DESIGNING AND DEVELOPING FIRE AND SMOKE ALARM SYSTEM USING ARDUINO INTERFACE WITH CALL AND SMS ALERT

Submitted by

TANISHA ROY

Registration number: 211-1211-0466-20 & Roll Number: 203211-11-0041

DEBOJIT PAL

Registration number: 211-1111-0438-20 & Roll Number 203211-21-0063

ANIMA MAHATO

Registration number: 211-1211-0436-20 & Roll Number: 203211-11-0034

Supervisor:

Prof. SASWATI CHAKRABORTY

Assistant Professor

Final Project Report submitted to the Department of Computer Science, Maharaja Manindra Chandra College, University of Calcutta in partial fulfillment of the requirements for the Degree of

B. Sc. IN COMPUTER SCIENCE

2022-2023 Semester VI

CERTIFICATE

This is to certify that Tanisha Roy, Debojit Pal and Anima Mahato have carried out their project entitled **"DESIGNING AND DEVELOPING FIRE AND SMOKE DETECTION SYSTEM USING ARDUINO INTERFACE WITH SMS AND CALL ALERT"** under my supervision as per the requirement for the degree of Bachelor of Science in the department of Computer Science, University of Calcutta. During this tenure they have learnt Arduino IDE and applied them independently. This report is not being submitted elsewhere for examination by other students except them.

. . . .

(Prof. Saswati Chakraborty) **External Examiner**

Dept. of Computer Science

Maharaja Manindra Chandra College

University of Calcutta

ACKNOWLEDGEMENT

With great pleasure we acknowledge that our respected teacher Prof. Saswati Chakraborty, Computer Science Department (C.U) has introduced us to this fascinating field of study and guided us for the same. We achieved a golden opportunity to work on such a wonderful project under her guidance.

We are especially thankful and in fact indebted to our guide Prof. Saswati Chakraborty for her wonderful cooperation in completing our project successfully. She had been a wonderful guide to us throughout our project work. We express our sincere gratitude to her. We would like to express our special thanks of gratitude to our teacher who gave us the golden opportunity to do this wonderful project on the topic (**DESIGNING AND DEVELOPING FIRE AND SMOKE DETECTION SYSTEM USING ARDUINO INTERFACE WITH CALL AND SMS ALERT**), which also helped us in doing a lot of Research and we came to know about so many new things. We would be ever grateful to our respected teacher Prof. Saswati Chakraborty for her valuable support.

. . . .

Tanisha Roy   *Debojit Pal*   *Anima Mahato*

Date:

Department of Computer Science

Maharaja Manindra Chandra College

University of Calcutta

CONTENTS

ABSTRACT

Fires are very detrimental, ranging from loss of property to even casualties. Inflorescence can be seen when the fire has grown, or smoke has risen from the building. Therefore, we need a tool that can detect an early smoke and fire source's presence not to spread. This study aims to design and build an efficient and affordable device, which can be applied to housing . This tool is intended to use Arduino Uno R3 as the central controller. A flame sensor can respond to infrared light emission in the modulation spectrum from 5 to 30 cycles per second Web as a server, and data storage on the Internet of Things. The experimental results show that when the fire sensor feels the fire , the system will make a call and when the smoke sensor reads the smoke, the system will send message to the user and alert the house. This research produces a tool that can detect the early presence that can notify the user's mobile device to minimize the spread of a more massive fire and smoke.

INTRODUCTION

1.1 INTRODUCTION

1.1.1 DOMAIN DESCRIPTION

Fire and smoke detection systems are essential in alerting people before fire engulfs their homes. However, fire alarm systems, today, require a lot of wiring and labor to be installed. This discourages users from installing them in their homes. Therefore, we are proposing an IoT based fire and smoke detection system that is easy to setup. The proposed system is consists of a microcontroller (Arduino Uno R3) connected to smoke and flame sensor that continuously sense the surrounding environment to detect the presence of fire and smoke. Once fire and smoke are detected by a node, it sends a signal to a centralized node that is triggered to make a call and sends an sms to the user and alert the house by producing a local alarm. The user can also get information about the status of his/her home via sending a SMS and call to the system. Simultaneously, Arduino sends the information to the GSM module SIM900A. GSM SIM900A will then the accompanying information to the mobile through message and call, where, approved individuals can take fitting measure so as to check the fire and smoke. The device which would enable the work force to get data identified with the area, where the fire is detected and exhaust fumes by turning on the exhaust fan.

1.1.2 MOTIVATION

- These accidents can occur from discarded cigarettes left on flammable materials gas leak from cylinder.
- The primary motivation for fire alarm system requirements in building and fire codes is to provide early notification to building occupants so they can exit the building, and to notify the fire service so it can respond to the fire.
- By detecting a fire quickly and accurately (i.e., by not sacrificing speed or causing false alarms) and providing early warning notification, a fire-detection system can limit the emission of toxic products created by combustion, as well as global-warming gases produced by the fire itself.
- This study aims to design and build an efficient and affordable device, which can be applied to housing or warehouse.

1.1.3 SCOPE

The proposed system, Automated fault tolerant fire and smoke detection and warning communicator system, is appropriate for various buildings sizes and kinds. Each premise can have its own installation that depends on its structure. However, the system design does not cover the open-door areas. The system has two main goals: the first goal is to make sure that an action is taken to prevent the fire by using automated direct notification to civil defense, while the second goal is to reduce the amount of resources loss and victims as much as possible by using automated notification to the owner of the facility and the neighbours of fire location.

1.2 BACKGROUND

Fire and smoke detectors are usually things that we take for granted. Most of us have these useful life-saving devices installed in our homes and places of work, and although we rely on them to work when we need them, few of us actually know much about fire and smoke detectors. Here at Protect & Detect, we want to make sure all property owners are fully informed about the fire alarm systems in place so here is a little about their history to get you started.

The very first fire alarm system was invented by Dr. William F. Channing and Moses Farmer in 1852, and their original design was two fire alarm boxes, each containing a telegraphic key with a handle attached. When someone detected a fire, they would crank the handle, which would then relay the details of the fire alarm box number to a central alarm station. The operator at the station would receive the message, which would then be forwarded to the fire department.Francis Robbins Upton was the man responsible for designing the first electric fire alarm, but when he patented the design in 1890, the patenting office mistakenly recorded it as the “Portable Electric Tire Alarm”. Oh dear!

Although the design was largely unpopular at first and received little attention, it soon became apparent that the need for this device was growing.

1.3 OVERVIEW AND BENEFITS

Fire and Smoke Detectors play a very important role in HAS(HOME AUTOMATION SYSTEM) .They help in detecting fire and smoke at an early stage and can help in saving lives. Commercial Fire detecting systems usually have an alarm signaling, with the help of a buzzer or Siren and smoke detecting system detect the smoke and exhaust fumes automatically by turning on the exhaust fan. We have designed an IoT based Fire Alerting System using flame and a smoke sensor which would not only signal the presence of fire in a particular premise but will also send related information through IoT.

Internet of Things (IoT) is basically the network of 'things' by which physical things can exchange data with the help of sensors, electronics, software, and connectivity. These systems do not require any human interaction. In this Arduino fire alarm system using fire and smoke sensor using the IoT project, we can send LIVE information like flame, Smoke Value detected by a particular device to the owner of the home authority whoever installs this device.

1.4 PROJECT OBJECTIVES

Nowadays, Fire and smoke detecting and Alerting system are very common in HOME AUTOMATION SYSTEM(HAS). They usually detect fire and smoke and alert people . But, what happens, when nobody is there to listen to the Alarm. Like when nobody is at home. So, to inform the home authority about fire incidents ,we have made this IoT Fire and smoke Detector using the Arduino project. This project can be further modified to notify the fire control department automatically. This IoT based project detects the nearby flame and smoke using an Infrared Flame Sensor and smoke sensor and then Arduino Triggers the relay to exhaust the smoke automatically. It also informs the authority using call and SMS. With the help of this project, you can easily understand the concept of a fire ad smoke detector and alert system.

LITERATURE REVIEW

"Development in building fire and smoke detection and evacuation system-a comprehensive review" by Gajanand S. Birajdar, Rajesh Singh, Anita Gehlot, Amit Kumar Thakur

Early detection of fire and smoke is important to save the life of human beings as well as property. In this case, multiple sensors need to be installed at different places of the building which plays a crucial role.

A system based on Lab view is proposed to detect and alert fire events in a portable community video surveillance system developed in can be used to detect people's motion. Environmental parameters such as temperature, the concentration of toxic gases is measured and monitored in using Arduino and sensors. In, Low-cost IoT based evacuation Service is implemented for real-life situations of Building Automation System (BAS) applications. The Mobile terminal-based system is discussed in which send SMS and make call and alert users to exit safely.

"*BuildingFire Evacuation: An IoT-Aided Perspective in the 5G Era*" by Hongqiang Fang, ORCID, Siuming Lo and Jacqueline T. Y. Lo

The IoT-aided building fire evacuation is a new concept of building evacuation mode which enhances the building fire evacuation, including the elevator evacuation process, by making the most appropriate evacuation strategies based on the real-time fire-ground information, such as fire environment and occupants' situations. With the maturity of IoT technologies and the appearance of more IoT application scenarios in the 5G era, the IoT-aided building fire evacuation will no longer be only a concept. It has great potential to be used in practice and play more important roles in future emergency response. Therefore, by focusing on IoT applications in building fire evacuation, this paper investigated the advantages and insufficiencies of current smart building fire evacuation systems. After that, a preliminary design of an IoT-aided building fire evacuation control system was proposed. The proposed IoT system was designed in the sequence of information needs, information sources and data transmission, and potential services and applications, which corresponds to the architecture of an IoT system. In addition, the advantages of the proposed system were also concluded. The proposed system could enable (1) monitoring of the fire environment and occupants' situations in buildings; (2) 3D-visualization of the building information; and (3) system-level-based

evacuation strategy planning. With the help of the proposed IoT system, safe and more efficient building fire evacuation will surely be enabled.

"*Automation systems in smart buildings*" by D. S. Vijayan, A. Leema Rose, S. Arvindan, J. Revathy& C. Amuthadevi

In the present scenario everyone turned into smart applications by including the intelligence into the applications and this reduces the burden of frequent interruption or control by the humans. Smartness makes the ability to interconnect more real time parameters by M2M (Machine to Machine) interaction and make wise decisions in harmful situations. Many low-cost devices are available in the market to collect the real time data and transmit them using Internet of Things (IoT). So, the control can be done remotely. In this work, few basic applications in smart buildings are going to be studied to analyze with the technical advances which can bring better solutions automatically. The details collected from different sensors will be useful for analytics and the need for smart design models for better buildings. The services needed in smart building such as security control, energy management, control and monitoring of HVAC system, water management, lighting systems, health system of elders and fire detection are going to be surveyed. The major objective of this study is to identify the issues faced by current methodologies with these applications and give a guideline for future research. From analyzing the relevant methods and needs it is observed that, the buildings construction and usage can depend on the applications, the smart system can respond intelligently for further events. In future all the traditional buildings will be automotive based on HVAC (Heating, ventilation, and air conditioning).

THEORY

3.1. IoT (INTERNET OF THINGS)

The Internet of things (IoT) describes physical objects (or groups of such objects) with sensors, processing ability, software, and other technologies that connect and exchange data with other devices and systems over the Internet or other communications networks. Internet of things has been considered a misnomer because devices do not need to be connected to the public internet, they only need to be connected to a network and be individually addressable.
IoT as a term has evolved long way as a result of convergence of multiple technologies, machine learning, embedded systems and commodity sensors. It is a system of interconnected devices assigned a UIDS, enabling data transfer and control of devices over a network. It reduced the necessity of actual interaction in order to control a device. It is an advanced automation and analytics system which exploits networking, sensing, big data, and artificial intelligence technology to deliver complete systems for a product or service. These systems allow greater transparency, control, and performance when applied to any industry or system.

3.1.1 Features of IoT

- **Intelligence**
  IoT comes with the combination of algorithms and computation, software & hardware that makes it smart. Ambient intelligence in IoT enhances its capabilities which facilitate the things to respond in an intelligent way to a particular situation and supports them in carrying out specific tasks. In spite of all the popularity of smart technologies, intelligence in IoT is only concerned as a means of interaction between devices, while user and device interaction are achieved by standard input methods and graphical user interface.
- **Connectivity**
  The heart and soul of IoT is its connectivity. Connectivity means the establishment of a connection between different devices (or nodes) so that they can communicate on their own. In IoT, various devices, sensors, computers, and data busses need to interact and communicate with each other. A fast, safe, and secure connection is a must for IoT to be of any business use. IoT also connects devices with cross-domain technology like cloud computing, artificial intelligence, and blockchain technology. We can connect them over radio waves, Wi-Fi, Bluetooth, or wires.

- **Dynamic Nature**

  The primary activity of Internet of Things is to collect data from its environment, this is achieved with the dynamic changes that take place around the devices. The state of these devices changes dynamically, example sleeping and waking up, connected and/or disconnected as well as the context of devices including temperature, location and speed. In addition to the state of the device, the number of devices also changes dynamically with a person, place and time.

- **Scaling**

  IoT systems are designed in such a way that the number of devices, sensors, or computers can be scaled up and down according to the need. An IoT system should be elastic enough so that it can handle workload during peak demand hours and can resort back to the normal state when the demand is low.

- **Sensing**

  IoT devices gather information about their surroundings (such as temperature, light, sound, acceleration, pressure) and then, after analyzing the data, take a decision. Thus, sensors help in automation by gathering information and taking actions that would otherwise, be done by humans. The raw data gathered, and the analysed data, serve as the basis of the functioning of IoT. For example, in an automatic door, sensors would collect data through sensors such as radar sensors and optical sensors. If it detects a person coming, it will open the door automatically. Some sensors used in IoT are- Humidity sensor, temperature sensor, Accelerometer, Gyroscope, Motion sensor, image sensor, level sensor, and Proximity sensor.

- **Analyzing**

  We know IoT gathers raw information through sensors, but why does IoT need data? What does IoT do with all that raw data? Data as such has no value of its own. It is meaningless and useless until it is purposefully processed to gain some meaningful insights from it. IoT gathers raw data to extract something meaningful out of it. Analyzing the raw data in terms of its structure, correlation, and usability is necessary because, if processed properly, it can be very useful. In the above-mentioned example of the automatic door for instance, after analyzing the data through sensors, it should be able to differentiate between a person and an animal.

- **Artificial Intelligence**

  IoT becomes a lot more useful when combined with artificial intelligence. For instance, if you are out of groceries, your smart refrigerator can notify you to bring some on your

way back home. Things like these have been made possible by the application of artificial intelligence. IoT devices collect raw data from their surroundings and convert them into something useful and insightful. The IoT devices and systems are also trained with various machine learning models so that they can better understand the changes in their surroundings and perform better.

- **Security**
  IoT devices are naturally vulnerable to security threats. As we gain efficiencies, novel experiences, and other benefits from the IoT, it would be a mistake to forget about security concerns associated with it. There is a high level of transparency and privacy issues with IoT. It is important to secure the endpoints, the networks, and the data that is transferred across all of it means creating a security paradigm.

3.1.2 ADVANTAGES of IoT

- **Communication**
  IoT encourages the communication between devices, also famously known as Machine-to-Machine (M2M) communication. Because of this, the physical devices are able to stay connected and hence the total transparency is available with lesser inefficiencies and greater quality.

- **Automation and Control**
  Due to physical objects getting connected and controlled digitally and centrally with wireless infrastructure, there is a large amount of automation and control in the workings. Without human intervention, the machines are able to communicate with each other leading to faster and timely output.

- **Information**

  It is obvious that having more information helps making better decisions. Whether it is mundane decisions as needing to know what to buy at the grocery store or if your company has enough widgets and supplies, knowledge is power and more knowledge is better.

- **Monitor Data**
  The second most obvious advantage of IoT is monitoring. Knowing the exact quantity of supplies or the air quality in your home, can further provide more information that could not have previously been collected easily. For instance, knowing that you are low on milk

or printer ink could save you another trip to the store in the near future. Furthermore, monitoring the expiration of products can and will improve safety.

- **Better Time Management**
  As hinted in the previous examples, the amount of time saved because of IoT could be quite large. And in today's modern life, we all could use more time.

- **Better Quality of Life**
  all the applications of this technology culminate in increased comfort,convenience,and better management, thereby improving the quality of life.

3.1.3 DISADVANTAGES of IoT

- **Compatibility**
  Currently, there is no international standard of compatibility for the tagging and monitoring equipment. I believe this disadvantage is the easiest to overcome. The manufacturing companies of these equipment just need to agree to a standard, such as Bluetooth, USB, etc. This is nothing new or innovative needed.

- **Complexity**
  As with all complex systems, there are more opportunities of failure. With the Internet of Things, failures could sky rocket. For instance, let's say that both you and your spouse each get a message saying that your milk has expired, and both of you stop at a store on your way home, and you both purchase milk. As a result, you and your spouse have purchased twice the amount that you both need. Or maybe a bug in the software ends up automatically ordering a new ink cartridge for your printer each and every hour for a few days, or at least after each power failure, when you only need a single replacement.

- **Safety**
  Imagine if a notorious hacker changes your prescription. Or if a store automatically ships you an equivalent product that you are allergic to, or a flavour that you do not like, or a product that is already expired. As a result, safety is ultimately in the hands of the consumer to verify any and all automation.
  As all the household appliances, industrial machinery, public sector services like water supply and transport, and many other devices all are connected to the Internet, a lot of information is available on it. This information is prone to attack by hackers. It would be

very disastrous if private and confidential information is accessed by unauthorized intruders.

- **Lesser Employment of Menial Staff**
  The unskilled workers and helpers may end up losing their jobs in the effect of automation of daily activities. This can lead to unemployment issues in the society. This is a problem with the advent of any technology and can be overcome with education. With daily activities getting automated, naturally, there will be fewer requirements of human resources, primarily, workers and less educated staff. This may create Unemployment issue in the society.

- **Technology Takes Control of Life**
  Our lives will be increasingly controlled by technology, and will be dependent on it. The younger generation is already addicted to technology for every little thing. We have to decide how much of our daily lives are we willing to mechanize and be controlled by technology.

3.1.4 APPLICATIONS GROUNDS OF IoT

- **Smart homes**
  One of the best and the most practical applications of IoT, smart homes really take both, convenience and home security, to the next level. Though there are different levels at which IoT is applied for smart homes, the best is the one that blends intelligent utility systems and entertainment together. For instance, your electricity meter with an IoT device giving you insights into your everyday water usage, your set-top box that allows you to record shows from remote, Automatic Illumination Systems, Advanced Locking Systems, and Connected Surveillance Systems all fit into this concept of smart homes. As IoT evolves, we can be sure that most of the devices will become smarter, enabling enhanced home security.3.1.4.3 Health care.

- **Traffic Management**
  Car traffic management in large cities can be greatly improved with the help of the Internet of Things (IoT). The Internet of Things helps us stay informed and improves traffic monitoring by allowing us to use our mobile phones as sensors to collect and share data from our vehicles through apps like Waze or Google Maps. This feeds and improves the data on the various routes to the same destination, distance, and estimated arrival time.

- **Agriculture**

  A greenhouse farming technique enhances the yield of crops by controlling environmental parameters. However, manual handling results in production loss, energy loss, and labour cost, making the process less effective. A greenhouse with embedded devices not only makes it easier to be monitored but also, enables us to control the climate inside it. Sensors measure different parameters according to the plant requirement and send it to the cloud. It, then, processes the data and applies a control action.

- **Industrial Automation**

  For a higher return of investment this field requires both fast developments and quality of products. This vitality thus coined the term IIOT. This whole schematic is re-engineered by IoT applications. Following are the domains of IoT applications in -

  - industrial automation
  - Factory Digitalization
  - Product flow Monitoring Inventory Management Safety and Security
  - Quality Control
  - Packaging optimization
  - Logistics and Supply Chain Optimization
  - Government and Safety

  IoT applied to government and safety allows improved law enforcement, defense, city planning, and economic management. The technology fills in the current gaps, corrects many current flaws, and expands the reach of these efforts. For example, IoT can help city planners have a clearer view of the impact of their design, and governments have a better idea of the local economy.

- **Self-driven Cars**

  These cars use several sensors and embedded systems connected to the Cloud and the internet to keep generating data and sending them to the Cloud for informed decision-making through Machine Learning. Though it will take a few more years for the technology to evolve completely and for countries to amend laws and policies, what we're witnessing right now is one of the best applications of IoT.

3.1.5 IoT SOFTWARE

IoT software addresses its key areas of networking and action through platforms, embedded systems, partner systems, and middleware. These individual and master applications are responsible for data collection, device integration, real-time analytics, and application and

process extension within the IOT network. They exploit integration with critical business systems (e.g., ordering systems, robotics, scheduling, and more) in the execution of related tasks.

- **Data Collection**
  This software manages sensing, measurements, light data filtering, light data security, and aggregation of data. It uses certain protocols to aid sensors in connecting with real-time, machine-to-machine networks. Then it collects data from multiple devices and distributes it in accordance with settings. It also works in reverse by distributing data over devices. The system eventually transmits all collected data to a central server.

- **Device Integration**
  Software supporting integration binds (dependent relationships) all system devices to create the body of the IoT system. It ensures the necessary cooperation and stable networking between devices. These applications are the defining software technology of the IoT network because without them, it is not an IoT system. They manage the various applications, protocols, and limitations of each device to allow communication.

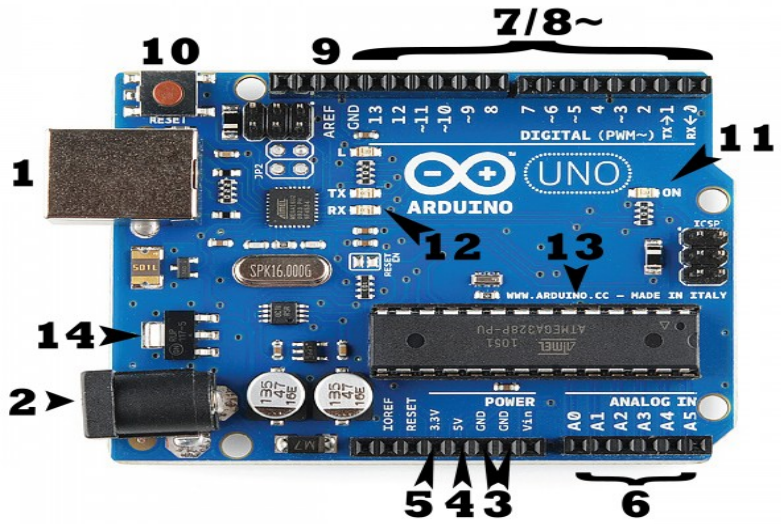
- **Real-Time Analytics**
  These applications take data or input from various devices and convert it into feasible actions or clear patterns for human analysis. They analyse information based on various settings and designs in order to perform automation-related tasks or provide the data required by industry.

3.2 ARDUINO

Arduino is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment) that runs on your computer, used to write and upload computer code to the physical board.

The Arduino platform has become quite popular with people just starting out with electronics, and for good reason. Unlike most previous programmable circuit boards, the Arduino does not need a separate piece of hardware (called a programmer) in order to load new code onto the board -- you can simply use a USB cable. Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program. Finally, Arduino provides a

standard form factor that breaks out the functions of the micro-controller into a more accessible package.



- **Power (USB / Barrel Jack)**

  Every Arduino board needs a way to be connected to a power source. The Arduino UNO can be powered from a USB cable coming from your computer or a wall power supply (like this) that is terminated in a barrel jack. In the picture above the USB connection is labeled(1) and the barrel jack is labeled(2) .

  The USB connection is also how you will load code onto your Arduino board. More on how to program with Arduino can be found in our Installing and Programming Arduino tutorial.

- **Pins**

  The pins on your Arduino are the places where you connect wires to construct a circuit (probably in conjuction with a breadboard and some wire. They usually have black plastic 'headers' that allow you to just plug a wire right into the board. The Arduino has several different kinds of pins, each of which is labeled on the board and used for different functions.

  - GND(3) : Short for 'Ground'. There are several GND pins on the Arduino, any of which can be used to ground your circuit.

- 5V(4) & 3.3V(5) : As you might guess, the 5V pin supplies 5 volts of power, and the 3.3V pin supplies 3.3 volts of power. Most of the simple components used with the Arduino run happily off of 5 or 3.3 volts.
- Analog (6): The area of pins under the 'Analog In' label (A0 through A5 on the UNO) are Analog In pins. These pins can read the signal from an analog sensor (like a temperature sensor) and convert it into a digital value that we can read.
- Digital(7) : Across from the analog pins are the digital pins (0 through 13 on the UNO). These pins can be used for both digital input (like telling if a button is pushed) and digital output (like powering an LED).
- PWM(8) : You may have noticed the tilde (~) next to some of the digital pins (3, 5, 6, 9, 10, and 11 on the UNO). These pins act as normal digital pins, but can also be used for something called Pulse-Width Modulation (PWM). We have a tutorial on PWM, but for now, think of these pins as being able to simulate analog output (like fading an LED in and out).
- AREF(9): Stands for Analog Reference. Most of the time you can leave this pin alone. It is sometimes used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

- **Reset Button**

  Just like the original Nintendo, the Arduino has a reset button(10). Pushing it will temporarily connect the reset pin to ground and restart any code that is loaded on the Arduino. This can be very useful if your code doesn't repeat, but you want to test it multiple times. Unlike the original Nintendo however, blowing on the Arduino doesn't usually fix any problems.

- **Power LED Indicator**

  Just beneath and to the right of the word "UNO" on your circuit board, there's a tiny LED next to the word 'ON' (11). This LED should light up whenever you plug your Arduino into a power source. If this light doesn't turn on, there's a good chance something is wrong. Time to re-check your circuit!

- **TX RX LEDs**

  TX is short for transmit, RX is short for receive. These markings appear quite a bit in electronics to indicate the pins responsible for serial communication. In our case, there are two places on the Arduino UNO where TX and RX appear -- once by digital pins 0 and

1, and a second time next to the TX and RX indicator LEDs (12). These LEDs will give us some nice visual indications whenever our Arduino is receiving or transmitting data (like when we're loading a new program onto the board).

- **Main IC**

  The black thing with all the metal legs is an IC, or Integrated Circuit(13) . Think of it as the brains of our Arduino. The main IC on the Arduino is slightly different from board type to board type, but is usually from the ATmega line of IC's from the ATMEL company. This can be important, as you may need to know the IC type (along with your board type) before loading up a new program from the Arduino software. This information can usually be found in writing on the top side of the IC. If you want to know more about the difference between various IC's, reading the datasheets is often a good idea.

- **Voltage Regulator**

  The voltage regulator (14)is not actually something you can (or should) interact with on the Arduino. But it is potentially useful to know that it is there and what it's for. The voltage regulator does exactly what it says -- it controls the amount of voltage that is let into the Arduino board. Think of it as a kind of gatekeeper; it will turn away an extra voltage that might harm the circuit. Of course, it has its limits, so don't hook up your Arduino to anything greater than 20 volts.

- **The Arduino Family**

  Arduino makes several different boards, each with different capabilities. In addition, part of being open source hardware means that others can modify and produce derivatives of Arduino boards that provide even more form factors and functionality. If you're not sure which one is right for your project, check this guide for some helpful hints. Here are a few options that are well-suited to someone new to the world of Arduino:

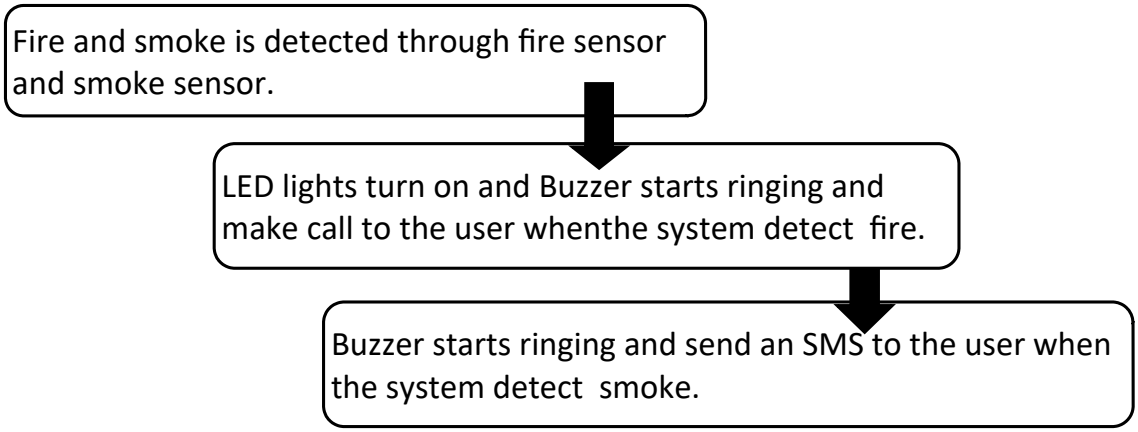
Arduino Uno (R3)

We use Arduino uno R3 for our project. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a USB connection, a power jack, a reset button and more. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

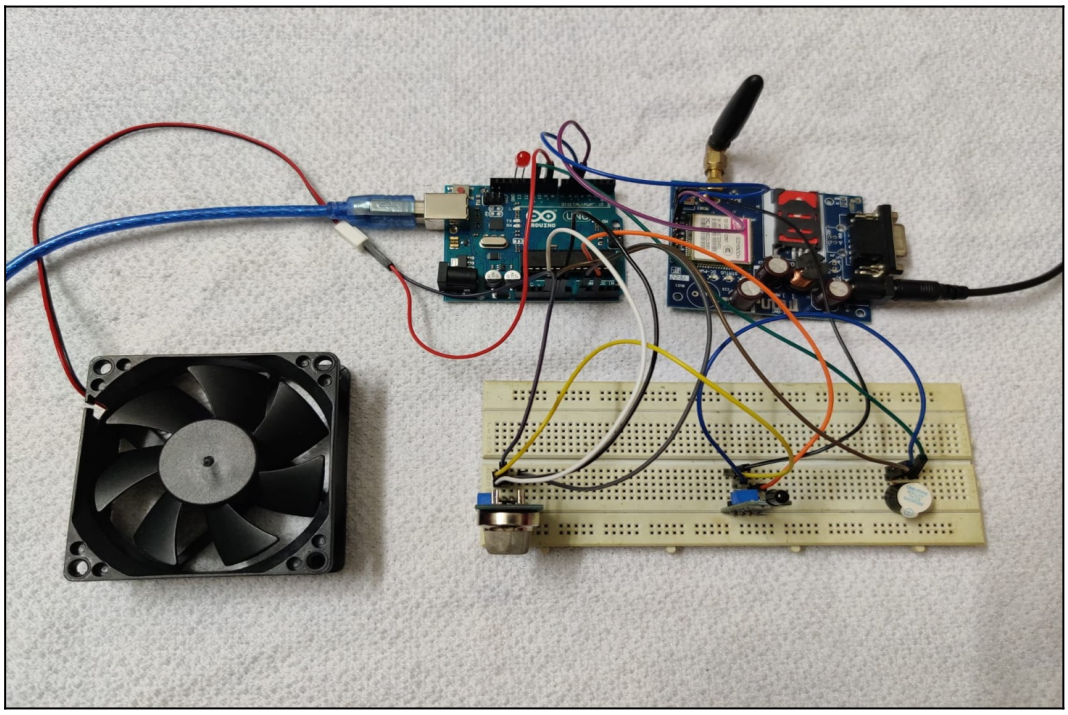
METHODOLOGY

4.1. ALGORITHM DESCRIPTION

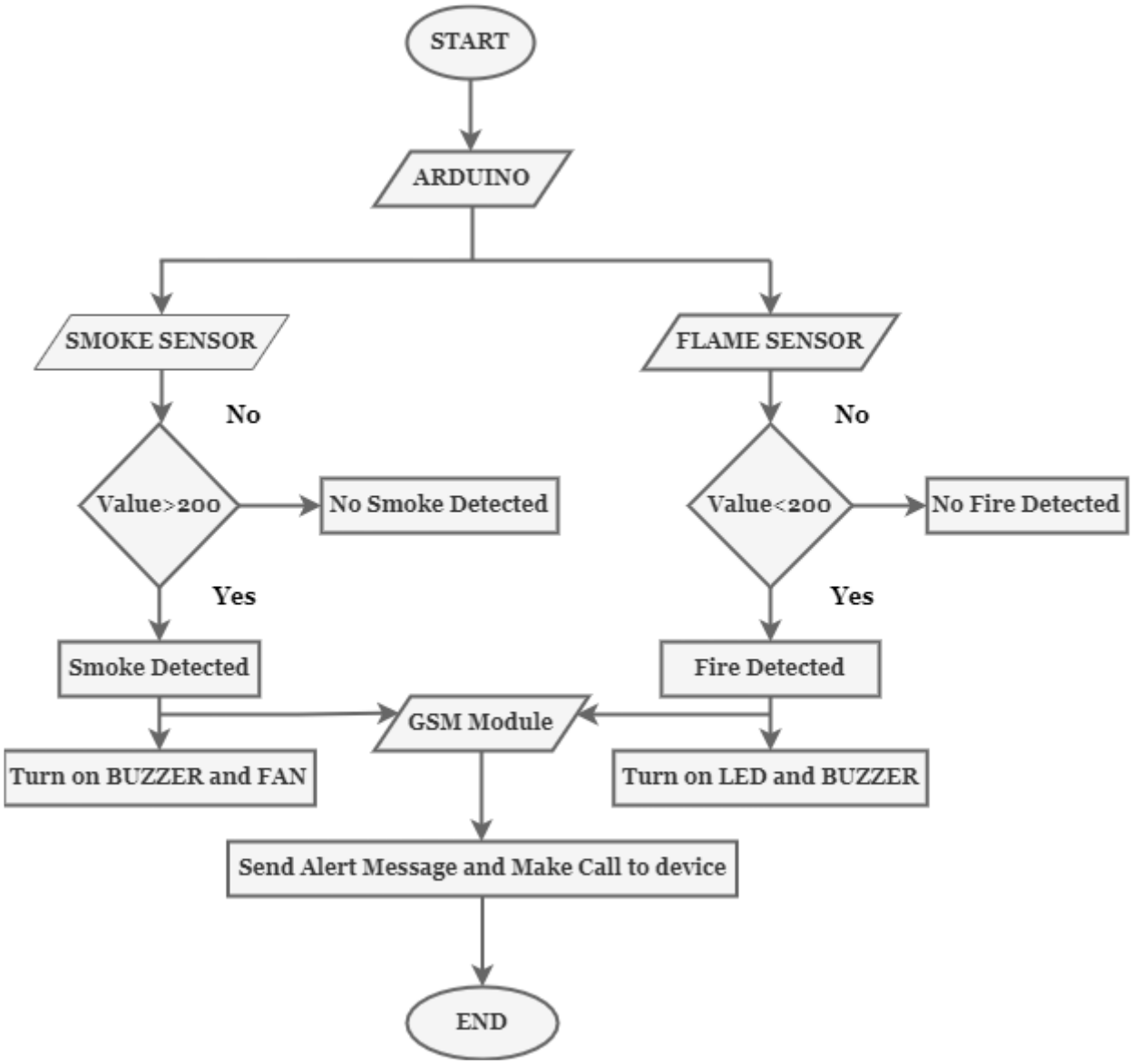
The following diagram describes the process of fire and smoke detection and the steps after that :



4.2. CIRCUIT DIAGRAM



4.3. FLOWCHART



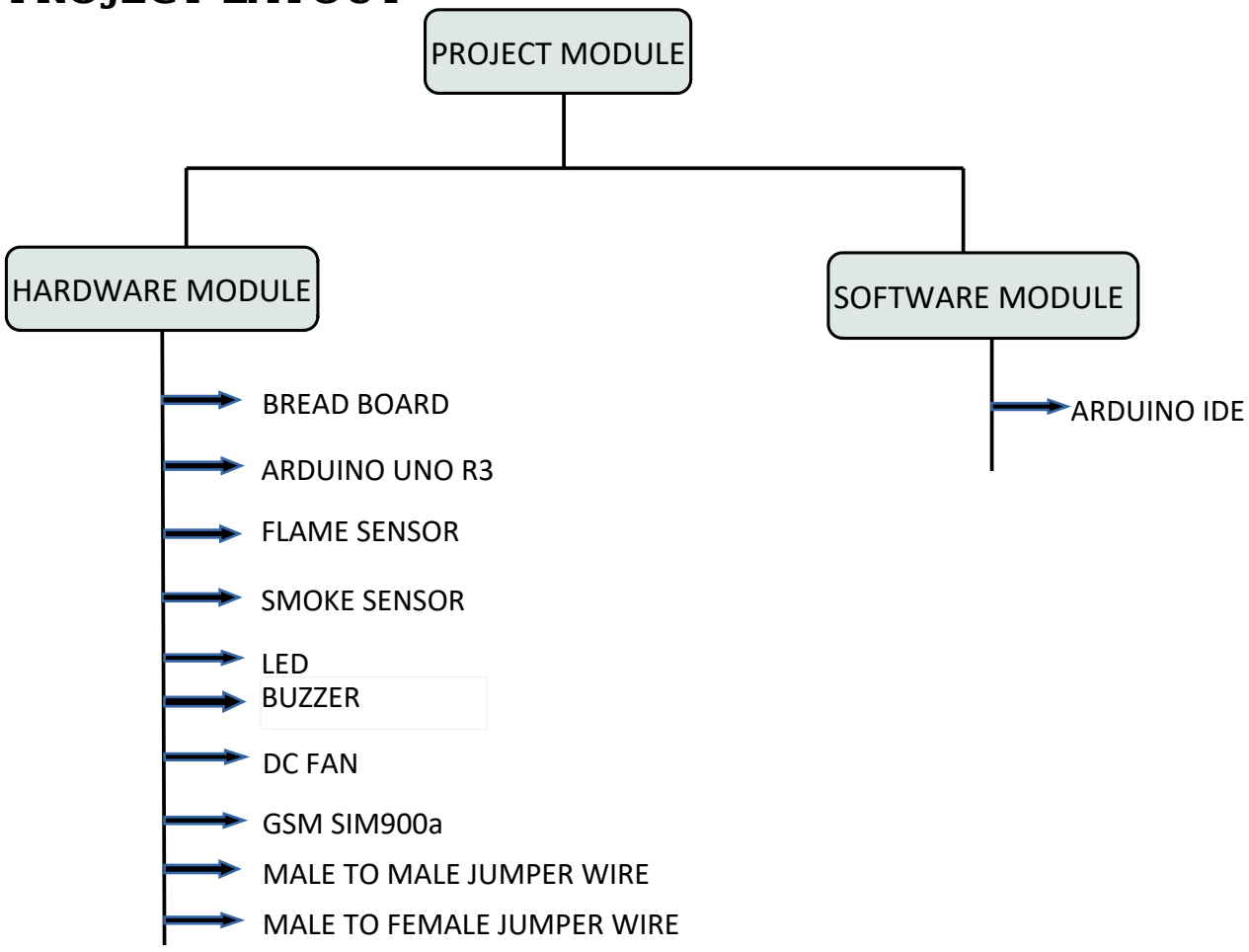
HARDWARE MODELING AND SETUP

5.1. MAIN FEATURES OF THE PROTOTYPE

The features of the developed prototype are:

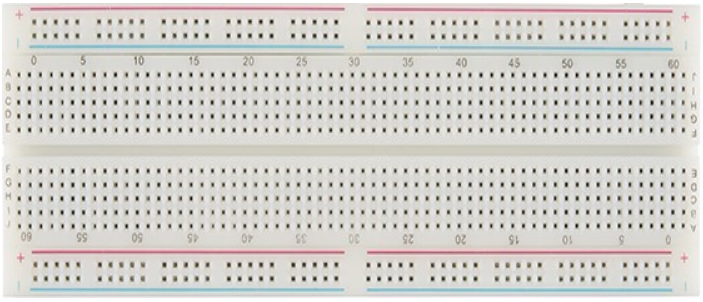
- Simple design easy to integrate into a verity of places and extend on further range.
- The prototype warns peoples when smoke ,fire or other fire related emergencies are detected.
- The prototype's alarms activated automatically from smoke and fire.
- Send SMS to the smartphone when fire is detected by the prototype.
- Make call to the smartphone when smoke is detected by the prototype.
- Displays the status of each time fire and smoke are detected on the application in smartphone.

5.2. PROJECT LAYOUT



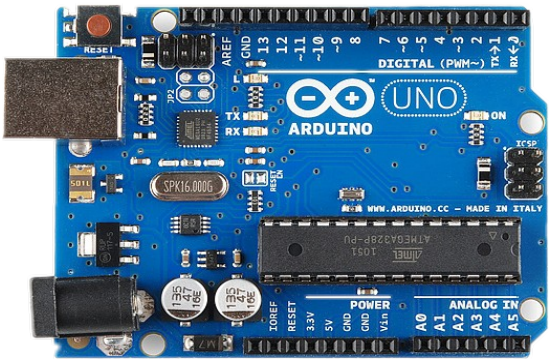
- **BREAD BOARD**

  A breadboard is a widely used tool to design and test circuit. You do not need to solder wires and components to make a circuit while using a bread board. It is easier to mount components & reuse them. Since, components are not soldered you can change your circuit design at any point without any hassle. It consist of an array of conductive metal clips encased in a box made of white ABS plastic, where each clip is insulated with another clips. There are a number of holes on the plastic box, arranged in a particular fashion. A typical bread board layout consists of two types of region also called strips. Bus strips and socket strips. Bus strips are usually used to provide power supply to the circuit. It consists of two columns, one for power voltage and other for ground.



- **ARDUINO UNO R3**

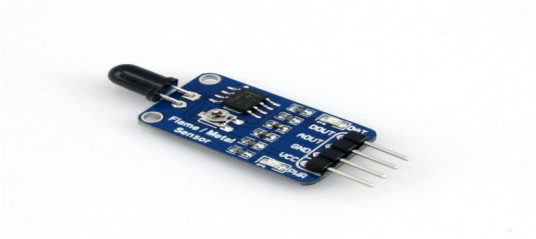
  Arduino UNO is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins , 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header and a reset button.



- **FLAME SENSOR**

  Flame sensor infrared receiver module ignition source detection module is Arduino compatible. It has 4 pins namely VCC, GND, DO- digital output and AO- analogue output. It is one kind of detector which is mainly designed for detecting as well as responding to the occurrence of a fire or flame. The flame detection response can depend on its fitting.

It includes an alarm system, a natural gas line, propane & a fire suppression system. This sensor is used in industrial boilers. The main function of this is to give authentication whether the boiler is properly working or not. The response of these sensors is faster as well as more accurate compare with a heat/smoke detector because of its mechanism while detecting the flame.



- **SMOKE SENSOR**

A smoke sensor which works on 5V DC and draws around 800mW. It has 4 pins namely VCC,GND, DO- digital output, AO- analogue output. It is one of the commonly used gas sensors in MQ sensor series. It is a Metal Oxide Semiconductor (MOS) type Gas Sensor. It is based upon change of resistance of the sensing material when the Gas comes in contact with the material. Using a simple voltage divider network, concentrations of gas can be detected.

- **LED**

Light-emitting diode(LED) is a semiconductor light source. Positive terminal and negative terminal are longer and shorter pins respectively. Emits light when an electric current flows through it. When current passes through an LED, the electrons recombine with holes emitting light in the process. LEDs allow the current to flow in the forward direction and blocks the current in the reverse direction.

- **BUZZER**

  Buzzer is a kind of voice device that converts audio model into sound signal. It is mainly used to prompt or alarm. According to different design and application, it can produce music sound, flute sound, buzzer, alarm sound, electric bell and other different sounds. Typical applications include siren, alarm device, fire alarm, air defense alarm, burglar alarm, timer, etc. It is widely used in household appliances, alarm system, automatic production line, low-voltage electrical equipment, electronic toys, game machines and other products and industries.

- **DC FAN**

  The dc fan are powered with a potential of fixed value . It's voltage values are 5v,12v,24v and 48v.

- **GSM SIM900a**

  It is a GSM modem. It is used for sending SMS messages, making phone call, connecting to the internet via GPRS, and much more.This is an ultra compact module. The SIM900A is a complete Dual-band GSM/GPRS solution in a SMT module which can be embedded in the customer applications allowing you to benefit from small dimensions and cost-effective solutions.Featuring an industry-standard interface, the SIM900A delivers GSM/GPRS 900/1800MHz performance for voice, SMS, Data, and Fax in a small form factor and with low power consumption. With a tiny configuration of 24mm x 24mm x 3 mm, SIM900A can fit almost all the space requirements in your applications, especially for slim and compact demand of design.

- **MALE TO MALE JUMPER WIRE**

  A very Flexible and easily detachable cable. Has 1 Pin male to the 1 pin male header with both ends.

- **MALE TO FEMALE JUMPER WIRE**

  very Flexible and easily detachable cable. Has 1 Pin female to the 1 pin male header .

5.3. COMPONENTS REQIRED FOR THE PROJECT

SL.NO	NAME OF THE COMPONENTS	QUANTITY
1.	Arduino Uno R3	1
2.	Breadboard	1
3.	Flame Sensor	1
4.	Smoke Sensor	1
5.	5v DC Fan	1
6.	Buzzer	1
7.	LED	1
8.	Male to Male jumper wire	10
9.	Male to Female jumper wire	3
10.	GSM SIM900a	1
11.	**USB Cables**	**1**

5.4. SETTING UP THE SYSTEM

5.4.1. SPECIFICATION OF THE ACTUAL COMPONETS USED IN THE CIRCUIT:

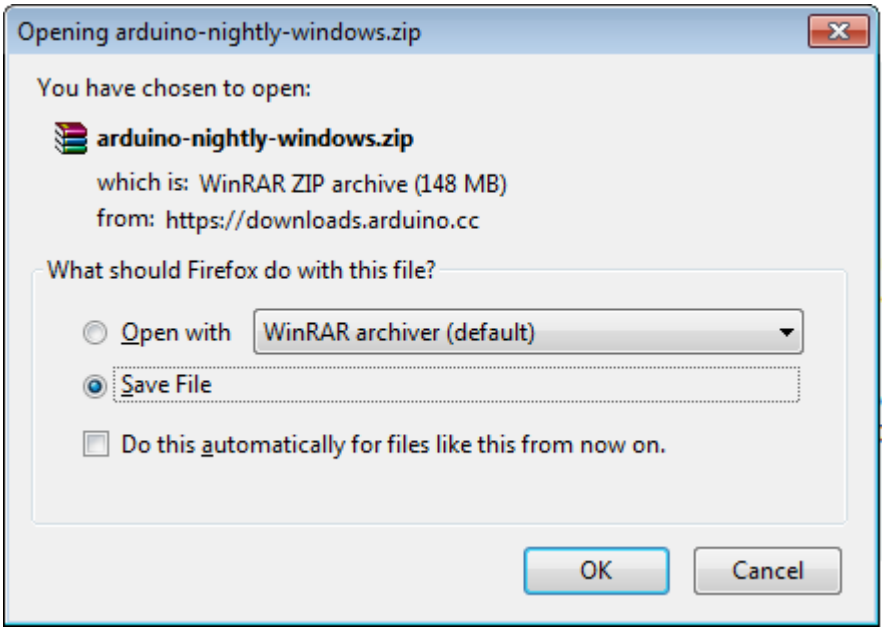
- **Arduino Uno R3**
- **Breadboard**
- **Flame Sensor**
- **Smoke Sensor**
- **5v DC Fan**
- **Buzzer**
- **LED**
- **Male to Male jumper wire**
- **Male to Female jumper wire**
- **GSM SIM900a**
- **USB Cables**

5.4.2. Interfacing Arduino Uno R3 with ARDUINO IDE

Set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1 – First we must need our Arduino board and a USB cable. In case we use Arduino UNO R3.

Step 2 – Download Arduino IDE Software.



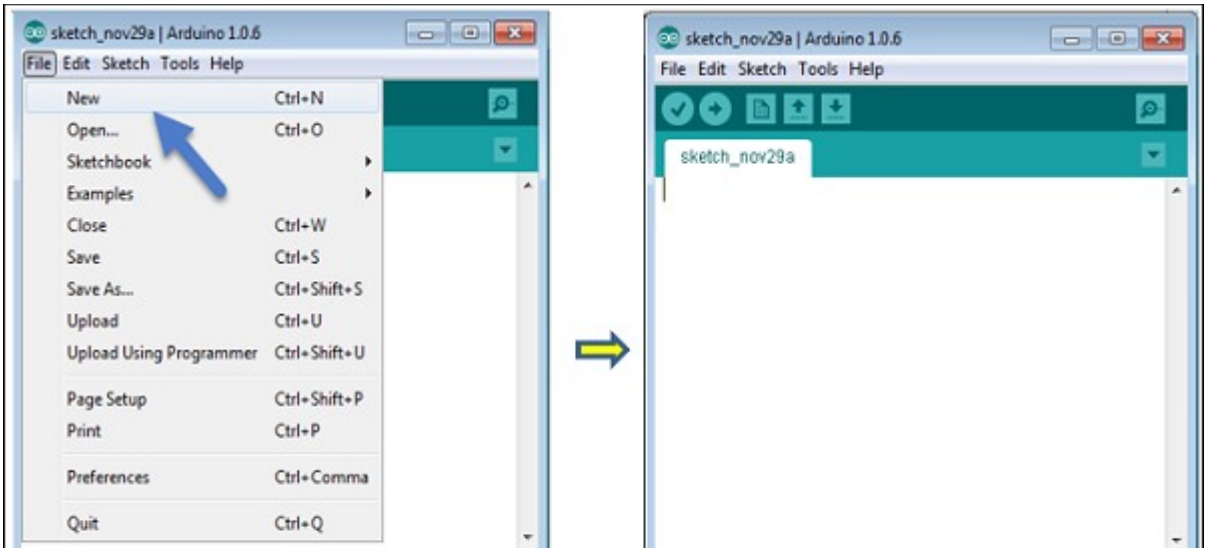
Step 3 – Power up the board.

Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

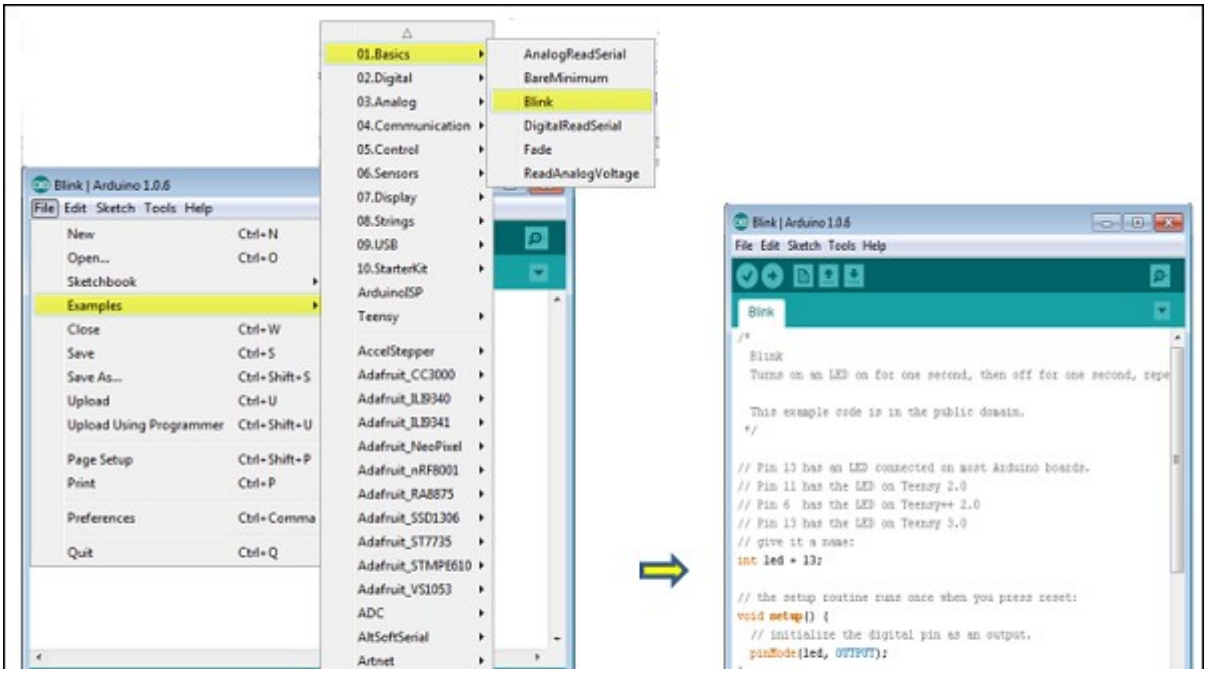
Step 4 – Open our first project.

Once the software starts, you have two options –

To create a new project, select File → New.



To open an existing project example, select File → Example → Basics → Blink.

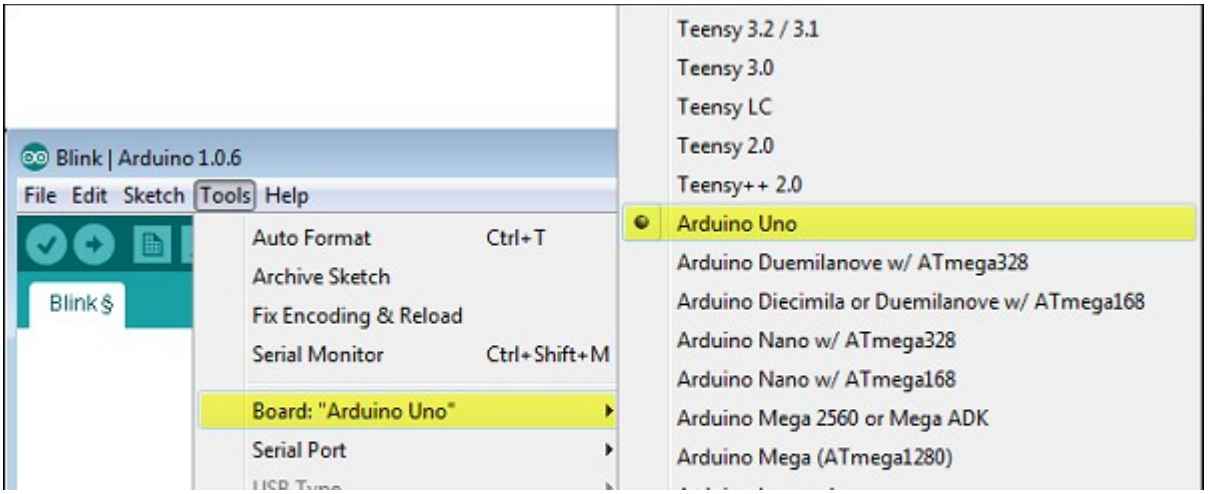


Here, we are selecting just one of the examples with the name Blink. It turns the LED on and off with some time delay. You can select any other example from the list.

Step 5 – Select the Arduino board.

To avoid any error while uploading 4our program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

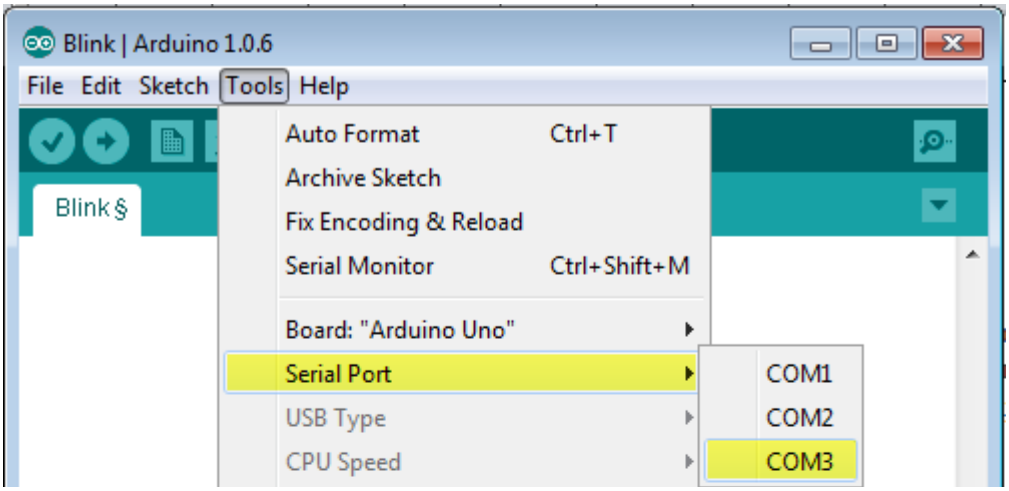
Go to Tools → Board and select your board.



Here, we have selected Arduino Uno board according to our tutorial, but we must select the name matching the board that you are using.

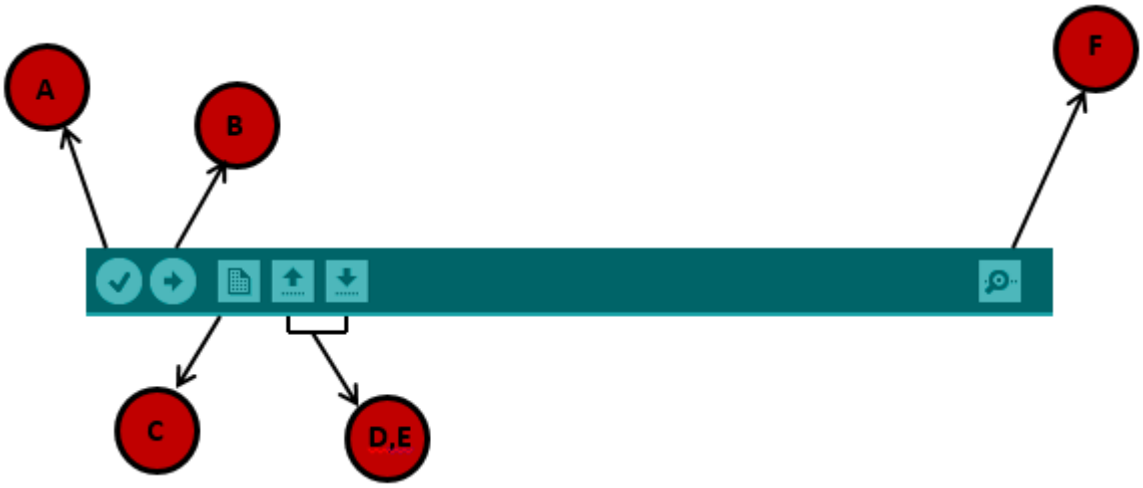
Step 6 – Select serial port.

Select the serial device of the Arduino board. Go to Tools → Serial Port menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, we can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.



Step 7 – Upload the program to your board.

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.



A– Used to check if there is any compilation error.

B– Used to upload a program to the Arduino board.

C– Shortcut used to create a new sketch.

D– Used to directly open one of the example sketch.

E– Used to save your sketch.

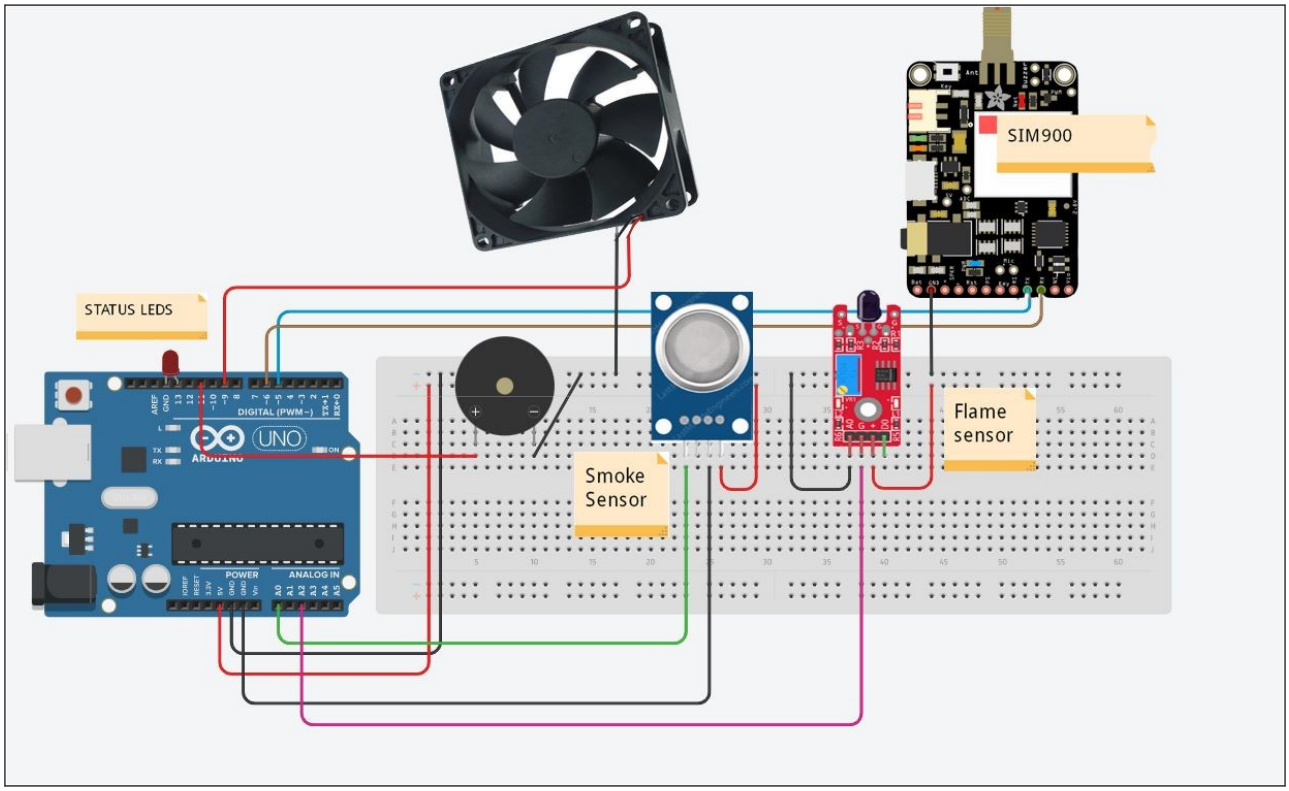
F– Serial monitor used to receive serial data from the board and send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

5.5. HARDWARE ASSEMBLY

Hardware assembly mainly includes connecting specific parts of the Arduino uno to the specific components. To make our project efficient we have used smoke sensor as well as fire sensor provided with the help of buzzer which will indicate the smoke and fire and one led to assure the working of the flame sensor .We connect a exhaust fan with out project which will turn on it the smoke sensor sense any smoke. To provide more safety we use call and sms alert in our project which is implemented using SOM900A GSM module. The connections are done part by part. First it comes to the flame sensor connections which detects any kind of flame produced nearby. The A0 pin of flame is connected to the A2 pin of arduino as it will display the extent of flame generated digitally. The GND of flame sensor is connected to the GND of arduino. On the other hand the VCC of flame sensor and arduino are connected. On the next part it comes to MQ2(SMOKE SENSOR) sensor assembling,the extent of smoke generated analogically as output. The A0 pin of MQ2 is connected to the A0 pin of arduino. Both the ground and VCC of MQ2 are connected to arduino. Thirdly it comes to led. The VCC of led is linked to 13 pin of arduino . Fourthly , buzzer which will act as alarm for any fire and smoke caught nearby. The VCC of buzzer is connected to 11 pin of arduino. The ground of flame sensor, buzzer and led are connected accordingly. The VCC and ground of the exhaust fan is connected with the pin 9 and ground of Arduino respectively.The TX, RX and ground of gsm module is connected with the pin 5, 6 and ground of Arduino respectively. For the flame and smoke sensor the thresh hold is set to 200 so that it can detect any nearby flame and smoke and gets activated.

5.6. PROPOSED DIAGRAM FOR THE PROTOTYPE

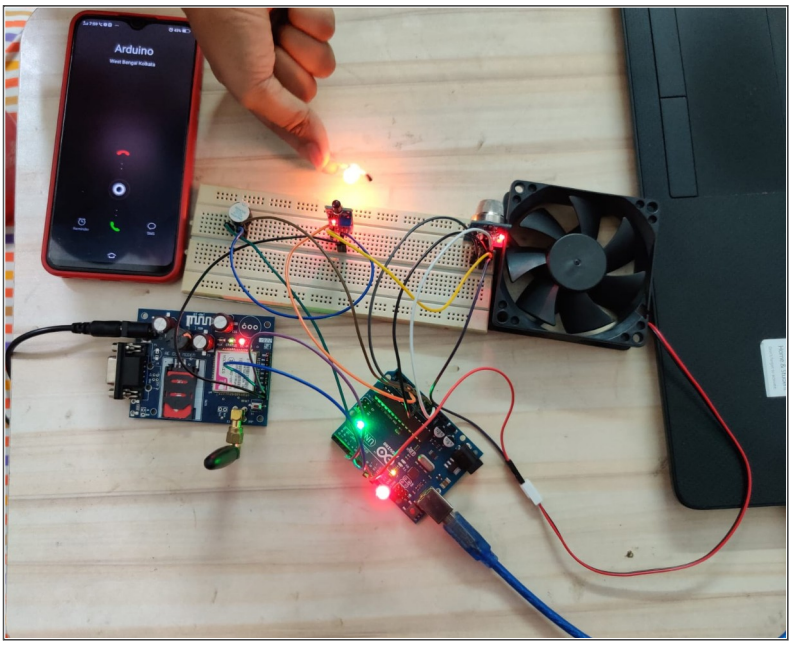


RESULT AND FUTURE SCOPE

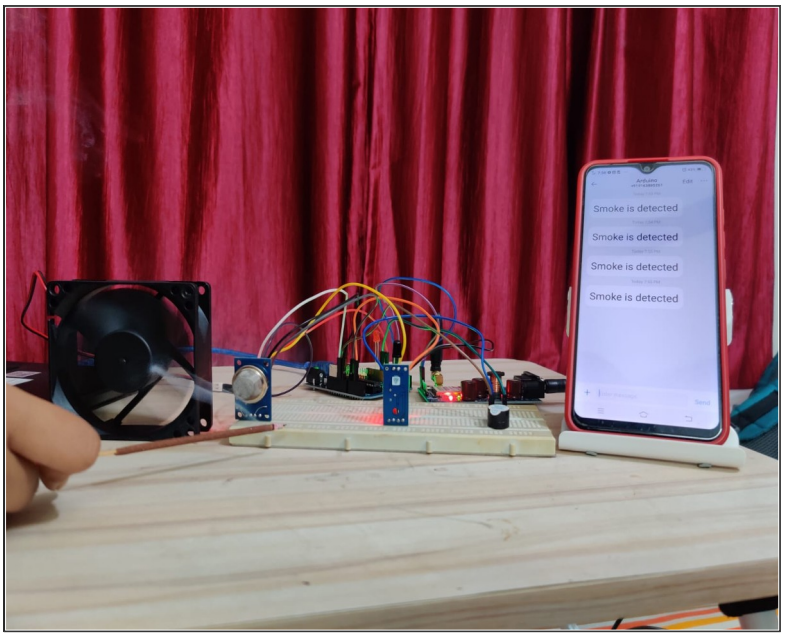
6.1. RESULTS

The experimental model was made according to the circuit diagram and the results were as expected. When the fire and smoke is detected the smoke and flame sensor are activated as a result the led lights turn on and buzzer starts ringing alerting the fire probed nearby. As a result when the smoke is detected we get a message and when the system detected fire we get a phone call through GSM module .

FOR FIRE :



FOR SMOKE:



6.2. LIMITATION

Our project runs well and good with proper power connection. In case power runs out then our project will fail to execute. In case user is in a place where network connection is poor then the call and message alert feature will not work. The sensors fitted in this prototype for sensing fire and smoke can only cover a particular area. As the power of the sensors used in this prototype model is very low they cannot cover a big area.

6.3. FURTHER ENHANCEMENT AND FUTURE SCOPE

- Fire accidents can be controlled to a great extent in a places such as forests,home,colleges industries,trains and some other public places.
- Fire accidents leads to death of excess of people,by using this technique we can save those life easily.
- To detect the chain smokers (which are hazardous to health).
- The future will be with the detection more for the products of combustion such as carbon monoxide, carbon dioxide,sulfur dioxide,nitrogen oxides in addition to heat and particulate matter.

6.4. CONCLUSION

- It is most efficient ,simple and less complex circuit which can be designed as small equipment ,which eases the portability.
- It can be used to provide security at house where storage and protection of various types of goods is main concern and works home security too.
- The fast automated fire alert notification between these units will help to save more people's lives and reduce physical losses that may be caused by the fire or the gas leakage.
- The important of these it gives sms and call when fire attacks to our home.

APPENDIX A

SOURCE CODE

- **SOURCE CODE :**

```
#include <SoftwareSerial.h>

SoftwareSerial mySerial(5, 6);
#define FAN 9
#define smokepin A1
const int ledpin=13;
const int buzpin=11;
const int flamepin=A2;
const int threshold=430;// sets threshold value for flame sensor
int flamesensvalue=0; // initialize flamesensor reading
float sensorValue;

void setup() {
Serial.begin(9600);
mySerial.begin(9600);

pinMode(ledpin,OUTPUT);
pinMode(flamepin,INPUT);
pinMode(smokepin,INPUT);
pinMode(buzpin,OUTPUT);
pinMode(FAN, OUTPUT);
Serial.println("MQ2 warming up!");
delay(20000); // allow the MQ2 to warm up
}

void loop() {
  Serial.print("smoke Sensor Value: ");
  Serial.println(analogRead(smokepin));
  Serial.print("flame Sensor Value: ");
  Serial.println(analogRead(flamepin));
```

```
flamesensvalue=analogRead(flamepin);
noTone(buzpin);
if (flamesensvalue<=threshold) {
digitalWrite(ledpin,HIGH);
digitalWrite(buzpin, HIGH);
SendCall();
delay(1000);
}

else{
digitalWrite(ledpin,LOW); //turns led off led and buzzer
noTone(buzpin);
}
noTone(buzpin);
if((analogRead(smokepin)>230))
{
  digitalWrite(buzpin, HIGH);
  analogWrite(FAN, 100);
  SendMessage();
  delay(2000); // wait 2s for next reading
}

else
{
  noTone(buzpin);
  analogWrite(FAN, 0);
}
delay(2000);

}

void SendMessage()
{
  Serial.println("initializing...");  //Message sent to Mobile

  mySerial.println("AT+CMGF=1");    //Sets the GSM Module in Text Mode
  updateSerial();
  mySerial.println("AT+CMGS=\"+918777218032\""); // Replace x with mobile number
```

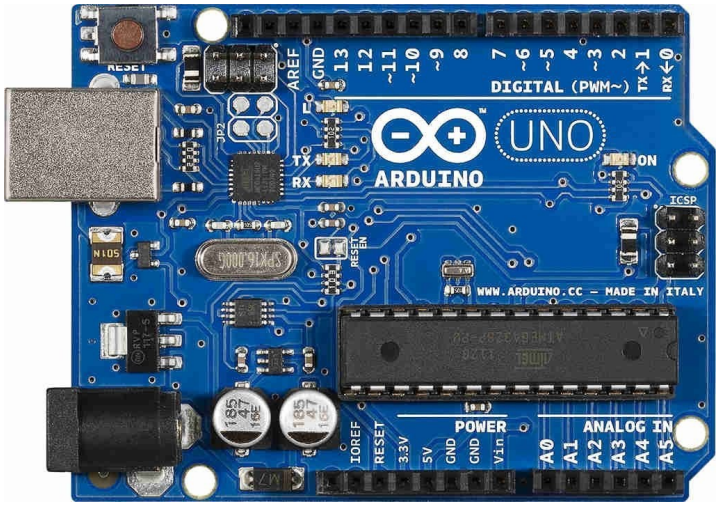
```
  updateSerial();
  mySerial.println("Smoke is detected");// The SMS text you want to send
  updateSerial();
  mySerial.write(26);

}

void SendCall()
{
  Serial.println("initializing...");  //Message sent to Mobile
  delay(1000);
  mySerial.println("AT");    //Sets the GSM Module in Text Mode
  updateSerial();
  mySerial.println("ATD+ +919547976339;"); // Replace x with mobile number
  updateSerial();
  delay(20000);
  mySerial.println("ATH");// The SMS text you want to send
  updateSerial();

}

void updateSerial()
{
  delay(500);
  while (Serial.available())
  {
    mySerial.write(Serial.read());//Forward what Serial received to Software Serial Port
  }
  while(mySerial.available())
  {
    Serial.write(mySerial.read());//Forward what Software Serial received to Serial Port
  }
}
```

APPENDIX B

DATASHEETS

Handson Technology

ATMega328P



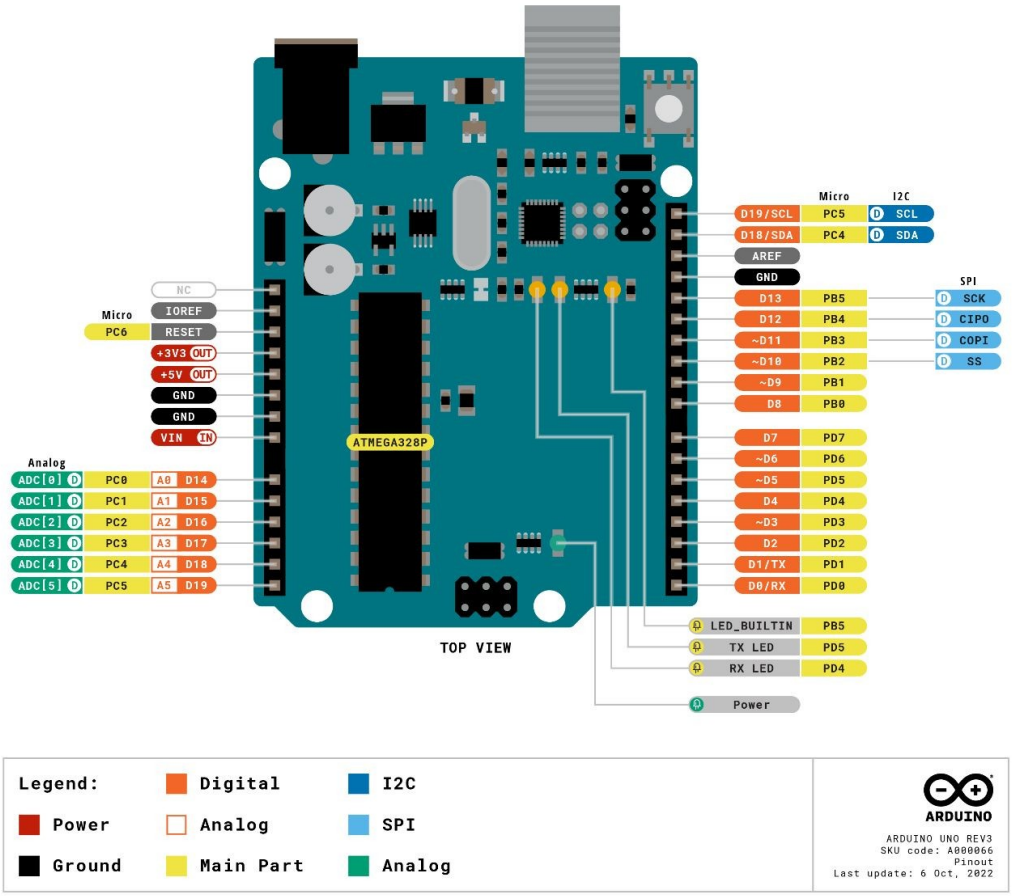
The Arduino UNO R3 is the perfect board to get familiar with electronics and coding. This versatile microcontroller is equipped with the well-known ATmega328P and the ATMega 16U2 Processor.

This board will give you a great first experience within the world of Arduino.ATmega328P has 14 digital input/output pins, 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header and a reset button.

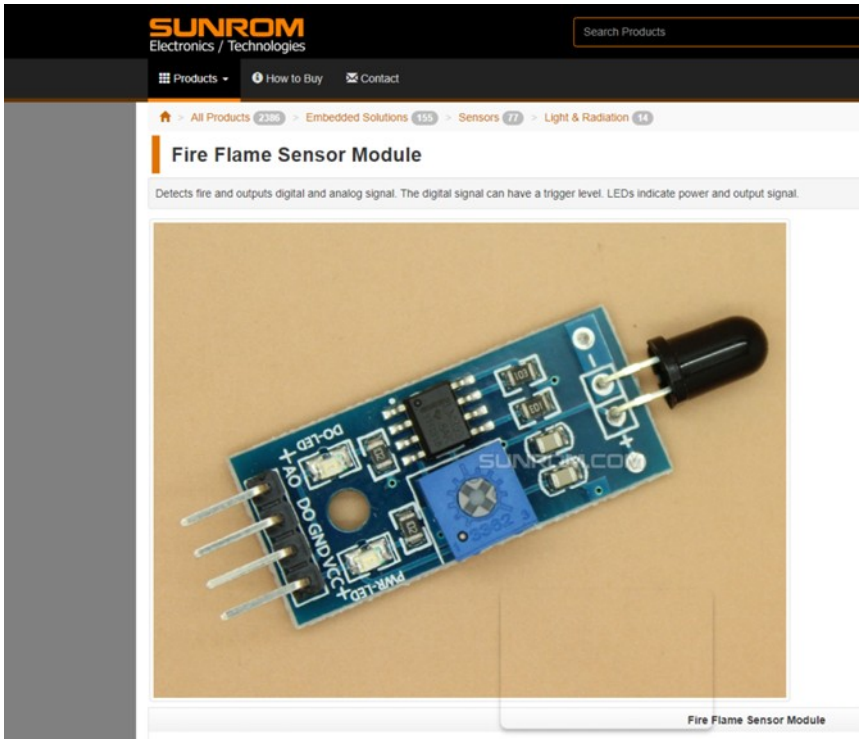
8.1.1. Specification

- **Memory**
  - AVR CPU at up to 16 MHz
  - 32KB Flash
  - 2KB SRAM
  - 1KB EEPROM
- **Security**
  - Power On Reset (POR)
  - Brown Out Detection (BOD)
- **Peripherals**
  - 2x 8-bit Timer/Counter with a dedicated period register and compare channels
  - 1x 16-bit Timer/Counter with a dedicated period register, input capture and compare channels
  - 1x USART with fractional baud rate generator and start-of-frame detection
  - 1x controller/peripheral Serial Peripheral Interface (SPI)
  - 1x Dual mode controller/peripheral I2C
  - 1x Analog Comparator (AC) with a scalable reference input
  - Watchdog Timer with separate on-chip oscillator
  - Six PWM channels
  - Interrupt and wake-up on pin change
- **ATMega16U2 Processor**
  - 8-bit AVR® RISC-based microcontroller
- **Memory**
  - 16 KB ISP Flash
  - 512B EEPROM
  - 512B SRAM debugWIRE interface for on-chip debugging and programming
- **Power**
  - 2.7-5.5 volts

8.1.2. Pin Details



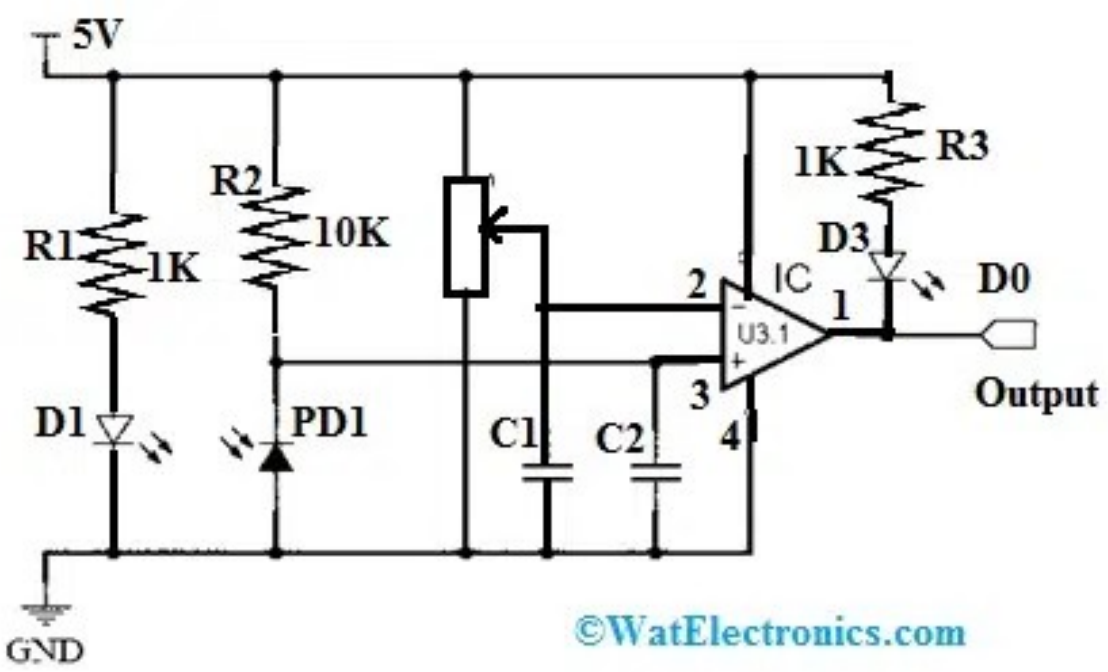
8.2. FLAME SENSOR



8.2.1. Specification

- Operating Voltage : 3.3V TO 5V DC
- Operating Current : 15ma
- Output Digital : 0v to 5v, adjustable trigger level from preset
- Output Analog : 0v to 5v based on infrared radiation from fire flame falling on the sensor
- LEDs indicating output and power
- PCB size : 3.2cm x 1.4cm
- LM393 based design

8.2.2. Board Schematic



8.2.3. Pin Details

- VCC : 3.3V to 5V DC
- GND : Ground
- DO: Digital Output
- AO : Analog Output

8.2.4. Working principle

Flame sensors use UV (Ultraviolet) or IR (Infra-Red) or UV-IR technology to identify flames below a second. These sensors react to a detected flame based on the installation, although it includes sounding an alarm, disabling a fuel line & activating a fire control system.

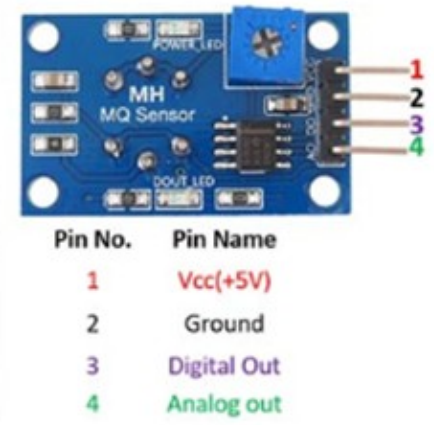
The flame sensor with UV technology works by simply sensing the UV radiation. Generally, all fires generate UV radiation at the ignition point so, in case of a fire, the sensor would become alert of it & generate a series of pulses that are changed by detector electronics and gives an alarm output.
The flame sensor with IR technology works by checking the IR spectral band for particular ornamentation that hot gases emit. But, this kind of device needs a flickering movement of the flame. Generally, infrared radiation is not only generated by flames but also radiated from lamps, ovens, etc. So, there is a high risk of a fake alarm
The sensor with UV-IR is capable of detecting both UV & IR radiations, thus it possesses both sensors. The combined sensor has a better rejection capacity for false alarms as compared to a separate UV/IR detector.

8.2.5. Applications

- Hydrogen stations
- Industrial heating
- Fire detection
- Fire alarm
- Fire fighting robot
- Drying systems
- Industrial gas turbines
- Domestic heating systems
- Gas-powered cooking devices

8.3. MQ2 Gas Sensor

MQ2 Gas sensor

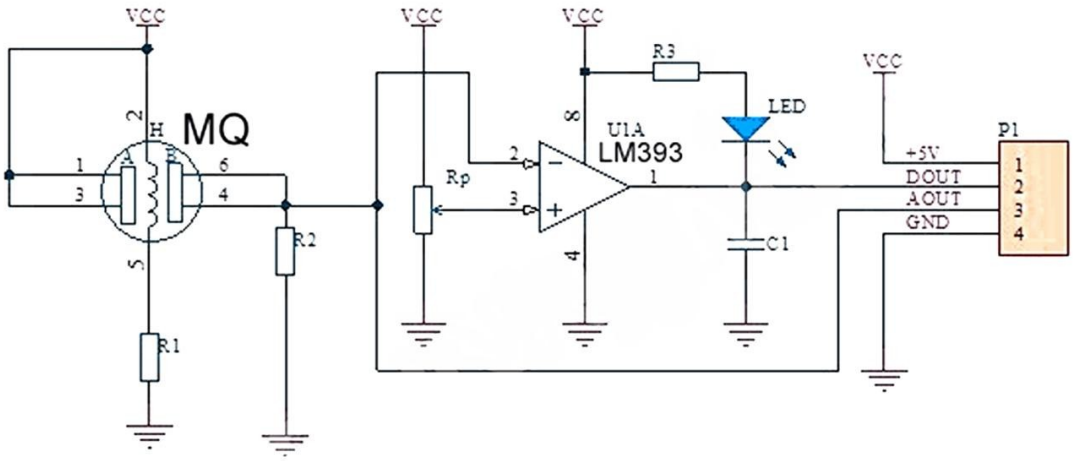


Pin No.	Pin Name
1	Vcc(+5V)
2	Ground
3	Digital Out
4	Analog out

8.3.1. Specification

- Operating Voltage : +5v
- Can be used to measure or detect LPG, Alcohol , Propane.Hydrogen, CO and even Methane
- Analog Output Voltage : 0V to 5V
- Digital Output Voltage : 0V or 5V (TTL Logic)
- Preheat dutation : 20 seconds
- Can be used as a Digital or Analog sensor
- The Sensitivity of Digital pin can be varied using the potentiometer

8.3.2. Board Schematic

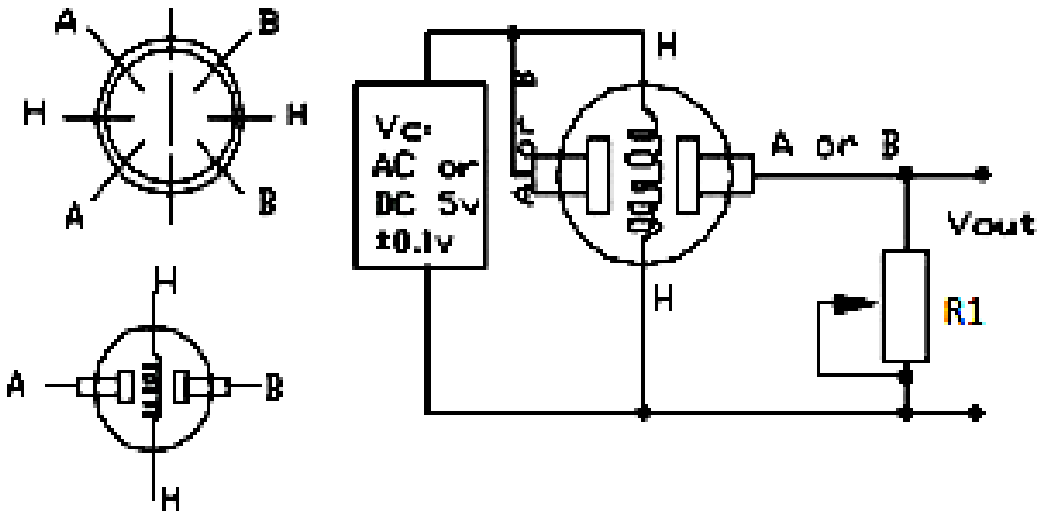


8.3.3. Pin Configuration

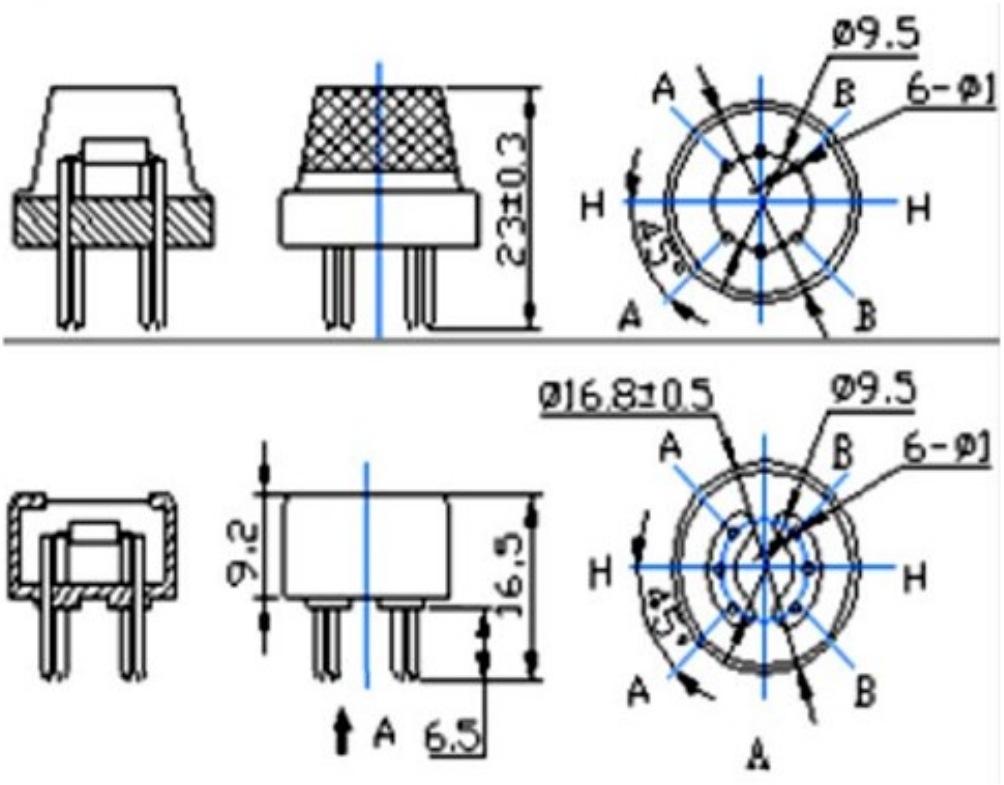
Pin no.	Pin Name	Description
1	Vcc	This pins power the module,typically the operating voltage is +5v
2	Ground	Used to connect the module to system ground
3	Digital output	You can also use this sensor to get digital output from this pin ,by setting a threshold value using the potentiometer
4	Analog output	This pin outputs 0-5v analog voltage based on the intensity of the gas

8.3.4. Working principle

The MQ2 has an electrochemical sensor, which changes its resistance for different concentrations of varied gasses. The sensor is connected in series with a variable resistor to form a voltage divider circuit (Fig 1), and the variable resistor is used to change sensitivity. When one of the above gaseous elements comes in contact with the sensor after heating, the sensor's resistance change. The change in the resistance changes the voltage across the sensor, and this voltage can be read by a microcontroller. The voltage value can be used to find the resistance of the sensor by knowing the reference voltage and the other resistor's resistance. The sensor has different sensitivity for different types of gasses.



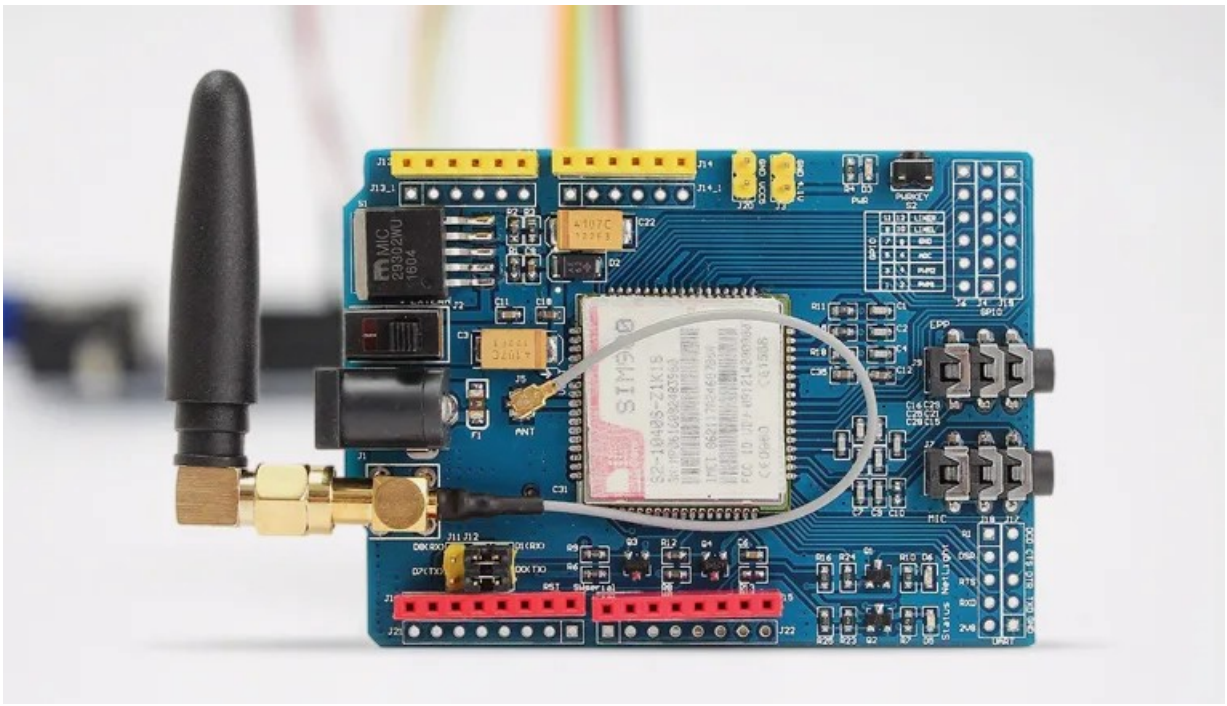
8.3.6. 2d Model of MQ-2 Gas Sensor



8.3.5. Applications

- Detects or measure Gases like LPG, Alcohol, Propane, Hydrogen, CO and even methane.
- Air quality monitor
- Safety standard maintenance
- Maintaining environment standards in hospitals

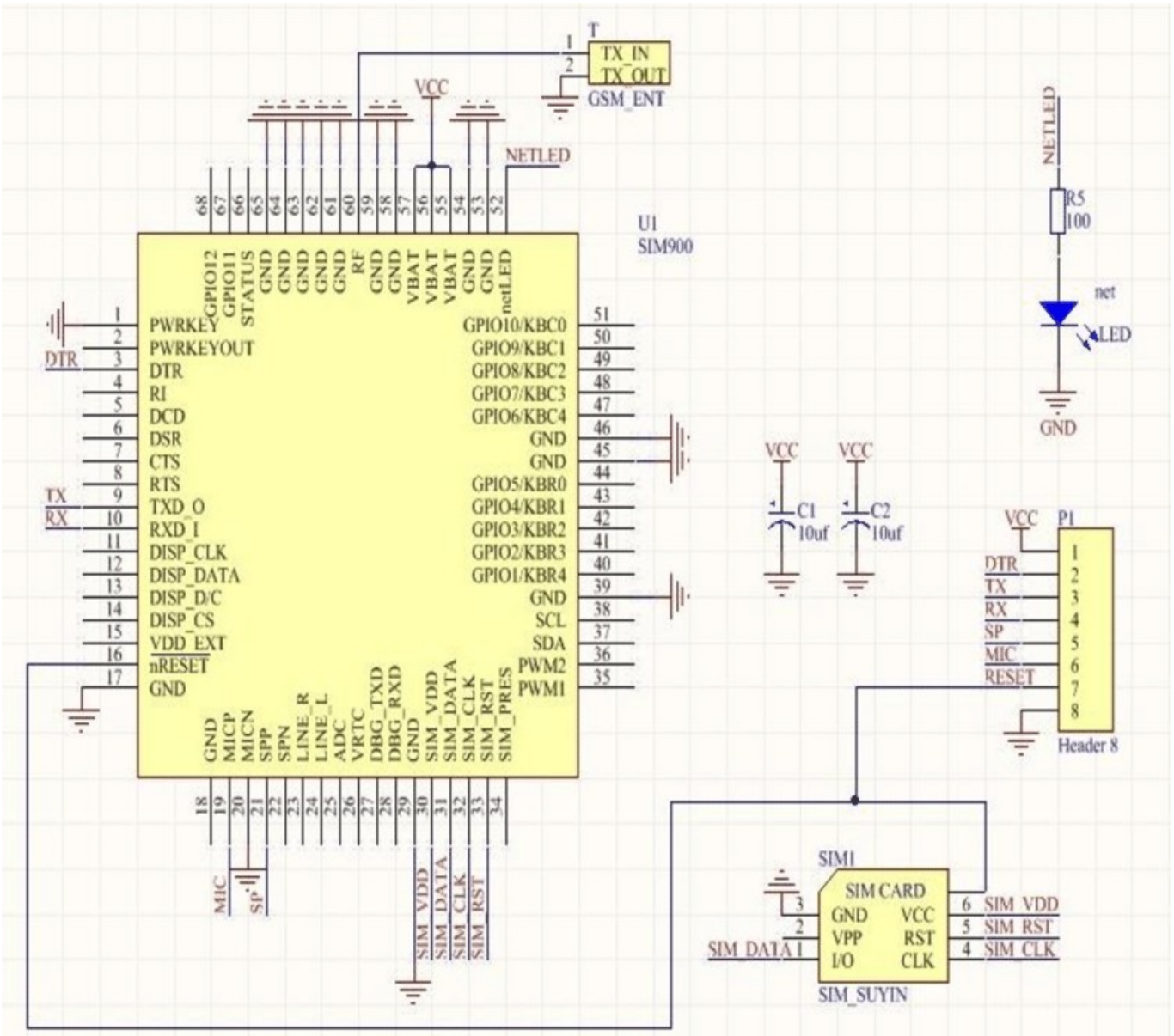
8.4. GSM SIM900a



8.4.1. Specification

- Single supply voltage: 3.4V – 4.5V
- Power saving mode: Typical power consumption in SLEEP mode is 1.5mA
- Frequency bands:SIM900A Dual-band: EGSM900, DCS1800. The SIM900A can search the two frequency bands automatically. The frequency bands also can be set by AT command.
- GSM class: Small MS
- GPRS connectivity:GPRS multi-slot class 10 (default) , GPRS multi-slot class 8 (option)
- Transmitting power: Class 4 (2W) at EGSM 900, Class 1 (1W) at DCS 1800
- Operating Temperature: -30°C to +80°C
- Storage Temperature: -5°C to +90°C
- DATA GPRS: download transfer max is 85.6KBps, Upload transfer max 42.8KBps
- Supports CSD, USSD, SMS, FAX
- Supports MIC and Audio Input

8.4.2. Board Schematic



8.4.3. PIN CONFIGURATION

Pin no.	Pin Name	Description
1	PWRKEY	Voltage input for PWRKEY. PWRKEY should be pulled low to power on or power off the system. The user should keep pressing the key for a short time when power on or power off the system because the system need margin time in order to assert the software.
2	PWRKEY_OUT	Connecting PWRKEY and PWRKEY_OUT for a short time then release also can power on or power off the module.
3	DTR	Data terminal Ready [Serial port]
4	RI	Ring indicator [Serial port]
5	DCD	Data carry detect [Serial port]
6	DSR	Data Set Ready [Serial port]
7	CTS	Clear to send [Serial port]
8	RTS	Request to send [Serial port]
9	TXD	Transmit data [Serial port]
10	RXD	Receive data [Serial port]
11	DISP _CLK	Clock for display [Display interface]
12	DISP_DATA	Display data output [Display interface]
13	DISP _D/C	Display data or command select [Display interface]
14	DISP _CS	Display Enable [Display interface]
15	VDD_EXT	2.8V output power supply
16	NRESET	External reset input
17,18,29,39,45,	GND	Ground

46,53,54,58,59, 61,62,63,64,65		
19	MIC_P	Microphone Positive
20	MIC_N	Microphone Negative
21	SPK_P	Speaker Positive
22	SPK_N	Speaker Negative
23	LINEIN_R	Right Channel input [External line inputs are available to directly mix or multiplex externally generated analog signals such as polyphonic tones from an external melody IC or music generated by an FM tuner IC or module.]
24	LINEIN_L	Left Channel Input
25	ADC	General purpose analog to digital converter.
26	VRTC	Current input for RTC when the battery is not supplied for the system. Current output for backup battery when the main battery is present and the backup battery is in low voltage state.
27	DBG_TXD	Transmit pin [Serial interface for debugging and firmware upgrade]
28	DBG_RXD	Receive pin [Serial interface for debugging and firmware upgrade]
30	SIM_VDD	Voltage supply for SIM card
31	SIM_DATA	SIM data output
32	SIM_CLK	SIM clock
33	SIM_RST	SIM reset
34	SIM_PRESENCE	SIM detect

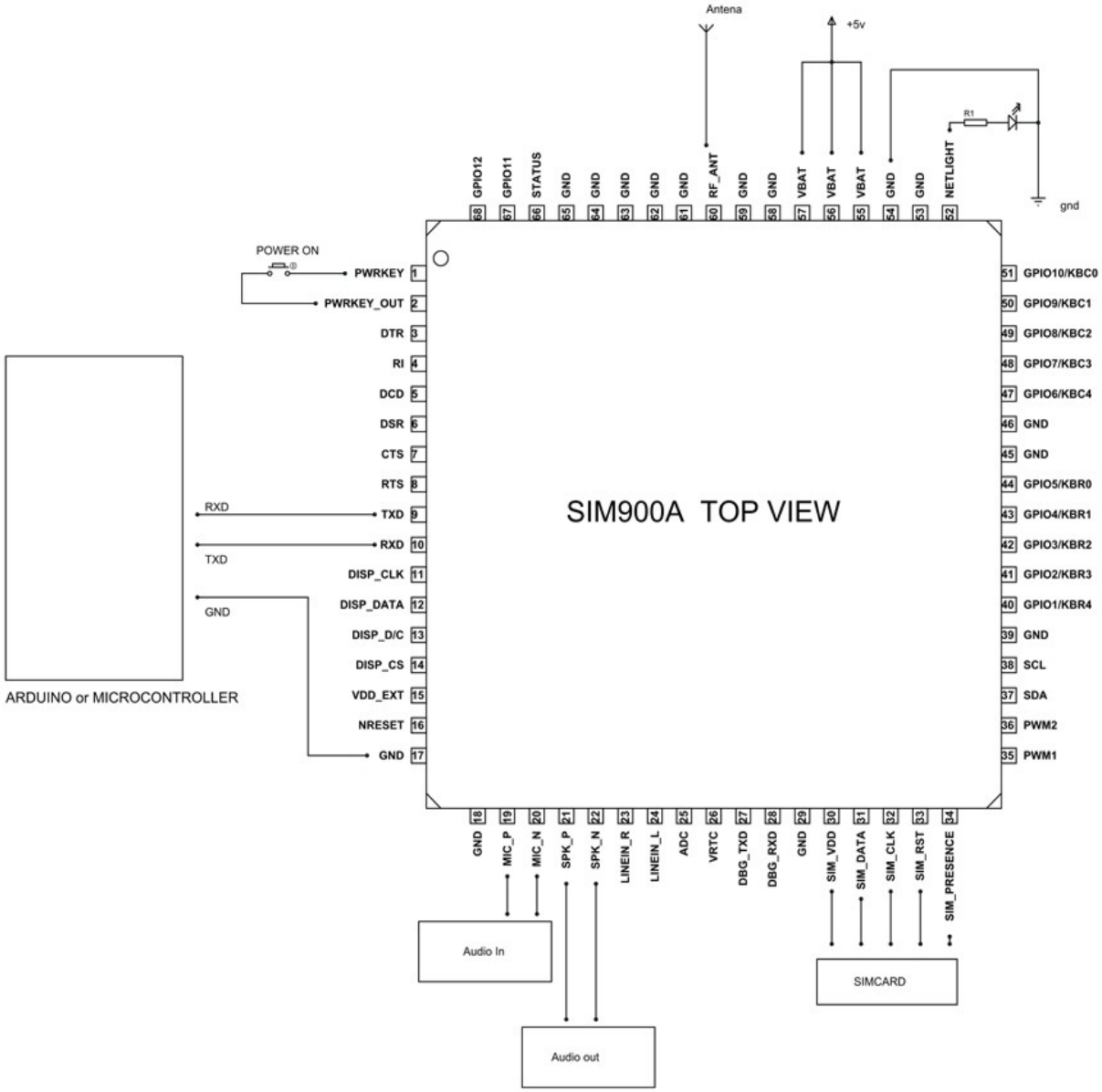
35	PWM1	PWM Output
36	PWM2	PWM Output
37	SDA	Serial Data [I2C]
38	SCL	Serial Clock [I2C]
40,41,42,43,44 & 47,48,49,50,51	KBR0 to KBR4 & KBC4 to KBC0	Keypad interface [ROWS & COLUMNS]
52	NETLIGHT	Indicate net status
55,56,57	VBAT	Three VBAT pins are dedicated to connect the supply voltage. The power supply of SIM900A has to be a single voltage source of VBAT= 3.4V to 4.5V. It must be able to provide sufficient current in a transmit burst which typically rises to 2A.
60	RF_ANT	Antenna connection
66	STATUS	Indicate working status
67	GPIO 11	General Purpose Input/output
68	GPIO 12	General Purpose Input/output

8.4.4. Working principle

As Shown below, the communication with this module is done through UART or RS232 Interface. The data is sent to the module or received from the module through UART interface.

The module is typically connected to +4.0V standard power supply. It can work on +4.5V regulated power and any higher voltage may damage the module. And the power source should be able to deliver a peak current of 2A. The UART interface is established as shown in figure. All you need to do is connect RXD of module to TXD of Arduino and TXD is connected to RXD of ARDUINO. The ground of controller and module must be connected for voltage reference. Here AUDIO IN is connected to MIC and AUDIO OUT is connected to a speaker or headset. And at last we need to connect a working GSM SIM card to the module. On powering the module, the NETLIGHT LED will blink periodically to state successful connection.

After all connections are done,we need to write a program for the microcontroller to exchange data with module. Since data exchange sequence between controller and module is really complex we will use libraries prewritten for the module. You can download libraries for controller or module through their websites. Using these libraries makes the communication easy. All you need to do is download these libraries and call them in programs. Once the header file is included, you can use simple commands in the program to tell the controller to send or receive data. The controller sends the data to the module through UART Interface based on protocol setup in libraries. The module sends this data to another GSM user using cellular network. If the module receives any data from the cellular network (or another GSM user) it will transmit it to controller through UART serial communication.This way we can use GSM900A module to establish cellular connection.



8.4.5. Applications

- Cellular Communication
- Robotics
- Mobile Phone Accessories
- Servers
- Computer Peripherals
- Automobile
- USB Dongles

REFERENCES

1. https://lastminuteengineers.com/electronics/arduino-projects/
2. https://learn.sparkfun.com/tutorials/what-is-an-arduino/all
3. https://www.electronicwings.com/arduino/sim900a-gsm-module-interfacing-with-arduino-uno
4. https://www.javatpoint.com/arduino-uno
5. https://www.electronicsforu.com/resources/gsm-module
6. Analytics for the Internet of Things (IoT): Intelligent analytics for your intelligent devices By Andrew Minteer
7. Enterprise IoT: Strategies and Best Practices for Connected Products and Services By Dirk Slama, Frank Phulmann, Jim Morrish, and Dr Rishi Bhatnagar