapping for the Atmega8, 168, and 328 is identical. Each of the 14 digital 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 2050k ohm. A maximum of 40mA is the value that must not be exceeded on any 1/0 pin to avoid permanent damage to the microcontroller.

**Technical specifications of ATmega328p.**

|  |  |
| --- | --- |
| Microcontroller | ATmega328p |
| Operating Voltage | 5 V |
| Input Voltage(recommended) | 7-12 V |
| Input Voltage(limit) | 6-20 V |
| Digital I/O Pins | 14 (~6 PWM pins) |
| PWM Digital I/O Pins | 6 |
| Analog Input Pins | 6 |
| DC Current per I/O Pin | 20 mA |
| DC Current per 3.3V Pin | 50 mA |
| Flash Memory | 32 KB (ATmega328P) of which 0.5KB used by bootloader |
| SRAM | 2 KB(ATmega328P) |
| EPROM | 1 KB(ATmega328P) |
| Clock Speed | 16 MHz |
| Length | 68.6 min |
| Width | 53.4 min |
| Weight | 25 g |

In addition, some pins have specialized functions: Serial - 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) FTL, serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB to-TTL Serial chip.

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

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

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

LED - 13. 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

TWI - A4 or SDA pin and AS or SCL pin. Support TWI communication using the Wire library.

The Uno has 6 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 analog Reference() function.

There are a couple of other pins on the board:

AREF Reference voltage for the analog inputs. Used with analog Reference ().

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

Communication

Arduino Uno has a number of facilities for communicating with a computer, another Arduino board, or other microcontrollers. The ATmega328 provides UART TIL (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, an 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 1).

A Software Serial library allows serial communication on any of the Uno’s digital pins.

The ATmega328 also supports 12C (TWI) and SPI communication. The Arduino Software (IDE) includes a Wire library to simplify use of the 12C bus; see the documentation for details. For SPI communication, use the SPI library.

Automatic (Software) Reset

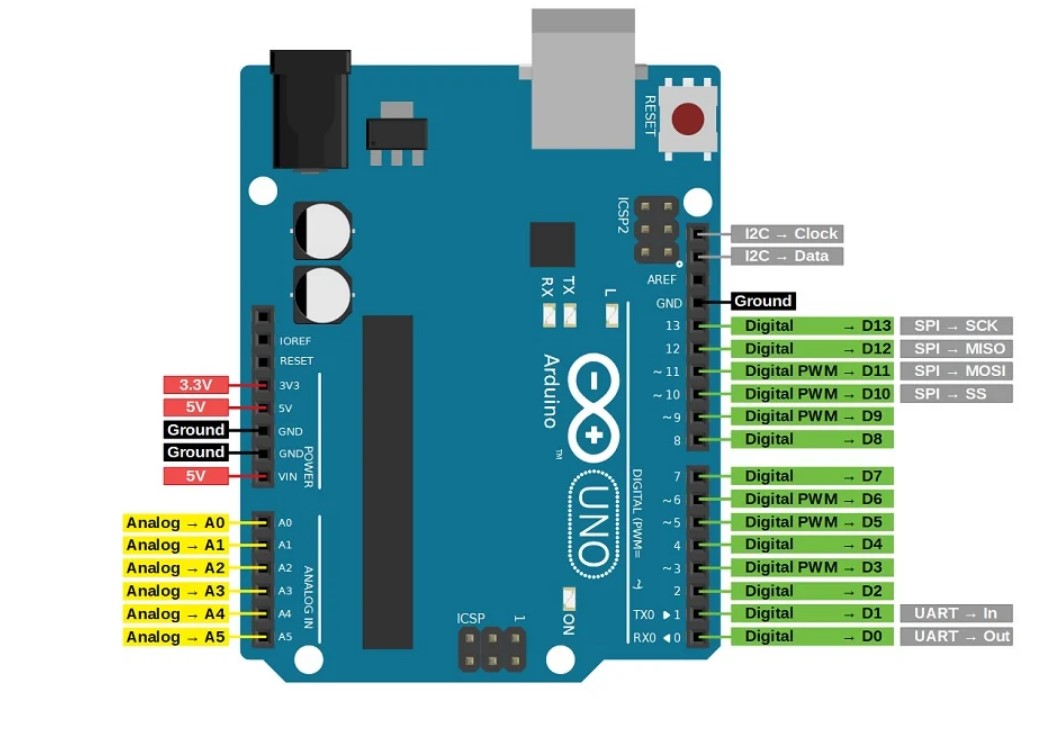
Rather than requiring a physical press of the reset button before an upload, the Arduino Uno board is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines: (DTR) of the ATmega8U2/16U2 is connected to the reset line of the ATmega328 via a 100 nano farad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino Software (IDE) uses this capability to allow you to upload code by simply pressing the upload button in the interface toolbar. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Uno is connected to either 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. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Uno board contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It’s labelled “RESET-EN”. You may also be able to disable the auto-reset by connecting a 110-ohm resistor from 5V to the reset line; see this forum thread for details.

Revisions

Revision 3 of the board has the following new features: 1.0 pinout: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible with both the board that uses the AVR, which operates with 5V and with the Arduino Due that operates with 3.3V. The second one is a not connected pin, that is reserved for future purposes. Stronger RESET circuit Atmega 16U2 replace the 8U2.



**Arduino UNO Pin Configuration**

**LCD DISPLAY:**

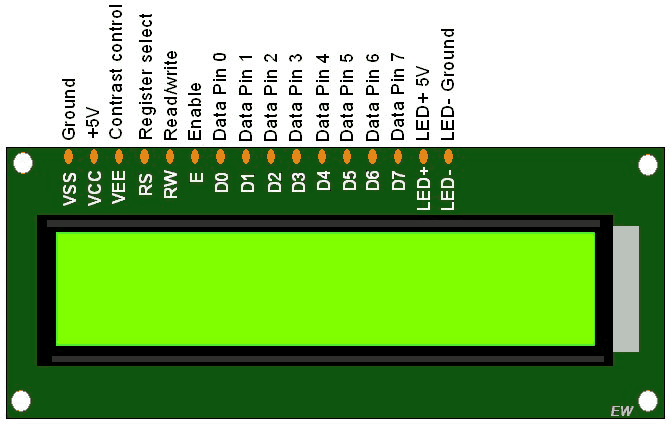
The term [LCD stands for liquid crystal display](https://www.elprocus.com/difference-alphanumeric-display-and-customized-lcd/). It is one kind of electronic display module used in an extensive range of applications like various circuits & devices like mobile phones, calculators, computers, TV sets, etc. These displays are mainly preferred for multi-segment [light-emitting diodes](https://www.elprocus.com/light-emitting-diode-led-working-application/) and seven segments. The main benefits of using this module are inexpensive; simply programmable, animations, and there are no limitations for displaying custom characters, special and even animations, etc.

The 16×2 LCD pinout is shown below.

* Pin1 (Ground/Source Pin): This is a GND pin of display, used to connect the GND terminal of the microcontroller unit or power source.
* Pin2 (VCC/Source Pin): This is the voltage supply pin of the display, used to connect the supply pin of the power source.
* Pin3 (V0/VEE/Control Pin): This pin regulates the difference of the display, used to connect a changeable POT that can supply 0 to 5V.
* Pin4 (Register Select/Control Pin): This pin toggles among command or data register, used to connect a microcontroller unit pin and obtains either 0 or 1(0 = data mode, and 1 = command mode).
* Pin5 (Read/Write/Control Pin): This pin toggles the display among the read or writes operation, and it is connected to a microcontroller unit pin to get either 0 or 1 (0 = Write Operation, and 1 = Read Operation).
* Pin 6 (Enable/Control Pin): This pin should be held high to execute Read/Write process, and it is connected to the microcontroller unit & constantly held high.
* Pins 7-14 (Data Pins): These pins are used to send data to the display. These pins are connected in two-wire modes like 4-wire mode and 8-wire mode. In 4-wire mode, only four pins are connected to the microcontroller unit like 0 to 3, whereas in 8-wire mode, 8-pins are connected to microcontroller unit like 0 to 7.
* Pin15 (+ve pin of the LED): This pin is connected to +5V
* Pin 16 (-ve pin of the LED): This pin is connected to GND.

### **Features of LCD16x2**

* The operating voltage of this LCD is 4.7V-5.3V
* It includes two rows where each row can produce 16-characters.
* The utilization of current is 1mA with no backlight
* Every character can be built with a 5×8-pixel box
* The alphanumeric LCDs alphabets & numbers
* Is display can work on two modes like 4-bit & 8-bit
* These are obtainable in Blue & Green Backlight
* It displays a few custom generated characters



**Lcd display.**

### Registers of LCD

A 16×2 LCD has two [registers](https://www.elprocus.com/know-about-types-of-registers-in-8051-microcontroller/) like data register and command register. The RS (register select) is mainly used to change from one register to another. When the register set is ‘0’, then it is known as command register. Similarly, when the register set is ‘1’, then it is known as data register.

**Command Register**

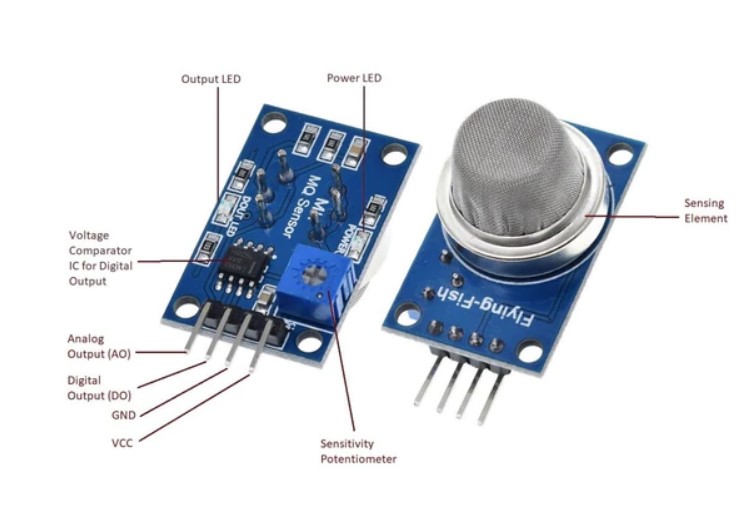
The main function of the command register is to store the instructions of command which are given to the display. So that predefined tasks can be performed such as clearing the display, initializing, set the cursor place, and display control. Here commands processing can occur within the register

**Data Register**

The main function of the data register is to store the information which is to be exhibited on the LCD screen. Here, the ASCII value of the character is the information which is to be exhibited on the screen of LCD. Whenever we send the information to LCD, it transmits to the data register, and then the process will be starting there. When register set =1, then the data register will be selected.

**GAS SENSOR:**

Sensitive material of MQ-2 gas sensor is SnO2, which with lower conductivity in clean air. When the target combustible gas exist, the sensor’s conductivity is higher along with the gas concentration rising. Convert change of conductivity to correspond output signal of gas concentration. MQ-2 gas sensor has highly sensitive to Propane and Hydrogen, also could be used to Methane and other combustible steam, it is with low cost and suitable for different application.



**Gas Sensor.**

Characteristics

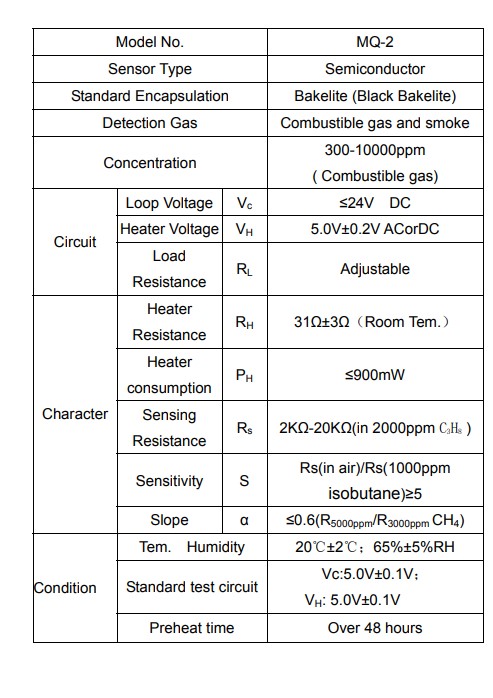
-Good sensitivity to Combustible gas in wide range

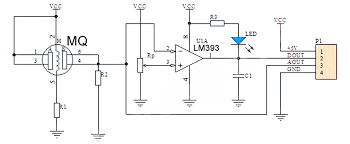
-High sensitivity to LPG, Propane and Hydrogen

-Long life and low cost

-Simple drive circuit

**Technical Data for gas sensor**





**Sensor Circuit Diagram**

The above figure 4.3.2. is basic test circuit of the sensor. The sensor needs to be put 2 voltage, heater voltage（VH） and test voltage（VC). VH used to supply certified working temperature to the sensor, while VC used to detect voltage (VRL) on load resistance （RL）whom is in series with sensor. The sensor has light polarity, Vc need DC power. VC and VH could use same power circuit with precondition to assure performance of sensor. In order to make the sensor with better performance, suitable RL value is needed: Power of Sensitivity body (Ps):

Ps = Vc ^ 2 × Rs / (Rs+RL) ^ 2

Resistance of sensor (Rs): Rs = (Vc/VRL-1) × RL

Working principle

When tin dioxide (semiconductor particles) is heated in air at high temperature, oxygen is adsorbed on the surface. In clean air, donor electrons in tin dioxide are attracted toward oxygen which is adsorbed on the surface of the sensing material. This prevents electric current flow. In the presence of reducing gases, the surface density of adsorbed oxygen decreases as it reacts with the reducing gases. Electrons are then released into the tin dioxide, allowing current to flow freely through the sensor. Since MQ2 Gas Sensor is not breadboard compatible, we do recommend this handy little breakout board. It’s very easy to use and comes with two different outputs. It not only provides a binary indication of the presence of combustible gases but also an analog representation of their concentration in air. The analog output voltage provided by the sensor changes in proportional to the concentration of smoke/gas. The greater the gas concentration, the higher is the output voltage; while lesser gas concentration results in low output voltage. The following animation illustrates the relationship between gas concentration and output voltage. The analog signal from MQ2 Gas sensor is further fed to LM393 High Precision Comparator (soldered on the bottom of the module), of course to digitize the signal. Along with the comparator is a little potentiometer you can turn to adjust the sensitivity of the sensor. You can use it to adjust the concentration of gas at which the sensor detects it.

Pin names



**Sensor Pin Configuration**

**Gas Sensor Application**

Module version of this sensor can be used without interfacing to any [microcontroller](https://www.elprocus.com/different-microcontrollers-used-in-automobiles/) and is useful when detecting only one particular gas. This can only detect the gas. But if ppm has to be calculated then the sensor should be used without module.

This sensor is also used for Air quality monitoring, Gas leak alarm and for maintaining environmental standards in hospitals. In industries, these are used to detect the leakage of harmful gases.

These sensors are used to detect the presence of gases in the air such as methane, butane, LPG and smoke but they are unable to distinguish between gases. Thus, they cannot tell which gas it is.

**BUZZER:**

The [buzzer](https://www.quisure.com/blog/faq/what-is-a-buzzer) is a sounding device that can convert audio signals into sound signals. It is usually powered by DC voltage. It is widely used in alarms, computers, printers and other electronic products as sound devices. It is mainly divided into piezoelectric buzzer and electromagnetic buzzer, represented by the letter "H" or "HA" in the circuit. According to different designs and uses, the buzzer can emit various sounds such as music, siren, buzzer, alarm, and electric bell.

### **Piezo buzzer**

The piezoelectric buzzer uses the piezoelectric effect of the piezoelectric ceramics and uses the pulse current to drive the vibration of the metal plate to generate sound. Piezoelectric buzzer is mainly composed of multi-resonator, piezoelectric plate, impedance matcher, resonance box, housing, etc. Some of the piezoelectric buzzers are also equipped with light-emitting diodes. The multi-resonator consists of transistors or integrated circuits. When the power supply is switched on (1.5~15V DC operating voltage), the multi-resonator oscillates and outputs 1.5~2.5kHz audio signal. The impedance matcher pushes the piezoelectric plate to generate sound. The piezoelectric plate is made of lead zirconate titanate or lead magnesium niobate piezoelectric ceramic, and silver electrodes are plated on both sides of the ceramic sheet. After being polarized and aged, the silver electrodes are bonded together with brass or stainless-steel sheets.

****

**Buzzer**

There are two types of piezo buzzers - transducers and indicators. Transducers consist of a casing, a piezoceramic element and a terminal. In order to operate a transducer, the user must send a square wave signal to the buzzer. Indicators consist of a casing, a piezoceramic element, a circuit board and a terminal. In order to operate an indicator, the user must send the buzzer a specified dc voltage.

**GSM:**

The Global System for Mobile Communications (GSM) is a standard developed by the [European Telecommunications Standards Institute](https://en.wikipedia.org/wiki/European_Telecommunications_Standards_Institute) (ETSI) to describe the protocols for second-generation ([2G](https://en.wikipedia.org/wiki/2G)) digital [cellular networks](https://en.wikipedia.org/wiki/Cellular_network) used by mobile devices such as mobile phones and tablets. It was first deployed in [Finland](https://en.wikipedia.org/wiki/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](https://en.wikipedia.org/wiki/1G)) analog cellular networks. The GSM standard originally described a digital, circuit-switched network optimized for [full duplex](https://en.wikipedia.org/wiki/Duplex_(telecommunications)#Full_duplex) voice [telephony](https://en.wikipedia.org/wiki/Telephony). This expanded over time to include data communications, first by [circuit-switched transport](https://en.wikipedia.org/wiki/Circuit_Switched_Data), then by [packet](https://en.wikipedia.org/wiki/Network_packet) data transport via [General Packet Radio Service](https://en.wikipedia.org/wiki/General_Packet_Radio_Service) (GPRS), and [Enhanced Data Rates for GSM Evolution](https://en.wikipedia.org/wiki/Enhanced_Data_Rates_for_GSM_Evolution) (EDGE). Subsequently, the [3GPP](https://en.wikipedia.org/wiki/3GPP) developed third-generation ([3G](https://en.wikipedia.org/wiki/3G)) [UMTS](https://en.wikipedia.org/wiki/UMTS) standards, followed by the fourth-generation ([4G](https://en.wikipedia.org/wiki/4G)) LTE Advanced and the fifth-generation [5G](https://en.wikipedia.org/wiki/5G) standards, which do not form part of the ETSI GSM standard.

"GSM" is a [trade mark](https://en.wikipedia.org/wiki/Trade_mark) owned by the [GSM Association](https://en.wikipedia.org/wiki/GSM_Association). It may also refer to the (initially) most common voice codec used, [Full Rate](https://en.wikipedia.org/wiki/Full_Rate). As a result of the network's widespread use across Europe, the acronym "GSM" was briefly used as a generic term for mobile phones in France, the Netherlands and in [Belgium](https://en.wikipedia.org/wiki/Belgium). A great number of people in Belgium still use it to date.

****

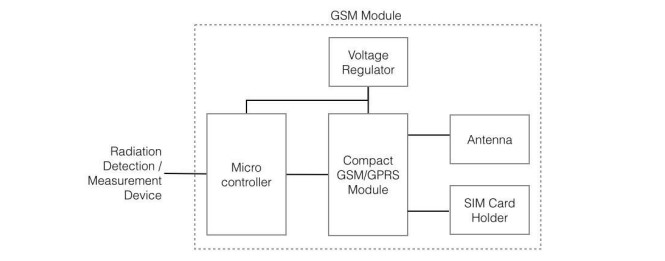
**GSM Module**

**System design**

Hardware and firmware development of the GSM module were done in parallel. Initially, the firmware was developed on Arduino platform; Arduino Uno and GSM Shield for Arduino. The Serial Monitor in Arduino IDE is a very useful tool to echo communication between the microcontroller and GSM shield. This tool enables programmer to monitor and verify firmware operation in GSM shield responds and SMS data processing. Hence, the proof-of-concept prototype is successfully developed on Arduino platform. However, the Arduino GSM library was not utilized; instead, the microcontroller communicates with GSM shield by using AT Command to avoid dependency on the library as well as to maintain the flexibility of the firmware. GSM modems are connected to your server or workstation with a serial port or USB connection. Connecting your GSM modem with a serial cable is a very secure way to use a GSM modem. No extra drivers are necessary and the cable can be fixed tightly to the server and the GSM modem. GSM modems are optimized for SMS and data transmission and often support advanced options like automatic resets and on-board software.

**Hardware Design**

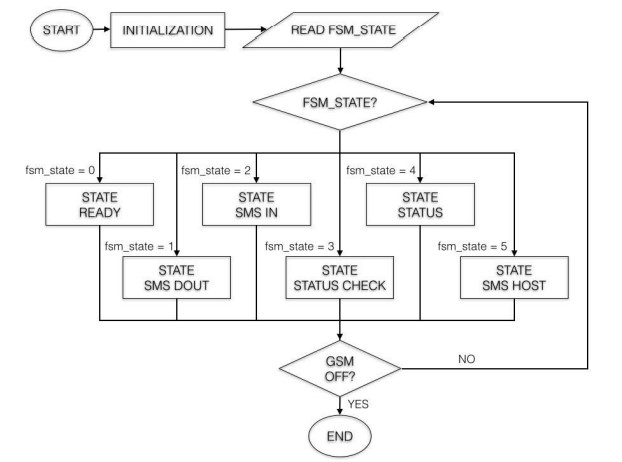
A customized PCB is designed to create a single board that housed all the necessary design blocks as shown in figure 1. The voltage regulator circuit provides voltage supply to all the components on board based on their respective voltage specifications. Microcontroller Atmega328P controls and synchronizes the operation of the module; it controls the operation of GSM module, and handles data and commands from the external device. Telit GL865-DUAL/QUAD V3 module (Telit) is a compact GSM/GPRS module that suits portable and battery operated device. This IC acts as GSM modem that transmits and receives all the SMS for the GSM module.



**Block diagram of GSM PCB Module**

**Firmware Design**

Communication between GSM module and microcontroller is done using AT commands by using standard serial connection. The GSM module is designed to enable wireless communication for radiation monitoring instrument intended for continuous data monitoring and emergency alert. There are three configuration parameters that are essential to complete the task; the Host number, time interval for data transmission, and threshold level for alert SMS. These parameters are stored in EEPROM of microcontroller. User will be able to change and update the configuration parameters via SMS. Firmware of GSM microcontroller is implemented as a finite state machine as shown as the state diagram. The firmware is responsible to handle task related to GSM/SMS communication with the Host server. A GSM module or a GPRS module is a chip or circuit that will be used to establish communication between a mobile device or a computing machine and a GSM or GPRS system. The modem (modulator-demodulator) is a critical part here these modules consist of a GSM module or GPRS modem powered by a [power supply circuit](https://www.electronicsforu.com/electronics-projects/plus-minus-5v-supply-from-9v-battery) and communication interfaces (like RS-232, USB 2.0, and others) for computer.



**GSM microcontroller firmware**

**AT Command and SMS PDU Processing**

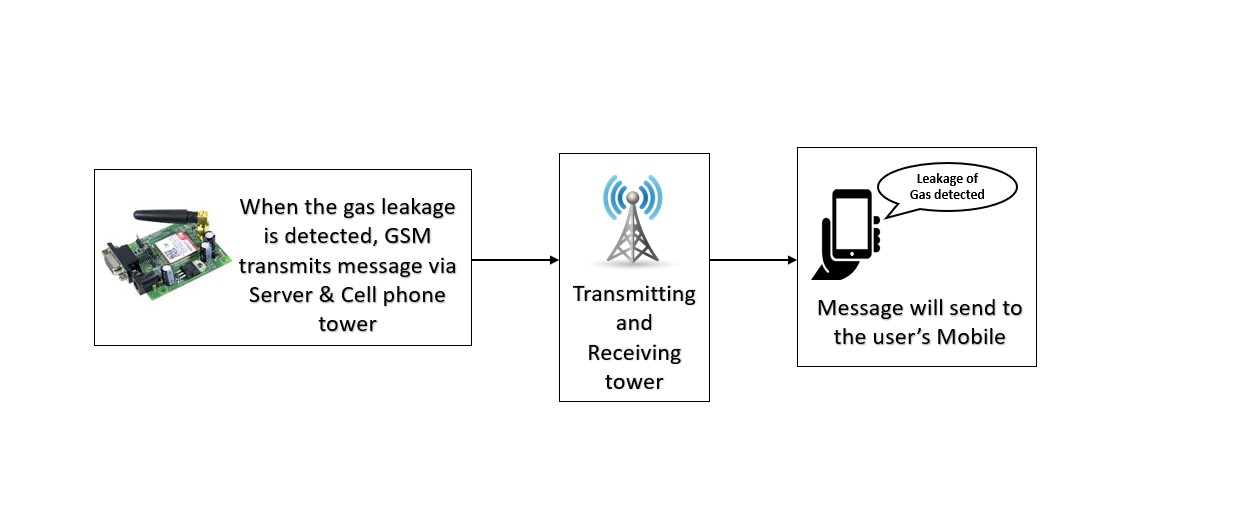
Communication between microcontroller and Telit is done by using AT Command via serial connection. AT commands are list of standard instructions that is used to control a modem. For example, AT+CMGD is a command used to instruct Telit to delete SMS. In order to simplify the SMS processing, AT+CMGF=1 is used to instruct Telit for SMS in text mode. In default state (St\_Ready), microcontroller frequently checks its serial receive buffer in case there is new notification or status from Telit GSM (ReadGSM). All incoming SMS is temporarily copied and stored in the firmware array buffer. It is crucial for the microcontroller to identify and process SMS intended for the system.

In this case, the module is expected to receive three types of SMS:

* SMS from Host or any sender to check the status
* SMS from Host to change configuration parameter (Host number, data time interval, threshold value for alert)
* Other unrelated SMS

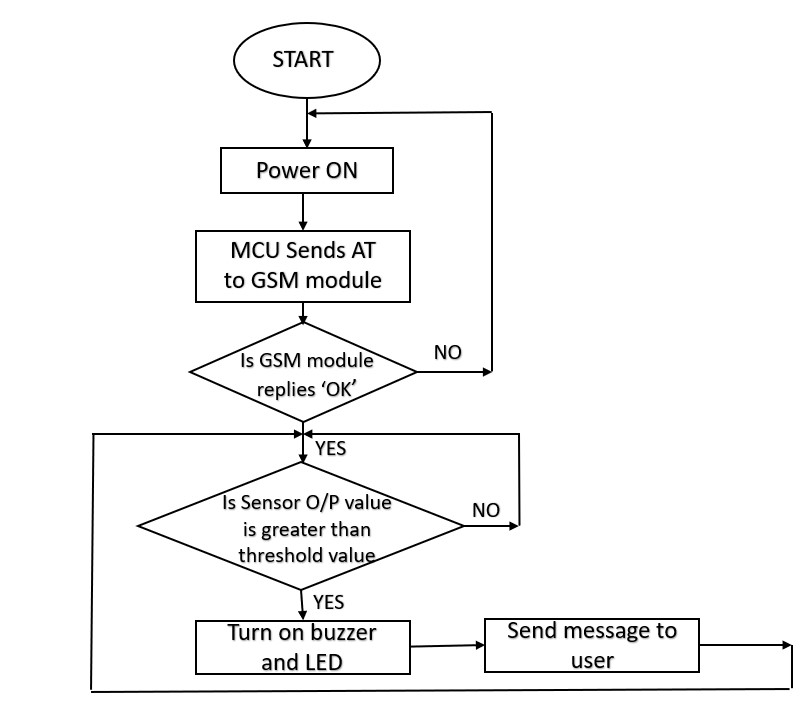
An example of a received SMS is as follows: +CMGL: 5, “REC UNREAD”, “+60123030224”,””,”2016/06/09 11:27:35+32” @HOST SCFG H+60162324045 I5 T3

The first line consists of information on location index of the SMS in the message storage area, status of the SMS message ("received unread", "received read", etcetera), originator phone number, and timestamp at which the SMS message arrived at the SMSC. The second line is the body of SMS that might contain request instruction or configuration parameter from the host. Hence, the SMS body is designed to contain markers or keywords that could be identified by the microcontroller. The keyword ‘@HOST’ is used to identify that the SMS is a host SMS and ‘SCFG’ is a marker for configuration SMS. Subsequently, the microcontroller will be able extract the information from the SMS and execute the next function respectively.



**Working principle**

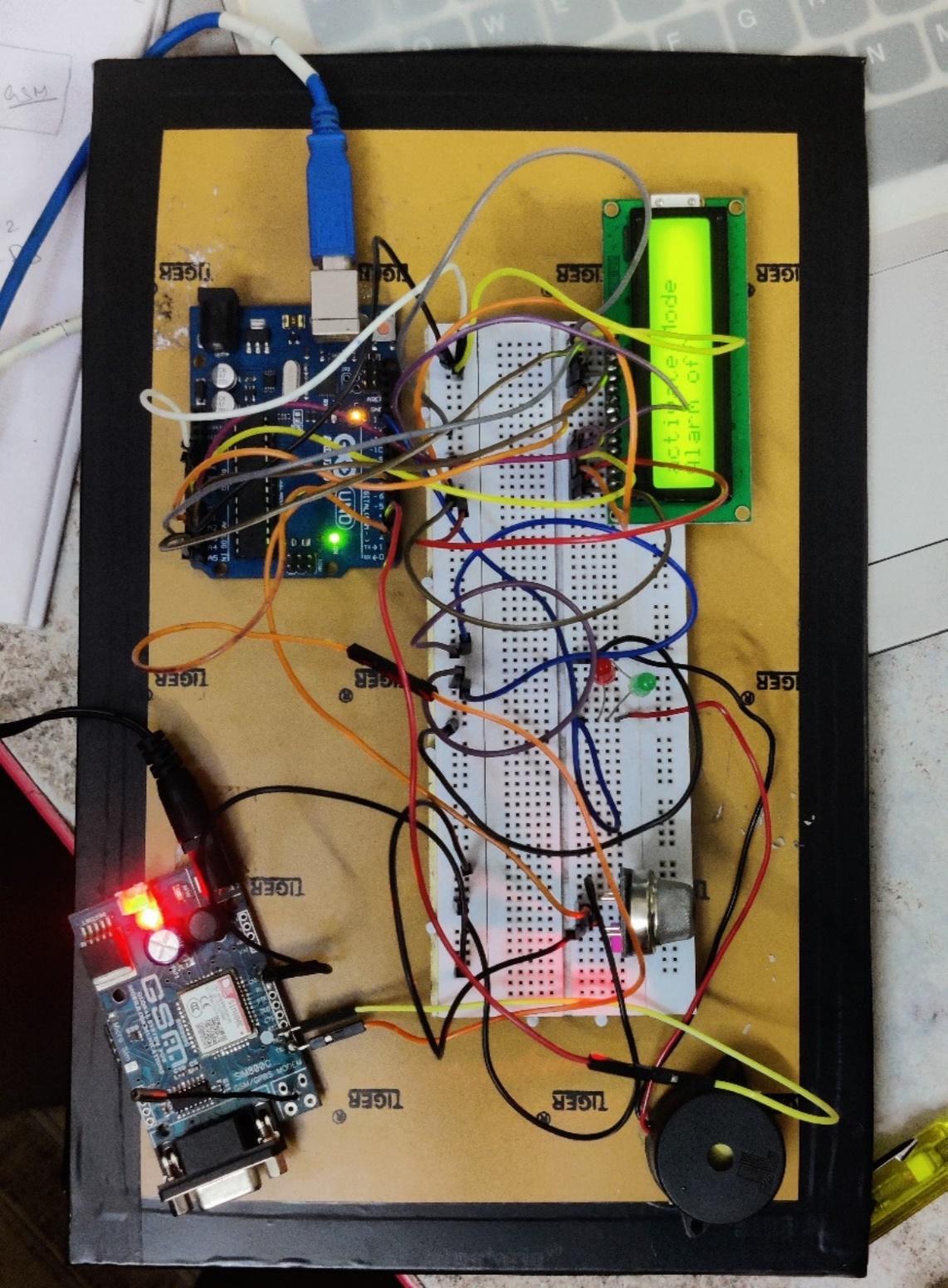
FLOW DIAGRAM



**RESULT AND DISCUSSION**

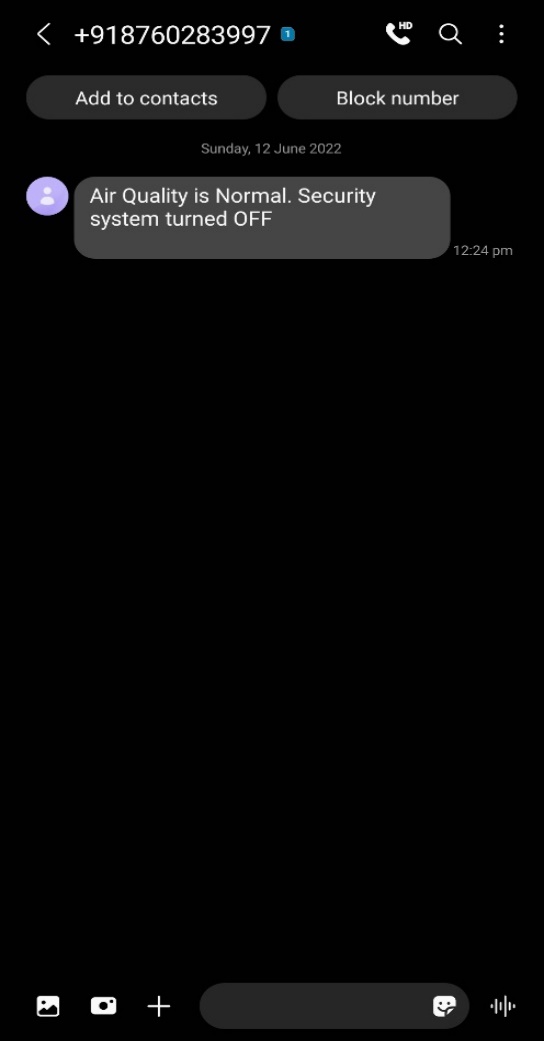
The output of the proposed system is as follows:

The Gas leakage detection system is designed by using following components such as Gas sensor, Arduino UNO, LCD display, Buzzer, GSM (Global System for Mobile Communication) which are connected in the breadboard.



**Gas leakage detector- Prototype**

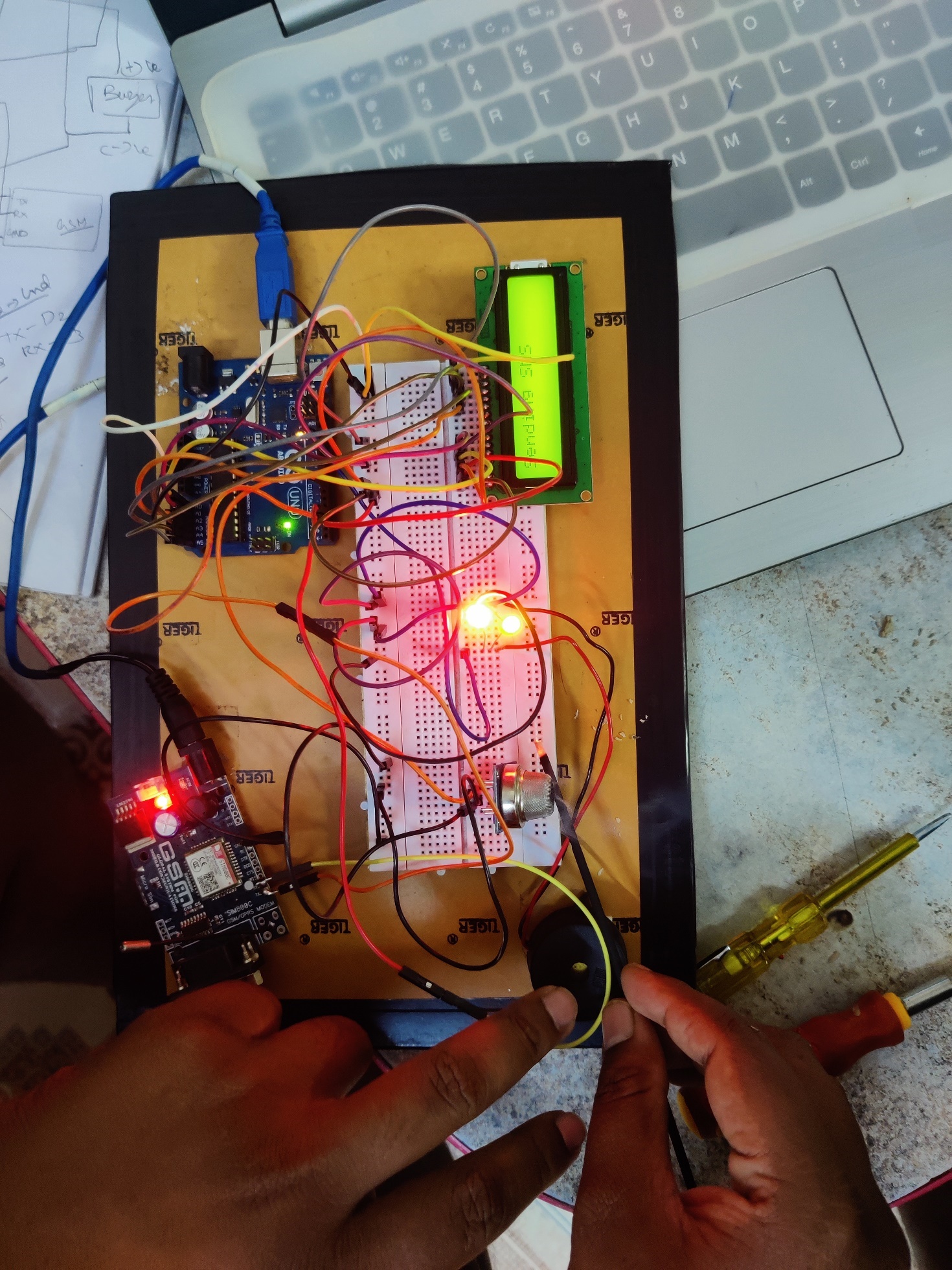
The above figure shows the prototype of gas leakage detector. In which the lcd display module is connected and it will display the status of the system, whether the security system is on or off. GSM (global system for mobile communication) module is connected and helps to communicate with the user through the mobile communication.



**Message sent to user**

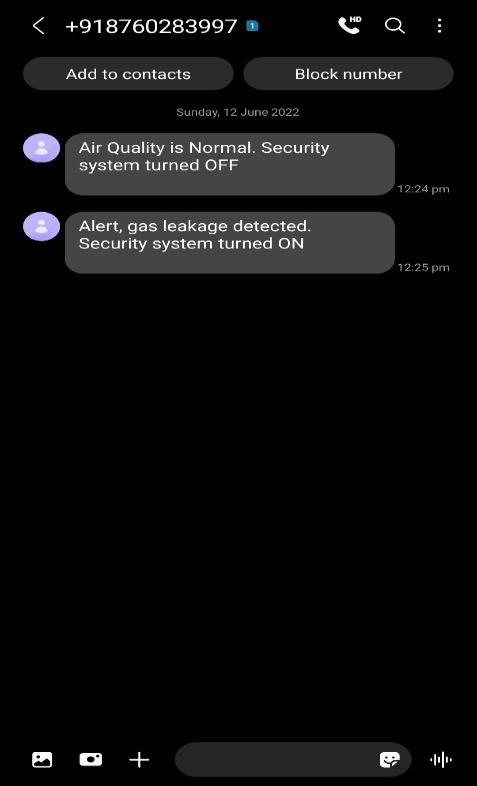
The above figure shows the system status message which is sent to user. Initially the system sends the status to the user with a help of GSM module, which can communicate with user mobile through mobile communication. It sends message to a user mobile as “Air Quality is Normal. Security system turned OFF”.

.



**Alert alarm using Buzzer**

The above figure shows the detection of smoke by a gas sensor. The smoke is detected by a gas sensor connected and it will send an input data to microcontroller and it will process and gives an alarm by using buzzer sound. LED begins to glow. And the sending status in displayed in LCD display connected.



**Alert message sent to user**

The above figure shows the alert message sent to user. The alert message,

“Air Quality is Normal. Security system turned OFF” is sent to user mobile number using GSM module which can communicate with mobile tower.

**CONCLUSION**

The proposed system demonstrates the gas leakage detection in air conditioner. Here the leakage of gas is detected by using a gas sensor. Due to tear and wear of air conditioner there may cause a holes and cracks in the condenser coil. The gas can be leaked from the holes and cracks. This may lead to short circuit in air conditioner and cause fire and emits the carbon monoxide gas which is an toxic, colourless, odourless and tasteless gas. In this kind of leakage, the gas leaked or smoke is detected by gas sensor and led starts to glow and alarm starts to make sound by using buzzer. And warning message is sent to user using GSM module which may transmits the message signal to the nearby mobile towers and then it will receive by a mobile phone.