**AN IOT BASED FOREST FIRE DETECTION SYSTEM USING SENSORS , GSM & GPS MODULE**

**A PROJECT REPORT**

***Submitted by***

**SARAVANAN K (312820205033)**

**ARUN KUMAR G(312820205007)**

***in partial fulfillment for the award of the degree***

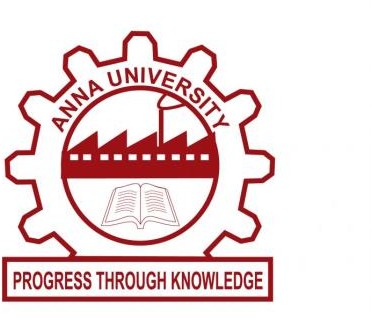
***of***

**BACHELOR OF TECHNOLOGY**

***In***

## INFORMATION TECHNOLOGY

**`AGNI COLLEGE OF TECHNOLOGY**

**THALAMBUR**

**ANNA UNIVERSITY: CHENNAI 600 025**

## MAY 2024

A picture containing text, clipart

Description automatically generatedLogo

Description automatically generatedANNA UNIVERSITY: CHENNAI 600 025

**BONAFIDE CERTIFICATE**

Certified that this project report **AN IOT BASED FOREST FIRE DETECTION SYSTEM USING SENSORS, GSM & GPS MODULE**

is the bonafide work of **SARAVANAN K (312820205033), ARUN KUMAR G (312820205007)** who carried out the project work under my supervision.

|  |  |
| --- | --- |
| **SIGNATURE** | **SIGNATURE** |
|  |  |
| Dr.S. GEERTHIK, M.E, Ph.D. | Mrs LEONILA, M.E, |
| **HEAD OF THE DEPARTMENT** | **SUPERVISOR** |
| Department of Information Technology | Department of Information Technology |
| Agni College Of Technology,  Thalambur, Chennai - 600130 | Agni College Of Technology,  Thalambur, Chennai - 600130 |

## Submitted for the project viva held on …………………………

**INTERNAL EXAMINER EXTERNAL EXAMINER**

## ACKNOWLEDGEMENT

We would like to express our deepest gratitude to the management of “**AGNI COLLEGE OF TECHNOLOGY”** and would like to thank our respected Principal **Dr. SRINIVASAN ALAVANDAR** for his words of inspiration and for providing necessary facilities to carry out our project work successfully.

We are immensely thankful to **Dr. S. GEERTHIK, M.E, Ph.D.,** Associate professor, Head of the Department, Information Technology for his words of wisdom and his constant source of inspiration.

I would like to offer our heartfelt thanks to our guide to **Mrs. LEONILA, M.E,** Assistant professor Department of Information Technology who molded us accordingly and gave valuable suggestion for completing our project work successfully.

We extended our warmest thanks to all the faculty members of our department for their assistance and we also thank all our friends who helped us in bringing out our project in good shape and form.

Finally, we express our sincere benevolence to our beloved parents for their perpetual encouragement and support in all endeavours.

## ABSTRACT

The design of a framework for tracking and preventing forest fires using IOT is presented in this project. The frequency and severity of forest fires continues to increase worldwide, there is an urgent need for technologies that will improve early detection and rapid response. This paper presents an IOT -based wildfire detection system designed to solve these problems. The system integrates a network of sensor nodes strategically placed in forest areas to create instant monitoring infrastructure. Sensor nodes are equipped with various environmental sensors such as temperature, humidity and smoke detectors to collect information about fire events. The nodes communicate wirelessly with the central IoT gateway, which integrates data and uses an IOT platform for data analysis and to notify. The IOT platform gathers different data’s it helps with the early stages of wildfires. Additionally, the system uses GSM module to provide accurate location information, allowing responders to quickly arrive at the emergency in a timely manner. The main goal of our device is to detect a forest fire and issue an alarm using IoT, Neo 6MGPS, and GSM, ESP32. The design of the system is made up of different types of sensors, these sensors play a major role in detecting the forest fire and used to alert the people nearby. Also, we are able to monitor the observed area in real-time via IOT webpage.

.

|  |  |  |
| --- | --- | --- |
|  | **TABLE OF CONTENTS** |  |
| **CHAPTER NO** | **TITLE** | **PAGE NO** |
|  |  |  |
|  | **ABSTRACT** | II |
|  | **LIST OF FIGURES** | IV |
|  | **LIST OF ABBREVIATIONS** | V |
| 1 | 1.1 Introduction | 1 |
|  | 1.2 Objective | 2 |
| 2 | **LITERATURE SURVEY** | 3 |
| 3 | **SYSTEM ANALYSIS** | 5 |
|  | 3.1 Existing system | 5 |
|  | 3.2 Proposed system | 5 |
|  | 3.3 Block diagram | 6 |
| 4 | **SYSTEM REQUIREMENTS** | 7 |
|  | 4.1 Hardware requirements | 7 |
|  | 4.2 Arduino uno | 7 |
|  | 4.2.1 Pin configuration | 10 |
|  | 4.2.2 Communication | 12 |
|  | 4.3 Node mcu | 13 |
|  | 4.3.1 Node mcu-board | 15 |
|  | 4.3.2 Specifications | 15 |
|  | 4.4 Fire sensor | 16 |
|  | 4.4.1 Specifications | 17 |
|  | 4.5 DHT11 Sensor | 17 |
|  | 4.5.1 Specifications | 19 |
|  | 4.6 Gas sensor | 20 |
|  | 4.6.1 Specifications | 22 |
|  | 4.6.2 Pin configuration | 23 |
|  | 4.7 GSM Modem | 25 |
|  | 4.7.1 Communication | 27 |
|  | 4.7.2 Working of GSM MODEM | 29 |
|  | 4.7.3 GSM modem Application | 30 |
|  | 4.8 GPS Module | 31 |
|  | 4.8.1 Neo 6m with Arduino | 33 |
|  | 4.8.2 Accuracy | 34 |
|  | 4.9 Software requirements | 35 |
|  | 4.9.1 Arduino IDE | 36 |
|  | 4.9.1.1 Working | 36 |
|  | 4.9.2 Proteus | 38 |
|  | 4.9.3 Language | 40 |
| 5 | **CODING** | 42 |
|  | 5.1 Arduino UNO | 42 |
|  | 5.2 IOT code | 48 |
| 6 | **RESULTS AND DISCUSSION** | 51 |
|  | 6.1 Output | 53 |
| 7 | **SUMMARY AND CONCLUSION** | 54 |
| 8 | **REFERENCES** | 57 |

|  |  |  |
| --- | --- | --- |
|  | **LIST OF FIGURES** |  |
| **FIGURE NO** | **TITLE NO** | **PAGE** |
|  |  |  |
| **3.1** | **Block diagram** | **6** |
| **3.2** | **Arduino UNO** | **8** |
| **4.1** | **NODE MCU (ESP8266)** | **15** |
| **4.2** | **Fire sensor** | **19** |
| **4.3** | **Temperature sensor** | **19** |
| **4.4** | **Gas sensor** | **21** |
| **4.5** | **GSM Module** | **26** |
| **4.6** | **GSM Network Diagram** | **26** |
| **4.7** | **GPS Module** | **32** |
| **4.8** | **GPS With Arduino** | **33** |
| **4.9** | **Arduino cable** | **36** |
| **4.10**  **4.11**  **4.12**  **6.1**  **6.2** | **Arduino IDE Code**  **IDE Port selection page**  **Uploading code over arduino**  **IOT webpage**  **Overall setup** | **37**  **37**  **38**  **52**  **53** |

## LIST OF ABBREVIATIONS

|  |  |
| --- | --- |
| GSM | GLOBAL SYSTEM FOR MOBILE COMMUNICATION |
| GPS | GLOBAL POSITIONING SYSTEM |
| IOT | INTERNET OF THINGS |
| CO2 | CARBONDIOXIDE |
| IR | INFRARED |
| IDE | INTERGRATED DEVELOPMENT ENVIRONMENT |
| AT | ATTENTION COMMAND |
| SMS | SHORT MESSAGE SERVICE |

**CHAPTER 1**

* 1. **INTRODUCTION**

Today, fire is one of the silent dangers among human society. Fire hazards can be fatal and denigrating for industrial and household security and also intimidate for human life. The best way to reduce these losses is to respond to the emergency situation as quick as possible, that need a standalone automatic fire detection system. This Fire alarm system provides the owner with the advantage of checking from distant location and taking immediate actions when an emergency message is received. For the demand of the market, it is easier to use and more safe than other type of fire alarm systems. It can use more easy and convenience. It can use whether the GSM line connection active. SMS based fire alarm system is very useful in remote locations where human interaction is limited. This proposed system can provide a safe, secure and efficient way for accidents. Using one of this detector low cost and quick-responsive fire/smoke detection and alarm system has been designed and implemented. The system is capable of sending alert messages via GSM network and activating siren at the premises. Due to the outbreak of fire that damage properties and life, there is need to locally design and construct of fire detector using GSM module, which can be building conveniently, offices, home also in industries which will be able to send text messages to the owner of the installed in premises. Furthermore, there is need to contribute to the economic growth of the nation and reduce the fire outbreak, while avoiding loss of properties. Lastly to put into practice the knowledge acquired in my field of studies. This paper shows how to implement step by step of GSM base fire alarm system.

The fire prevention technologies are constantly innovating along with the growing science and technology, because fire is a serious threat to human life and properties. The appearance of the fire is often accompanied by the generation of smoke. Actually in the early stages of the fire, a lot of smoke is produced due to the fuel temperature and combustion is not high enough. Smoke cannot be covered by a large area, and the mobility is relatively high. If the fire can be detected before the occurrence of fire and sent an early warning to the nearest fire station, the occurrence of fire and a large area spreading of fire can be avoided so as to reduce people's material and financial damage.

**1.2 OBJECTIVE**

Fire can make major hazards in this hectic world. All buildings and vehicles used in public transportation have fire prevention and fire protection systems due to the accelerated number in the fire incidents. Also, many of the firms conduct a mock fire drill in every occurrence of months to protect their employees from the fire. This would help them to understand what to do or what not to do when a fire situation happens. Forests are one of the main factors in balancing the ecology. It is very harmful when a fire occurs in a forest. But most of the time, the detection of forest fire happens when it spread over a wide region. Sometimes, it could not be possible to stop the fire. As a result, the damage of the environment is higher than predictable. The emission of large amount of carbon dioxide (CO2) from the forest fire damages the environment. As well as it would lead to complete disappearance of rare species in the world (Alkhatib, 2014). Also, it can make an impact on the weather, and this make major issues like earthquakes, heavy rains, floods and so on.

**CHAPTER 2**

**LITERATURE SURVEY**

**[1] Project Title:** A Smart Real Time Fire and Smoke Detection System.

**Authors:** 1Mr. Aneesh . A, 2 Mr. Austine Cyriac.

Fire plays an important role in our daily life, but it is also a serious threat to human life and properties. Along with the growing science and technology, fire prevention technologies are also constantly innovating. This system is designed to detect the fire and smoke at the early stage and notify the nearest fire station through a push notification. The notification contains the fire or smoke warning and the location information. Whenever a fire occurs, the fire sensor senses the fire at very early stage itself. In case of a smoke, the smoke sensor detects it and sends a notification saying there is a chance of fire which helps the fighters to subdue it before the fire causes damage at a greater extend. Tensilica ESP 8266 processor is used as the brain of the system. This processor contains a built in Wi-Fi and is available at a cheap cost

**[2] Project Title:** Enhanced Wireless Control System for Smoke and Fire Detection.

**Authors:** G. Rashedul Qayum, S.M Ehsanul Amin Md Kamrul Hussain

A smoke detector or smoke alarm is a device that detects smoke and issues an alarm to alert nearby people that there is a potential fire. Smoke alarms are self contained devices that incorporate a means of detecting a fire (smoke detector) and giving a warning (alarm). They are about the size of a hand and are normally fitted to the ceiling. They can detect fires in their early stages and give you those precious minutes to enable you and your family to leave your house in safety.

**[3] Project Title:** Microcontroller Based Fire Alarm System Using Sensory and Monitoring System

**Authors:** S. Chaudhary and N. K. Joshi

Fire alarm system plays an important role in maintaining and monitoring the safe of all kind environments and situations. However the usability of many existing fire alarm system is well known but could he produce with high cost. Subsequently, it is not affordable for the low income users. This paper will discuss the design and implementation of a fire alarm system using the microcontroller which is been produced with low cost and with effective outcome. The outcome of this fire alarm system is almost the same compared to the existing fire alarm system in the market which is been produced with higher cost.

**[4] Project Title:** Automatic Smoke Detector and Fire Alarm System.

**Authors:** J. Vyas and P. Modi

In this paper the proposed work is designed to monitor the smoke and heat and to activate the speaker by using the light dependent resistor(LDR) based on PIC microcontroller. It is comprised of a combination of electrical/electronic devices/equipment’s working together to detect the presence of fire and alert people through audio or visual medium after detection. These alarms may be activated from smoke detectors which, when detects fire. Then, it automatically operates a relay which can be used to switch on a motor which is started to pump the water to spray on fire. Test results from the proposed system show that the automatic fire alarm system achieves the design requirements. In this paper, the simulation work is carried out with PROTEUS software and programming has been done with C coding

## CHAPTER 3

**SYSTEM ANALYSIS**

## Existing system

Numerous solutions have been proposed and implemented for this problem. Most common systems used in field work are video surveillance systems. Video cameras are sensitive to smoke only in day time.

Fire sensitive cameras at night, using IR thermal imaging cameras for heat flux detecting and using backscattering of laser light, detect the smoke particles.

This fire alert system has a few limitations because of environmental conditions like dust particles, mist, shadows and so on. Another method is automated picture capturing of fires in forest. Capturing can be done by the cameras which are placed on top of towers.

A motor was introduced to give a coverage view on the forest and for its movement (Basu et al., 2018). Captured pictures are processed using program or MATLAB simulation and matching with references taken at beginning stage.

This alert system has limitation of false caution rate and visual cameras installed on towers are of high cost

## Proposed system

## 

Fire Security is primary concern for everyone. Individuals, industries, home appliances are taking general precaution about fire safety.  But forest fire monitoring always in less priority.

The main aim of this project is to continuously monitor the forest from fire accidents occur, the system should alert the siren and also inform immediately to the person concerned by using wireless communication system GSM.

This system totally eliminates a person who has to monitor the house or industry all the time. Here we proposed solution like forest fire monitoring system using GSM

## 3.3Block diagram

## 

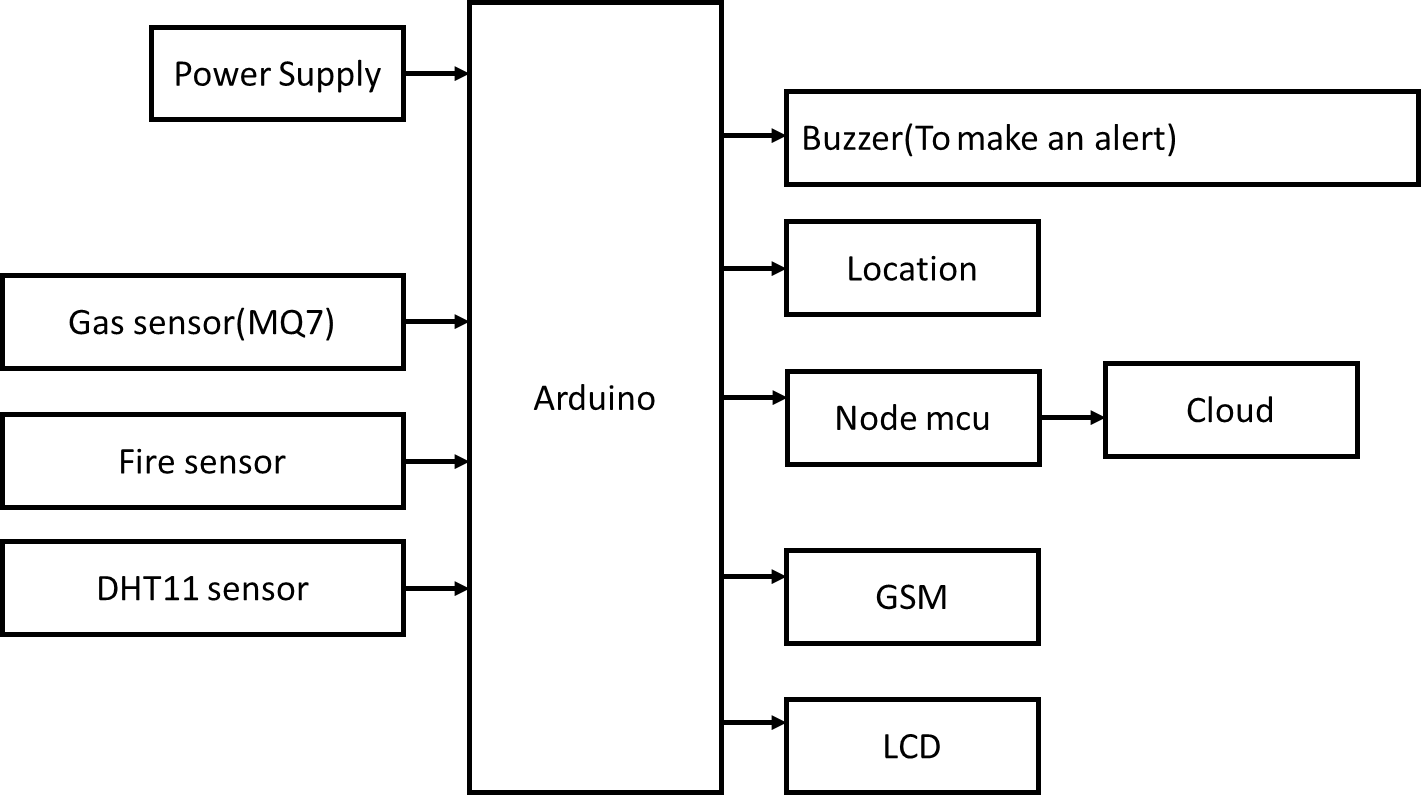
****

Fig 3.1 Block diagram

## CHAPTER 4

**SYSTEM REQUIREMENTS**

## HARDWARE REQUIREMENTS

## Arduino

## Node mcu

## Gas sensor(MQ-7)

## Fire sensor

## DHT-11 sensor

## LCD

## Buzzer

## GSM

## GPS

## 4.2 ARDUINO UNO AND ITS PROGRAMMING

Arduino is a tool for making computers that can sense and control more of the physical world than your desktop computer. It's an open-source physical computing platform based on a simple microcontroller board, and a development environment for writing software for the board.Arduino can be used to develop interactive objects, taking inputs from a variety of switches or sensors, and controlling a variety of lights, motors, and other physical outputs. Arduino projects can be stand-alone, or they can be communicate with software running on your computer. The boards can be assembled by hand or purchased preassembled; the open-source IDE can be downloaded for free.The Arduino programming language is an implementation of Wiring, a similar physical computing platform, which is based on the Processing multimedia programming environment.

****

## Fig 3.2 Arduino Uno

## The Arduino microcontroller is an easy to use yet powerful single board computer that has gained considerable traction in the hobby and professional market. The Arduino is open-source, which means hardware is reasonably priced and development software is free. This guide is for students in ME 2011, or students anywhere who are confronting the Arduino for the first time. For advanced Arduino users, prowl the web; there are lots of resources.This guide covers the Arduino Uno board (Spark fun DEV-09950, $29.95), a good choice for students and educators. With the Arduino board, you can write programs and create interface circuits to read switches and other sensors, and to control motors and lights with very little effort.

## The Duemilanove board features an Atmel ATmega328 microcontroller operating at 5 V with 2 Kb of RAM, 32 Kb of flash memory for storing programs and 1 Kb of EEPROM for storing parameters. The clock speed is 16 MHz, which translates to about executing about 300,000 lines of C source code per second. The board has 14 digital I/O pins and 6 analog input pins. There is a USB connector for talking to the host computer and a DC power jack for connecting an external 6-20 V power source, for example a 9 V battery, when running a program while not connected to the host computer. Headers are provided for interfacing to the I/O pins using 22 g solid wire or header connectors.

## The Arduino programming language is a simplified version of C/C++. If you know C, programming the Arduino will be familiar. If you do not know C, no need to worry as only a few commands are needed to perform useful functions. An important feature of the Arduino is that you can create a control program on the host PC, download it to the Arduino and it will run automatically. Remove the USB cable connection to the PC, and the program will still run from the top each time you push the reset button. Remove the battery and put the Arduino board in a closet for six months. When you reconnect the battery, the last program you stored will run. This means that you connect the board to the host PC to develop and debug your program, but once that is done, you no longer need the PC to run the program.

## The Arduino Uno is a microcontroller board based on the ATmega328.It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

## `4.2.1 PIN CONFIGURATION

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

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

The power pins are as follows:

**VIN**. The input voltage to the Arduino board when it's using an external power source (asopposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

**5V**. this pin outputs a regulated 5V from the regulator on the board. The board can besupplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board.

**3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50mA.

**GND**. Ground pins.

**IOREF.** This pin on the Arduino board provides the voltage reference with which themicrocontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs for working with the 5V or 3.3V.

**Memory**

The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM.

**Input and Output**

Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode(), digital Write( ), and digital Read( ) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull- up resistor (disconnected by default) of 20-50 kohms. In addition, some pins have specialized functions:

**Serial: 0 (RX) and 1 (TX). Used to receive (RX) and Trans mit (TX) TTL serial data.**

These pins are connected to the corresponding pins of the ATmega8U2 USB-to- TTL Serial chip.

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

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

**SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication using the SPI library.

**LED: 13.** There is a built- in LED connected to digital pin 13. When the pin is HIGH value,the LED is on, when the pin is LOW, it's off. The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analog Reference () function. Additionally, some pins have specialized functionality:

**TWI: A4 or SDA pin and A5 or SCL pin.** Support TWI communication using the Wire library.

There are a couple of other pins on the board:

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

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

**4.2.2 Communication**

Microcontrollers depend on a host computer for developing and compiling programs. The software used on the host computer is known as an integrated development environment, or IDE. For the Arduino, the development environment is based on the open source Processing platform (www.processing.org) which is described by its creators as a “programming language and environment for people who want to program images, animation, and interactions.“ The Arduino programming language leverages an open source project known as Wiring (wiring.org.co). The Arduino language is based on good old- fashioned C. If you are unfamiliar with this language, don’t worry; it’s not hard to learn, and the Arduino IDE provides some feedback when you make mistakes in your programs.

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a inf file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A Software Serial library allows for serial communication on any of the Uno's digital pins. The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation for details. For SPI communication, use the SPI library.

As you go through the list of programming statements available in the Arduino IDE (choose Help->Reference), you might think there isn’t much power for doing t hings like running servos, operating stepper motors, reading potentiometers, or displaying text on an LCD. Like most any language based on C, the Arduino supports the notion of “libraries” code.

Repositories that extend core programming functionality. Libraries let you re- use code without having to physically copy and paste it into all your programs. The standard Arduino software installation comes with several libraries you may use, and you can download others from the Arduino support pages and from third-party websites that publish Arduino library code.

**4.3 NODE MCU**

NodeMCU is an open-source Lua based firmware and development board specially targeted for IoT based Applications. It includes firmware that runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module Programming NodeMCU ESP8266 with Arduino IDE. The NodeMCU Development Board can be easily programmed with Arduino IDE since it is easy to use.

Programming NodeMCU with the Arduino IDE will hardly take 5-10 minutes. All you need is the Arduino IDE, a USB cable and the NodeMCU board itself. You can check this Getting Started Tutorial for NodeMCU to prepare your Arduino IDE for NodeMCU. NodeMCU is an open source firmware for which open source prototyping board designs are available. The name "NodeMCU" combines "node" and "MCU" (micro-controller unit).[8] The term "NodeMCU" strictly speaking refers to the firmware rather than the associated development kits.

The firmware uses the Lua scripting language. The firmware is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. It uses many open source projects, such as lua-cjson[9] and SPIFFS.[10] Due to resource constraints, users need to select the modules relevant for their project and build a firmware tailored to their needs. Support for the 32-bit ESP32 has also been implemented.

The prototyping hardware typically used is a circuit board functioning as a dual in-line package (DIP) which integrates a USB controller with a smaller surface-mounted board containing the MCU and antenna. The choice of the DIP format allows for easy prototyping on breadboards. The design was initially based on the ESP-12 module of the ESP8266, which is a Wi-Fi SoC integrated with a Tensilica Xtensa LX106 core, widely used in IoT applications.

As Arduino.cc began developing new MCU boards based on non-AVR processors like the ARM/SAM MCU and used in the Arduino Due, they needed to modify the Arduino IDE so that it would be relatively easy to change the IDE to support alternate toolchains to allow Arduino C/C++ to be compiled for these new processors. They did this with the introduction of the Board Manager and the SAM Core. A "core" is the collection of software components required by the Board Manager and the Arduino IDE to compile an Arduino C/C++ source file for the target MCU's machine language. Some ESP8266 enthusiasts developed an Arduino core for the ESP8266 WiFi SoC, popularly called the "ESP8266 Core for the Arduino IDE".[17] This has become a leading software development platform for the various ESP8266-based modules and development boards, including NodeMCUs. NodeMCU is an open-source LUA based firmware developed for the ESP8266 wifi chip. By exploring functionality with the ESP8266 chip, NodeMCU firmware comes with the ESP8266 Development board/kit i.e. NodeMCU Development board.

**4.3.1 NodeMCU Development Board/kit v0.9 (Version1)**

Since NodeMCU is an open-source platform, its hardware design is open for edit/modify/build. NodeMCU Dev Kit/board consist of ESP8266 wifi enabled chip. The ESP8266 is a low-cost Wi-Fi chip developed by Espressif Systems with TCP/IP protocol. For more information about ESP8266, you can refer to the ESP8266 WiFi Module. There is Version2 (V2) available for NodeMCU Dev Kit i.e. NodeMCU Development Board v1.0 (Version2), which usually comes in black coloured PCB.

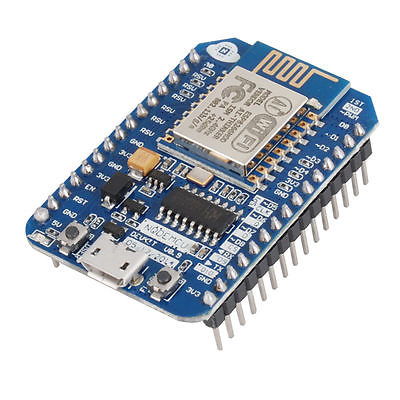


Fig 4.1 Node mcu

**4.3.2 NodeMCU ESP8266 Specifications & Features**

* Microcontroller: Tensilica 32-bit RISC CPU Xtensa LX106
* Operating Voltage: 3.3V
* Input Voltage: 7-12V
* Digital I/O Pins (DIO): 16
* Analog Input Pins (ADC): 1
* UARTs: 1
* SPIs: 1
* I2Cs: 1
* Flash Memory: 4 MB
* SRAM: 64 KB
* Clock Speed: 80 MHz
* USB-TTL based on CP2102 is included onboard, Enabling Plug n Play
* PCB Antenna
* Small Sized module to fit smartly inside your IoT projects

## 4.4 Fire Sensor Module

## Flame sensor is the most sensitive to ordinary light that is why its reaction is generally used as flame alarm purposes. This module can detect flame or wavelength in 760 nm to 1100 nm range of light source. Small plate output interface can and single-chip can be directly connected to the microcomputer IO port. The sensor and flame should keep a certain distance to avoid high temperature damage to the sensor. The shortest test distance is 80 cm, if the flame is bigger, test it with farther distance. The detection angle is 60 degrees so the flame spectrum is especially sensitive. The detection angle is 60 degrees so the flame spectrum is especially sensitive.

## 

## 

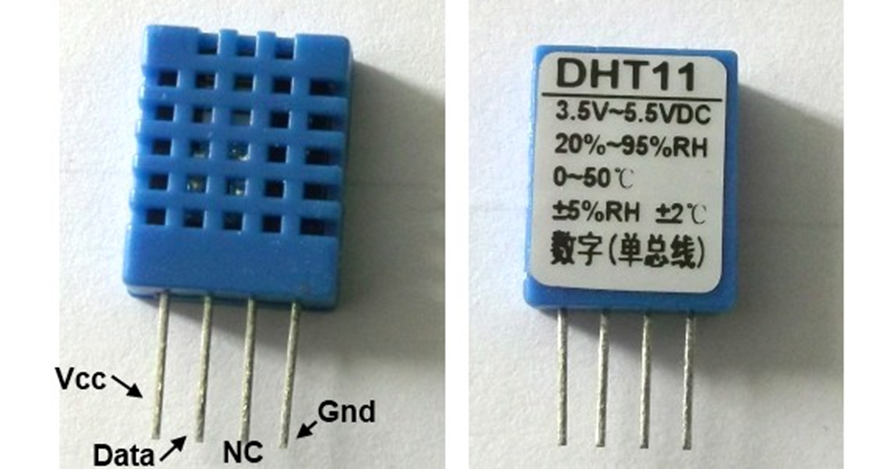
## Fig 4.2 Fire sensor

**4.4.1 Specifications**

* On-board LM393 voltage comparator chip and infrared sensing probe.
* Support 5V/3.3V voltage input.
* On-board signal output indication, output effective signal is high level, and the same time the indicator light up, output signal can directly connect with microcontroller IO.
* Signal detection sensitivity can be adjusted.
* Reserved a line voltage compare circuit (P3 is leaded out).
* PCB size: 30(mm) x15(mm).

**4.5 DHT11 sensor**

Measurement and control of temperature and relative humidity finds applications in numerous areas. These days devices are available which have both temperature and humidity sensors with signal conditioning, ADC, calibration and communication interface all built inside them. The use of such smart sensors greatly simplify the design and reduces the overall cost. We discussed in past about Humidity and temperature measurements with Sensirion’s SHT1x/SHT7x sensors. These sensors are capable of measuring both temperature and relative humidity and provide fully calibrated digital outputs. While SHT1x/SHT7x are very accurate sensors, they are still expensive for hobbyists use. This articles discusses the DHT11 sensor which also provides calibrated digital outputs for temperature and humidity but is relatively lot cheaper than the Sensirion sensors. The DHT11 sensor uses a proprietary 1-wire protocol which we will be exploring here and implementing with the PIC16F628A microcontroller that will receive the temperature and humidity values from the sensor and display them on a 16×2 character LCD. The DHT11 sensor comes in a single row 4-pin package and operates from 3.5 to 5.5V power supply. It can measure temperature from 0-50 °C with an accuracy of ±2°C and relative humidity ranging from 20-95% with an accuracy of  ±5%. The sensor provides fully calibrated digital outputs for the two measurements. It has got its own proprietary 1-wire protocol, and therefore, the communication between the sensor and a microcontroller is not possible through a direct interface with any of its peripherals. The protocol must be implemented in the firmware of the MCU with precise timing required by the sensor. The following timing diagrams describe the data transfer protocol between a MCU and the DHT11 sensor. The MCU initiates data transmission by issuing a “Start” signal. The MCU pin must be configured as output for this purpose. The MCU first pulls the data line low for at least 18 ms and then pulls it high for next 20-40? s before it releases it.

****

**Fig 4.3 DHT11 Sensor**

**4.5.1 Specifications**

* Humidity: 20% to 90% RH (Relative Humidity)
* Accuracy:
* Temperature: ±2°C
* Humidity: ±5% RH
* Resolution:
* Temperature: 1°C
* Humidity: 1% RH
* Operating Voltage: Typically operates at 3.3V to 5V DC.
* Output Signal: Digital output (Single-Wire Digital Interface)
* Sampling Rate: 1 reading every 2 seconds
* Power Consumption:
* Standby: < 1µA
* Active: 0.5mA during conversion
* Size: Small form factor, typically packaged in a black plastic housing with three pins for connection (VCC, DATA, GND).
* Response Time:
* Temperature: < 5 seconds
* Humidity: < 5 seconds
* Long-Term Stability: The sensor provides reliable and stable readings over extended periods of use.
* Calibration: The sensor is factory calibrated.
* Interchangeability: Multiple DHT11 sensors can be used interchangeably without calibration adjustments.
* Compatibility: Widely compatible with microcontrollers and development boards like Arduino, Raspberry Pi, etc.
* Applications: Commonly used in weather stations, HVAC (Heating, Ventilation, and Air Conditioning) systems, environmental monitoring, home automation, and other projects requiring temperature and humidity sensing.
* Limitations: While the DHT11 sensor is cost-effective and easy to use, it has limitations in terms of accuracy and response time compared to more advanced sensors like the DHT22 or the SHT series. Additionally, it is not designed for use in extreme conditions or high-precision applications.
* Measurement Range:Temperature: 0°C to 50°C (32°F to 122°F)

**4.6 Gas sensor**

In current technology scenario, monitoring of gases produced is very important. From home appliances such as air conditioners to electric chimneys and safety systems at industries monitoring of gases is very crucial. Gas sensors are very important part of such systems.  Small like a nose, gas sensors spontaneously react to the gas present, thus keeping the system updated about any alterations that occur in the concentration of molecules at gaseous state. Gas sensors are available in wide specifications depending on the sensitivity levels, type of gas to be sensed, physical dimensions and numerous other factors. This Insight covers a methane gas sensor that can sense gases such as ammonia which might get produced from methane. When a gas interacts with this sensor, it is first ionized into its constituents and is then adsorbed by the sensing element. This adsorption creates a potential difference on the element which is conveyed to the processor unit through output pins in form of current.

The gas sensor module consists of a steel exoskeleton under which a sensing element is housed. This sensing element is subjected to current through connecting leads. This current is known as heating current through it, the gases coming close to the sensing element get ionized and are absorbed by the sensing element. This changes the resistance of the sensing element which alters the value of the current going out of it.

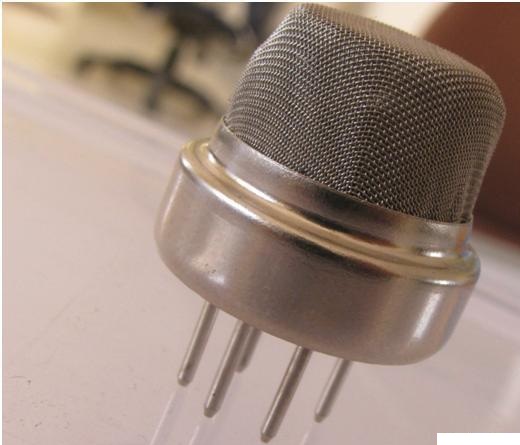


Fig 4.4 gas sensor

Image shows the hollow sensing element which is made up from Aluminium Oxide based ceramic and has a coating of tin oxide. Using a ceramic substrate increases the heating efficiency and tin oxide, being sensitive towards adsorbing desired gas’ components (in this case methane and its products) suffices as sensing coating.The leads responsible for heating the sensing element are connected through Nickel-Chromium, well known conductive alloy. Leads responsible for output signals are connected using platinum wires which convey small changes in the current that passes through the sensing element. The platinum wires are connected to the body of the sensing element while Nickel-Chromium wires pass through its hollow structure. While other wires are attached to the outer body of the element, Nickel-Chromium wires are placed inside the element in a spring shaped. Image shows coiled part of the wire which is placed on the inside of the hollow ceramic.

**4.6.1 Specifications**

* Measurement Range: Indicates the range within which the sensor can accurately detect the concentration of the target gas.
* Sensitivity: Describes the sensor's ability to detect small concentrations of the target gas within its measurement range.
* Response Time: Refers to the time taken by the sensor to detect and respond to changes in gas concentration.
* Accuracy: Specifies the sensor's accuracy in measuring gas concentration, usually expressed as a percentage of the measured value.
* Resolution: Indicates the smallest change in gas concentration that the sensor can detect.
* Operating Temperature and Humidity Range: Defines the environmental conditions under which the sensor can operate effectively.
* Warm-Up Time: Specifies the time required for the sensor to stabilize and provide accurate readings after being powered on.
* Power Consumption: Describes the amount of power consumed by the sensor during operation.
* Output Signal: Indicates the type of output signal produced by the sensor, which could be analog voltage, digital, or both.
* Operating Voltage: Specifies the voltage range required to power the sensor.
* Life Span: Estimates the expected lifetime of the sensor under normal operating conditions.
* Size and Form Factor: Defines the physical dimensions and package style of the sensor, which can vary based on the type and manufacturer.
* Certifications: Some sensors may have certifications indicating compliance with specific industry standards or safety regulations.
* Applications: Describes the intended use cases or applications for which the sensor is suitable, such as industrial safety, environmental monitoring, automotive, etc.
* Maintenance Requirements: Indicates any regular maintenance or calibration needed to ensure the sensor's continued accuracy and reliability.
* Gas Detected: Specifies the type of gas the sensor is designed to detect, such as CO, CH4, NH3, H2, etc.

**4.6.2 Pin configuration**

1. "Start" and "Response" signals.
2. The 40-bit data from the sensor has the following structure.
3. Data (40-bit) = Integer Byte of RH + Decimal Byte of RH + Integer Byte of Temp. + Decimal Byte of Temp. + Checksum Byte
4. For DHT11 sensor, the decimal bytes of temperature and humidity measurements are always zero. Therefore, the first and third bytes of received data actually give the numeric values of the measured relative humidity (%) and temperature (°C). The last byte is the checksum byte which is used to make sure that the data transfer has happened without any error. If all the five bytes are transferred successfully then the checksum byte must be equal to the last 8 bits of the sum of the first four bytes, i.e.,

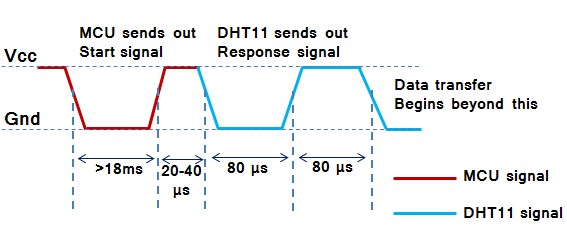
[](http://embedded-lab.com/blog/wp-content/uploads/2012/01/TimingDiagram1.jpg)

Fig 4.4 signals

1. Checksum = Last 8 bits of (Integer Byte of RH + Decimal Byte of RH + Integer Byte of Temp. + Decimal Byte of Temp.)
2. Now lets talk about the most important thing, which is signalling for transmitting “0″ and “1″. In order to send a bit of data, the sensor first pulls the line low for 50 ?s. Then it raises the line to high for 26-28 ?s if it has to send “0″, or for 70 ?s if the bit to be transmitted is “1″. So it is the width of the positive pulse that carries information about 1 and 0.

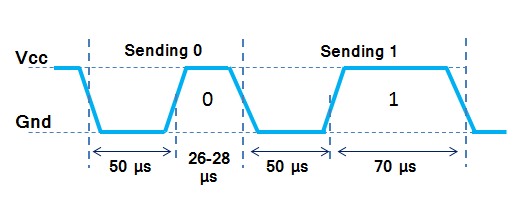
[](http://embedded-lab.com/blog/wp-content/uploads/2012/01/TimingDiagram2.jpg)

Fig 4.5 Signals

1. Timing difference for transmitting "1s" and "0s"

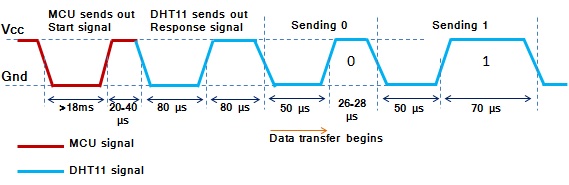
[](http://embedded-lab.com/blog/wp-content/uploads/2012/01/CombinedTiming.jpg)

Fig 4.6 Signals

1. Start, Response and Data signals in sequence
2. At the end of the last transmitted bit, the sensor pulls the data line low for 50s and then releases it. The DHT11 sensor requires an external pull-up resistor to be connected between its Vcc and the data line so that under idle condition, the data line is always pulled high. After finishing the data transmission and releasing the data line, the DHT11 sensor goes to the low-power consumption mode until a new “Start” signal arrives from the MCU.

**4.7 GSM MODEM:**

Global system for mobile communication (GSM) is a globally accepted standard for digital cellular communication. GSM is the name of a standardization group established in 1982 to create a common European mobile telephone standard that would formulate specifications for a pan-European mobile cellular radio system operating at 900 MHz.

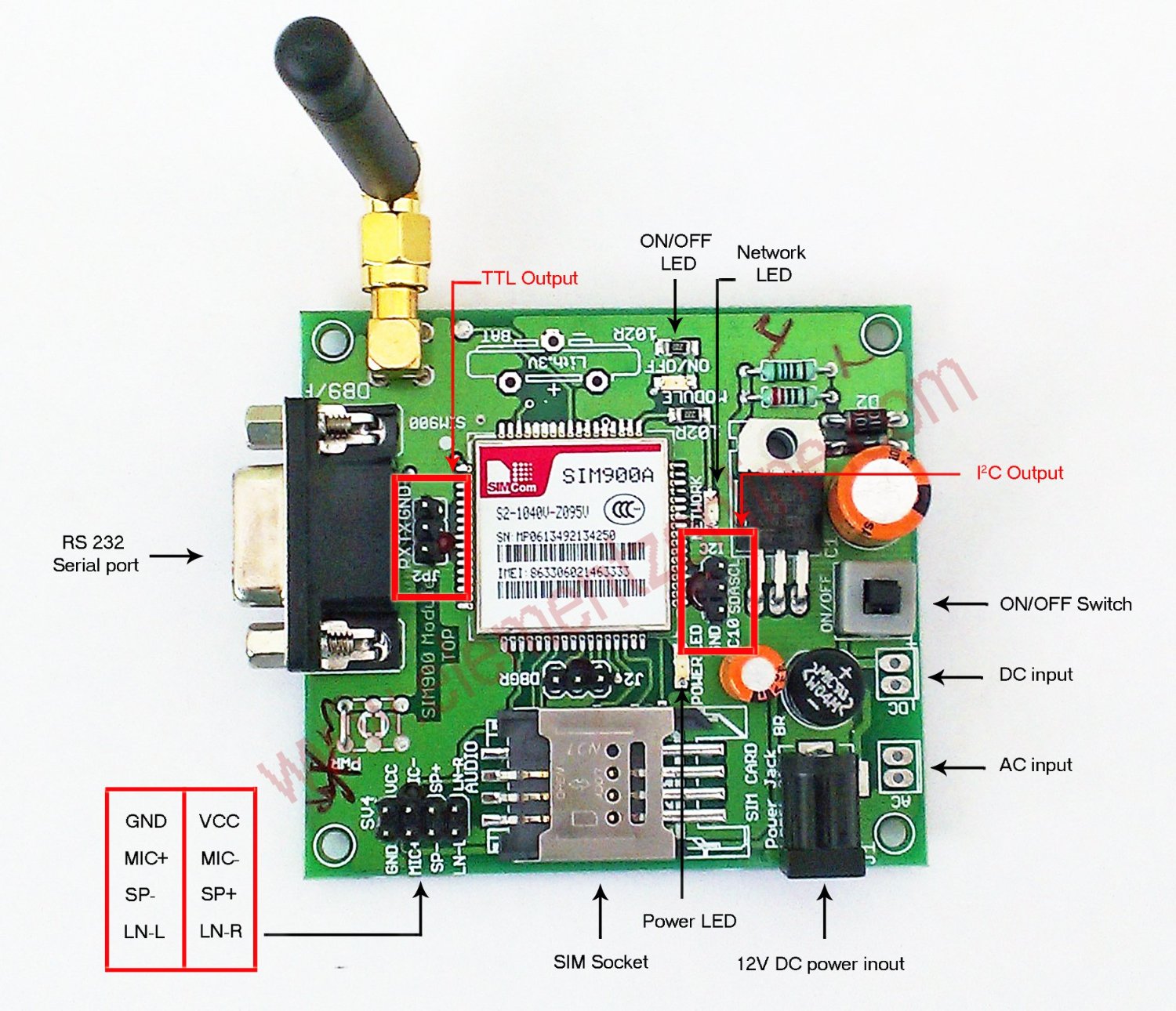


Fig 4.7 GSM modem

GSM provides recommendations, not requirements. The GSM specifications define the functions and interface requirements in detail but do not address the hardware. The reason for this is to limit the designers as little as possible but still to make it possible for the operators to buy equipment from different suppliers. The GSM network is divided into three major systems: the switching system (SS), the base station system (BSS), and the operation and support system (OSS). The basic GSM network elements are shown in below figure .

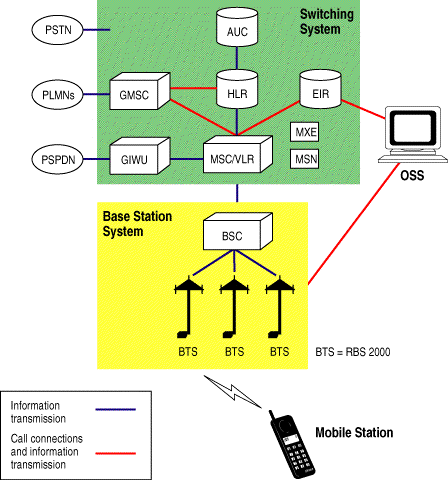


Fig 4.8 GSM Network diagram

A GSM modem is a wireless modem that works with a GSM wireless network. A wireless modem behaves like a dial-up modem. The main difference between them is that a dial-up modem sends and receives data through a fixed telephone line while a wireless modem sends and receives data through radio waves.A GSM modem can be an external device or a PC Card / PCMCIA Card. Typically, an external GSM modem is connected to a computer through a serial cable or a USB cable. A GSM modem in the form of a PC Card / PCMCIA Card is designed for use with a laptop computer. It should be inserted into one of the PC Card / PCMCIA Card slots of a laptop computer. Like a GSM mobile phone, a GSM modem requires a SIM card from a wireless carrier in order to operate.As mentioned in earlier sections of this SMS tutorial, computers use AT commands to control modems. Both GSM modems and dial-up modems support a common set of standard AT commands. You can use a GSM modem just like a dial-up modem.In addition to the standard AT commands, GSM modems support an extended set of AT commands. These extended AT commands are defined in the GSM standards. With the extended AT commands, you can do things like

* 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.

The number of SMS messages that can be processed by a GSM modem per minute is very low -- only about six to ten SMS messages per minute.

**4.7.1 APPLICATION OF GSM MODULE COMMUNICATION**

If your application needs one or more of the following features, GSM will be more cost-effective then other communication systems.Short Data Size.You data size per transaction should be small like 1-3 lines. e.g. banking transaction data, sales/purchase data, consignment tracking data, updates. These small but important transaction data can be sent through SMS messaging which cost even less then a local telephone call or sometimes free of cost worldwide. Hence with negligible cost you are able to send critical information to your head office located anywhere in the world from multiple points.

You can also transfer faxes, large data through GSM but this will be as or more costly compared to landline networks. If you have multiple data collections points situated all over your city, state, country or worldwide you will benefit the most. The data can be sent from multiple points like your branch offices, business associates, warehouses, and agents with devices like GSM modems connected to PCs, GSM electronic terminals and Mobile phones. Many a times some places like warehouses may be situated at remote location may not have landline or internet but you will have GSM network still available easily.If your business require high uptime and availability GSM is best suitable for you as GSM mobile networks have high uptime compared to landline, internet and other communication mediums. Also in situations where you expect that someone may sabotage your communication systems by cutting wires or taping landlines, you can depend on GSM wireless communication. GSM SMS messaging can handle large number of transaction in a very short time. You can receive large number SMS messages on your server like e-mails without internet connectivity. E-mails normally get delayed a lot but SMS messages are almost instantaneous for instant transactions. Consider situation like shop owners doing credit card transaction with GSM technology instead of conventional landlines. time you find local transaction servers busy as these servers use multiple telephone lines to take care of multiple transactions, whereas one GSM connection is enough to handle hundreds of transaction.GSM technology allows mobility, GSM terminals, modems can be just picked and installed at other location unlike telephone lines. Also you can be mobile with GSM terminals and can also communicate with server using your mobile phone. You can just purchase the GSM hardware like modems, terminals and mobile handsets, insert SIM cards, configure software and you are ready for GSM communication.

**4.7.2 Working of GSM MODEM**

In a forest fire detection system, GSM technology serves as the vital link for transmitting real-time alerts and data to relevant authorities and stakeholders. As sensors monitor environmental conditions for signs of fire outbreaks, GSM-enabled modules swiftly relay any detected anomalies to a central control unit. Once confirmed, precise alerts containing critical information, including the fire's location and severity, are formatted and transmitted via SMS or GPRS to designated recipients such as fire departments and emergency response teams. This immediate communication enables rapid mobilization of resources and coordinated firefighting efforts. Additionally, GSM facilitates remote monitoring and control of the system, allowing for proactive maintenance and enhancing overall effectiveness in mitigating the devastating impact of forest fires.

In conjunction with the sensor network, GSM technology provides a robust and reliable means of communication even in remote forested areas where traditional communication infrastructure may be lacking. By leveraging existing mobile networks, the system ensures widespread coverage, enabling timely alerts to be disseminated to relevant parties regardless of their location. This ubiquitous connectivity is essential for effective forest fire management, as it facilitates rapid response coordination and ensures that no critical information is lost in transit. Moreover, the integration of GSM technology allows for scalability and flexibility, enabling the system to adapt to varying environmental conditions and operational requirements.

Furthermore, GSM-enabled forest fire detection systems often incorporate features for data logging and analysis, capturing historical trends and patterns in fire incidents. This data-driven approach enables authorities to refine their predictive models and enhance preparedness strategies based on past occurrences. By leveraging insights gleaned from data analysis, forest management agencies can proactively identify high-risk areas, allocate resources efficiently, and implement preventive measures to mitigate the impact of potential fire outbreaks. Ultimately, the combination of GSM technology with advanced analytics empowers stakeholders with the foresight and agility needed to safeguard forests and communities against the ever-present threat of wildfires.

**4.7.3 GSM Modem Applications**

The GSM/GPRS Modem comes with a serial interface through which the modem can be controlled using AT command interface. An antenna and a power adapter are provided. The basic segregation of working of the modem is as under

* Voice calls
* SMS
* GSM Data calls
* GPRS

**Voice calls:**

Voice calls are not an application area to be targeted. In future if interfaces like a microphone and speaker are provided for some applications then this canbe considered.  
  
**SMS:**

SMS is an area where the modem can be used to provide features like:

* These SMS can be transmitted on certain trigger events in an automation system
* SMS can also be used in areas where small text information has to be sent. The transmitter can be an automation system or machines like vending machines, collection machines or applications like positioning systems where it keeps.
* The navigator keeps on sending SMS at particular time intervals. SMS can be a solution where GSM data call or GPRS services are not available.

**Access control devices:**

Now access control devices can communicate with servers and security staff through SMS messaging. Complete log of transaction is available at the head-office Server instantly without any wiring involved and device can instantly alert security personnel on their mobile phone in case of any problem. RaviRaj Technologies is introducing this technology in all Fingerprint Access control and time attendance products.

**4.8 GPS NEO-6m Module**

Satellite Signal Reception: The NEO-6M module receives signals from multiple GPS satellites, as well as other satellite constellations like GLONASS, Galileo, and BeiDou. These satellites transmit signals containing precise timing information and their own orbital data.

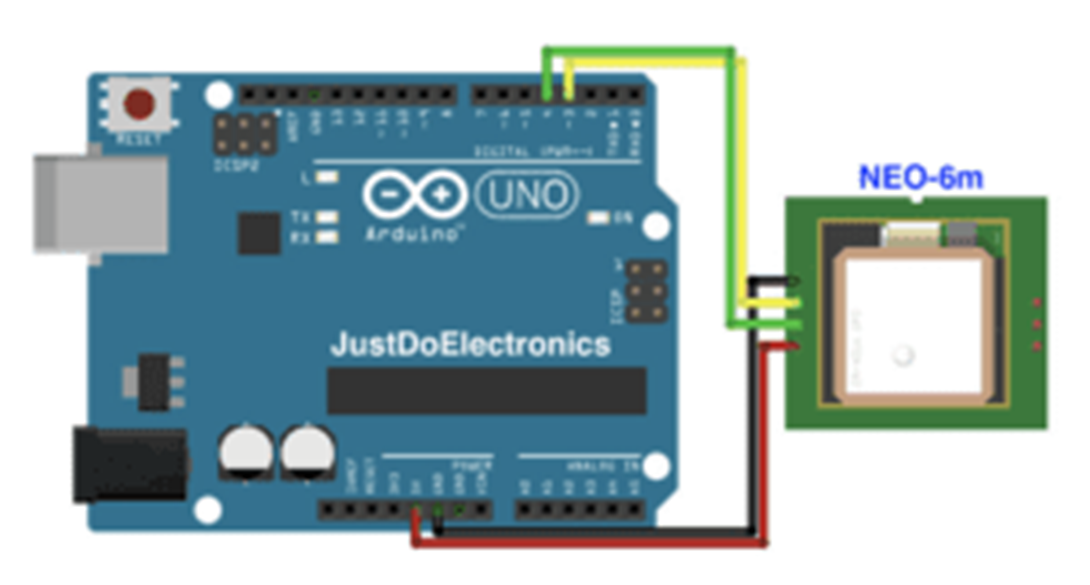
* Signal Processing: The module processes the received satellite signals to extract essential information. It uses a combination of algorithms, including trilateration, to determine the distance between the module and each satellite. By analyzing the time it takes for the signals to travel from the satellites to the module, the module can calculate the module’s position
* Position Calculation: They take information from multiple satellites to calculate their precise position on the Earth’s surface. Calculated distances from different satellites, the module can determine the latitude, longitude, and altitude of its position.
* Data Output: Once the module has determined its position, it formats the information into NMEA (National Marine Electronics Association) sentences. GPS parameters, such as latitude, longitude, altitude, speed, and time.
* Configuration and Control: The NEO-6M module can be configured using AT (Attention) commands sent over the serial interface.
* Operating temperature range: -40 TO 85°CUART TTL socket
* EEPROM to save configuration settings
* Rechargeable battery for Backup
* The cold start time of 38 s and the Hot start time of 1 s
* Supply voltage: 3.3 V
* Configurable Baud rates 115200 and Baud rates 9600
* SuperSense ® Indoor GPS: -162 dBm tracking sensitivity
* Support SBAS (WAAS, EGNOS, MSAS, GAGAN)
* Separated 18X18mm GPS antenna.



Fig 4.9 GPS Module

* Outdoor Recreation: The module is employed in outdoor recreational devices like hiking Multi-Constellation Support: The NEO-6M module is capable of receiving signals from multiple satellite constellations, including GPS (USA), GLONASS (Russia), Galileo (Europe), and BeiDou (China). This multi-constellation support enhances positioning accuracy and reliability by accessing signals from a larger number of satellites.
* High Sensitivity: The module has a high receiver sensitivity, allowing it to acquire and track satellite signals even in challenging environments with weak signal conditions. This sensitivity enables reliable positioning performance in urban areas, dense foliage, and other areas where signal reception can be difficult.
* Application Of GPS NEO-6m Module
* Vehicle Tracking: They are used in vehicle tracking systems, real-time monitoring of the vehicle’s location, speed, and direction.
* Personal Navigation Devices: The module can be integrated into handheld navigation devices, such as GPS receivers, smartphones, and smartwatches, providing users with accurate location information.
* GPS units, cycling computers, and sports watches.

**4.8.1 GPS NEO-6m With Arduino**

****

**Fig 4.10 GPS with arduino**

**4.8.2 Accuracy**

In the vast expanse of forests, where every second counts in averting disaster, precision is paramount. In the realm of forest fire management, the integration of IoT technology with GPS accuracy has emerged as a game-changer. By amalgamating real-time data collection, advanced analytics, and swift response mechanisms, this fusion has revolutionized how we combat wildfires, mitigating their devastating impact on the environment and communities.

Imagine a scenario where a dense forest, teeming with life, is suddenly engulfed in flames. In such critical moments, traditional methods of fire detection may prove inadequate. This is where IoT-enabled GPS systems step in, offering unparalleled accuracy and efficiency. These systems, equipped with sensors strategically dispersed throughout the forest, continuously monitor various parameters such as temperature, humidity, and wind speed. This constant stream of data is then analyzed in real-time, enabling early detection of potential fire outbreaks.

However, the true power of IoT-enabled GPS accuracy lies not only in detection but also in response coordination. As soon as a fire is detected, precise location data is transmitted to centralized control centers, equipped with state-of-the-art monitoring systems. These centers act as nerve centers, orchestrating a synchronized response effort. Simultaneously, alerts are dispatched to relevant authorities and firefighting teams, guiding them swiftly to the exact location of the fire.

Moreover, the integration of GPS accuracy with IoT extends beyond mere detection and response. It facilitates proactive measures to contain and suppress wildfires before they escalate. Advanced predictive analytics algorithms analyze historical data, terrain characteristics, and weather patterns to forecast areas prone to fire outbreaks. This foresight allows for preemptive measures such as controlled burns and strategic deployment of resources, minimizing the risk of catastrophic wildfires.

Furthermore, the efficacy of IoT-enabled GPS accuracy extends to post-fire assessment and recovery. Drones equipped with high-resolution cameras and GPS capabilities survey the affected areas, providing invaluable insights into the extent of damage and areas requiring rehabilitation. This data aids in formulating comprehensive recovery plans and implementing targeted restoration efforts to restore the ecosystem's balance.

In essence, the marriage of IoT technology with GPS accuracy heralds a new era in forest fire management. It empowers us with the tools and insights needed to combat wildfires with unprecedented precision and efficiency. By leveraging real-time data, advanced analytics, and swift response mechanisms, we not only protect our forests and wildlife but also safeguard human lives and livelihoods against the ravages of these infernos. As we continue to harness the potential of technology, we move closer to a future where the threat of wildfires is met with resilience, resourcefulness, and unwavering resolve.

**4.9 SOFTWARE REQUIREMENTS**

* **Arduino IDE**
* **Embedded c programming**
* **Proteus based circuit diagram**

**4.9.1 Arduino IDE:**

Arduino IDE is an integrated development environment. Can be used over different operating systems or platforms i.e. Windows, macOS, Linux.

IDE is used for writing and uploading the code of the hardware for programming the microcontroller accordingly. The IDE consists of the different software libraries from the previous wiring project.

**4.9.1.1Working:**

Step 1. Connecting the Arduino device with the computer using a USB.



Fig 4.11 Arduino cable

Step 2. Writing the code in the preferred language.

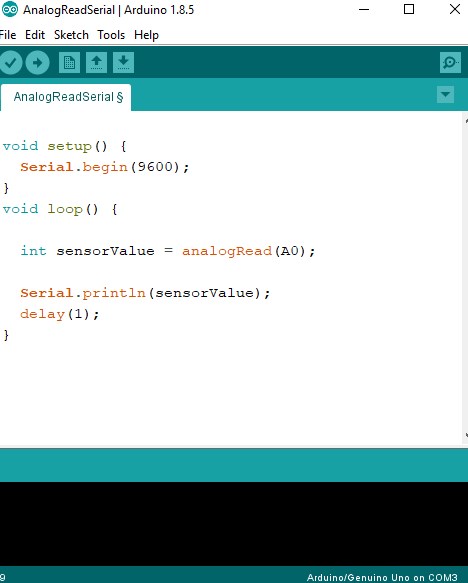


Fig 4.12 Arduino IDE Code

Step 3. Selecting the port for list of ports

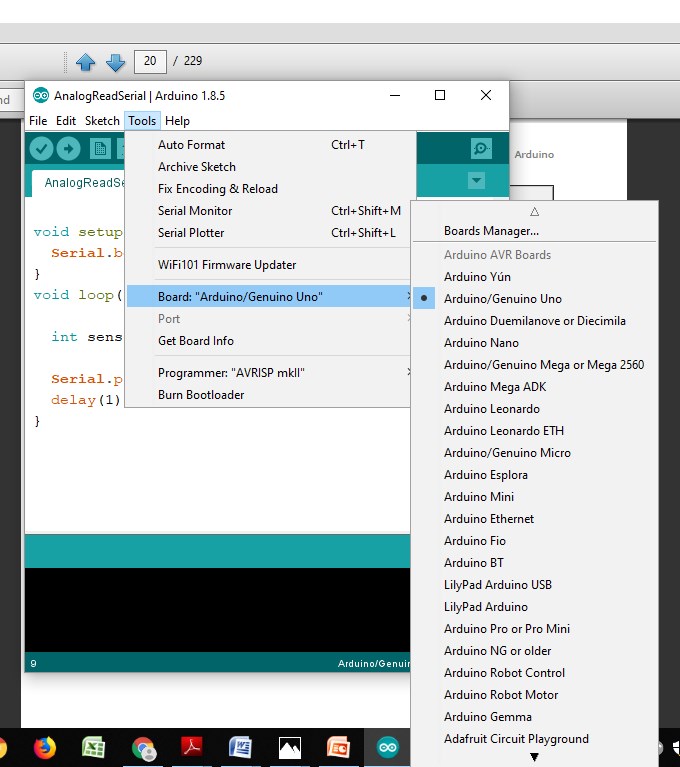


Fig 4.13 IDE port selection page

Step 4. Uploading the code over to the Arduino UNO microcontroller.

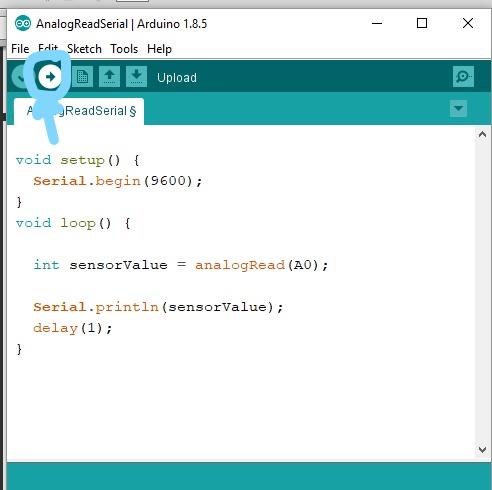


Fig 4.14 Uploading code over Arduino

**4.9.2 PROTEUS SOFTWARE**:

Proteus 8 is a best simulation software for various designs with microcontroller. It is mainly popular because of availability of almost all microcontrollers in it. So it is a handy tool to test programs and embedded designs for electronics hobbyist. You can simulate your programming of microcontroller in Proteus 8 Simulation Software. After simulating your circuit in Proteus 8 Software you can directly make PCB design with it so it could be a all in one package for students and hobbyists. So I think now you have a little bit idea about what is proteus software. Proteus is a Virtual System Modelling (VSM) that combines circuit simulation, animated components and microprocessor models to co-simulate the complete microcontroller based designs. This is the perfect tool for engineers to test their microcontroller designs before constructing a physical prototype in real time. This program allows users to interact with the design using on-screen indicators and LED and LCD displays and, if attached to the PC, switches and buttons. Proteus comes with extensive debugging features, single stepping and variable display for a neat design prior to hardware prototyping. Proteus is the program to use when you want to simulate the interaction between software running on a microcontroller and any analog or digital electronic device connected to it. Proteus was initially created as a multiplatform (DOS, Windows, Unix) system utility, to manipulate text and binary files and to create CGI scripts. The language was later focused on Windows, by adding hundreds of specialized functions for: network and serial communication, database interrogation, system service creation, console applications, keyboard emulation, ISAPI scripting (for IIS). Most of these additional functions are only available in the Windows flavor of the interpreter, even though a Linux version is still available.

Proteus was designed to be practical (easy to use, efficient, complete), readable and consistent.

* powerful string manipulation;
* comprehensibility of Proteus scripts;
* Availability of advanced data structures: arrays, queues (single or double), stacks, bit maps, sets, AVL trees.
* The language can be extended by adding user functions written in Proteus or DLLs created in C/C++.

**4.9.3 LANGUAGE FEATURES:**

Proteus has a fully functional, procedural approach; variables are untyped, do not need to be declared, can be local or public and can be passed by value or by reference; all the typical control structures are available (if-then-else; for-next; while-loop; repeat-until; switch-case);new functions can be defined and used as native functions.Data types supported by Proteus are only three: integer numbers, floating point numbers and strings. Access to advanced data structures (files, arrays, queues, stacks, AVL trees, sets and so on) takes place by using handles, i.e. integer numbers returned by item creation functions. Type declaration is unnecessary: variable type is determined by the function applied – Proteus converts on the fly every variable when needed and holds previous data renderings, to avoid performance degradation caused by repeated conversions. There is no need to add parenthesis in expressions to determine the evaluation order, because the language is fully functional (there are no operators).

Proteus includes hundreds of functions for:

* + accessing file system;
  + sorting data;
  + manipulating dates and strings;
  + interacting with the user (console functions)
  + Calculating logical and mathematical expressions.

Proteus supports associative arrays (called sets) and AVL trees, which are very useful and powerful to quickly sort and lookup values.The functional approach and the extensive library of built-in functions allow to write very short but powerful scripts; to keep them comprehensible, medium-length keywords were adopted.The user, besides writing new high-level functions in Proteus, can add new functions in C/C++ by following the guidelines and using the templates available in the software development kit; the new functions can be invoked exactly the same way as the predefined ones, passing expressions by value or variables by reference. Proteus is an interpreted language: programs are loaded into memory, pre-compiled and run; since the number of built-in functions is large, execution speed is usually very good and often comparable to that of compiled programs. One of the most interesting features of Proteus is the possibility of running scripts as services or ISAPI scripts. Running a Proteus script as a service, started as soon as the operating system has finished loading, gives many advantages: no user needs to login to start the script; a service can be run with different privileges so that it cannot be stopped by a user. This is very useful to protect critical processes in industrial environments (data collection, device monitoring), or to avoid that the operator inadvertently closes a utility (keyboard emulation).The ISAPI version of Proteus can be used to create scripts run through Internet Information Services and is equipped with specific functions to cooperate with the web server. For intellectual property protection Proteus provides: digital signature of the scripts, by using the development key (which is unique);the option to enable or disable the execution of a script (or part of it) by using the key of the customer. Proteus is appreciated because it is relatively easy to write short, powerful and comprehensible scripts; the large number of built-in functions, together with the examples in the manual, keep low the learning curve.

The development environment includes a source code editor with syntax highlighting and a context-sensitive guide. Proteus does not need to be installed: the interpreter is a single executable (below 400 Kb) that does not require additional DLLs to be run on recent Windows systems.

**CHAPTER 5**

**CODING**

**5.1 ARDUINO UNO**

#include <dht.h>

#include <dht.h> // Include library

#define alcohol A2 // Defines pin number to which the sensor is connected

dht DHT; // Creates a DHT object

#include<SoftwareSerial.h>

#include <LiquidCrystal.h>

SoftwareSerial mySerial(6,7);

const int rs = 8, en = 9, d4 = 10, d5 = 11, d6 = 12, d7 = 13;

LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

int gas = A0;

void setup() {

Serial.begin(9600);

mySerial.begin(9600);

pinMode(2,INPUT);//ir

pinMode(A0,INPUT);

//attachInterrupt(digitalPinToInterrupt(2), change, LOW);

pinMode(13,OUTPUT);//buzzer

digitalWrite(13,HIGH);delay(500);

digitalWrite(13,LOW);delay(500);

}

void loop() {

int readData = DHT.read11(alcohol);

int temp = DHT.temperature; // Read temperature

float h = DHT.humidity; // Read humidity

int Fire = !digitalRead(2);

int Gas = analogRead(A0);

Serial.print(" temp :");

Serial.println(temp);

Serial.print(" Fire :");

Serial.println(Fire);

Serial.print(" Gas :");

Serial.println(Gas);

if (Fire == 1)

{

digitalWrite(13,HIGH);delay(2000);digitalWrite(13,LOW);delay(2000);

}

else if (Gas > 350)

{

digitalWrite(13,HIGH);delay(2000);digitalWrite(13,LOW);delay(2000);

}

else if (temp > 38)

{

digitalWrite(13,HIGH);delay(2000);digitalWrite(13,LOW);delay(2000);

}

String data= String(temp)+"@"+String(Gas)+"#"+String(Fire)+"$";

Serial.println(data);

delay(100);

if(Fire == 1)

{

MakeCall1();

delay(15000);

MakeCall();

delay(150000);

SendMessage();

delay(15000);

SendMessage1();

delay(15000);

Serial.print("EMERGENCY!!!!!");

lcd.clear();

lcd.setCursor(0,0);

lcd.print("EMERGENCY!!!!!");

lcd.setCursor(0,1);

lcd.print("SMS/CallingSend");

Serial.println("Received");

}

else if (Gas > 450)

{

digitalWrite(13,HIGH);

delay(1000);

}

else

{

digitalWrite(13,LOW);

//Serial.println("Normal State");

lcd.clear();

lcd.setCursor(0,0);

//lcd.print("Normal State");

delay(1000);

}

}

void SendMessage()

{

Serial.println("I am in send");

mySerial.println("AT+CMGF=1"); //Sets the GSM Module in Text Mode

delay(1000); // Delay of 1000 milli seconds or 1 second

mySerial.println("AT+CMGS=\"+918838647593\"\r"); // Replace x with mobile number

delay(1000);

mySerial.println("Emergency Alert Forest Fire Detected!!!!!!");// The SMS text you want to send

delay(1000);

mySerial.println("https://maps.app.goo.gl/F7Tk9smy7FPHWVW59");

delay(1000);

mySerial.println((char)26);// ASCII code of CTRL+Z

delay(1000);

//mySerial.println((char)26);// ASCII code of CTRL+Z

}

void SendMessage1()

{

Serial.println("I am in send");

mySerial.println("AT+CMGF=2"); //Sets the GSM Module in Text Mode

delay(1000); // Delay of 1000 milli seconds or 1 second

mySerial.println("AT+CMGS=\"+918838647593\"\r"); // Replace x with mobile number

delay(1000);

mySerial.println("Emergency Alert Forest Fire Detected!!!");

delay(1000);

mySerial.println("Lat : 12.922915, Log : 80.127457");

delay(1000);

mySerial.println((char)26);// ASCII code of CTRL+Z

delay(1000);

}

void MakeCall()

{

mySerial.println("ATH");

delay(1000);

mySerial.println("ATD+918838647593;"); // ATDxxxxxxxxxx; -- watch out here for semicolon at the end!!

delay(1000);

mySerial.println("Calling "); // print response over serial port

delay(1000);

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("CALLING");

lcd.setCursor(0, 1);

lcd.print("+919344887029");

Serial.println("918838647593");

}

void MakeCall1()

{

mySerial.println("ATH");

delay(1000);

mySerial.println("ATD+918838647593;"); // ATDxxxxxxxxxx; -- watch out here for semicolon at the end!!

delay(1000);

mySerial.println("Calling "); // print response over serial port

delay(1000);

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("CALLING");

lcd.setCursor(0, 1);

lcd.print("+918838647593");

Serial.println("918838647593");

}

**5.2 IOT CODE (ESP32)**

#include "ThingSpeak.h"

#include <ESP8266WiFi.h>

char ssid[] = "Redmi 10A"; // your network SSID (name)

char pass[] = "www12345"; // your network password

int keyIndex = 0; // your network key Index number (needed only for WEP)

WiFiClient client;

unsigned long myChannelNumber = 2535861;

const char \* myWriteAPIKey = "ETTW9TBERNBDUJQ1";

String myStatus = "";

void setup()

{

Serial.begin(9600); // Initialize serial

WiFi.mode(WIFI\_STA);

ThingSpeak.begin(client); // Initialize ThingSpeak

// Connect or reconnect to WiFi

if (WiFi.status() != WL\_CONNECTED)

{

Serial.print("Attempting to connect to SSID: ");

Serial.println(ssid);

while (WiFi.status() != WL\_CONNECTED)

{

WiFi.begin(ssid, pass); // Connect to WPA/WPA2 network. Change this line if using open or WEP network

Serial.print(".");

delay(5000);

}

Serial.println("\nConnected.");

}

}

void loop()

{

String data;

while (Serial.available())

{

data = Serial.readStringUntil('@');

data.trim();

String temp = data;

data = Serial.readStringUntil('#');

data.trim();

String Gas = data;

data = Serial.readStringUntil('%');

data.trim();

String Fire = data;

Serial.print("temp level :");

Serial.println(temp);

Serial.print("Gas Sensor:");

Serial.println(Gas);

Serial.print("Fire Level:");

Serial.println(Fire);

// set the fields with the values

ThingSpeak.setField(1, temp);

ThingSpeak.setField(2, Gas);

ThingSpeak.setField(3, Fire);

ThingSpeak.setStatus(myStatus);

// write to the ThingSpeak channel

int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey );

if (x == 200)

{

Serial.println("Channel update successful.");

}

else

{

Serial.println("Problem updating channel. HTTP error code " + String(x));

}

delay(100); // Wait 20 seconds to update the channel again

}

}

**CHAPTER 6**

**RESULTS AND DISCUSSION**

The fire detection system employs three sensors - fire, temperature, and smoke detection sensors, which relay information to the Arduino UNO board upon detecting changes in the surroundings. Once confirmed by the Arduino, connected components including the IOT module ESP8266, GSM module, and a buzzer, are activated for output. The primary alert mechanism utilizes the ESP8266, allowing real-time global monitoring through an internet-connected webpage. The Arduino sends temperature, smoke, and geographic coordinates data to the ESP8266, which uploads it to the webpage. However, internet access is necessary for this feature. A secondary alert via mobile phone notification is facilitated by the GSM module, sending alerts and geographic coordinates to the designated person's mobile number without internet dependency. Additionally, a buzzer is incorporated as a traditional local alert method, sounding when the sensors detect fire-related changes, ensuring notification to those without access to internet or mobile signals. This multi-layered approach enables swift response and coordination in fire emergencies, catering to various communication needs and scenarios.

A GSM module is mounted over the circuit in-order to pass mobile notification to a person in-charge. As same as for IOT module, once the temperature rises above threshold temperature and if smoke is detected, through GSM module the arduino send a mobile notification to the mobile number that is provided. The message will be denoting the alert and also the exact geometric co-ordinates latitude and longitude will be send to the person’s mobile number. The exact geometric co-ordinates latitude and longitude will be provided by the GPS module connected to the circuit. This will be even more helpful to know the forest fire as soon as possible. As this part uses GSM module, even without internet we can get the alert. Connecting to internet is not mandatory for GSM module and mobile alert. Once the person in-charge gets the alert he/she can take necessary

actions accordingly. Even though we have high level alert systems connected to the module, we need to have a traditional method to alert people in the local who don’t have access to above methods. We need a method to alert local fire station or people. This method should we able to work even without internet and mobile signal in-case if they are not available. For the last 3rdtype to alert system we have connected a buzzer along with the circuitry. As the sensor senses the rise in temperature and smoke and fire, the sensors sends the information/data to the arduino Uno which in turn on the buzzer. The buzzer starts alarming so that the local people gets alerted.

**6.1 OUTPUT**

The IOT Webpage output is shown below

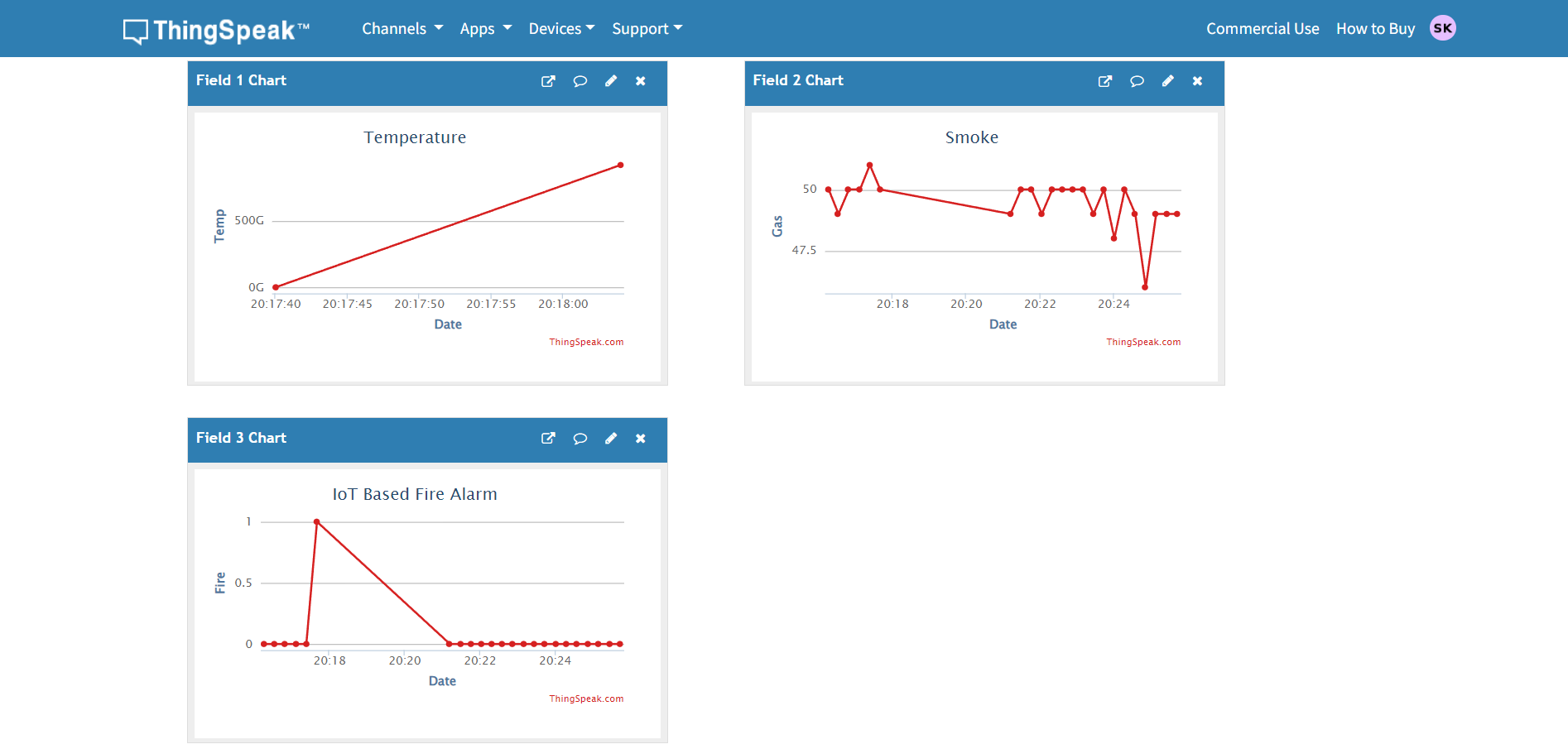


Fig 6.1 IOT Webpage

This is the image of IOT webpage developed with PHP. This webpage is connected with IOT module ESP8266 in the circuit. The webpage stores the data logs that contains the data from the ESP8266. The smoke value, temperature value from the sensors and geometric co-ordinates (latitude and longitude) from the GPS module connected with the module will be uploaded in real-time. Also, we can see the date and time of upload. This can be viewed and monitored in real-time anywhere around the globe.

The overall setup for forest fire detection system

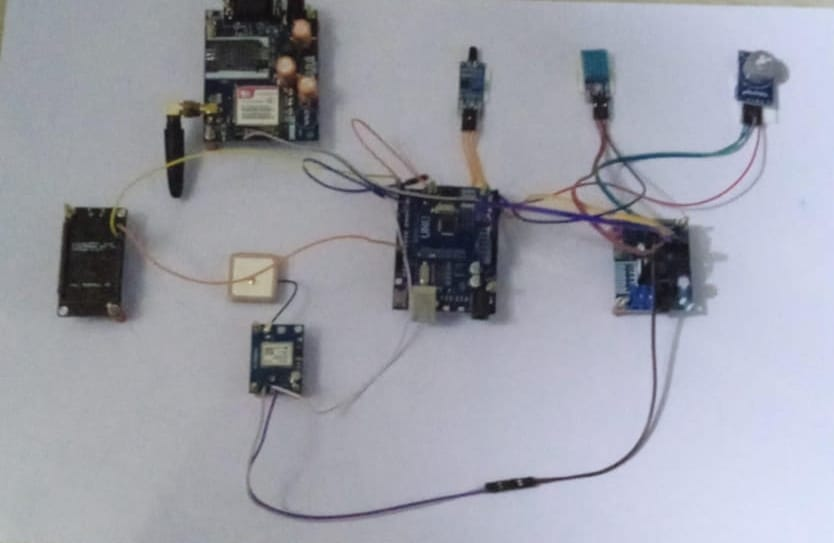


Fig 6.2 Overall setup.

## CHAPTER 7

## SUMMARY AND CONCLUSION

## We have used two modules: one for fire detection with the help of Gas sensor (MQ-7) and the other is for the location, humidity condition and temperature Here we used one alarm near to fire so as the fire detected in that area fire alarm will give the sound and alert the animals and people working there to leave that place safely. GPS and gas sensor is synchronized in such a way that as the gas sensor detects the fire GPS will send its location along with the gas sensor reading. This data will be send by GSM sensor and receive by GSM sensor on the receiving side and then to data base in this way we can protect our wildlife, forest and people working there before fire occurs. New technologies and concepts developed in such as advanced multi- function sensors, computer vision systems and wireless sensors, real-time control via the Internet, and integrated building service systems.Wireless sensor network are increasingly applied in the field of fire safety and monitoring. In addition, wireless sensor technology has a broad application background in the field of real time forest fire monitoring. If this type of project is implemented by the Government then many lives can be saved including humans and animals as well as valuable trees. It can also save capitals, time and manpower. Implementation of this project can solve many of the environmental issues.

## Despite the fact that our machine is self-contained and independent, other variables that might influence the hardware were checked over time. It would be introduced in limited woodland regions where forest fires are a frequent phenomenon. To survive all of the temperature variations that can impact its operation, the device must be resilient.The equipment we used is one of the most effective and well-known for detecting forest fires in a short period of time. We experimented in forest-like environments, but the true challenge will come when we implement in a wide area in real time. At extreme weather conditions, the signal to the GSM module might be unreliable and also in deep forests area’s the signal may be unreachable. In future we can overcome this drawback using eminent technology.With the use of this circuitry, the forest fire can be detected and prevented(in case of minor fires)using servo motor. The locals can be alerted with the help of buzzer, the fire department head will be notified via phone message. Also we can monitor all the sensor values in real-time through webpage, all around the globe. As a conclusion, our work setup can be implemented in real-time environment.

## REFERENCES

[1] National Semiconductor Corporation DS005516 www.national.com, November 2000

[2] Microchip received QS-9000 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona in July 1999 and Mountain View, California in March 2002.

[3] L. Ravindran, M. G. Mariam and C. A. Aliza, "Microcontroller Based Fire Alarm System Using Sensory and Monitoring System", International Conference on Engineering and JCT (ICEJ 2007)

[4] Karwan MUHEDEN, Ebubekir ERDEM, Sercan VANÇİN, "Design and Implementation of the Mobile Fire Alarm System Using Wireless Sensor Networks", Department of Computer Engineering Firat University.

[5] Ashwitha A, Arjun K, Prashanth H. L, '' GSM Enabled Smart Fire Alarm Controlling System with SMS Alert", International Journal of Computer Applications (0975 – 8887) Volume 147 – No.11, August 2016

[6] R. Sathishkumar, M. Vinothkumar, Devaraj Varatharaj, S. Rajesh, S. M. Gowthaman, " Design And Development Of Automatic Fire Detection Using Sms And Voice Alert System", International Journal of Scientific & Engineering Research, Volume 7, Issue 5, 114 ISSN 2229-5518, May-2016.

[7] Dhruvajyoti Paul, Arnab Ghosh, Soumya Jyoti Banerjee, "GSM Based Fire Sensor Alarm Using Arduino", International Journal of Scientific & Engineering Research, Volume 7, Issue 4, 259 ISSN 2229-5518, April-2016.

[8] Onengiye M. Georgewill, Chukwunazo J. Ezeofor, "Design and Implementation of SMS-Based Industrial/Homes Gas Leakage Monitoring & Detection Alarm System" International Journal of Engineering Trends and Technology (IJETT) – Volume 35 Number 9 - May 2016.

[9] R. O. Okeke, M. Ehikhamenle, '' Design And Simulation Of Gas And Fire Detector And Alarm System With Water Sprinkle", International Journal of Engineering Research and General Science Volume 5, Issue 1, ISSN 2091-2730, January-February 2017

[10] Digvijay Singh, Neetika Sharma, Mehak Gupta, Shubham Sharma,'' Development of System for Early Fire Detection using Arduino UNO ", International Journal of Engineering Science and Computing, May 2017.