A project report on

REAL TIME GAS LEAKAGE AND FIRE SENSING SYSTEM FOR INDUSTRIAL SECURITY

Submitted in partial fulfillment for the award of the degree of

Integrated MTECH

by

N S M KARTHIK - 19MIS7023

K V KARTHIK - 19MIS7001

M V S G PARDHU - 19MIS7099

AMARAVATI

SCOPE

July,2022

CERTIFICATE

This is to certify that the Summer Project titled **"REAL TIME GAS LEAKAGE AND FIRE SENSING SYSTEM FOR INDUSTRIAL SECURITY"** that is being Submitted by **M V S G PARDHU – 19MIS7099** is in partial fulfillment of the requirements for the award of **Master of Technology (Integrated 5 Year) software engineering**, is a record of bonafide work done under my guidance. The contents of this Project work, in full or in parts, have neither been taken from any other source nor have been submitted to any other Institute or University for award of any degree or diploma and the same is certified.

Signature of the Guide
(Dr. D SUMATHI)

The thesis is satisfactory

Approved by

PROGRAM CHAIR
M. Tech. SE

DEAN
School Of Computer Science and Engineering

ACKNOWLEDGEMENT

It is my pleasure to express with deep sense of gratitude to **Dr. D Sumathi**, Associate Professor Grade – 1, SCOPE, VIT-AP, for her constant guidance, continual encouragement, understanding; more than all, she taught me patience in my endeavor. My association with her is not confined to academics only, but it is a great opportunity on my part of work with an intellectual and expert in the field of Internet of Things.

I would like to express my gratitude to **Dr. G. Viswanathan**, **Dr. Sekar Viswanathan**, **Sankar Viswanathan**, **G. V. Selvam, Kadhambari**, **S. Viswanathan**, **Dr. S. V. Kota Reddy**, and **Dr. Sudha S V**, School of Computer Science and Engineering, for providing with an environment to work in and for his inspiration during the tenure of the course.

In jubilant mood I express ingeniously my whole-hearted thanks to **Dr. Reeja** Program Chair Master of Technology in Integrated Software Engineering (MTSE), Professor Grade-1, all teaching staff and members working as limbs of our university for their not-self-centered enthusiasm coupled with timely encouragements showered on me with zeal, which prompted the acquirement of the requisite knowledge to finalize my course study successfully. I would like to thank my parents for their support.

It is indeed a pleasure to thank my friends who persuaded and encouraged me to take up and complete this task. At last, but not least, I express my gratitude and appreciation to all those who have helped me directly or indirectly toward the successful completion of this project.

Place: Amaravati
Date: 30/07/22

N S M KARTHIK - 19MIS7023

K V KARTHIK - 19MIS7001

M V S G PARDHU - 19MIS7099

ABSTRACT

Industrial disasters are caused either by human / technical failures. Negligence, incompetence & accidents in managing the prescribed systems leads to heavy losses. Particularly gas leakage and fire accidents can be fatal and leads to material loss. The major cause of explosion, firing or suffocation is majorly based on the physical/chemical properties such as toxicity, density, flammability, etc. In parallel to technological upgradations, industrial disasters are also incremental. The fatality is on high side due to the release of toxic pollutants, gases and also due to fire accidents in recent years.

Estimations of death vary from 3,800 to as many as 16,000, but the government figures an estimation of 15,000 killed over the years due to the exposure to toxic gases. Some harmful gases being heavier than air, do not disperse easily. It may then lead to choking when inhaled and may cause explosion.

Therefore, the working environment should be routinely checked and maintained in order to prevent such disassembly. The system designed is to very useful in the industries as it helps to detect the gas leakage and fire explosions using various sensors and immediately informs to the concerned authorities without human involvement. The data is further stored in cloud for further references.

LIST OF FIGURES

CONTENTS

CHAPTER 1 – INTRODUCTION

Technological and industrial development promotes living standards and provides us with the tools and science to cope with emergencies, but it creates new ones. Due to the extensive usage of chemicals, there are continually new chemical mishaps that result in significant financial losses and harm to people. Between 1985 and 1997, the European Chemical Accident Center recorded more than 300 significant chemical incidents. According to Kalra, Henderson, and Raines, the Bhopal methyl isocyanate accident in 1984, which resulted in 2000–5000 fatalities and tens of thousands of injuries, was the deadliest chemical accident in recent memory (1985). One of the most recent catastrophes in this regard was the explosion of the ammonium nitrate train in Nishabor, Iran, which left 300 people dead and more than 500 injured. Leal, Gustavo, and Santiago all remarked that more research has been done in recent years on mishaps brought on by the emission of combustible liquids and gases, notably vapour cloud explosions.

Significant efforts are being made to increase security among the technologically advanced countries as a result of the militant organizations' increased access to chemical and biological weapons. The use of microcontroller equipment is generally on the rise in the fire and gas detection industry. Gas detectors are used in a wide variety of applications today, including process and quality monitoring, safety, and environmental compliance. Because gas detectors that aid in prevention are more dependable and more affordable than alternate methods. There are many manufacturers who can provide the right equipment to protect persons at risk of hazardous chemicals in their atmosphere, as evidenced by the large variety of poisonous and flammable gas detection systems described in recent publications.

For the protection of persons and property in all kinds of societies and on all kinds of manufacturing platforms, gas monitoring could be a crucial component of municipal safety systems. A City Hazardous Gas Monitoring was created for this project in order to identify hazardous and combustible gas leaks in their early phases.

An intelligent alert system known as a gas leakage detector uses automatic field sounders and automatic phoning to notify inhabitants and previously designated authorities when dangerous gases are accumulating in the city atmosphere at specific locations.

At the moment, modern homes are built and designed to be airtight spaces. Standard home fuel-burning furnaces need a lot of oxygen to finish the fuel combustion process. Carbon monoxide gas is produced when heating fuels like natural gas are burned partially. Colorless, odourless, tasteless, and invisible gas known as carbon monoxide is a very hazardous cumulative toxicant. Haemoglobin exchanges oxygen for carbon monoxide molecules when carbon monoxide is breathed into the body. A person who continues to breathe in carbon monoxide will eventually endure breathing difficulties, sickness, brain damage, and even death as more and more oxygen molecules are replaced.

Installing a carbon monoxide detector in a bedroom, where inhabitants spend a significant amount of time engaging in the particularly sensitive activity of sleeping, is an efficient way to monitor the presence of CO in a residential building. A portable multiple warning system with an alarm and a smoke or gas detector is revealed in To Viewing.

However, the gas detector only offers one kind of auditory alarm to the user to warn of harmful quantities of gases like carbon monoxide. The user is not given any specific gas detection information by this device, nor are any safety instructions given in accordance with the gas concentration detected. The microcontroller-based device offers quick, precise, and ongoing gas monitoring in harsh settings. Protection of residences, workplaces, buildings, and other facilities is crucial.

The project involves developing a system that uses an alarm, a gas sensor, a fire sensor, GSM, and an Arduino ATMEGA 328 microcontroller to warn occupants of dangerous gas concentrations. The gas sensor produces co-relatable sensor signals after detecting the concentration of a certain dangerous gas. The controller automatically sends an SMS to the authorized person's mobile phone because it is operationally connected to the gas sensor and GSM.

The demonstration module is built with a MQ2 Gas sensor, which has a high sensitivity to methane, propane, and butane. This makes it perfect for monitoring natural gas and LPG, as well as a fire sensor for detecting fire accidents and transmitting the information to the fire station. The microcontroller serves as the system's brain or operational hub.

These days, GSM-based project tasks are becoming more and more popular thanks to the special features the GSM network offers. The network was created based on European standards and is known as the GSM (Global System for Mobile Communications).

Many concepts, ranging from simple to complicated systems, are developed using this technology for a variety of applications, but this GSM-based system is extremely distinctive and helpful for both home and commercial applications to control appliances and receive accident information.

1.2 Objective of the project

General Objective

To layout and acquire a project “Advanced Gas Leakage and Fire Detection System”.

Specific Objective

To create and set up an SMS-centered Alert method that sends SMS alert missives to limit mobile number entry inside the Arduino programme, as well as to design and acquire a project that will create a sound alarm during gas outflow and fire explosion and rest the Methane outflow or any other such petroleum-based gaseous substance that are toxic, can be discovered using MQ2s.

1.3 Outline of the report

This report includes comprehensive information on all the project's components. The following parts are used:

- Arduino UNO
- Gas sensor
- IR Sensor
- GSM Module
- Buzzer

Each component is described in distinct chapters with a thorough report:

Chapter 1	contains introduction and Objective of the project.
Chapter 2	contains analysis of existing and proposed systems.
Chapter 3	contains information about Arduino UNO.
Chapter 4	contains information about Gas sensor.
Chapter 5	contains information about IR sensor.
Chapter 6	contains information about Buzzer.
Chapter 7	contains information about GSM module.
Chapter 8	contains block diagram.
Chapter 9	describes about software used.
Chapter 10	contains information about working module & Code.
Chapter 11	describe about the results of all components used in the developed system.
Chapter 12	describes about conclusion and scope for future.
Chapter 13	contains bibliography and appendix.

CHAPTER 2-ANALYSIS

2.1 Existing System

Existing system uses gas and fire detectors to detect gas leakage and fire explosion. The signal is intimidated to the pivot/control room. After the signal is received, the control room depute teams to the specific leak points to control.

2.2 Disadvantage of existing system

Disadvantage of current technology is due to negligence, incompetence and difficulty of implementation of rules and regulations prescribed.

The primary drawback of the current approach is the time-consuming human intervention.

2.3Proposed System

Methane outflow or any other petroleum-based gaseous substance that is harmful can be discovered using MQ2 sensor and fire explosion is recognized using IR sensor in order to layout and acquire project that will perceive gas outflow like Methane leak, Butane leak, and LPG leak.

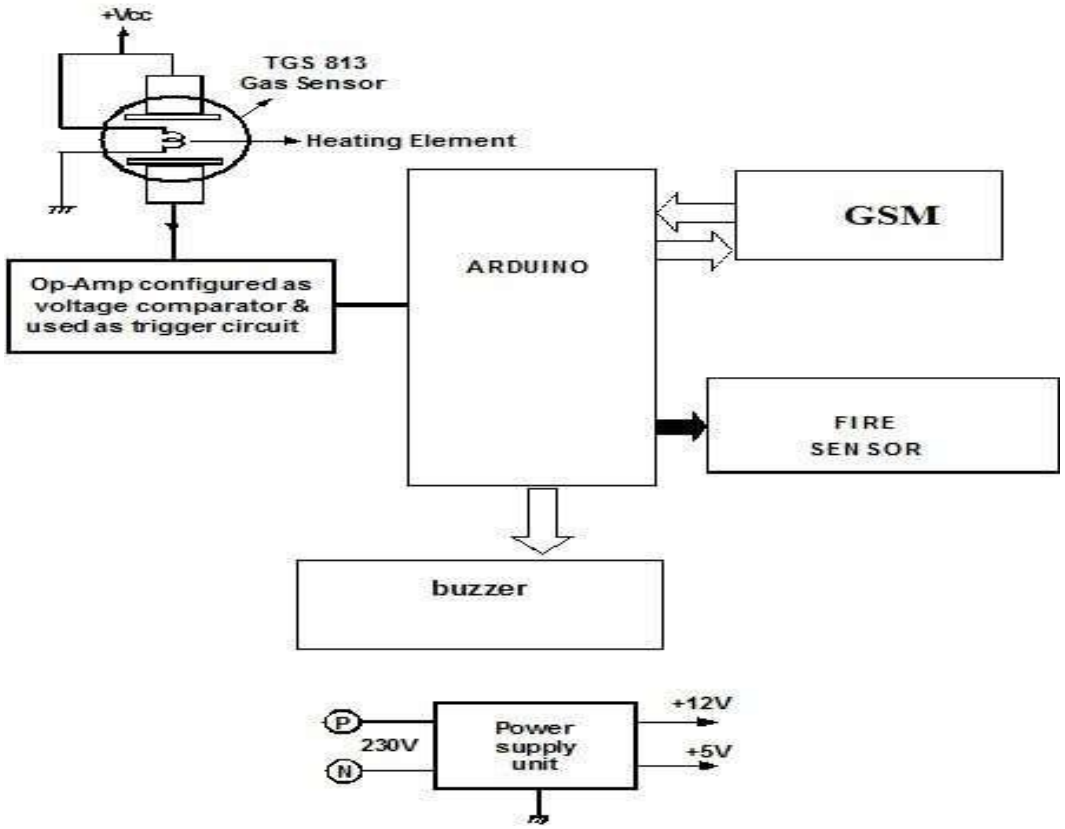
In order to design and acquire a project that will create a sound alarm during gas outflow and fire explosion, an SMS-centered Alert technique is built to send SMS alert missives to restrict cellphone number enter within the Arduino software. The data is kept in Thingspeak cloud for future use.

2.4 Advantages over Existing System

The main advantage of proposed system over existing system is that, the process can be carried without human intervention which in turn reduces human error The process is however less time consuming and the data which is uploaded in the cloud can be referred for further investigations.

CHAPTER 3 - BLOCK DIAGRAM DESCRIPTION

3.1 Block Diagram



3.2 Description

The system consists of a gas sensor, IR sensor, Buzzer, GSM module & an Arduino uno.Software used are- Arduino IDE.

A gas detector is a tool that can identify different gases in a space and can alert users to the presence of potentially dangerous gases. Combustible, toxic (poisonous), and CO2 gases can all be found with the aid of gas detectors. The MQ2 universal gas sensor is utilized here.

Similar to this, IR sensors are used to detect fire. The fire in the area is discovered using an IR LED (Detector). The employment of an IR diode, which emits hot gases with a distinctive spectral pattern in the IR region, is a straightforward and portable method of detecting the presence of fire in any fire accident.

The outputs of both the sensors are connected to the Arduino. When the un-usualities are detected, the signal is sent to the Arduino.

The buzzer is triggered to alert the surroundings and the GSM module is used tosend alert messages to the concerned authorities.

CHAPTER 4-COMPONENTS

4.1 List of Components

- Arduino UNO
- Gas sensor
- IR Sensor
- GSM Module
- Buzzer

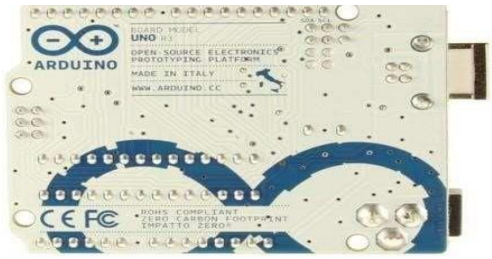
4.2 Arduino Uno

4.2.1 Introduction

A microcontroller board called the Arduino Uno is based on the ATmega328. It contains a 16 MHz ceramic resonator, 6 analogue inputs, 14 digital input/output pins (of which 6 can be used as PWM outputs), a USB port, a power jack, an ICSP header, and a reset button. Everything needed to support the micro-controller is included; all you need to do to get started is connect it to a computer with a USB cable, power it with an AC-to-DC adapter, or use a battery. The FTDI USB-to-serial driver chip is not used by the Uno, which is how it differentiates from all earlier boards. In its place, a USB to serial converter built using the Atmega16U2 (or Atmega8U2 up to version R2) is featured.

Arduino Uno R3 Front

Fig 4.1 Arduino Front



Arduino Uno R3 Back

Fig 4.2 Arduino Back

4.2.2 Pin Configuration

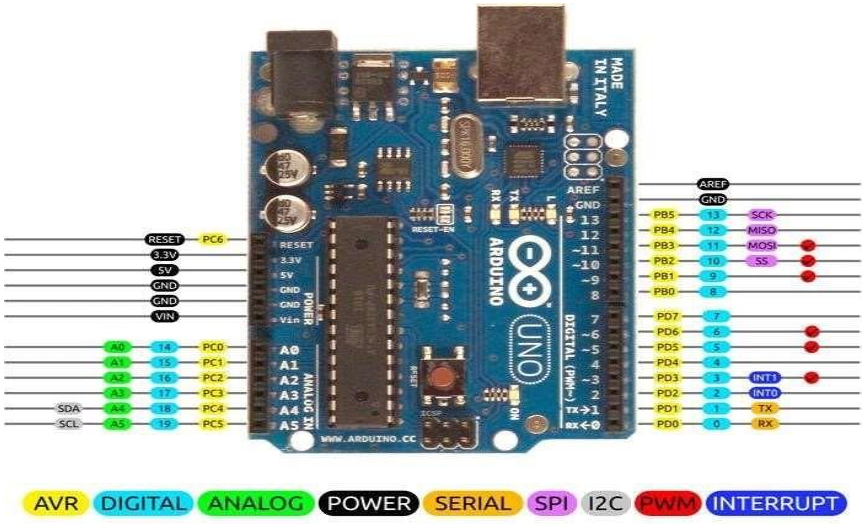


Fig 4.3 Pin Diagram

The board's third revision adds the following additional features: SDA and SCL pins, which are next to the AREF pin and two additional new pins, the IOREF, that enable the shields to adapt to the voltage supplied by the board, are added to the 1.0 pinout. Shields will eventually work with both the Arduino Due, which runs on 3.3V, and the board that uses the AVR, which runs on 5V. The second pin is unconnected and is set aside for a future use.

The name "Uno" denotes the imminent release of Arduino and translates to "one" in Italian.

Moving forward, the reference versions of Arduino will be the Uno and version 1.0.
The Uno is the most recent in a line of USB Arduino boards and the platform's reference design; see below for a comparison with an earlier iteration.

Current models use an ATmega328, however the Arduino reference design can also use an Atmega8, 168, or 328.

Every single one of the three CPUs has the same pin layout.

4.2.3 Features

Micro-controller	**ATmega328**
Operating Voltage	5V
Input Voltage(recommended)	7-12V
Input Voltage(limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40mA
DC Current for3.3V Pin	50mA
Flash Memory	32KB
SRAM	2KB
EEPROM	1KB
Clock Speed	16MHz

4.2.4 Schematic & Reference Design

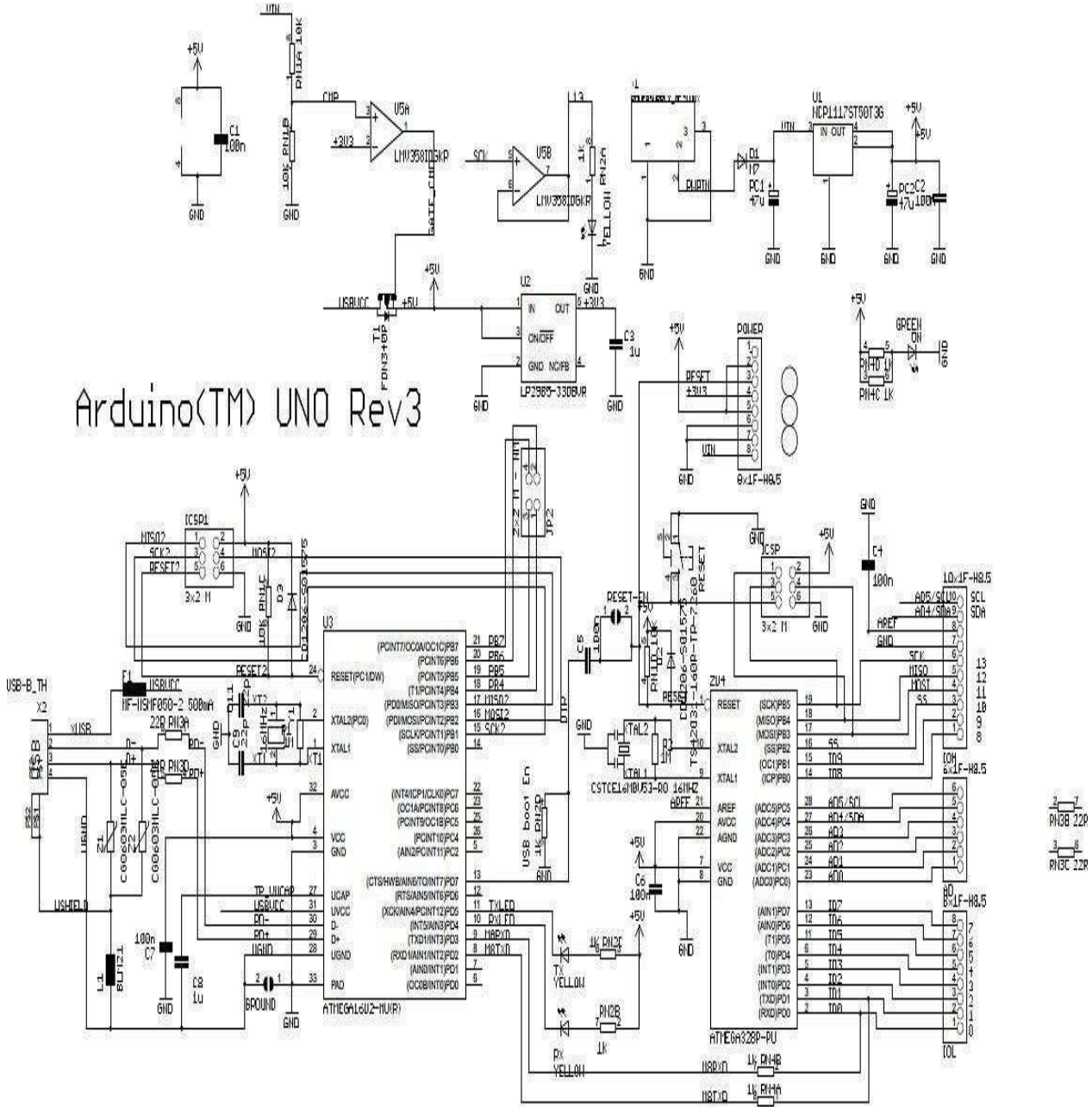


Fig 4.4 Schematic & Reference Diagram

4.2.5 Power

Either an external power source or the USB connection can be used to power the Arduino Uno.

The power source is picked for you automatically.

External (non-USB) power sources include batteries and wall warts that convert AC current to DC. A 2.1mm center-positive plug can be used to connect the adapter by inserting it into the board's power connector. The Gnd and Vin pin headers of the POWER connection can accept battery leads.

A 6-to-20-volt external supply can be used to power the board. The 5V pin, however, may deliver less than five volts if supplied with less than seven volts, and the board may become unstable. The voltage regulator runs the danger of overheating and breaking the board if more than 12V is applied. The suggested voltage range is 7 to 12 volts. These are the power pins:

- **VIN-** Input voltage of the Arduino board when utilizing an external power supply (as opposed to 5 volts from the USB connection or other regulated power source). This pin can be used to feed voltage to or access voltage that has been supplied by the power jack.
- **5V-** This pin provides a regulated 5V output from the board's regulator. The board's VIN pin, the USB connector (5V), or the DC power jack (7–12V) can all be used to supply power to it (7-12V). Bypassing the regulator by applying power to the 5V or 3.3V pins can harm your board. We do not suggest it.
- **3V3-** an internal regulator-generated 3.3-volt supply. A 50 mA maximum current consumption is allowed.
- **GND-** Ground pins.
- **IOREF-** This Arduino board pin serves as the voltage reference for the microcontroller. A correctly built shield can read the IOREF pin voltage and then select the appropriate power supply or enable voltage translators on the outputs to work with 5V or 3.3V.

4.2.6 Memory

The Atmega328 has 32 KB of memory (with 0.5 KB used for the bootloader). Additionally, it features 1 KB of EEPROM and 2 KB of SRAM (which can be read and written with the EEPROM library).

4.2.7 Input and Output

The Uno's 14 digital pins can all be used as inputs or outputs by utilizing the pinMode(), digitalWrite(), and digitalRead() routines. They use 5 volts to work. Each pin includes a 20–50 k internal pull-up resistor that is unconnected by default and has a maximum current capacity of 40 mA.

Additionally, several pins perform specific tasks:

- **Serial: 0 (RX) and 1 (TX).**
  used to transmit and receive TTL serial data (RX and TX). The ATmega8U2 USB-to-TEL Serial chip's equivalent pins are connected to these pins.

- **External Interrupts: 2 and 3.**
  These pins can be set up to initiate an interrupt in response to low values, rising or falling edges, or value changes. Details can be found in the attachInterrupt() function.

- **PWM: 3, 5, 6, 9, 10, and 11.**
  The analogWrite() method outputs an 8-bit PWM signal.

- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).**
  The SPI library supports SPI communication on these pins.

- **LED: 13.**
  Digital pin 13 is wired to a built-in LED When the pin has a HIGH value, the LED is on, and when the pin has a LOW value, the LED is off.

- The Uno features six analogue inputs with the designations A0 through A5, each of which offers a resolution of 10 bits (i.e., 1024 different values). Using the AREF pin and the analogReference() function, the upper limit of their measurement range can be changed from the default range of ground to 5 volts.

Additionally, certain pins are designed to do specific tasks:

- **TWI: A4 or SDA pin and A5 or SCL pin:**
  Utilize the Wire library to support TWI communication.

Other pins on the board include the following:

- **AREF:**
  The analogue inputs' reference voltage. when coupled with analogReference ().

- **Reset:**
  The microcontroller can be reset by bringing this line to LOW. Usually applied to shields that obstruct the board's reset button.

4.2.8 Communication

A computer, another Arduino, or other micro-controllers can all be communicated with using the Arduino Uno's many communications features. On digital pins 0 (RX) and 1, the ATmega328 provides UART TTL (5V) serial connection (TX).

This serial communication is routed through USB by an ATmega16U2 on the board, which is seen by computer software as a virtual com port.

There is no need for an external driver because the '16U2 firmware uses the built-in USB COM drivers. On Windows, however, a.inf file is necessary. Using the serial monitor included in the Arduino software, simple textual data can be sent to and received from the Arduino board.

The RX and TX LEDs on the board will blink when data is transmitted using the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and1).

Any digital pin on the Uno can be used for serial communication thanks to the serial library. Communication over I2C (TWI) and SPI is also supported by the ATmega328. To make using the I2C bus simpler, the Arduino software comes with the Wire library; check the documentation for more information. Use the SPI library to carry out SPI communication.

4.2.9 Programming

The Arduino software can be used to program the Arduino Uno.

You may upload new code to the Arduino Uno's ATmega328 without using an external hardware programmer thanks to the bootloader that is already pre-burned on it. It communicates via the original STK500 protocol (reference, C header files).

See these instructions for more information on how to use the ICSP (In-Circuit Serial Programming) header to program the microcontroller instead of using the bootloader.

The firmware source code for the ATmega16U2 (or 8U2 in the rev1 and rev2 boards) is accessible. A DFU bootloader is pre-installed on the ATmega16U2/8U2, and it can be triggered by:

1. Resetting the8U2 after connecting the solder jumper on Rev1 boards, which is located close to the Italy map.
2. The 8U2/16U2 HWB line is pulled to ground by a resistor on Rev2 or later boards, making it simpler to enter DFU mode.

The next step is to load a fresh firmware using Atmel's FLIP software for Windows or the DFU programmer for Mac OS X and Linux. Alternately, you might utilize an outside programmer and the ISP header (overwriting the DFU bootloader). For more details, see this user-contributed instructional.

4.2.10 Automatic (Software)Reset

The Arduino Uno is made in a way that enables it to be reset by software running on a connected computer, as opposed to requiring a physical press of the reset button prior to an upload. A 100 nano farad capacitor is used to connect one of the ATmega8U2/16U2's hardware flow control lines (DTR) to the ATmega328's reset line. The reset line lowers for a long enough period of time when this line is asserted (taken low) to reset the chip. You can upload code using the upload button in the Arduino environment by using this feature of the Arduino program.

As a result, the bootloader's timeout can be shortened because the upload's beginning and the lowering of DTR can be perfectly timed.

This configuration has additional effects. The Uno resets every time a connection is made to it from software when it is connected to a computer running either Mac OS X or Linux (via USB). The bootloader is active on the Uno for the next half-second or so. The initial few bytes of data transmitted to the board after a connection is established will be intercepted, despite the fact that it is configured to disregard invalid data (i.e., anything other than an upload of new code).

Make sure that the software with which it talks waits a second after opening the connection and before providing any configuration or other data when a sketch running on the board initially gets it. To turn off the auto-reset, a trace on the Uno can be severed. To re-enable the trace, solder the pads on either side of it together. It has the label "RESET-EN." By adding a 110-ohm resistor from 5V to the reset line, you should be able to stop the auto-reset as well.

4.2.11 USB Over current Protection

The Arduino Uno contains a re-settable poly fuse that guards against shorts and overcurrent at the USB ports on your computer. The fuse offers an additional layer of safety even though the majority of computers have their own internal safeguards. The fuse will immediately cut off the connection if more than 500 mA is applied to the USB port until the short or overload is resolved.

4.2.12 Physical Characteristics

The USB connector and power jack protrude beyond the maximum length and breadth of the Uno PCB, which are 2.7 and 2.1 inches, respectively. The board can be fastened to a surface or container with the help of four screw holes. Digital pins 7 and 8 are separated by 160 mil (0.16"), which is not an even multiple of the other pins' distance of 100 mil.

The following are some of the uses for Arduino Uno.

- Prototyping for Do-It-Yourself projects uses the Arduino Uno.
- When creating code-based control projects
- creation of an automation system
    - Creating fundamental circuit designs.

Thus, the Arduino Uno data sheet is the main topic here. Finally, we can infer from the information provided above that this is an 8-bit ATmega328P micro-controller. It has many parts, including a voltage regulator, crystal oscillator, and serial communication for the microcontroller. In addition to a USB port, this board has analogue I/P pins 6, digital I/O pins 14, a power barrel jack, a reset button, and an ICSP header.

4.3 Gas Sensor

4.3.1 Introduction

To prevent or minimize mishaps involving explosions or poisoning, it is crucial to have accurate and quick toxic gas detection, alerting, and monitoring capabilities. There are toxic gases everywhere. These include carbon monoxide, hydrogen sulphide, chlorine, bromine, hydrogen chloride, hydrogen fluoride, nitric oxide, nitrogen dioxide, sulphur dioxide, ammonia, hydrogen cyanide, phosgene, benzene, formaldehyde, methyl bromide, arsine, phosphine, boranes, silane, and germane. To identify these gases in the past, particular color-changing reagents were used in difficult and time-consuming colorimetric or more sophisticated chromatographic methods.

But in the past ten to twenty years, technology has advanced quickly, and additional sensors have been created for the quick, accurate detection of a variety of dangerous chemicals.

Fig 4.5 Gas Sensor

Fig 4.6 MQ2 sensor pins

The method of finding potentially dangerous gas leaks using a variety of sensors is known as gas leak detection. When a harmful chemical is present, these sensors often use an audible warning to warn people. Infrared point sensors, ultrasonic sensors, electrochemical sensors, and semiconductor sensors are some of the common types of sensors utilized today.

Industrial plants, refineries, wastewater treatment facilities, automobiles, and areas surrounding the home all have these sensors. They are also employed for a broad variety of applications.

4.3.2 Semiconductor Detectors

When a gas comes into touch with a semiconductor sensor, a chemical reaction occurs, allowing the sensor to detect gases. The most prevalent substance used in semiconductor sensors is tin dioxide, and when it comes into contact with the monitored gas, the electrical resistance in the sensor decreases. Tin dioxide generally has a resistance of 50 k in air, however this resistance can decrease to only 3.5 k in the presence of 1% methane.

The gas concentration is determined using this variation in resistance. To find dangerous gases like carbon monoxide as well as hydrogen, oxygen, and alcohol, semiconductor sensors are frequently utilized. Carbon monoxide sensors are one of the most typical applications for semiconductor sensors. In breathalyzers, they are also utilized. Semiconductor sensors operate in a narrower range than infrared point or ultrasonic detectors because they need to come into touch with the gas in order to detect it.

4.3.3 MQ2 SENSOR:

A versatile sensor with good sensitivity properties for a variety of gases is MQ2. The intended operating voltage for this device is a regulated 5V supply. Methane, propane, and butane detection is the gas sensor's best-suited use, which makes it a superior sensor for gas leak detectors. The operational amplifier (opamp), which is setup as a voltage comparator, is triggered by the sensor output.

MQ-2 Combustible Gas Semiconductor Sensor The MQ-2 gas sensor's sensitive component is SnO2, which has a reduced conductivity in clean air. The conductivity of the sensor increases along with the concentration of the target flammable gas when it is present. Please use a straightforward electrical circuit to translate changes in conductivity into a signal that corresponds to gas concentration. The MQ-2 gas sensor is affordable, versatile, and very sensitive to LPG, Propane, and Hydrogen as well as to Methane and other flammable gases.

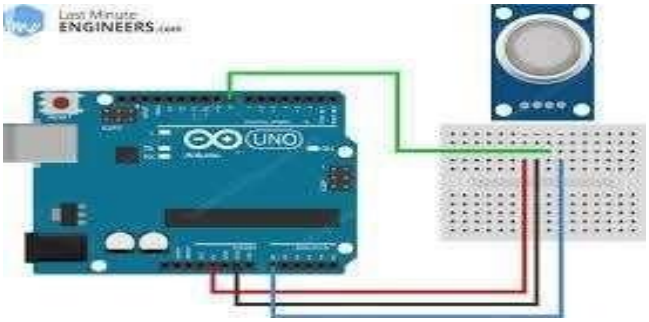


Fig 4.7 Interfacing MQ2 sensor with Arduino

4.3.4 Character Configuration

- Excellent sensitivity to flammable gas across a broad range
- Extreme sensitivity to hydrogen, propane, and LPG
- Low cost and long life
- Basic drive circuit

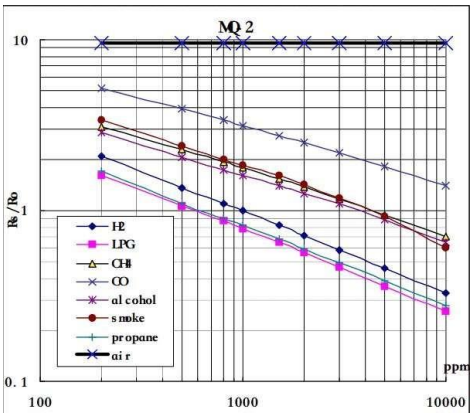


Fig 4.8 Sensitivity Characteristics

4.3.5 Application

- A gas leak detector for the home
- Gas detector for industrial combustibles
- A mobile gas detector

4.4 IR Flame Sensor

4.4.1 Introduction

An electrical device that monitors and detects infrared radiation in its environment is called an infrared (IR) sensor. William Herchel, an astronomer, made the unintentional discovery of infrared radiation in 1800. He saw that the temperature was highest just beyond the red light as he measured the temperatures of each colour of light (separated by a prism). The wavelength of infrared radiation is longer than that of visible light, making it invisible to the human eye (though it is still on the same electromagnetic spectrum). Infrared radiation is produced by everything that emits heat (i.e., everything with a temperature higher than about five degrees Kelvin).

Infrared sensors come in active and passive varieties. Infrared radiation is both produced and detected by active infrared sensors. A light emitting diode (LED) and a receiver are the two components of an active IR sensor. The receiver detects the infrared light from the LED that reflects off an object as it gets close to the sensor. Obstacle detection systems commonly use active IR sensors as proximity sensors (such as in robots).

Fig 4.9 IR Flame Sensor

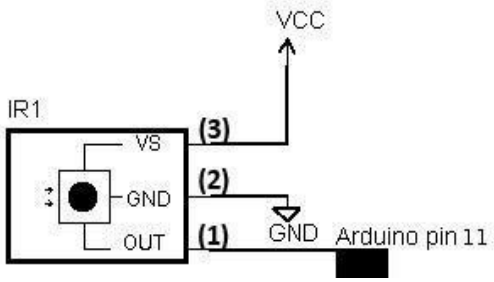


Fig 4.10 IR Flame sensor Pin Configuration

4.4.2 IR LED For Detecting Fire

The fire in the area is discovered using an IR LED (Detector). The employment of an IR diode, which emits hot gases with a distinctive spectral pattern in the IR region, is a straightforward and portable method of detecting the presence of fire in any fire accident.

The fire detection module employs an IR sensor and a transistor that serves as an amplifier. The device, which weighs around 5 grammes, is simple to mount where it is needed. It produces a lot of output when a fire is detected. Based on this output, a suitable action can be done. An internal LED onboard the device provides a visible indicator of output. It has an average response time of 1 sec and can operate independently or in conjunction with a number of networked safety systems to form a reliable fire monitoring system.

IR detectors are appropriate for locations where combustion sources may cause strong, smokey fires. They may function up to 60 meters away from the fire sources. IR detectors can be used both indoors and outdoors, unlike other detectors that can only be used indoors. In the environment where they are mounted, IR detectors are not affected by radiation from the sun, welding, or other hot items.

They are integrated in numerous high-tech detectors to stop IR detectors from setting off false alerts. The icing on the cake is that IR detectors are less expensive and more dependable than other detectors on the market.

4.4.3 Advantages of infrared flame detectors

The IR flame is appropriate for a chilly, harsh atmosphere. It can function effectively in snowy or icy circumstances thanks to a unique thermal optical window.

They are programmable for all situations, uses, and specifications and are perhaps the most robust flame detector now on the market. Changing the device's configuration parameter is a simple way to accomplish this. Highway addressable remote transducers can be used to carry out numerous monitoring duties, change these parameters, and perform maintenance.

You won't need to purchase as many infrared detectors as you might because of their extensive reach.

They are fairly pricey to buy and maintain but very dependable. Infrared detectors are typically made to function independently while being connected to an automatic fire suppression system or an alert.

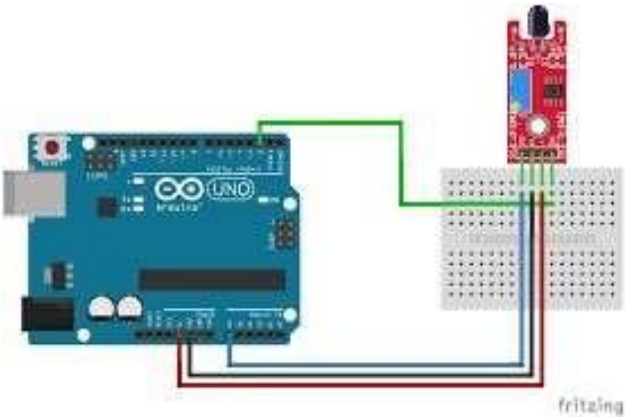


Fig 4.11 IR Flame sensor interfacing with Arduino

4.5 Buzzer

4.5.1 Introduction

A mechanical, electromechanical, or piezoelectric audio signalling device is a buzzer or beeper (piezo for short). Buzzers and beepers are frequently used as alarm clocks, timers, and to validate human input such a mouse click or keyboard.

Fig 4.12 Buzzer

Pin Number	Pin Name	*Description*
1	**Positive**	***Identified by (+) symbol or longer terminal lead. Can be powered by 6V DC***
2	**Negative**	***Identified by short terminal lead. Typicallyconnected to the ground of the circuit***

Fig 4.13 Buzzer Pin Configuration

4.5.2 Features and Specifications

- Rated Voltage: 6V DC
- Operating Voltage: 4-8V DC
- Rated current: <30mA
- Sound Type: Continuous Beep
- Resonant Frequency: ~2300 Hz
- Small and neat sealed package
- Breadboard friendly

4.5.3 How to use a Buzzer

An inexpensive yet effective part to include sound characteristics in our project or system is a buzzer. Since it has a 2-pin construction that is relatively small and compact, it can be utilized on PCBs and breadboards with ease, making it a common component in most electronic applications.

Buzzers come in two different categories and are widely available. The buzzer that is being demonstrated here is a straightforward one that, when powered, emits a continuous Beeeeeeppp sound; the readymade buzzer, on the other hand, is more obtrusive and produces a beep. Beep. Beep. Due to an inbuilt oscillating circuit, it produces sound. However, the one displayed here is the most popular because it can be altered with the aid of additional circuits to conveniently suit in our application.

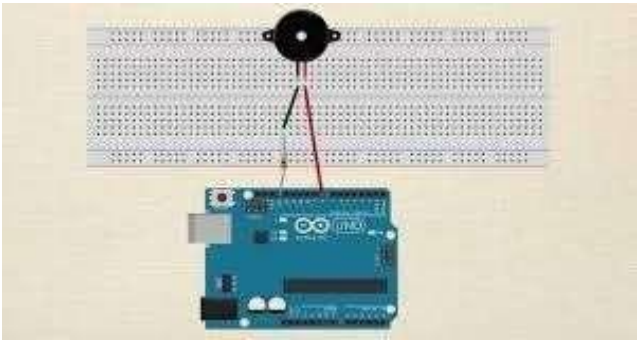


Fig 4.14 Interfacing Buzzer with Arduino

4.5.4 Applications

1. Alarming Circuits, where anything must alarm the user
2. Communications devices
3. Automobile devices
4. Portable devices, given their small size

4.6 GSM Module

4.6.1 Introduction

The Global System for Mobile Communication is also known as GSM. A technology developed in 1985 by the French company Group Special Mobile. Although this communication method was first created for private talks, it is now used for a wide variety of purposes. Since wireless communications is the only technology without a range constraint, any controlled or monitored equipment can be used anywhere in the world.

Mobile phone service is provided by cellular radio through the use of a network of cell sites spread out across a large area. A radio transceiver and base station controller are found in a cell site, which controls, sends, and receives traffic from mobiles in its vicinity to a cell phone switch.

Additionally, it uses a tower and its antennas to connect to a mobile communication switching office, a distant cellular switch (MTSO). This MTSO routes calls from landline phones to wireless subscribers, switches call between cells as subscribers' mobile devices cross cell borders, and verifies subscribers' wireless identities before they can make calls.

GSM calls can either be voice or data-based. Half-rate, full-rate, and enhanced full-rate audio codes are used for voice calls. The cell phone can function as a 9600-bps modem for receiving data calls. It employs time division multiple access and digital technology for transmission. GSM technology is constantly developing and has advanced significantly over the last ten years. In the coming years, it will go through an even more drastic change.

Fig 4.15 GSM Module

The frequency bands designated for GSM services in various nations are GSM-400, GSM-800, GSM-900, GSM-1800, and GSM-R. The two frequency bands that are most often used around the world are GSM-900 and GSM-1800.

The down link frequency range for the GSM-900 is 935-960 MHz, and the up link frequency range is 895-915 MHz. A 200KHz separation separates the 124 pairs of simplex channels that make up this frequency band. A specified set of simplex channels is given to a certain network provider.

GSM uses a digital air interface as its interface type. Before transmission, the analogue voice signals are transformed to digital signals. The GSM RF carrier can manage up to 8 MS subscribers at once. 270 Kbps is the transmission speed.

The GSM modem resembles a mobile phone. It has a tray to house the SIM card and a serial port for use with a serial communicating device. It provides its own instructions for doing things like sending, receiving, and deleting messages. These commands are transmitted through the microcontroller.

The foundation of the GSM mobile telecommunications service is a network of interconnected radio cells that completely enclose the service area and enable subscriber operation wherever inside it.

Radiophones had only one transmitter that could cover the entire service area before the invention of the cellular idea. Instead of using a single, huge transmitter to cover a broad region, cellular telephony makes use of numerous, smaller ones. The main issue is how to manage the case where someone using a phone in one cell walks outside of that cell's coverage area.

The service area was so broad because the radiophone service had no way to recover dropped calls. In cellular telephony, the issue is resolved by transferring the call to the following cell. Although this process is entirely automatic and doesn't need any additional user involvement, it is a difficult technical function that needs a lot of computing power to get a speedy response.

The Mobile Station, Base Station Subsystem, and Network Subsystem make up the bulk of a GSM system's functional design. Each subsystem is made up of functional units that interact with one another via various interfaces according to predetermined protocols.

The base station subsystem manages the radio link with the mobile station, which is carried by the subscriber.

The network subsystem manages mobile services, such as authentication, and switches calls between mobile and other fixed or mobile network users. The Mobile Services Switching Center is the key component of this subsystem.

4.6.2 Technical details

Mobile devices connect to the GSM cellular network by looking for nearby cells since it is one. GSM networks work in a variety of frequency bands (separated into GSM frequency ranges for 2G and UMTS frequency bands for 3G).

The majority of 2G GSM networks use the 900 MHz or 1800 MHz bands for operation.

Because the 900 and 1800 MHz frequency channels were already taken, certain American nations (including Canada and the United States) use the 850 MHz and 1900 MHz bands instead.

The bulk of 3G GSM networks in Europe operate in the 2100 MHz frequency band. The harder to find 400 and 450 MHz frequency bands are now available when these frequencies were previously reserved for first-generation systems in some countries.

Fig 4.16 GSM Specifications

GSM-900 has 124 RF channels (channel numbers 1 through 124) spaced at 200 kHz and uses 890-915 MHz for information transfer from the mobile station to the base station (uplink) and 935-960 MHz for information transfer in the other direction (downlink). 45 MHz of duplex spacing is employed.

The GSM-900 band has been expanded in several nations to encompass a wider frequency range. This "extended GSM," or E-GSM, adds 50 channels (channels 975 to 1023 and 0) to the original GSM-900 band by using 880-915 MHz (uplink) and 925-960 MHz (downlink) frequencies. To enable eight full-rate or sixteen half-rate speech channels per radio frequency channel, time division multiplexing is required. A TDMA frame is made up of eight radio timeslots, giving rise to eight burst periods. In the same timeslot, half rate channels use different frames.

The frame time is 4.615 milliseconds, and the channel data rate for all 8 channels is 270.833 kbit/s. In GSM850/900 and GSM1800/1900, the handset's transmission power is restricted to a maximum of 2 watts and 1 watt, respectively.

To fit 3.1 kHz audio into 5.6–13 kbit/s, GSM has employed a variety of voice codes. Initially, two codes termed Half Rate (5.6 kbit/s) and Full Rate (13 kbit/s), which were designated for different types of data channels, were utilized.

These employed a linear predictive coding-based technique (LPC). In addition to being effective with bitrates, these codes also made it simpler to distinguish the more crucial audio components, enabling the air interface layer to give them more priority and better protection.

To further enhance GSM, the Enhanced Full Rate (EFR) codec, a 12.2 kbit/s coding that makes use of a full rate channel, was implemented in 1997.

Last but not least, with the advent of UMTS, EFR was refactored into AMR-Narrowband, a variable-rate codec that, when used on full rate channels, offers excellent quality and interference resistance, and, when used on half-rate channels in favorable radio conditions, offers less robust but still respectable quality.

Macro, micro, pico, femto, and umbrella cells are the five main cell sizes that can be found in a GSM network. Each cell has a different coverage area depending on the implementation environment. Macro cells can be thought of as areas where the base station antenna is mounted on a pole or a structure higher than the typical roof line. Micro cells, which are frequently utilized in metropolitan settings, are cells with antenna heights below the average roof top level.

Picocells are tiny cells with a few dozen meters of covering diameter that are primarily utilized inside. Femtocells, which link to the service provider's network through a broadband internet connection, are cells intended for usage in home or small commercial settings. In order to fill in coverage gaps and cover shaded areas of smaller cells, umbrella cells are used.

The horizontal radius of a cell can range from a few hundred meters to several tens of kilometer's depending on antenna height, antenna gain, and propagation circumstances. The maximum practical distance supported by the GSM standard is 35 kilometers (22 mi).

There are numerous ways to apply the idea of an extended cell, in which the radius of the cell may double or even increase depending on the antenna system, the terrain's features, and the timing advance.

An indoor picocell base station or an interior repeater with scattered indoor antennas fed through power splitters may be utilized to provide the radio signals from an outdoor antenna to the separate indoor distributed antenna system. These are generally used when a large amount of call capacity is required indoors, such as in malls or airports. This is not a need, though, as adjacent cell radio frequencies can potentially penetrate buildings to offer inside coverage.

Gaussian minimum-shift keying (GMSK), a type of continuous phase frequency shift keying, is the modulation used in GSM. The signal that will be modulated onto the carrier in GMSK is smoothed with a Gaussian low-pass filter before being delivered to a frequency modulator, greatly decreasing interference to surrounding channels (adjacent channel interference).

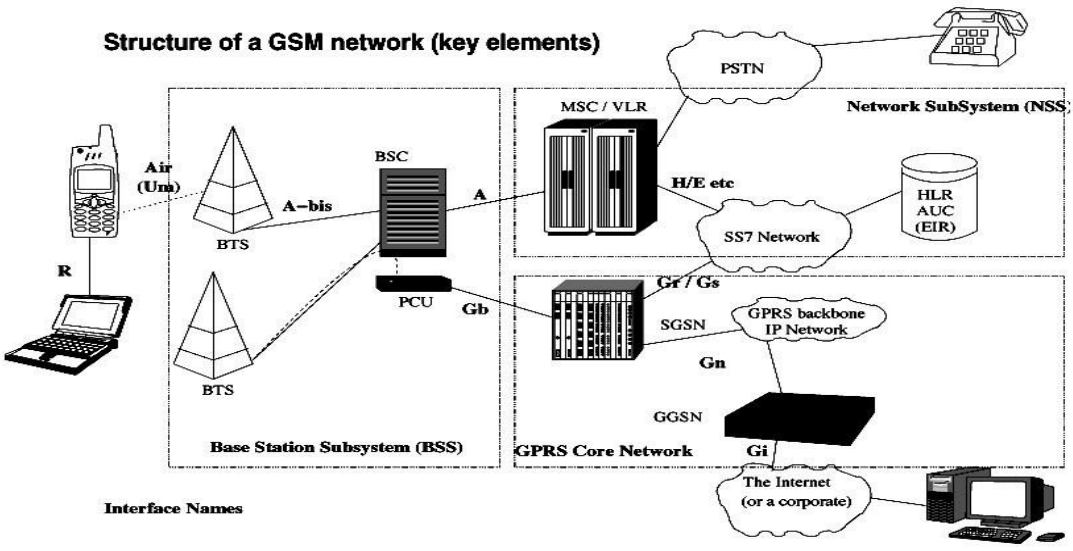


Fig 4.17 Structure of GSM network

4.6.3 Applications of GSM Modem

The most well-known mobile platform worldwide is GSM. GSM technology is used by mobile phones with SIM cards to connect you with your loved ones, friends, and coworkers. The following are benefits of GSM networks over standard landline telephony systems:

1. Mobility
2. Easy availability
3. High uptime

The majority of GSM calls are made to loved ones, friends, and coworkers for professional purposes. We use the communication features of landline telephones for Internet, e-mail, data connectivity, remote monitoring, computer-to-computer communication, and security systems. In a similar manner, we are able to use GSM technology and gain from its advantages.

A wireless modem that operates with a GSM wireless network is known as a GSM modem. Similar like a dial-up modem, a wireless modem operates. The fundamental distinction between both is that a wireless modem uses radio waves to send and receive data, whereas a dial-up modem uses a fixed telephone connection.

A GSM modem might be a PC Card/PCMCIA Card or an external device. Typically, a serial or USB cable is used to connect an external GSM modem to a computer. A laptop computer can use a GSM modem that is in the form of a PC Card or PCMCIA Card. It needs to be put into one of the laptop's PC Card or PCMCIA Card slots. A SIM card from a wireless carrier is required for a GSM modem in order to function, just as a GSM mobile phone.

AT instructions are used by computers to operate modems. A similar set of common standard AT commands is supported by dial-up and GSM modems alike. A GSM modem functions similarly to a dial-up modem. In addition to the standard AT commands, GSM modems provide a wider range of AT commands. The GSM specifications define these enhanced AT commands.

With the enhanced AT commands, we are able to:

- Reading, writing and deleting SMS messages.
- Sending SMS messages.
- Monitoring the signal strength.
- Monitoring the charging status and charge level of the battery.
- Reading, writing and searching phone book entries.

Only roughly six to ten SMS messages may be handled by a GSM modem per minute, which is a relatively low rate.

4.6.4 GSM Features

The Subscriber Identity Module is one of the outstanding features (SIM). The subscriber's identification number, a list of the networks and countries in which they are eligible for service, privacy keys, etc. are all stored on the SIM, which serves as a memory device. To activate service from any GSM phone, you need a four-digit personal identification number, or SIM. SIMs are available as plug-in modules that are portable and detachable or as smart cards that can be placed into GSM phones.

The GSM system's second feature is its on-the-air privacy protection. Digital data is encrypted to ensure privacy using a unique secret cryptographic key that is only known to the cellular carrier and that changes over time.

4.6.5 GSM Interfaces

The different interfaces used in GSM listed as follows:

1. **GSM radio air interface:** This is where MS and BTSs interface.
2. **Abis interface:** The Abis interface is the one that joins a BTS to a BSC. This is in charge of transporting traffic and maintenance data.
3. **An interface:** This is where a BSC and an MSC interface.

4.6.6 GSM Channels

There are two types of GSM logical channels:

1. **Traffic Channels:** Digitally encoded user speech or data is transmitted via these channels.
2. **Control Channels:** Through these channels, directives for signalling and synchronization between BS and MS are sent.

4.6.7 GSM Services

The GSM services in different spheres are listed as follows:

1. Data services include packet switched traffic and computer to computer communication.
2. services provided over the phone, including faxing. The GSM standard also supports videotex and teletex.
3. Teleservices offered by GSM comprise traffic that started on mobile devices and conventional mobile telephony.
4. Additional features include caller line identification, call waiting, and SMS services.

CHAPTER 5-CIRCUIT EXPLNATION

5.1 Circuit Diagram

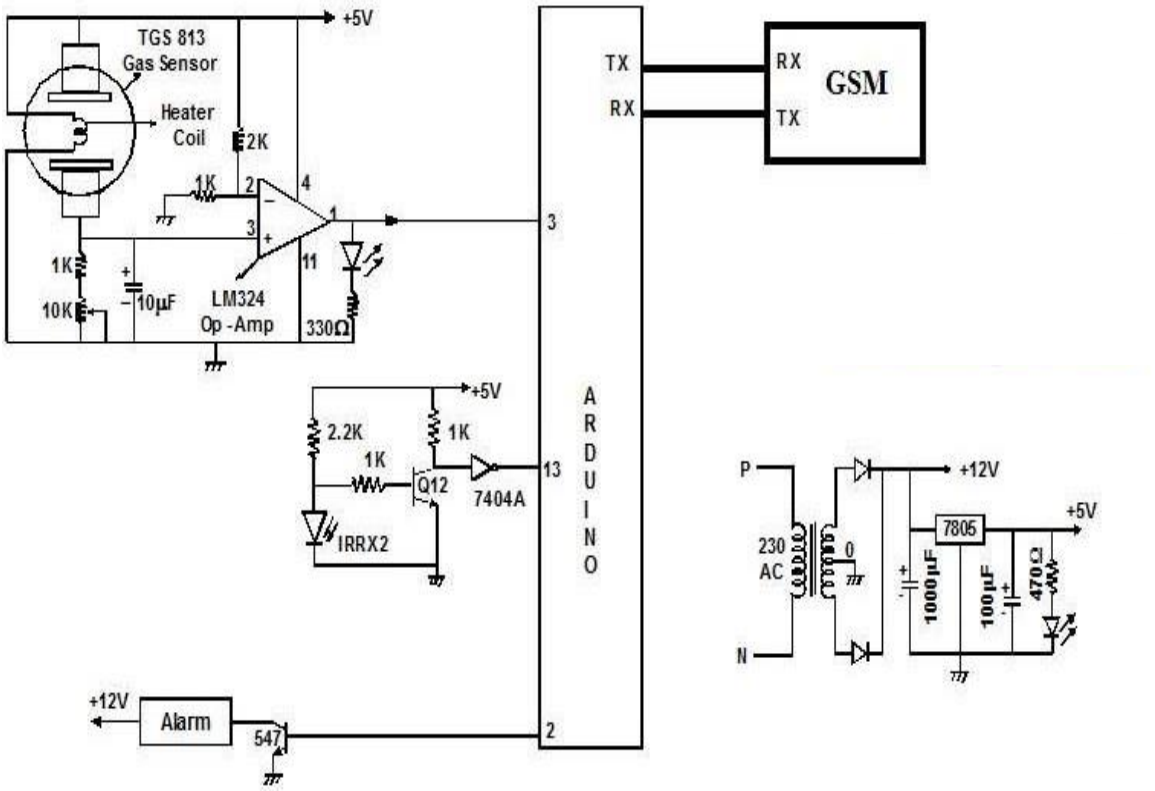


Fig 5.1 Circuit Diagram

5.2 Working

The project's ultimate goal is to create a finished Gas leakage and detection system. With the expansion of the internet everywhere, the adoption of the Internet of Things can further improve security. The item is made out of an outer wooden box-shaped case that serves as storage for the Arduino controller, MQ2 sensor, IR sensor, GSM Module, and a Buzzer.

The Arduino Uno, which serves as the control's brain, is coupled to the gas and fire sensors. MQ2, a versatile gas sensor, is utilized here as a gas sensor. The usage of an optical sensor and IR detector is similar to how fire is detected.

The operational amplifier (op-amp), which is set up as a voltage comparator, is triggered by the output of the gas sensor. When a sensor detects any harmful gases, it generates a potential that is greater than the reference voltage and as a result, the comparator output will become high. This comparison is done with the help of a reference voltage generated at one input of the comparator.

Similar to this, an IR sensor's photodiode (the IR receiver) on the sensor module will pick up a tiny bit of infrared light when a fire burns. The voltage from the sensor varies along with the variation in light intensity level on the sensor. The Arduino uno controller detects the fire status with the increase in voltage from the sensor. Afterward, check for a change in voltage across the IR receiver using an Op-Amp.

The controller receives this high signal from the gas sensor or flame sensor, and upon receiving it, promptly communicates this information to the relevant authorities for an immediate rescue operation via GSM, i.e., SMS to the number specified in the program.

A transformer is a sort of electrical device that moves electrical power from one electrical circuit to another without changing frequency. Transformers effectively change the voltage of AC electricity.

The voltage is regulated using a voltage regulator IC. Most regulators include some level of automatic overcurrent (also known as "overload protection") and thermal (also known as "thermal protection") protection. The product is affordable and adaptable to different environments. This process moves very swiftly and potentially stop additional gas leaks and fire blasts.

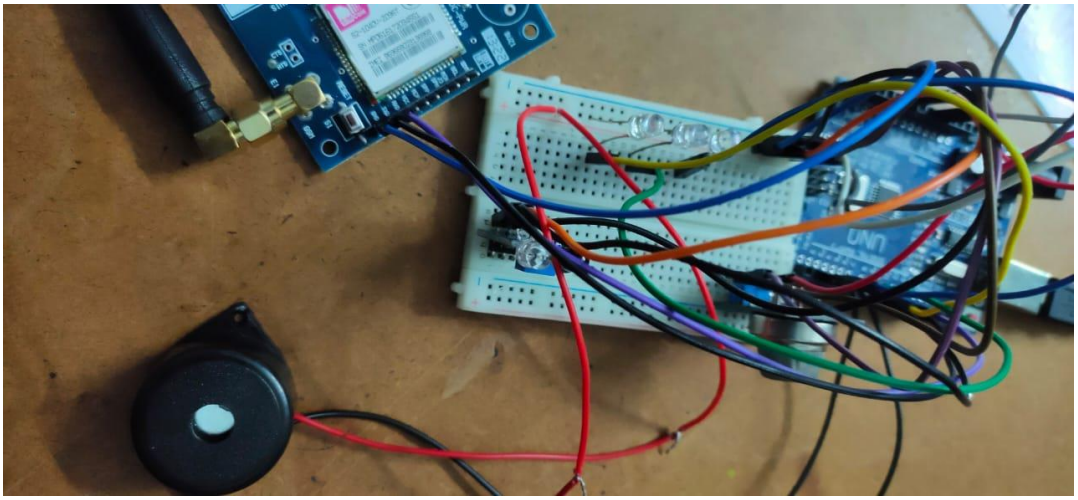


Fig 5.2 Working Module

CHAPTER 6-SOFTWARE

6.1 ARDUINO IDE

6.1.1 Introduction

For Windows, macOS, and Linux, the Arduino Integrated Development Environment (IDE) is a cross-platform program that uses C and C++ functions. It is used to create and upload applications to boards that are compatible with Arduino as well as other vendor development boards with the aid of third-party cores. The source code for the IDE is licensed under the GNU General Public License, version. The C and C++ programming languages are supported by the Arduino IDE using special code organization principles.

A software library from the Wiring project, which offers numerous standard input and output operations, is provided by the Arduino IDE. For the sketch to start and the main program loop, user-written code only needs two fundamental functions, which are combined with a program stub main() to create an executable cyclic executive program using the GNU toolchain, which is also distributed with the IDE. The executable code is transformed via the Arduino IDE's use of avrdude into a text file with hexadecimal encoding, which is then loaded into the Arduino board by a loader program in the firmware.

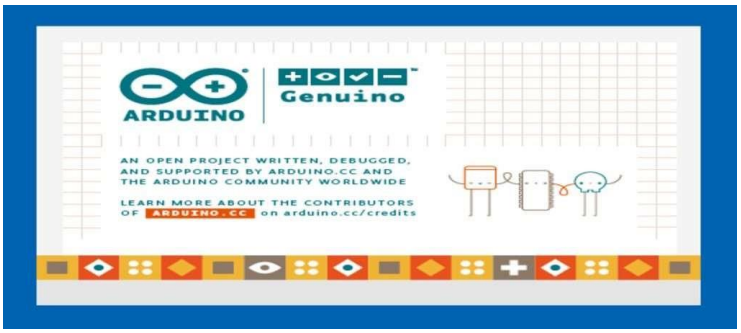


Fig 6.1 Arduino IDE

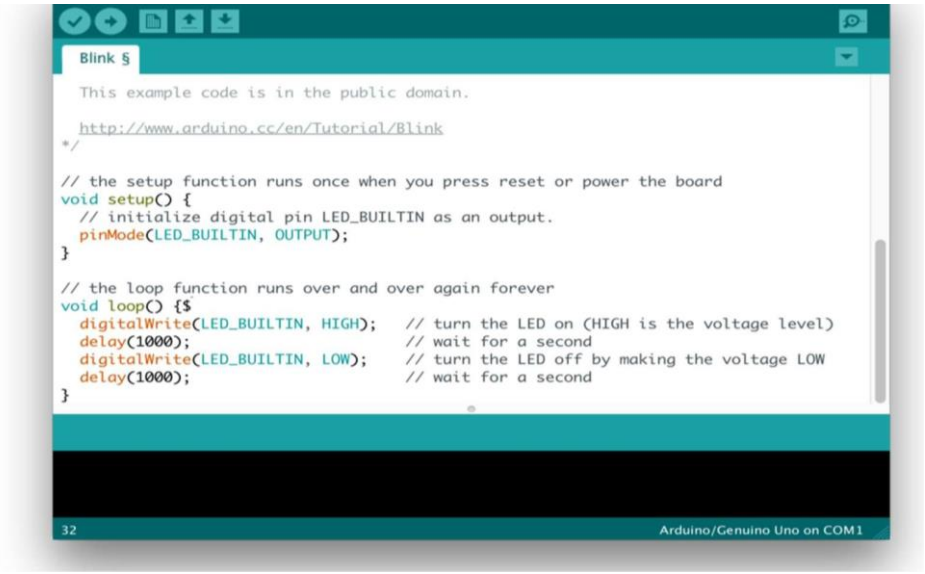


Fig 6.2 Arduino Example

6.1.2 Setting up Arduino IDE software:

Step 1: Download and Install the IDE

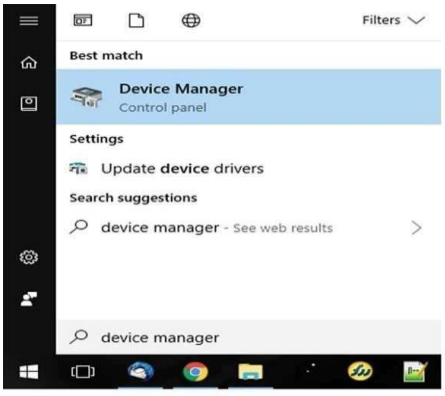
From the official Arduino website, you can download the IDE. The Arduino board is compatible with the majority of computers that have a USB port since it employs a USB to serial converter to interact with the host computer. You will obviously need the IDE initially. Fortunately, the Arduino creators have made the IDE available in a variety of operating system versions, including Windows, Mac, and Linux. If you do not have Windows 10, make sure to get the correct version of the IDE because we will be using it in this tutorial.

Fig 6.3 Arduino 1.8.5

Install the IDE after downloading it, making careful to enable the majority (if not all) of the settings, INCLUDING the drivers.

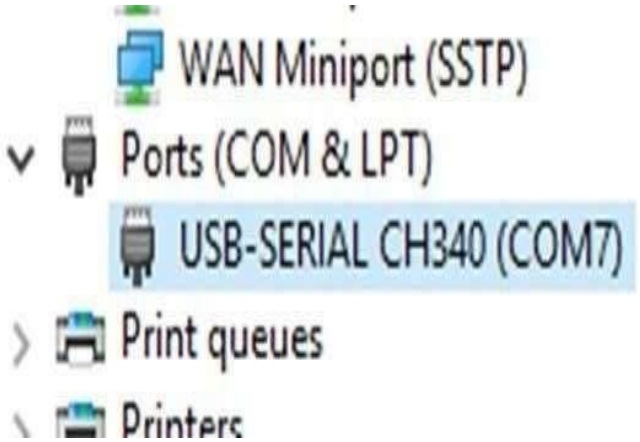
Step 2: Get the Arduino COM Port Number

The Arduino Uno board must then be connected to the PC. A USB B connection is used for this. We no longer need to supply power to the Arduino because of the great world of USB, which offers 5V up to 2A. The operating system ought to identify the board as a standard COM port once the Arduino is connected (for example, my Arduino Uno uses a CH340G, which is an RS-232 serial to USB converter). Once it is identified, we must ascertain the port number that was given to it. Entering "device manager" into Windows Search and choosing Device Manager when it appears is the simplest way to do this. In Windows 10, where to find the device management option



Look for a device in the "Ports (COM & LPT)" section of the Device Manager window; more often than not, the Arduino will be the only item there. The Arduino appears in my Device Manager as COM7 (I know this because CH340 is in the device name).

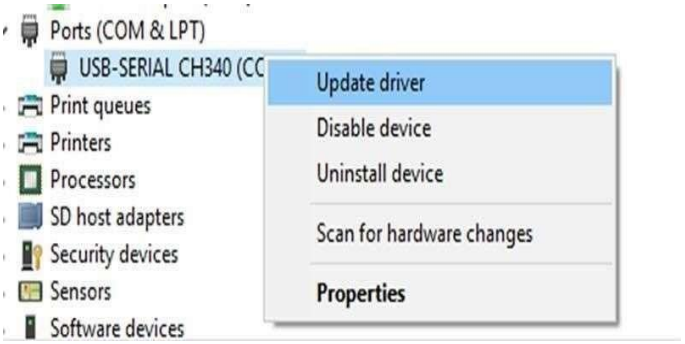
In my situation, the Arduino is a CH340, and it appears on COM7 (port 7).



Be forewarned—the Arduino may not always be instantly identified. Should your Arduino not be detected. then uninstall the driver, take out the Arduino, put it back in, look for the unidentified device, click "Update driver" in the context menu, and then select "Search automatically."

This ought to solve 99 out of 100 issues.

Update the driver if the Arduino is not recognized.



Click "Search automatically" in the window that displays.

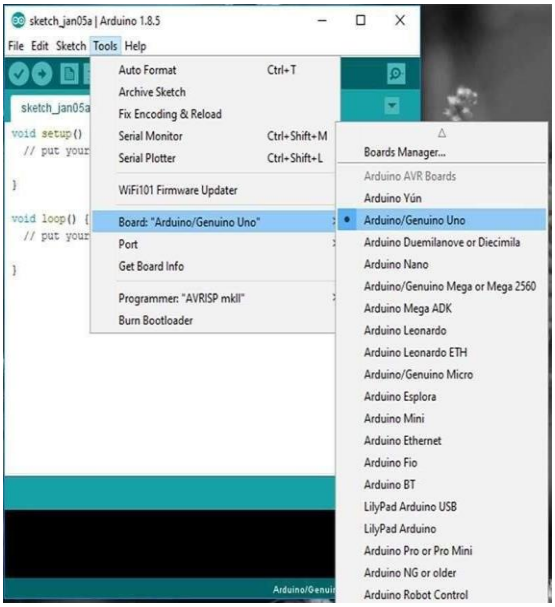
With COM ports, Windows may occasionally be a big hassle since it can mysteriously change their numbers between connections. In other words, your Arduino might be on port 7 (as seen here) one day, but Windows might change it to a different port number the next. According to what I understand, this occurs when you attach more COM ports to your machine (which I do frequently).

So, if your Arduino isn't showing up on the port you typically use, simply open Device Manager, check the port it's truly on, and update your driver if necessary.

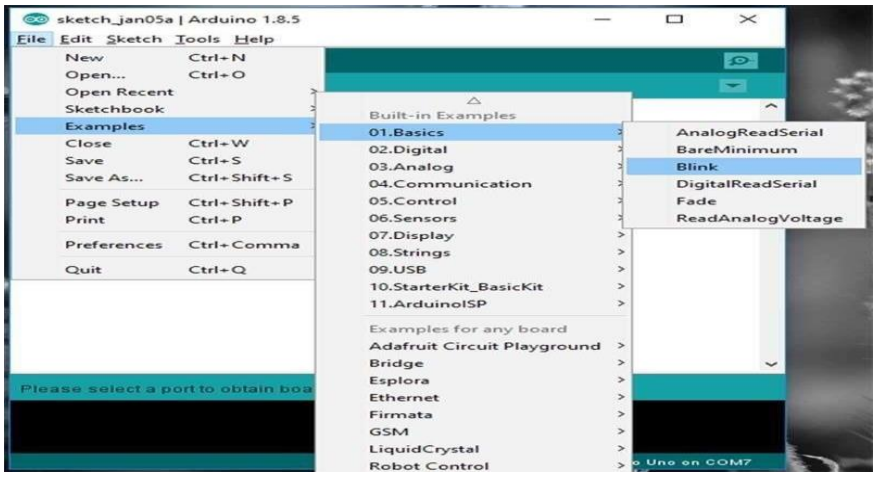
Step 3: **Configure the IDE**

It's time to load the Arduino IDE and set it up to use the same device and port now that we know what COM port the Arduino is on. The IDE should first be loaded. Navigate to Tools > Board > Arduino Uno once it has loaded. You must choose the correct board if, however, you are using a different board (one other than the Arduino Uno)!

Identify the board you are using for the IDE.



The IDE needs to know which COM port the Arduino is connected to next. Go to Tools > Port > COM7 to accomplish this. Of course, use a different port if your Arduino is on that one.



Step 4: Loading a Basic Example

For the purpose of simplicity, we'll load an example project that the Arduino IDE comes with. As a result of this example, the on-board LED will continuously flicker for one second. Click File > Examples > 01. Basics > Blink to load this example.

Load the blink example.

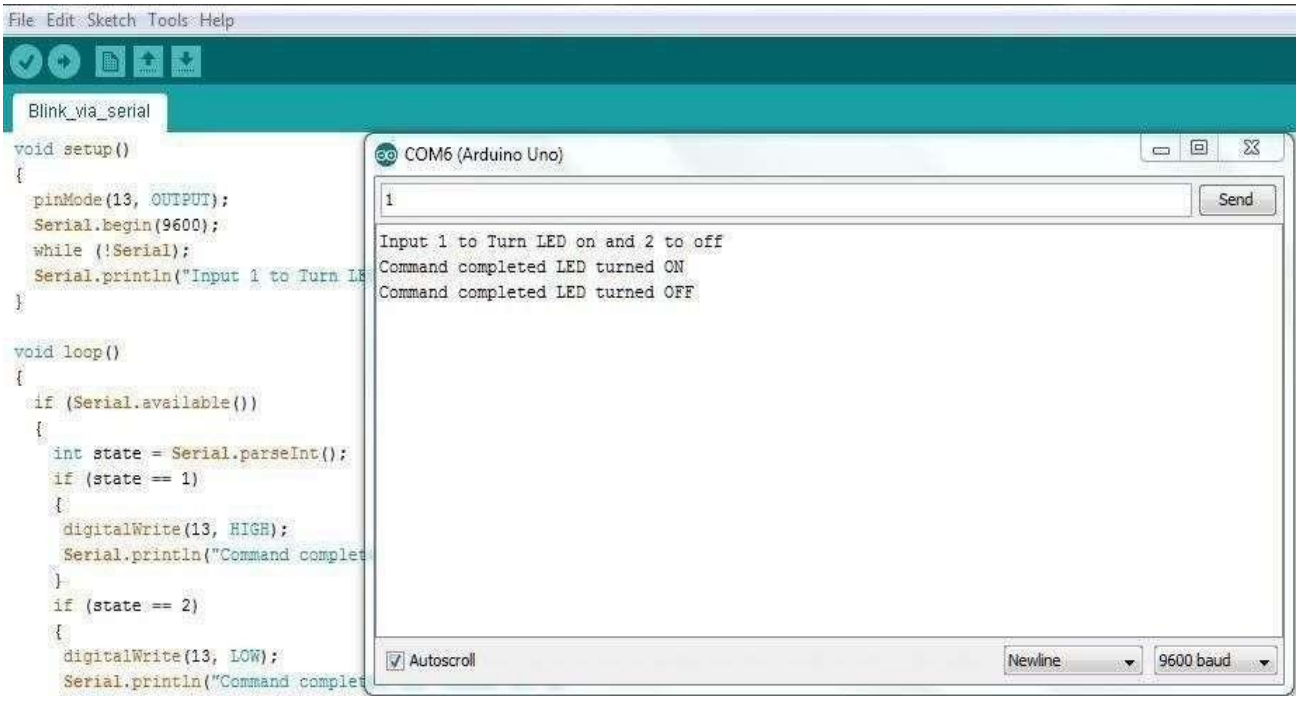
It's time to check and upload the code now that the example has loaded. The code is checked for faults at the verify stage, after which the Arduino is prepared for uploading the code. The binary data that was produced from the code is actually uploaded to the Arduino through the serial port during the upload stage.

Use the checkbox in the upper left window to validate and build the code. The Arduino code will be assembled when you click "Verify."

The output pane at the base of the IDE should display the following message if the compilation stage was successful. You might also get a notice that looks similar, but does not contain the terms "ERROR" and "WARNING." This compilation is effective.

You must now upload the code to the Arduino Uno when it has been compiled. Click the arrow next to the check mark to accomplish this.

The "Upload" button will upload your code to the Arduino.



After the code has been uploaded, the written code's output can be checked in the serial monitor or with other output devices such an LCD, LED, buzzer, and so on.

6.1.3 Conclusion

The Arduino is an effective tool for prototyping for numerous reasons, including the absence of a specialist programmer, the abundance of libraries available, and the ease of use of its IDE. In this project, we just managed to get a light to blink, but you can expect much more in the future. Try your hand at conversing on the internet, collecting measurements, interacting with displays, and perhaps even using AI.

6.2 CODE

```
#include <String.h>
#include <SoftwareSerial.h>

SoftwareSerial gsm(10,11);
int Led1 = 3; // for GAS
int Led2 = 4; // for FIRE
int greenLed = 8;
int buzzer1 = 5; // for GAS
int buzzer2 = 6; // for FIRE
int gasPin = A0;
int flamePin = 2;
int gasSensorThres = 250; // Your threshold value

void setup() {
  Serial.begin(9600);
  gsm.begin(9600);
  Serial.println("System is Initializing..");
  pinMode(Led1, OUTPUT);
  pinMode(Led2, OUTPUT);
  pinMode(greenLed, OUTPUT);
  pinMode(buzzer1, OUTPUT);
  pinMode(buzzer2, OUTPUT);
  pinMode(gasPin, INPUT);
  pinMode(flamePin, INPUT);
  delay(5000); // wait for 5 seconds

}
```

```
void loop() {
  int gasSensor = analogRead(gasPin);
  int flameSensor = digitalRead(flamePin);

  Serial.print("gasPin Value: ");
  Serial.println(gasSensor);
  Serial.print("flamePin Value: ");
  Serial.println(flameSensor);
  delay(1000);

  if (gasSensor > gasSensorThres  && flameSensor==LOW){

   digitalWrite(Led1, HIGH);
   tone(buzzer1, 5000); //the buzzer sound frequency at 5000 Hz
   digitalWrite(Led2, HIGH);
   tone(buzzer2, 5000); //the buzzer sound frequency at 5000 Hz
   digitalWrite(greenLed, LOW);
  }

     else if (gasSensor > gasSensorThres){

    Serial.println("GAS DETECTED");

    Serial.begin(9600);
    gsm.begin(9600);
    gsm.println("AT+CMGF=1");
    delay(1000);
    gsm.println("AT+CMGS=\"+917993602506\"\r");
    delay(1000);
```

```
  gsm.println("ALERT PLS,HARMFUL GASES DETECTED");

  delay(100);
  gsm.println((char)26);
  delay(1000);

  digitalWrite(Led1, HIGH);
  tone(buzzer1, 5000); //the buzzer sound frequency at 5000 Hz
  digitalWrite(Led2, LOW);
  noTone(buzzer2);
  digitalWrite(greenLed, LOW);
}


else if (flameSensor==LOW){ // HIGH MEANS NO FLAME

  Serial.println("FIRE DETECTED");

  Serial.begin(9600);
  gsm.begin(9600);
  gsm.println("AT+CMGF=1");
  delay(1000);
  gsm.println("AT+CMGS=\"+917993602506\"\r");
  delay(1000);
  gsm.println("ALERT PLS,FIRE DETECTED");
  delay(100);
  gsm.println((char)26);
  delay(1000);
```

```
    digitalWrite(Led1, LOW);
    noTone(buzzer1);
    digitalWrite(Led2, HIGH);
    tone(buzzer2, 5000); //the buzzer sound frequency at 5000 Hz
    digitalWrite(greenLed, LOW);
    }


  else{

    digitalWrite(Led1, LOW);
    digitalWrite(Led2, LOW);
    noTone(buzzer1);
    noTone(buzzer2);
    digitalWrite(greenLed, HIGH);
    Serial.println("----------------------------------------------No Danger Detected----------------
------------------------------");
   }
}
```

CHAPTER 7-RESULT

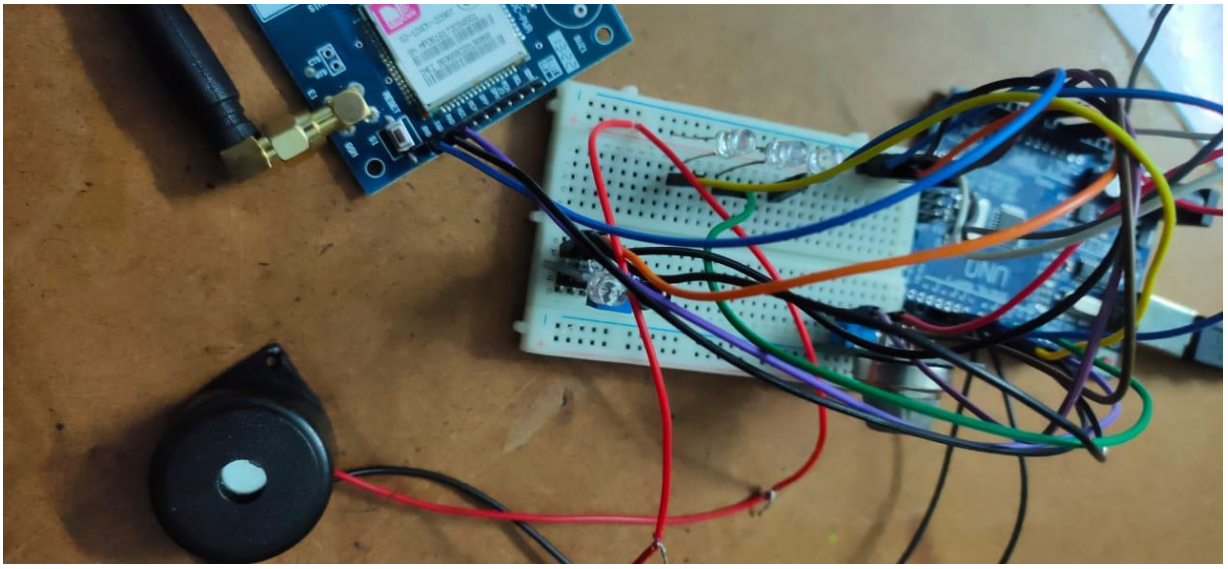


Fig 7.1 Prototype

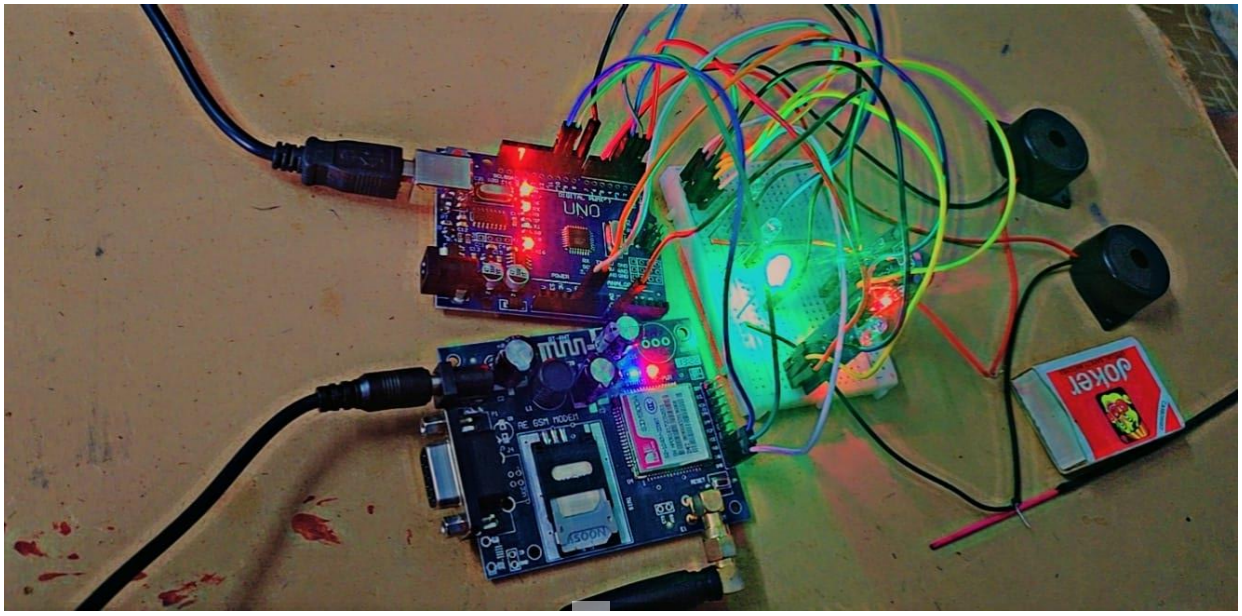


Fig 7.2 Prototyp letecting gas and fire

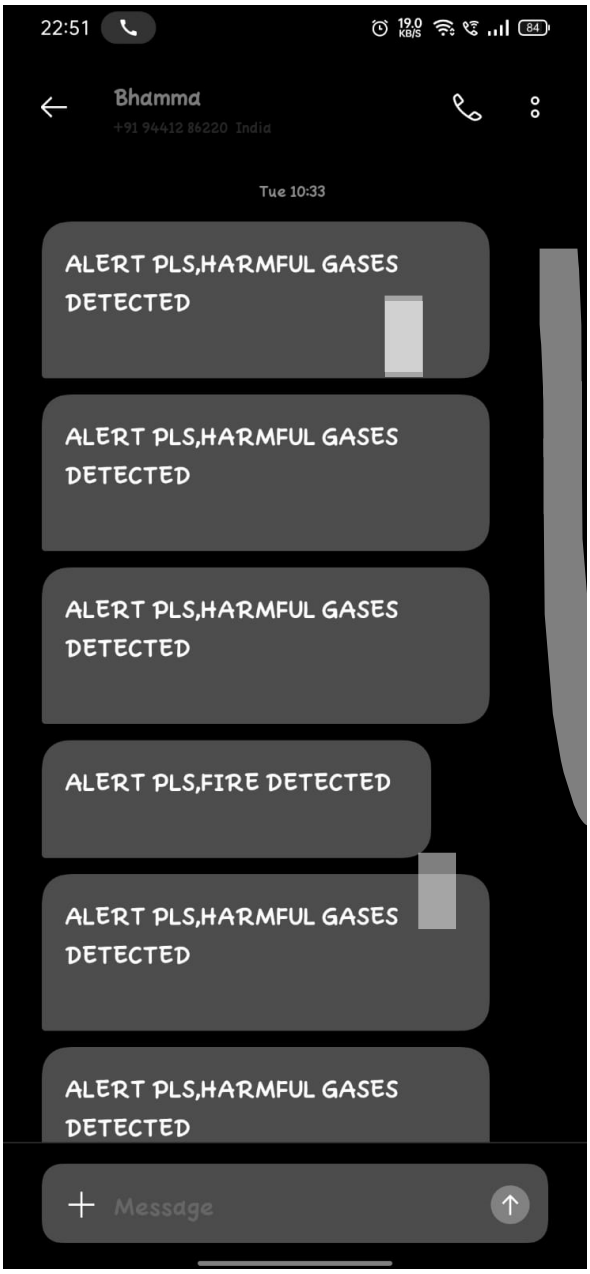


Fig 7.3 SMS Alert

CHAPTER 8 - CONCLUSION & FUTURE SCOPE

Real Time Gas Leakage & Fire Sensing System for Industrial Security will therefore be very helpful in preventing any risk brought on by gas leaks or fire explosions and is useful as part of safety to avoid the threats that can bring negative results. Additionally, it will raise industry safety standards.

Successful project work includes the design, testing, and fabrication of a demo unit. In addition to the factors already employed in this project, more parameters, such as temperature etc., might be added to this project work in order to prevent fire mishaps.

A voice module with other languages can be added to this project as a future addition in addition to the buzzer, making the alarm to the surroundings even clearer. With the help of GSM technology, data may be sent anywhere in the world and uploaded to a cloud storage system that can be accessed from any location in the world. The network of that specific mobile phone, however, is the only item that needs to be examined.

With only few adjustments, the system employed in this research can be applied to real applications.

BIBILOGRAPHY

- Intro to Embedded systems, By Shibu,2009.
- Getting Started with Arduino; Massimo Banzi, Michael Shiloh
- GRIET Arduino manual.
- Programming Arduino by Simon Monk
- https://en.wikipedia.org/wiki/Arduino
- https://en.wikipedia.org/wiki/Gas_detector
- https://en.wikipedia.org/wiki/Infrared_detector
- https://en.wikipedia.org/wiki/GSM
- https://ieeexplore.ieee.org/document/8822055
- https://ieeexplore.ieee.org/document/7972304
- https://www.arduino.cc/en/main/software
- https://en.wikipedia.org/wiki/Arduino_IDE