IoT Based Industrial Fault Gas, Smoke, Fire Detection and Temperature Monitoring System

A report submitted to the Department of Electrical and Electronic Engineering, Sonargaon University of Bangladesh in partial fulfillment of the requirements for the Award of Degree of Bachelor of Science in Electrical and Electronic Engineering.

Submitted by

Rahatuzzaman Ringko	ID: EEE - 1901016087
Md. Alamin	ID: EEE - 1901016088
Md. Ariful Islam	ID: EEE - 1901016100
Tania Begum	ID: EEE - 1901016187
Samim Miah	ID: EEE - 1701010146



Supervised by

Jannatul Nazrana

Lecturer

Department of Electrical & Electronic Engineering

Sonargaon University (SU)

Dhaka-1215, Bangladesh

September, 2022

LETTER OF TRANSMITTAL

September, 2022

To
Jannatul Nazrana
Lecturer
Department of Electrical & Electronic Engineering.
Sonargaon University of Bangladesh

Subject: Submission of Project Report.

Dear Sir,
We are pleased to submit the project report on "**IoT Based Industrial Fault Gas, Smoke, Fire Detection and Temperature Monitoring System".** It was a great pleasure to work on such an important topic. This project has been done as per instruction of your supervision and according to the requirements of the Sonargaon University.

We expect that the project will be accepted by the concerned authority we will remain happy to further explanation that you may feel necessary in this regard.

Thank You
Sincerely yours,

Rahatuzzaman Ringko
EEE - 1901016087

Md. Alamin
EEE - 1901016088

Md. Ariful Islam
EEE - 1901016100

Tania Begum
EEE - 1901016187

Samim Miah
EEE - 1701010146

DECLARATION

We do hereby solemnly declare that, the work presented here in this project report has been carried out by us and has not been previously submitted to any University/ Organization for award of any degree or certificate

We hereby ensure that the works that has been prevented here does not breach any existing copyright.

We further undertake to indemnify the university against any loss or damage arising from breach of the foregoing obligation.

Rahatuzzaman Ringko
EEE - 1901016087

Md. Alamin
EEE - 1901016088

Md. Ariful Islam
EEE - 1901016100

Tania Begum
EEE - 1901016187

Samim Miah
EEE - 1701010146

ACKNOWLEDGEMENT

First, we started in the name of almighty Allah. This thesis is accomplished under the supervision of **Jannatul Nazrana,** Lecturer, Department of Electrical & Electronic Engineering, Sonargaon University. It is a great pleasure to acknowledge our profound gratitude and respect to our supervisor for this consistent guidance, encouragement, helpful suggestion, constructive criticism and endless patience through the progress of this work. The successful completion of this thesis would not have been possible without his persistent motivation and continuous guidance.

The authors are also grateful to **Prof. Dr. M. Bashir Uddin, Professsor**, Head of the Department of Electrical & Electronic Engineering and all respect teachers of the Electrical & Electronic Engineering Department for their co-operation and significant help for completing the thesis work successfully.

[Authors]

Rahatuzzaman Ringko
EEE - 1901016087

Md. Alamin
EEE - 1901016088

Md. Ariful Islam
EEE - 1901016100

Tania Begum
EEE - 1901016187

Samim Miah
EEE - 1701010146

ABSTARCT

The IoT industry protection system using Node MCU controller is a system designed to protect industries from losses due to accidents using internet of things. Gas leakages may lead to fires leading to huge industrial losses, also instant fire detection is needed in case of furnace blasts or other conditions. Also low lighting in industries may create improper work conditions increasing the probability of accidents. The system makes use of Node MCU controller to achieve this functionality. The system makes use of temperature sensing along with light and gas sensing to detect fire, gas leakage. The system consists of fire, gas and temperature sensors interfaced with Node MCU controller and LCD screen. The sensor data is constantly scanned to record values and check for fire, gas leakage, temperature and then this data is transmitted online. The wi-fi module is used to achieve internet functionality. If there create any kind of incident about gas leakage, fire and temperature high then the system get a signal to buzzer and it will ring. Beside of this work a notification will be send in mobile phone of authors. In this situation author can shut down the load through IoT System.

Keywords : Node MCU, Temperature sensor, Flame sensor, Gas sensor, IoT, Industrial Safety Alert System.

TABLE OF CONTENTS

Letter of Transmittal		ii
Declaration		iii
Acknowledgement		iv
Abstract		v
Table of Contents		vi
Table of Figure		viii
CHAPTER-1	**INTRODUCTION**	
1.1 Introduction		1
1.2 Objective		2
1.3 Structure of the Projects		3
CHAPTER 2	**LITERATURE REVIEW**	
2.1 Introduction		4
2.2 Literature Review		4
2.3 Summary		6
CHAPTER-3	**HARDWARE AND SOFTWARE ANALYSIS**	
3.1 Node MCU		7
3.2 SMPS		10
3.3 Voltage Regulator		11
3.4 Resistor		12
3.5 Flame Sensor		13
3.6 Gas Sensor		14
3.7 Temperature Sensor		16
3.8 Buzzer		17
3.9 LCD Display		18
3.10 Relay		19
3.11 Arduino IDE		22
3.12 Proteus Software		26

3.13 Blynk Apps 27

CHAPTER-4 METHODOLOGY

4.1 Methodology 30
4.2 Working Step Chart 30
4.3 Block Diagram 31
4.4 Schematic Diagram 31
4.5 Working Principle 32
4.6 Our Final Project View 33

CHAPTER-5 RESULT AND DISCUSSION

5.1 Result 34
5.2 Advantage 34
5.3 Application 35
5.4 Limitation 35
5.5 Discussion 35

CHAPTER-6 CONCLUSION

6.1 Conclusion 37
6.2 Future Scope 37

Reference 38
Appendix 40

LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE NO
3.1	Node MCU	7
3.2	Node MCU Schematic Diagram	8
3.3	Node MCU Pin Out	9
3.4	SMPS Power Supply	10
3.5	Voltage Regulator	12
3.6	Flame Sensor	14
3.7	MQ2 Sensor	14
3.8	Temperature Sensor	16
3.9	Buzzer	17
3.10	LCD Display	18
3.11	Relay	19
3.12	Transistor Switching Circuit	20
3.13	Relay Module	20
3.14	Pin Diagram of Relay Module	21
3.15	Main Voltage Connection	22
3.16	Arduino Software Interface	23
3.17	Proteus Software Interface	27
3.18	Blynk Apps	27
4.1	Working Step Apps	30
4.2	Block Diagram	31
4.3	Schematic Diagram	31
4.4	Our Final System Overview (Top View)	33
4.5	Our Final System Overview (Front View)	33

CHAPTER 1

INTRODUCTION

1.1 Introduction

As the society is growing with various developments, the outmoded forms of storing various food products in cold store rooms is failing to satisfy human needs. As the society is growing with various developments, the outmoded forms of storing various food products in cold store rooms fails to satisfy human need. Through the monitoring of the temperature and humidity inside cold storage rooms, the goodness of the products can be ensured for a longer time. Recent research has revealed that operations of wireless sensor systems are largely affected by their on-board temperature [1]. We can implement sensors in wide area over the machines and instruments and control and monitor the circumstances by using concept of IoT [2].

In this paper we have designed a gateway which will be the central part of this whole system. The function of the gateway is to gather data, process them, upload them and process user control information. If the network connection is not established then the data will be stored and upon reestablishment of the network it will be uploaded. The terms of "things" in the IoT vision is very broad and includes a variety of physical elements. The terms of things include portable personal items such as smart phones, tablets and digital cameras. With the rapid increase in the number of user of and digital cameras. Furthermore, IoT includes elements of our environments (be it home, car or office), and things equipped with RFID tags connected to a gateway device. From those mentioned so far, a huge number of devices and things will be connected to the Internet, each providing data and information and some even services.

Gas leakage is a serious problem and nowadays it is observed in many places like residences, industries, and vehicles like Compressed Natural Gas (CNG), buses, cars, etc. It is noticed that due to gas leakage, dangerous accidents occur.[3] The Liquefied petroleum gas (LPG), or propane, is a flammable mixture of hydrocarbon gases used as fuel in many applications like homes, hostels, industries, automobiles, and vehicles because of its desirable properties which include high calorific value, less smoke, less soot, and meager harm to the environment. Liquid petroleum gas (LPG) is highly inflammable

and can burn even at some distance from the source of leakage.[4] This energy source is primarily composed of propane and butane which are highly flammable chemical compounds. These gases can catch fire easily. In homes, LPG is used mainly for cooking purposes.

When a leak occurs, the leaked gases may lead to an explosion. Gas leakage leads to various accidents resulting in both material loss and human injuries. Home fires have been occurring frequently and the threat to human lives and properties has been growing in recent years.[5] The risks of explosion, fire, suffocation are based on their physical properties such toxicity, flammability, etc. The number of deaths due to the explosion of gas cylinders has been increasing in recent years. The Bhopal gas tragedy is an example of accidents due to gas leakage.

The reason for such explosions is due to substandard cylinders, old valves, no regular checking of gas cylinders, worn out regulators and a lack of awareness of handling gas cylinders.[6] Therefore, the gas leakage should be detected and controlled to protect people from danger. An odorant such as ethane thiol is added to LPG, so that leaks can be detected easily by most people. However, some people who have a reduced sense of smell may not be able to rely upon this inherent safety mechanism.[7] A gas leakage detector becomes vital and helps to protect people from the dangers of gas leakage. A number of research papers have been published on gas leakage detection techniques.

1.2 Objective

In the course of design and installation of a **IoT Industrial Fault Gas, Smoke, Fire Detection & Temperature Monitoring System**, the following criteria are set to fulfill its objectives:

- To study of the **IoT Industrial Fault Gas, Smoke, Fire Detection & Temperature Monitoring System**
- To design and construct of a temperature, fire and gas detection system.
- To implement and study of take proper step to reduce these incident.
- To take necessary notes from the project for future improvements.

1.3 Structure of the Project

This Project is organized as follows:

Chapter 1 Introduction: The first chapter contains the statement of the introduction, our background study for the project, objectives of the study, and the project outline.

Chapter 2 Literature Review: The chapter two contains our literature review part.

Chapter 3 Hardware and Software Analysis: Chapter three describes the theoretical model. Here we mainly discuss about Hardware and software development of our project etc.

Chapter 4 Methodology: Chapter three describes the theoretical model. Here we mainly discuss about project methodology, project working step, Block Diagram, Circuit Diagram, working principle and our project final image etc.

Chapter 5 Result and Discussion: Chapter four deals with the result and discussion and discuss about our project advantages , limitations and application.

Chapter 5 Conclusion: Chapter five all about our project conclusion and future scope.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In this section topics related to **IoT Based Industrial Fault Gas, Smoke, Fire Detection and Temperature Monitoring System** are included. These provide a sampling of problems appropriate for application of **IoT Based Industrial Fault Gas, Smoke, Fire Detection and Temperature Monitoring System**. The references are summarized below.

2.2 Literature Review

In this section topics related to A Real Time Fire and Smoke Detection System are included. These provide a sampling of problems appropriate for application of A Real Time Fire and Smoke Detection System. The references are summarized below.

[1] H. S. Raju, Sanath Shenoy Siemens Technologies and Services Private Limited in "Real-Time remote monitoring and operation of commercial Devices using IoT and Cloud" 978-1-5090-5256-1/16/31.00- 2016 IEEE have explained recent times there has been significant advances in managing differing types of sensors and industrial devices by IoT (Internet of Things) protocol. Alongside the availability of massive amount of processing power provided by the Cloud new opportunities have emerged for complete automation of commercial devices. IoT features a vast application in several sectors and domains which are yet to be explored. In industry domain, industrial automation is need of the hour to extend the time to plug with high grade quality and enhanced productivity. During this paper we explain proven ways to utilize the capabilities of Cloud and IoT to regulate the device and analyze the info generated by them.

[2] "IoT-based Integration of IEC 61131 Industrial Automation Systems: The case of UML4IoT" by Foivos Christoulakis, 22 Page 20-28 © MANTECH PUBLICATIONS 2018. All Rights Reserved, Kleanthis Thramboulidis- 978-1-5090-0873-5/16/$31.00 ©2016 IEEE have explained that Internet of Things (IoT) plays a key role in the new

generation of industrial automation systems (IASs).Evolving IoT standards if effectively used may address many challenges in the development of IASs. However, the use of the IoT and the REST architectural paradigm that IoT is based on, is not an easy task for the automation engineer. In this paper, a model driven system engineering process is adopted for IASs and it is extended to exploit IoT standardization efforts in IEC 61131 based system. IoT is considered as an enabling technology for the integration of cyber physical and cyber components of the system and humans, bringing into the industrial automation domain the benefits of this technology. A UML profile for IoT is exploited to automate the generation process of the IoT wrapper, i.e., the software layer that is required on top of the IEC 61131 cyber part of the cyber physical component to expose its functionality to the modern IoT environment. A prototype implementation and performance measurements prove the feasibility of the presented approach.

A. Need for Wireless Fire Detection Systems using IOT: The interconnection of physical devices embedded with electronics, software and sensors, which is capable of collecting required data from the surrounding and sent those data over internet is called IOT. The fire detection composed of all of the techniques and processes that ensures the early detection of a fire. There are three main categories Smoke detection, Flame detection and Temperature detection. The automatic fire detection system will do a real-time surveillance. Mainly an automatic fire alarm system is designed for high-rise buildings. To do early extinguishing of a fire disaster, large numbers of detectors are deployed in buildings, which will periodically measure smoke concentration or temperature. There are many techniques we had been already used to detect fire, such as fire detection using sensors, image processing, CCTV technology.

B. Unified Smoke and Flame Detection for Intelligent Surveillance System: This paper explains the way of unification of flame and smoke detection algorithms by merging the common steps into a single processing flow. Smoke detection and flame detection can be done using color probability, boundary roughness, edge density and area variability filtering processes. The obtained preliminary results show that, by processing smoke and flame in parallel, we can reduce the processing time. The results obtained from unified processing part can be used to feed as input for different tasks of intelligent surveillance systems.

C. Design and Implementation of Smoke Early Warning System: In this paper the author proposes a simple hardware design, which consists of different fire sensors that enables every user to use this wireless fire security system. The challenges, while designing the nodes with different types of fire sensors are discussed and the methods to overcome this problem is analyzed. The system is interfaced with the many types of sensors to sense different fire sources like gas leakage, smoke, and heat. This circuit is fully controlled by the PIC micro-controller. All the sensors and detectors are wired to PIC micro-controller. The PIC micro-controller will periodically monitor all the connected sensors and when it senses any security problem then notification sent to the PC central monitoring station wirelessly. The distance should be in a range of 300m indoor/1500m outdoor using zigbee. Also gas, light, smoke, IR, temperature & humidity sensors can be interfaced with this micro-controller to find out abnormal fire conditions in the surroundings in all possible ways. ARM7 based micro-controller can also be used to build such systems.

2.3 Summary

We try to do this project by reading the above literature, and we have been able to make our project successful by reducing the mistakes of last year's project.

CHAPTER 3

HARDWARE AND SOFTWARE ANALYSIS

3.1 Node MCU

Node MCU is an open source firmware for which open source prototyping board designs are available. The name "Node MCU" combines "node" and "MCU" (micro-controller unit). The term "Node MCU" strictly speaking refers to the firmware rather than the associated development kits. Both the firmware and prototyping board designs are open source.

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 and SPIFFS. 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.

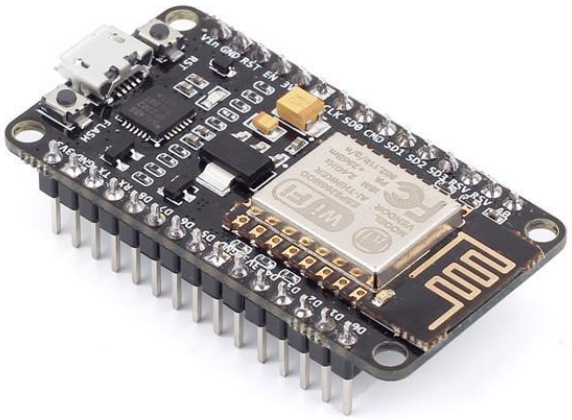


Figure 3.1: Node MCU

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 was 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.

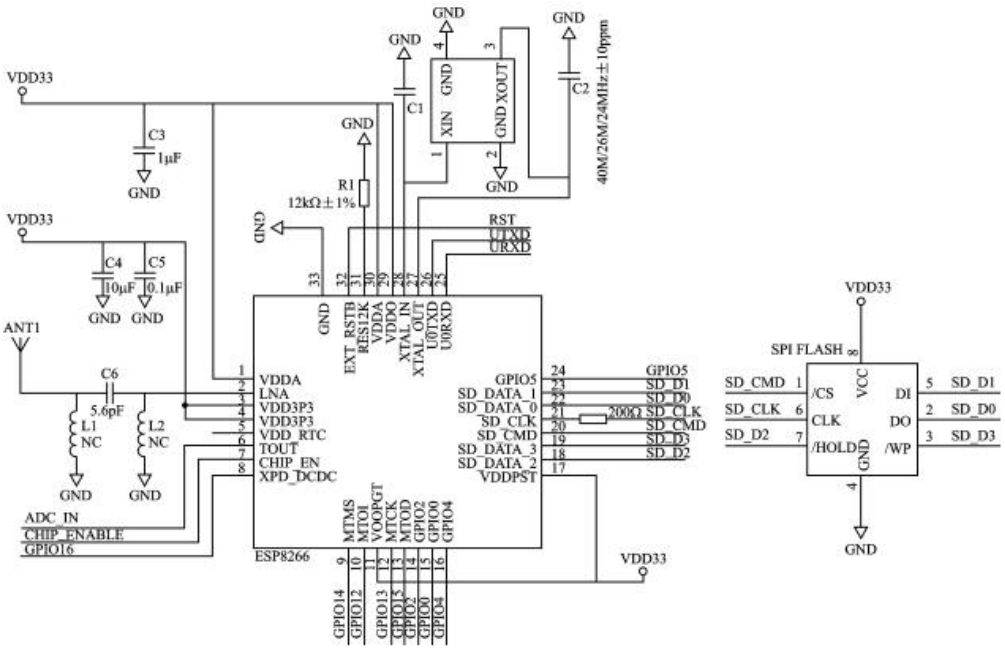


Figure 3.2: Node MCU Schematic Diagram

This an open source IoT platform. It includes firmware which runs on the ESP8266 Wi-Fi SoC from Express if Systems, and hardware which is based on the ESP-12 module. The term "Node MCU" by default refers to the firmware rather than the development kits. The firmware uses the Luascripting language. It is based on the eLua project, and built on the Espress if Non-OS SDK for ESP8266. Node MCU was created shortly after the ESP8266 came out. On December 30, 2013, Espressif Systems began production of the ESP8266. The ESP8266 is a Wi-Fi SoC integrated with a Tensilica Xtensa LX106 core, widely used in IoT applications (see related projects). Node MCU started on 13 Oct 2014, when Hong committed the first file of node mcu-firmware to GitHub. Two months later, the project expanded to include an open-hardware platform when developer Huang R committed the gerber file of an ESP8266 board, named devkit v0.9. Later that month, Tuan PM ported MQTT client library from Contiki to the ESP8266 SoC platform, and committed to Node MCU project, then Node MCU was able to support the MQTT IoT protocol, using Lua to access the MQTT broker. Another important update was made on

30 Jan 2015, when Devsaurus ported the u8glibto Node MCU project, enabling Node MCU to easily drive LCD, Screen, OLED, even VGA displays. In summer 2015 the creators abandoned the firmware project and a group of independent contributors took over. By summer 2016 the Node MCU included more than 40 different modules. Due to resource constraints users need to select the modules relevant for their project and build a firmware tailored to their needs.

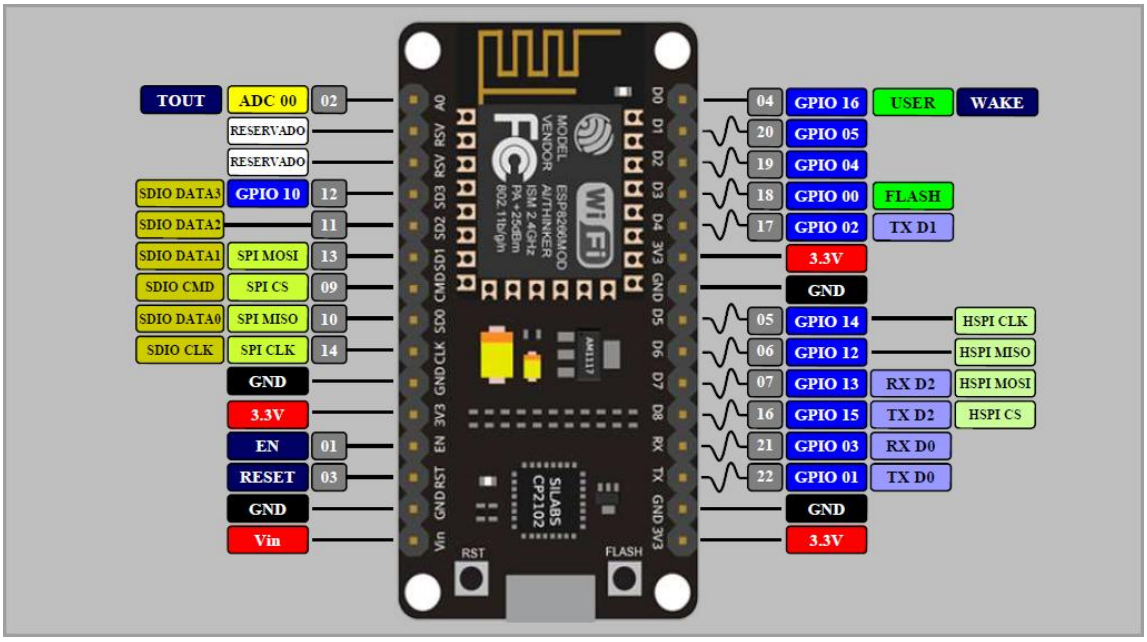


Figure 3.3: Node MCU Pin Out

Node MCU V3 ESP8266 ESP-12E is WiFi development board that helps you to prototype your IoT product with few Lua script lines, or through Arduino IDE. The board is based on ESP8266 ESP-12E variant, unlike other ESP-12E, you won't need to buy a separate breakout board, usb to serial adapter, or even solder it to a PCB to get started, you will only need a usb cable (Micro USB).

Features

1. Communication interface voltage: 3.3V.
2. Antenna type: Built-in PCB antenna is available.
3. Wireless 802.11 b/g/n standard
4. WiFi at 2.4GHz, support WPA / WPA2 security mode

5. Support STA/AP/STA + AP three operating modes
6. Built-in TCP/IP protocol stack to support multiple TCP Client connections (5 MAX)
7. D0 ~ D8, SD1 ~ SD3: used as GPIO, PWM, IIC, etc., port driver capability 15mA
8. AD0: 1 channel ADC
9. Power input: 4.5V ~ 9V (10VMAX), USB-powered
10. Current: continuous transmission: ≈70mA (200mA MAX), Standby: <200uA
11. Transfer rate: 110-460800bps
12. Support UART / GPIO data communication interface
13. Remote firmware upgrade (OTA)
14. Flash size: 4MByte.

3.2 SMPS

A switched-mode power supply is an electronic power supply that incorporates a switching regulator to convert electrical power efficiently. Like other power supplies, an SMPS transfers power from a DC or AC source to DC loads, such as a personal computer, while converting voltage and current characteristics.



Figure 3.4: SMPS Power Supply

A switched-mode power supply (switching-mode power supply, switch-mode power supply, switched power supply, SMPS, or switcher) is an electronic power supply that incorporates a switching regulator to convert electrical power efficiently.

Like other power supplies, an SMPS transfers power from a DC or AC source (often mains power) to DC loads, such as a personal computer, while converting voltage and current characteristics. Unlike a linear power supply, the pass transistor of a switching-mode supply continually switches between low-dissipation, full-on and full-off states, and spends very little time in the high dissipation transitions, which minimizes wasted energy. Ideally, a switched-mode power supply dissipates no power. Voltage regulation is achieved by varying the ratio of on-to-off time. In contrast, a linear power supply regulates the output voltage by continually dissipating power in the pass transistor. This higher power conversion efficiency is an important advantage of a switched-mode power supply. Switched-mode power supplies may also be substantially smaller and lighter than a linear supply due to the smaller transformer size and weight.

Switching regulators are used as replacements for linear regulators when higher efficiency, smaller size or lighter weight are required. They are, however, more complicated; their switching currents can cause electrical noise problems if not carefully suppressed, and simple designs may have a poor power factor.

3.3 5V Regulator IC

Voltage sources in a circuit may have fluctuations resulting in not providing fixed voltage outputs. A voltage regulator IC maintains the output voltage at a constant value. 7805 IC, a member of 78xx series of fixed linear voltage regulators used to maintain such fluctuations, is a popular voltage regulator integrated circuit (IC). The xx in 78xx indicates the output voltage it provides. 7805 IC provides +5 volts regulated power supply with provisions to add a heat sink.

7805 IC Rating

- Input voltage range 7V- 35V
- Current rating Ic = 1A
- Output voltage range VMax=5.2V ,VMin=4.8V

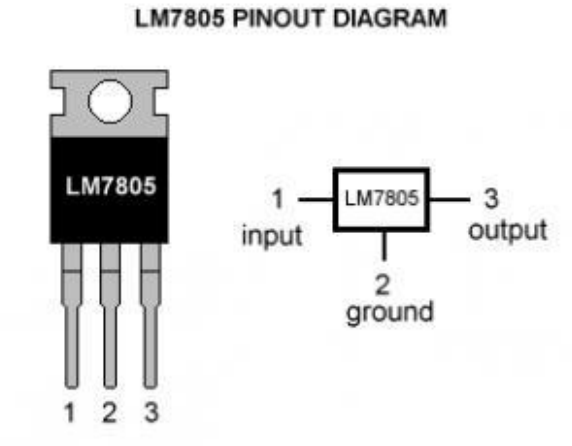
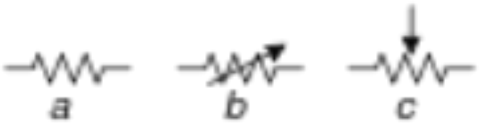


Figure 3.5: 5V Regulator IC

3.4 Resistor

A **resistor** is a passive two-terminal electrical component that implements electrical resistance as a circuit element. In electronic circuits, resistors are used to reduce current flow, adjust signal levels, to divide voltages, bias active elements, and terminate transmission lines, among other uses. High-power resistors that can dissipate many watts of electrical power as heat, may be used as part of motor controls, in power distribution systems, or as test loads for generators. Fixed resistors have resistances that only change slightly with temperature, time or operating voltage. Variable resistors can be used to adjust circuit elements (such as a volume control or a lamp dimmer), or as sensing devices for heat, light, humidity, force, or chemical activity.

Two typical schematic diagram symbols are as follows:



(a) resistor, (b) rheostat (variable resistor), and (c) potentiometer



IEC resistor symbol

One common scheme is the RKM code following IEC 60062. It avoids using a decimal separator and replaces the decimal separator with a letter loosely associated with SI prefixes corresponding with the part's resistance. For example, *8K2* as part marking code, in a circuit diagram or in a bill of materials (BOM) indicates a resistor value of 8.2 kΩ. Additional zeros imply a tighter tolerance, for example *15M0* for three significant digits. When the value can be expressed without the need for a prefix (that is, multiplicator 1), an "R" is used instead of the decimal separator. For example, *1R2* indicates 1.2 Ω, and *18R* indicates 18 Ω.

Specifications:

- Resistance: 220 Ohms
- Power (Watts): 0.25W, 1/4W
- Temperature Coefficient: 350ppm/Celcius
- Tolerance: +/- 5%
- Case: Axial
- Size: 0.094" Dia x 0.248" L (2.40mm x 6.30mm)

3.5 Flame Sensor

The flame sensor can detect flame and infrared light sources with wavelengths ranging from 760 nm to 1100 nm. It uses the LM393 comparator chip, which gives a clean, stable digital output signal and driving ability of 15 mA. This flame detector that can be used in fire alarms and other fire detecting devices.

Specifications

- LM393 comparator chip
- Detection Range: 760 nm to 1100 nm
- Operating Voltage: 3.3 V to 5 V
- Maximum Output Current: 15 mA
- Digital Outputs: 0 and 1
- Detection Angle: about 60 degrees

- Adjustable sensitivity via potentiometer
- LED lights indicators: power (red) and
- digital switching output (green)
- Fixed bolt holes for easy installation

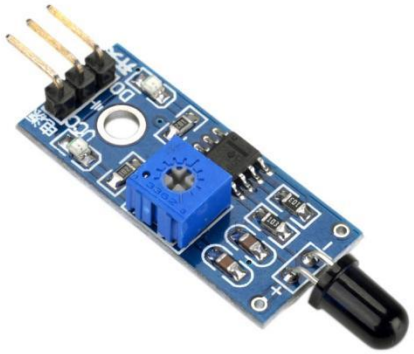


Figure 3.6: Flame Sensor

3.6 MQ 2 Gas Sensor

The utility model can be used for gas leakage monitoring devices in families and factories, and is suitable for the detection of liquefied petroleum gas, butane, propane, methane, Hydrogen, smoke, etc. This is a very easy to use low cost semiconductor Gas sensor Module with analog and digital output.



Figure 3.7: MQ 2 Gas Sensor

Features:

- Adopt high quality double panel design, with power indication and TTL signal output indication.
- It has DO switch signal (TTL) output and AO analog signal output.
- TTL output valid signal is low level. When the output is low, the signal light is on, and the micro controller or relay module can be directly connected.
- The analog output voltage increases with the concentration, the higher the voltage.
- It has better sensitivity to liquefied petroleum gas, natural gas, urban gas and smoke.
- MQ-2 MQ2 Smoke Gas LPG Butane Methane Sensor Detector Module
- With four screw holes, easy to locate.
- Product size: 32 (L), *20 (W), *22 (H)
- With long service life and reliable stability.
- Fast response recovery features

Specifications:

- Input voltage: DC5V
- Power dissipation (current): 150mA
- DO output: TTL, numeric quantities 0 and 1 (0.1 and 5V)
- AO output: 0.1-0.3V (relatively pollution-free), the highest concentration of about 4V voltage
- Special reminder: after the sensor is energized, you need to preheat 20S or so, the data to be stable, sensor heating is a normal phenomenon, because the internal heating wire, if hot, it is not normal.

Connection mode:

- VCC: power supply positive (5V)
- GND: power supply negative pole
- DO:TTL switch signal output
- AO: analog signal output
- Functions: This version supporting test procedures

- Using chips: AT89S52
- Crystal oscillator: 11.0592MHZ
- Since this Gas Sensor module is sensitive to smoke it can be used in for fire detection. MQ2 Gas Sensor is also sensitive to flammable/combustible gasses like LPG, Propane & Hydrogen.
- Baud rate: 9600

3.7 Temperature Sensor

The DHT11 is a basic, ultra-low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air, and spits out a digital signal on the data pin (no analog input pins needed). It's fairly simple to use, but requires careful timing to grab data.

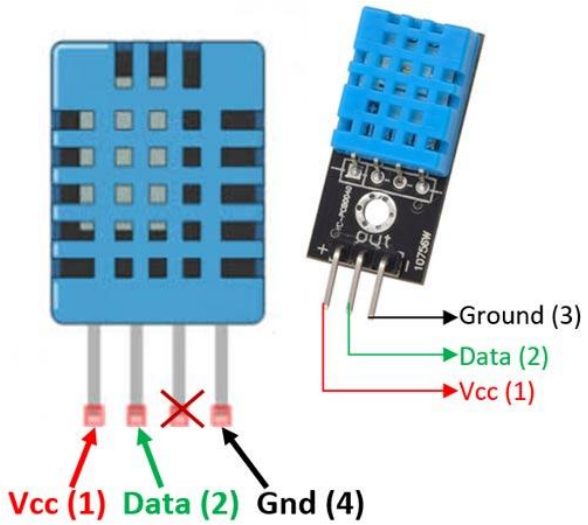


Figure 3.8: DHT11 Temperature Sensor

DHT11 Specifications:

- Operating Voltage: 3.5V to 5.5V
- Operating current: 0.3mA (measuring) 60uA (standby)
- Output: Serial data

- Temperature Range: 0°C to 50°C
- Humidity Range: 20% to 90%
- Resolution: Temperature and Humidity both are 16-bit
- Accuracy: ±1°C and ±1%

3.8 Buzzer

Piezo elements are actually one of the coolest crystals you'll play with. Flex them, they generate a voltage (pretty high, actually). Pass power through them, and they'll flex. Do it fast enough, and they'll start to make a sound! So besides making annoying little beepy noises, you could use it as a sensor to detect a knock or other similar vibration.



Figure 3.9: Buzzer

Specifications

- Operating Voltage: 1.0 ~ 20.0V
- Rated Voltage: 3.0V
- Diameter: Ø12.5mm
- Total Height: 6.3mm
- Capacitance: 15,000pF
- Current Consumption: ≤2mA

- Resonance Frequency: 4,000Hz
- Sound Pressure Level at 10cm: ≥75db
- Self Drive: No

3.9 LCD Display

LCD (Liquid Crystal Display) screen is an electronic display module and find a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. These modules are preferred over seven segments and other multi segment LEDs. The reasons being LCDs are economical; easily programmable, have no limitation of displaying special & even custom characters (unlike in seven segments), animations and so on. A 16x2 LCD means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. This LCD has two registers, namely, Command and Data. The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc.

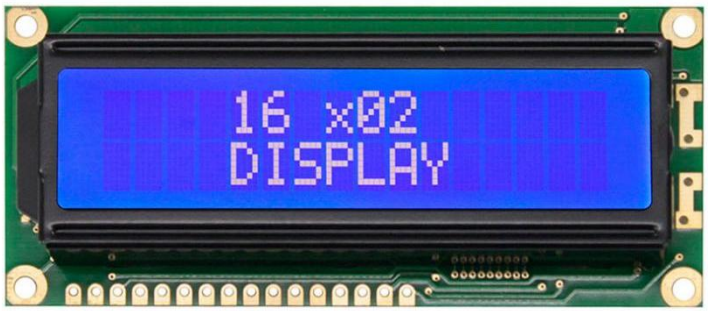


Figure 3.10: 16*2 LCD Display

Features of LCD Display:

Built-in controller (KS 0066 or Equivalent) + 5V power supply (Also available for + 3V) 1/16 duty cycle B/L to be driven by pin 1, pin 2 or pin 15, pin 16 or A.K (LED) N.V. optional for + 3V power supply.

3.10 Relay

A relay is an electrically operated switch. Many relays use an electromagnet to mechanically operate a switch, but other operating principles are also used, such as solid-state relays. Relays are used where it is necessary to control a circuit by a separate low-power signal, or where several circuits must be controlled by one signal. The first relays were used in long distance telegraph circuits as amplifiers: they repeated the signal coming in from one circuit and re-transmitted it on another circuit. Relays were used extensively in telephone exchanges and early computers to perform logical operations.



Figure 3.11: Relay

A type of relay that can handle the high power required to directly control an electric motor or other loads is called a contactor. Solid-state relays control power circuits with no moving parts, instead using a semiconductor device to perform switching. Relays with calibrated operating characteristics and sometimes multiple operating coils are used to protect electrical circuits from overload or faults; in modern electric power systems these functions are performed by digital instruments still called "protective relays".

Magnetic latching relays require one pulse of coil power to move their contacts in one direction, and another, redirected pulse to move them back. Repeated pulses from the same input have no effect. Magnetic latching relays are useful in applications where interrupted power should not be able to transition the contacts.

Magnetic latching relays can have either single or dual coils. On a single coil device, the relay will operate in one direction when power is applied with one polarity, and will reset when the polarity is reversed. On a dual coil device, when polarized voltage is applied to

the reset coil the contacts will transition. AC controlled magnetic latch relays have single coils that employ steering diodes to differentiate between operate and reset commands.

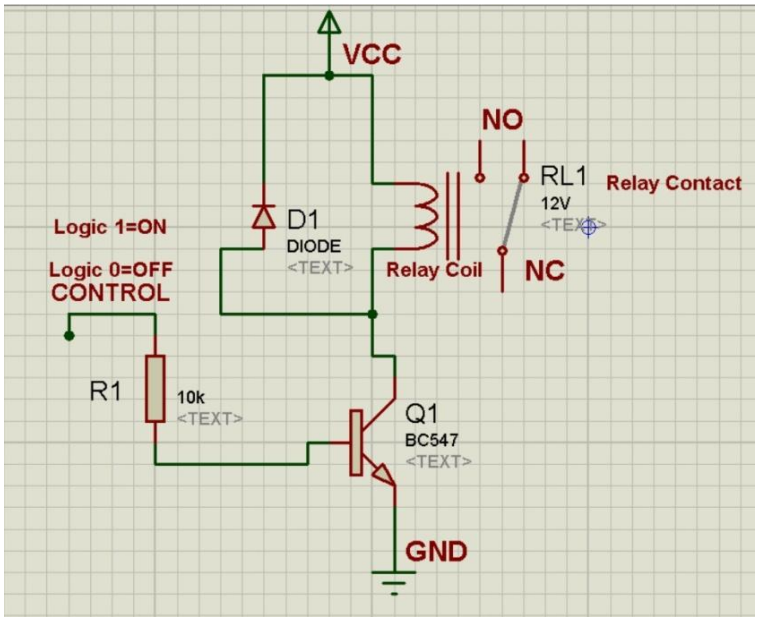


Figure 3.12: Transistor Switching Circuit.

The circuit above is called a low-side switch, because the switch – our transistor – is on the low (ground) side of the circuit. Alternatively, we can use a PNP transistor to create a high-side switch: Similar to the NPN circuit, the base is our input, and the emitter is tied to a constant voltage. A relay is an electrically operated switch of mains voltage. It means that it can be turned on or off, letting the current go through or not. Controlling a relay with the Arduino is as simple as controlling an output such as an LED. The relay module is the one in the figure below.

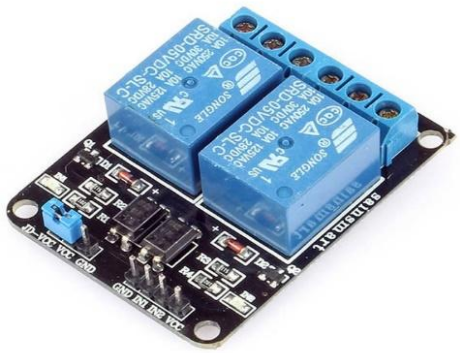


Figure 3.13: Relay Module

This module has two channels (those blue cubes). There are other varieties with one, four and eight channels.

Mains voltage connections:

In relation to mains voltage, relays have 3 possible connections:

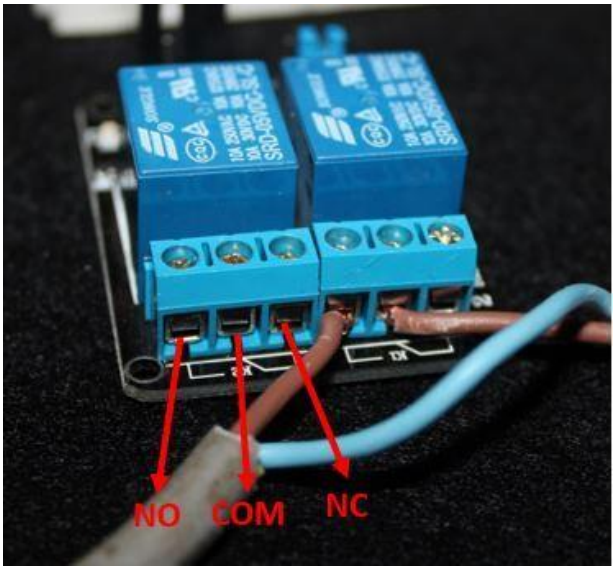


Figure 3.14: Pin diagram of Relay Module

COM: common pin

NO (Normally Open): there is no contact between the common pin and the normally open pin. So, when you trigger the relay, it connects to the COM pin and supply is provided to a load

NC (Normally Closed): there is contact between the common pin and the normally closed pin. There is always connection between the COM and NC pins, even when the relay is turned off. When you trigger the relay, the circuit is opened and there is no supply provided to a load.

Pin wiring:

The connections between the relay module and the Arduino are really simple:

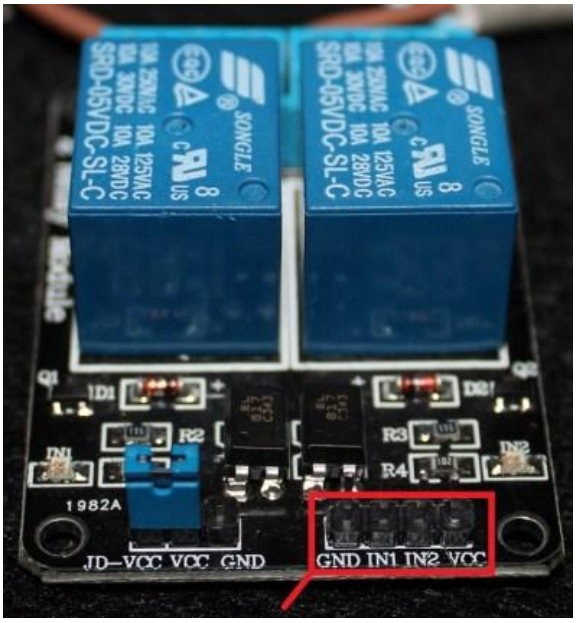


Figure 3.15: Main Voltage Connection

GND: goes to ground

IN1: controls the first relay (it will be connected to an Arduino digital pin)

IN2: controls the second relay (it should be connected to an Arduino digital pin if you are using this second relay. Otherwise, you don't need to connect it) **VCC**: goes to 5V

3.11 Arduino software

The smart microcontroller unit named as Arduino Nano can be programmed with the Arduino software. There is no any requirement for installing other software rather than Arduino. Firstly, Select "Arduino Nano from the Tools, Board menu (according to the microcontroller on your board). The IC used named as ATmega328 on the Arduino Nano comes pre burned with a boot loader that allows you to upload new code to it without the use of an external hardware programme.

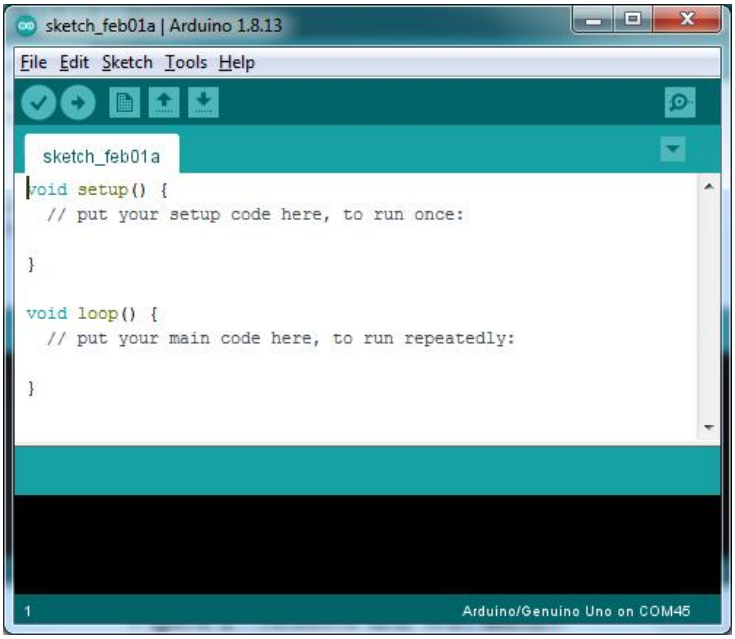


Figure 3.16: Arduino Software Interface IDE.

Communication is using the original STK500 protocol (reference, C header files). We can also bypass the boot loader and programs the microcontroller through the ICSP (In Circuit Serial Programming) header. The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available. The ATmega16U2/8U2 is loaded with a DFU boot loader, which can be activated by: On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2. On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode.

The Arduino Nano is one of the latest smart microcontroller units and has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL at (5V) with serial communication, which is available on digital pins 0 -(RX) for receive the data and pin no.1 (TX) for transmit the data. An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, an .inf file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board.

Writing Sketches

Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension .ino. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom righthand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

Sketchbook

The Arduino Software (IDE) uses the concept of a sketchbook: a standard place to store your programs (or sketches). The sketches in your sketchbook can be opened from the File > Sketchbook menu or from the Open button on the toolbar. The first time you run the Arduino software, it will automatically create a directory for your sketchbook. You can view or change the location of the sketchbook location from with the Preferences dialog.

Beginning with version 1.0, files are saved with a .ino file extension. Previous versions use the .pde extension. You may still open .pde named files in version 1.0 and later, the software will automatically rename the extension to .ino.

Tabs, Multiple Files, and Compilation

Allows you to manage sketches with more than one file (each of which appears in its own tab). These can be normal Arduino code files (no visible extension), C files (.c extension), C++ files (.cpp), or header files (.h).

Uploading

Before uploading your sketch, you need to select the correct items from the Tools > Board

and Tools > Port menus. The boards are described below. On the Mac, the serial port is probably something like /dev/tty.usbmodem241 (for an Uno or Mega2560 or Leonardo) or /dev/tty.usbserial-1B1 (for a Duemilanove or earlier USB board), or /dev/tty.USA19QW1b1P1.1 (for a serial board connected with a Keyspan USB-to-Serial adapter). On Windows, it's probably COM1 or COM2 (for a serial board) or COM4, COM5, COM7, or higher (for a USB board) - to find out, you look for USB serial device in the ports section of the Windows Device Manager. On Linux, it should be /dev/ttyACMx , /dev/ttyUSBx or similar. Once you've selected the correct serial port and board, press the upload button in the toolbar or select the Upload item from the Sketch menu. Current Arduino boards will reset automatically and begin the upload. With older boards (pre-Diecimila) that lack auto-reset, you'll need to press the reset button on the board just before starting the upload. On most boards, you'll see the RX and TX LEDs blink as the sketch is uploaded. The Arduino Software (IDE) will display a message when the upload is complete, or show an error.

When you upload a sketch, you're using the Arduino bootloader, a small program that has been loaded on to the microcontroller on your board. It allows you to upload code without using any additional hardware. The bootloader is active for a few seconds when the board resets; then it starts whichever sketch was most recently uploaded to the microcontroller. The bootloader will blink the on-board (pin 13) LED when it starts (i.e. when the board resets).

Libraries

Libraries provide extra functionality for use in sketches, e.g. working with hardware or manipulating data. To use a library in a sketch, select it from the Sketch > Import Library menu. This will insert one or more #include statements at the top of the sketch and compile the library with your sketch. Because libraries are uploaded to the board with your sketch, they increase the amount of space it takes up. If a sketch no longer needs a library, simply delete its #include statements from the top of your code.

Third-Party Hardware

Support for third-party hardware can be added to the hardware directory of your sketchbook directory. Platforms installed there may include board definitions (which appear in the board menu), core libraries, bootloaders, and programmer definitions. To install, create the hardware directory, then unzip the third-party platform into its own sub-directory. (Don't use "arduino" as the sub-directory name or you'll override the built-in Arduino platform.) To uninstall, simply delete its directory. For details on creating packages for third-party hardware, see the Arduino IDE 1.5 3rd party Hardware specification.

3.12 Proteus

The Proteus Design Suite is a proprietary software tool suite used primarily for electronic design automation. The software is used mainly by electronic design engineers and technicians to create schematics and electronic prints for manufacturing printed circuit boards.

The first version of what is now the Proteus Design Suite was called PC-B and was written by the company chairman, John Jameson, for DOS in 1988. Schematic Capture support followed in 1990, with a port to the Windows environment shortly thereafter. Mixed mode SPICE Simulation was first integrated into Proteus in 1996 and microcontroller simulation then arrived in Proteus in 1998. Shape based auto routing was added in 2002 and 2006 saw another major product update with 3D Board Visualization. More recently, a dedicated IDE for simulation was added in 2011 and MCAD import/export was included in 2015. Support for high speed design was added in 2017.Feature led product releases are typically biannual, while maintenance based service packs are released as required.

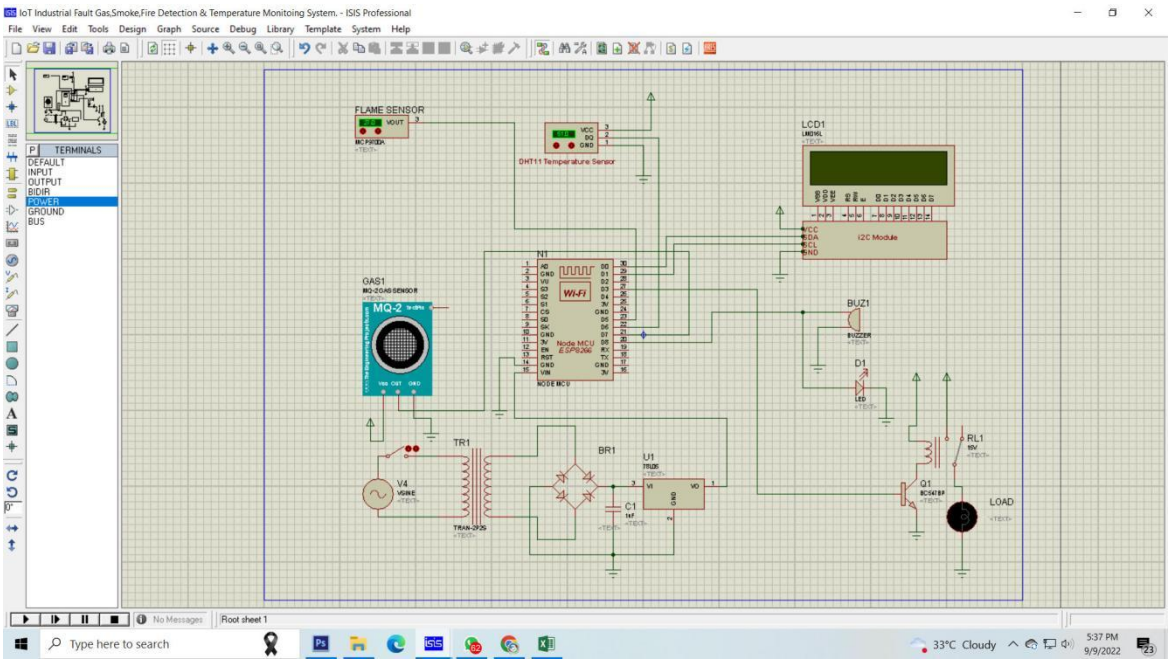


Figure 3.17: Proteus Software Interface.

3.19 Blynk Apps

Blynk is an Internet-of-Things platform designed to make development and implementation of smart IoT devices quick and easy. It can be used to read, store, and visualize sensor data and control hardware remotely.



Figure 3.18: Blynk App

Internet of Things has been all the buzz lately and more and more devices are being talking to internet every day. With the rise of such amazing technology, the risk of security has also increased substantially. Some of the major concerns in IoT are:

- If IoT devices are sending your data to the internet, the communication needs to be closed and encrypted which cannot be possible without using a dedicated and closed server which is really hard to manage.
- The IoT devices also need to be responsive and again, that is not possible without a server with low latency and high responsiveness.
- In IoT, the platform needs to be compatible with many different types of hardware architecture and devices, so that it doesn't restrict its users with single type of hardware with limited capabilities.
- Keeping in view the problems mentioned above, Blynk is the perfect solution for all these problems. Blynk consists of the following three major components:

Blynk App Details –

The mobile app developed by Blynk works as a control panel for visualizing and controlling your hardware. It is available for both Android and iOS. The app offers a very productive interface and various different widgets for different purposes. Blynk works on a currency of its own called energy. New users get 2000 amount of Blynk energy with a free Blynk account and this energy is used to buy and deploy widgets in the projects.

Blynk Server –

The most amazing component of the Blynk Platform which makes it all possible is the Blynk Server. Blynk offers a secure, responsive and centralized cloud service through its server which allows all of this communication between the devices. The Blynk server is also available as open source so you can literally make your own server and make it even more secure with a little tinkering.

Blynk Library –

The key feature of Blynk platform which makes it scalable and amazing, is the Blynk Library. The Blynk Library makes it possible to connect your hardware and get itup and running in a blink. The support for multiple hardware devices including Arduino,ESP8266 and Raspberry Pi is included in the library and it also makes it

possible to connect with hardware through many different ways of communication like Wi-Fi, Bluetooth, BLE,USB and GSM.

Features

- Similar API & UI for all supported hardware & devices
- Connection to the cloud using:
    - Wi-Fi
    - Bluetooth and BLE
    - Ethernet
    - USB (Serial)
    - GSM
    - Set of easy-to-use Widgets
- Direct pin manipulation with no code writing
- Easy to integrate and add new functionality using virtual pins
- History data monitoring via Super Chart widget
- Device-to-Device communication using Bridge Widget
- Sending emails, tweets, push notifications, etc.

CHAPTER 4

METHODOLOGY

4.1 Our methodologies for the project

Our methodologies for the project:

- Creating an idea for design and construction of an **IoT Industrial Fault Gas, Smoke, Fire Detection & Temperature Monitoring System**. And designing a block diagram & circuit diagram to know which components we need to construct it.
- Collecting all the components of our system.
- Setting up all the components in a PCB board & then soldering. Lastly, assembling all the blocks in a board and to run the system & for checking purposes.

4.2 Working Step Chart

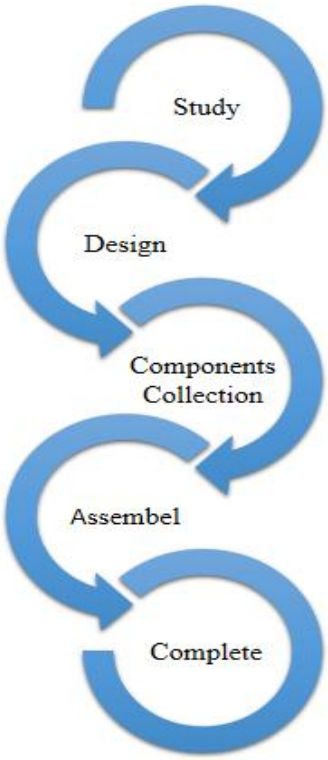


Figure 4.1: Working Step Chart

4.3 Block Diagram

The way of whole project works is that we take 220V AC power from the supply voltage and then feed it to a Switch Mode Power Supply or in short SMPS module. The SMPS simply converts the 220V AC to a pure DC of 5V 7amp. We will use this 5V DC output from the SMPS to run our controller, motor and other units.

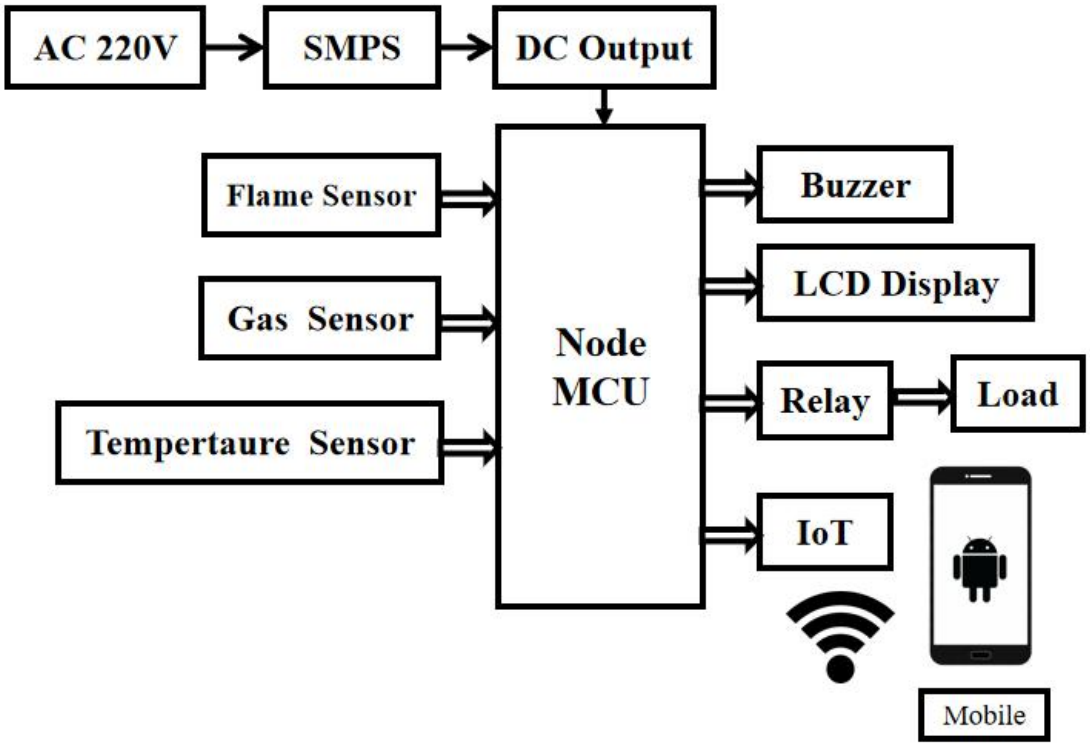


Figure 4.2: Block Diagram of Our System

4.4 Schematic Diagram

The schematic diagram here is representing the electrical circuit and the components of the project. Here we have used standardized symbols and lines.

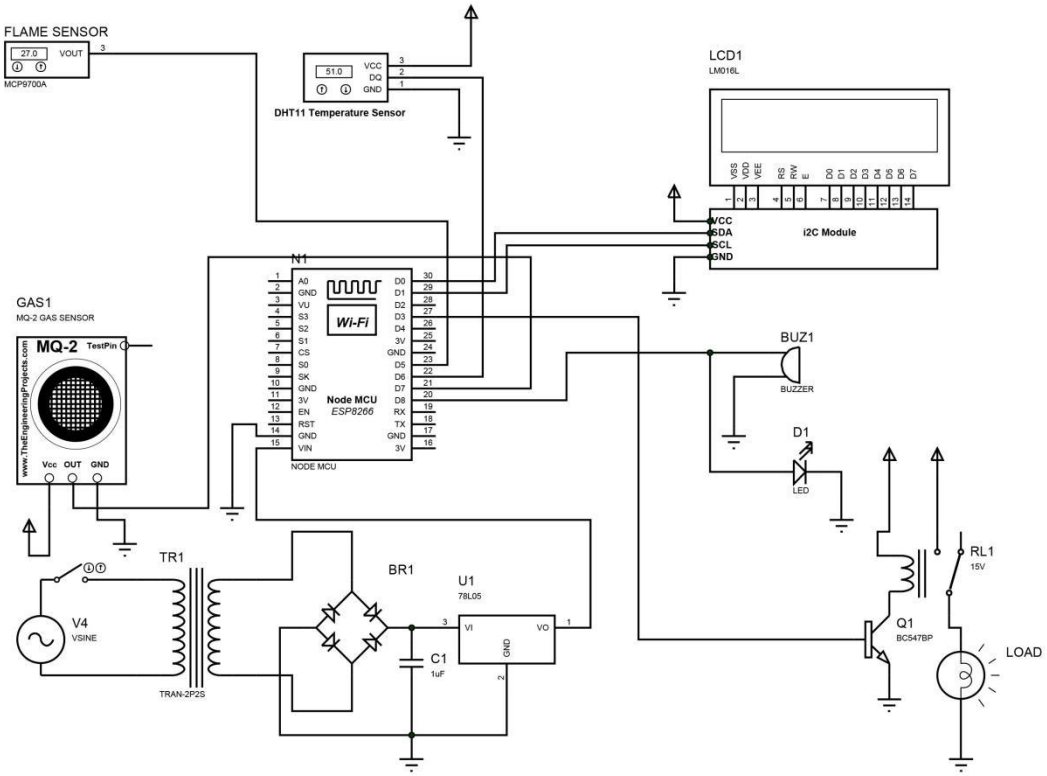


Figure 4.3: Schematic Diagram of the Project

4.5 Working Principle

The main brain of our system is the Node MCU. The way of whole project works is that we take 220V (rms) AC power from the supply voltage and then feed it to a Switch Mode Power Supply or in short SMPS module. The SMPS simply converts the 220V AC to a pure DC of 5V. We will use this 5V DC output from the SMPS to run our micro-controller, sensor and other units. Here Node MCU connected with SMPS through Voltage Input (Vin) pin. Sensor collect the reading data and send it to controller for next step. A temperature sensor detect the industrial inside temperature, flame sensor was used for detect the fire, MQ2 sensor is detect the gas. When there creates a fire then firstly detect it by flame sensor, after that when there creates some gas then it will detect by the MQ2 sensor. Then it will ring a buzzer for alerting system. All kind of operational output will be display in a LCD Display and Mobile Apps Display. This is the main process of our system.

4.6 Our Final System View

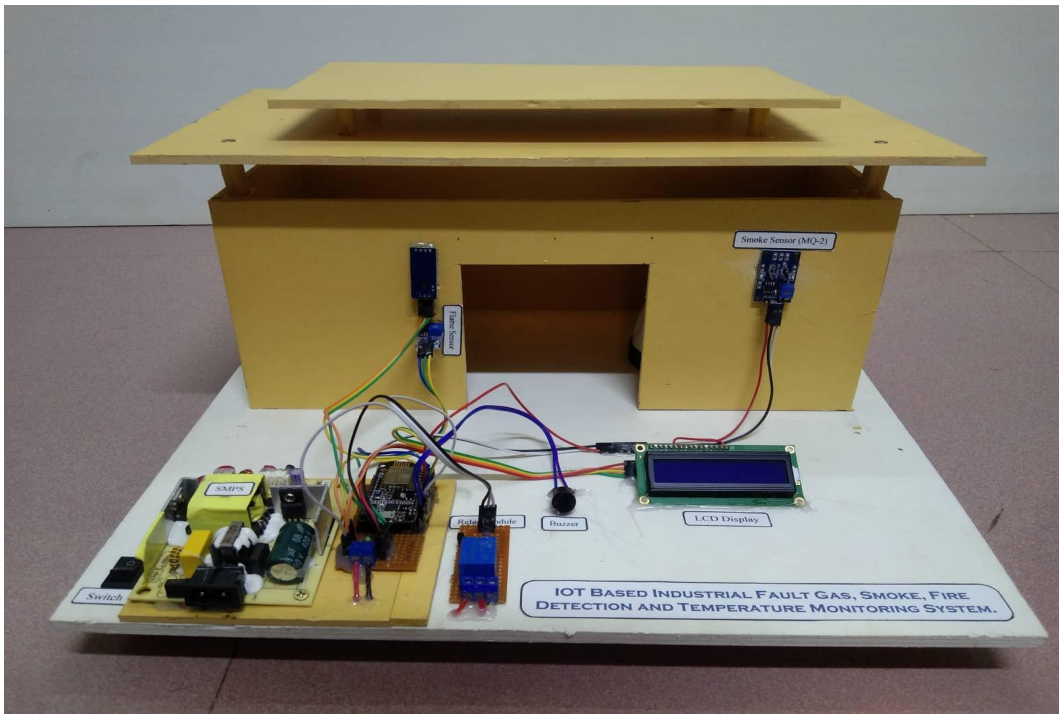


Figure 4.4: Our Final System Overview (Front View)

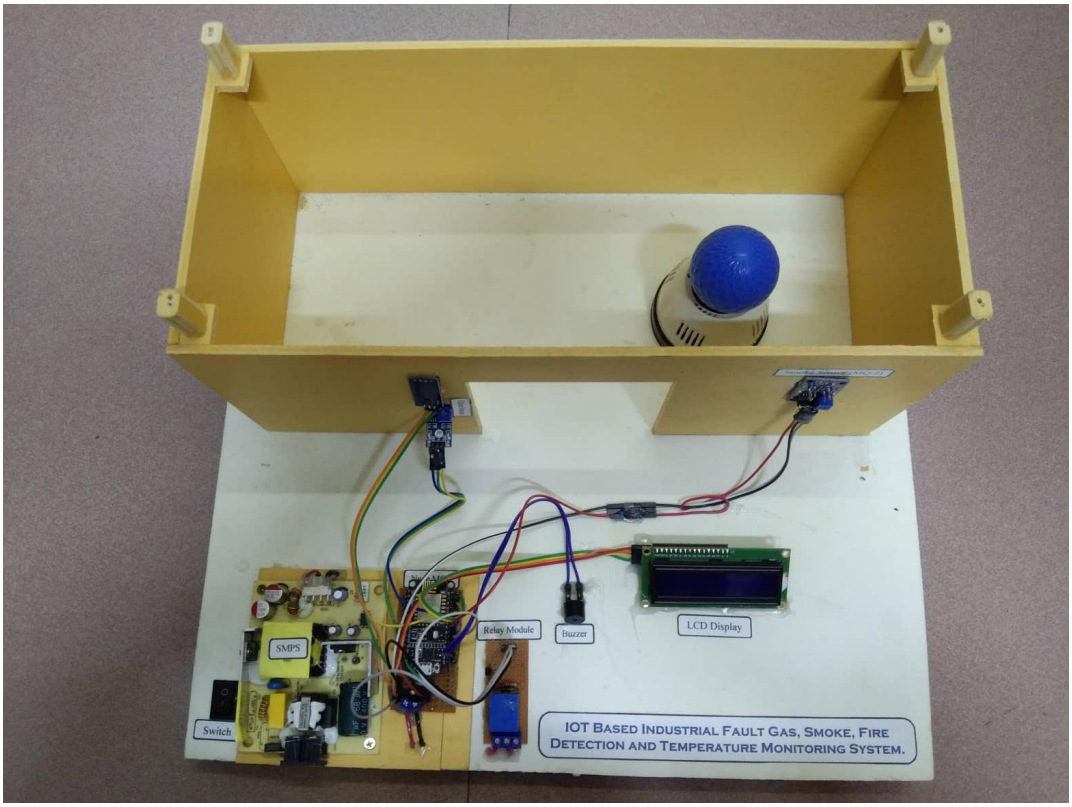


Figure 4.5: Our Final System Overview (Top View)

CHAPTER 5
RESULT AND DISCUSSION

5.1 Result

This chapter contains the results obtained and discussion about the full project. Our project is temperature, smoke and gas detection system. In our project making we used PVC boards for total hardware making. After finally completing this project, we ran it & we observed the output of this project. We can see that it is working well as expected. After making our project we observe it very careful. It works as we desire. Our project give output perfectly and all equipment are work perfectly. We check how much it works and we get perfect output from this project.

- ❑ Finally, we have completed our project successfully & check our project its run accurately according to our objective.
- ❑ At first, we start our system.
- ❑ Then it was show industrial inside temperature in LCD display and mobile display.
- ❑ Then if there create any fire and gas or smoke, incident then it will sense and ring a buzzer for alert and load automatically off.
- ❑ Beside of this work this system also send a notification in mobile apps.

5.2 Advantage

There are certainly many advantages of our project and some of the major ones have been given below:

- Can measure the temperature of inside industry.
- Can detect gas and fire incident automatically.
- Also send notification in IoT mobile apps.
- This project is easy to use
- Cost effective

- User-friendly
- Automatic and requires less human interaction
- Can be operated virtually from anywhere in the world
- The whole system consumes very little energy.

5.3 Application

This project has applications in many fields due its necessity. We have selected a few of them and they are given below:

- It can be used in any educational institutions
- It can be used in any industry or corporate offices
- It can be used in Hospitals
- It can be used in residential areas

5.4 Limitation

Though this project has many advantages but it has some limitations as well and they are listed below:

- Used cheap Chinese products for the prototype so there's some processing delay present in the circuit
- This project can now be only used for small scale purposes.
- Must be have internet to get IoT notification.

5.5 Discussion

Temperature, Fire and Gas Detection System is really needed for country right now because it can very well save people from unwanted harms. This project lets the user control the unwanted fire and gas related incident. Our project is simple to use but very smart. It doesn't require much energy to run as well. The components that were used to build the system is cheap and available so maintenance is not issue. This type of machine reduces human efforts and also provides accurate results. While doing our project, we did face some difficulties such as faulty wires, coding problem, structure building problems etc. But at the end, the results we came up with were up to the mark. As we have some

limitations in our project, so it means that there are lots of things that we can do with this project in the future and we are already focusing on that.

CHAPTER 6
CONCLUSION

6.1 Conclusion

On our system, we read three different parameters for detecting fire, these three parameters are flame, smoke, and temperature this will helps to improve the accuracy of detecting fire. It will help to detect fire detection on any condition more accurately and fast. As most of the fire detection systems sense any one parameter for detecting fire and there is a chance to take more time for detection of fire until that sensor particular parameter does not exceed above the threshold value. So, we try to overcome this issue by sensing different parameters for this reason we interface different parameters for sensing detection of fire. Also, this system is based on IoT technologies so real time sensor data and alert messages send to the concerned person so our system detects fire faster and accurately.

6.2 Future Scope

The model can be improved by making some changes in the hardware and components.
Some suggestions are given below-

- In future development this project can be develop by more sensor & alarm system.
- In future we will add camera monitoring system.

REFERENCE

[1] D. Gutmacher, U. Hoefer, and J. Wöllenstein, “Gas sensor technologies for fire detection,” *Sensors Actuators, B Chem.*, 2012, doi: 10.1016/j.snb.2011.11.053.

[2] P. S. Murvay and I. Silea, “A survey on gas leak detection and localization techniques,” *Journal of Loss Prevention in the Process Industries*. 2012, doi: 10.1016/j.jlp.2012.05.010.

[3] S. J. Chen, D. C. Hovde, K. A. Peterson, and A. W. Marshall, “Fire detection using smoke and gas sensors,” *Fire Saf. J.*, 2007, doi: 10.1016/j.firesaf.2007.01.006.

[4] E. Scorsone, A. M. Pisanelli, and K. C. Persaud, “Development of an electronic nose for fire detection,” *Sensors Actuators, B Chem.*, 2006, doi: 10.1016/j.snb.2005.12.059.

[5] S. Rappert and R. Müller, “Odor compounds in waste gas emissions from agricultural operations and food industries,” *Waste Manag.*, 2005, doi: 10.1016/j.wasman.2005.07.008.

[6] W. Krüll, R. Tobera, I. Willms, H. Essen, and N. Von Wahl, “Early forest fire detection and verification using optical smoke, gas and microwave sensors,” in *Procedia Engineering*, 2012, doi: 10.1016/j.proeng.2012.08.208.

[7] D. Gutmacher, C. Foelmli, W. Vollenweider, U. Hoefer, and J. Wöllenstein, “Comparison of gas sensor technologies for fire gas detection,” in *Procedia Engineering*, 2011, doi: 10.1016/j.proeng.2011.12.276.

[8] H. Jin, L. Zhang, W. Liang, and Q. Ding, “Integrated leakage detection and localization model for gas pipelines based on the acoustic wave method,” *J. Loss*

[9] Reshma Shinde, Ritika Pardeshi, Archana Vishwakarma and Nayan Barhate, ” Need for Wireless Fire Detection Systems using IOT ” , International Research

Journal of Engineering and Technology, Volume 04, Issue 01, pp-1078-1081, January 2017.

[10] Alexander Filonenko, Danilo Caceres Hernandez, Ajmal Shahbaz and Kang-Hyun Jo , ” Unied Smoke and Flame Detection for Intelligent Surveillance System ” , IEEE, Volume 08, Issue 03, pp-953-957, May 2016.

[11] Xiaofei Ji, Xuan Xie, Xinmeng Zuo and Jiangtao Cao, ” Design and Implementation of Smoke Early Warning System ” , IEEE, Volume 03, Issue 17, pp-351-356, July 2017.

Appendix