

AN EMBEDDED APPLICATION FOR DETECTING ADVANCE NEUROLOGICAL AND BREATHING DISORDERS BY AIR QUALITY MEASUREMENT IN REAL TIME

Submitted in partial fulfilment of the requirements for the award of
Bachelor of Engineering degree in Electronics and Communication
Engineering
by

GADDAM SAI SAMHITHA (Reg. No. 37130126)
DANTU DHANISHA (Reg. No. 37130108)



DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

SCHOOL OF ELECTRICAL AND ELECTRONICS

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

APRIL - 2021



SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**



Accredited with "A" grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **GADDAM SAI SAMHITHA** (Reg. No.37130126) and **DANTU DHANISHA** (Reg. No.37130126) who carried out the project entitled "**AN EMBEDDED APPLICATION FOR DETECTING ADVANCE NEUROLOGICAL AND BREATHING DISORDERS BY AIR QUALITY MEASUREMENT IN REAL TIME**" under my supervision from.

Internal Guide

Dr. S.LAKSHMI, M.E., Associate Professor

Head of the Department

Dr. T. RAVI, M.E., Ph.D.

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

We **GADDAM SAI SAMHITHA (Reg. No.37130126)** and **DANTU DHANISHA (Reg. No.37130108)** hereby declare that the Project Report entitled “**AN EMBEDDED APPLICATION FOR DETECTING ADVANCE NEUROLOGICAL AND BREATHING DISORDERS BY AIR QUALITY MEASUREMENT IN REAL TIME**” done by us under the guidance of **Dr. S.LAKSHMI,M.E.,Associate Professor** is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering degree in **ELECTRONICS AND COMMUNICATION ENGINEERING**.

DATE:

PLACE:

1.

2.

SIGNATURE OF THE CANDIDATES

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to Board of Management of SATHYABAMA for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

We convey our thanks to **Dr. N.M. NANDITHA, M.E., Ph.D.**, Dean, School of Electrical and Electronics and **Dr. T. RAVI, M.E., Ph.D.**, Head of the Department, Dept. of Electronics and Communication Engineering for providing us necessary support and details at the right time during the progressive reviews.

We would like to express our sincere and deep sense of gratitude to our Project Guide **Dr. S. LAKSHMI, M.E., Associate Professor** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

We wish to express our thanks to all Teaching and Non-teaching staff members of the Department of Electronics and Communication Engineering who were helpful in many ways for the completion of the project.

ABSTRACT

Environmental and health issues have risen as living conditions and economic growth have improved. The haze weather phenomenon is becoming a serious problem as air quality continues to deteriorate. It has been discovered that there is a possible connection or causal relationship between exposure to air pollution and neurological diseases. The haze weather phenomenon is integrated into fog as an extreme weather warning in many geographical areas. The design and implementation of an embedded multisensory system for air pollution and air quality monitoring with disease prevention warnings and alarms is presented in this project. This device is made up of a microcontroller and various sensors for measuring gases in the atmosphere. This system also includes a model to predict diseases in a more efficient way other than the present conventional methods. Moreover an immediate alert system is also included with display and enables data collection of these sensors for future machine learning and artificial intelligence systems.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	i
	LIST OF FIGURES	v
	LIST OF ABBREVIATIONS	vi
1	INTRODUCTION	1
2	LITERATURE SURVEY	2
3	EXISTING SYSTEM	6
4	PROPOSED SYSTEM	7
	4.1 Block diagram of proposed System	7
	4.2 Arduino	8
	4.2.1 Digital	9
	4.2.2 Analog	10
	4.2.3 Output signals	11
	4.2.4 Input signals	12
	4.3 Gas sensors	15
	4.3.1 MQ3 sensor	15
	4.3.1.1 Features of MQ3 Alcohol sensor	15
	4.3.2 MQ4 sensor	16
	4.3.2.1 MQ4 Methane gas Sensor –	
	General Specifications	16
	4.3.3 MQ6 sensor	17
	4.3.3.1 LPG gas sensor features	17

4.3.4 MQ7 sensor	17
4.3.5 MQ8 sensor	18
4.3.5.1 Features	18
4.3.5.2 Main Applications	19
4.3.6 MQ135 sensor	19
4.3.6.1 Applications	19
4.4 Buck Converter	20
4.4.1 Working of a Buck Converter	20
4.5 ESP32 microcontroller	21
4.5.1 Programming	22
4.5.2 Other chips in eESP32 Family	23
4.6 Display module	23
4.7 Arduino Software	24
4.7.1 Writing sketches	24
4.7.2 Tools	24
4.7.3 Help	25
4.7.4 Sketchbook	26
4.7.5 Tabs, Multiple files, and compilation	26
4.7.6 Uploading	26
4.7.7 Libraries	27
4.7.8 Third-party Hardware	27
4.7.9 Serial monitor	28
4.7.10 Preferences	28
4.7.11 Language Support	28

	4.7.12 Boards	29
	4.8 Node red	29
	4.8.1 Browser-based Flow editing	32
	4.9 Neurological Diseases	33
	4.9.1 Acute spinalcord Injury	33
	4.9.2 Alzheimers Disease	34
	4.9.3 Ataxia	34
	4.9.3.1 Symptoms	35
	4.9.4 Amyotrophic Lateral Sclerosis	35
	4.9.4.1 Symptoms	35
	4.9.5 Bell's Palsy	36
	4.9.5.1 Symptoms	36
	4.9.5.2 Risk Factors	37
	4.9.6 Brain Tumors	37
	4.9.6.1 Symptoms	37
	4.9.7 Cerebral Aneurysm	38
	4.9.7.1 Symptoms	38
	4.9.7.2 Risk Factors	38
	4.9.8 Epilepsy and Seizures	39
	4.9.8.1 Causes of epilepsy	39
	4.9.8.2 Symptoms	39
5	RESULT	41
6	CONCLUSION	45
	REFERENCES	

LIST OF FIGURES

FIGURE NO	TITLE	PAGE NO
4.1	Block diagram of proposed system	7
4.2	Arduino UNO	8
4.2.4	PIN Configuration of Arduino UNO	13
4.3.1	MQ3 sensor	15
4.3.2	MQ4 sensor	16
4.3.3	MQ6 Sensor	17
4.3.4	MQ7sensor	17
4.3.5	MQ8 sensor	18
4.3.6	MQ135 Sensor	19
4.4	Buck Converter	20
4.4.1.1	Step-1 Working	20
4.4.1.2	Step-2 Working	21
4.5	ESP32 Microcontroller	21
4.5.1	ESP32 Function Block Diagram	22
4.6	Ili9341 Display Module	23
4.9	The main types of nodes-input output and processing	30
4.8.1.1	Browser-based flow editing	32
4.8.1.2	Using json file save the document	33

LIST OF ABBREVIATIONS

MQTT	Message Queuing Telemetry Transport
NodeMCU	Node Microcontroller Unit
TFT	Thin Film Transistor
SPI	Serial Peripheral Interface
IDE	Integrated Development Environment
JSON File	Java Script Object Notation
ALS	AMYOTROPHIC LATERAL SCLEROSIS

CHAPTER 1

INTRODUCTION

In this project We will develop a low-cost device capable of detecting neurological disorder in advance in this project. To analyse and give alerts based on the sensor data.To provide a healthy environment for breathing using data from sensors and analytics. As we already have a Portable Multi-Sensory Module for Neurological Disease Prevention MQTT Based Air Quality Measurement System using NodeMCU and Node-RED, we can use it to monitor air quality. We are proposing Using low cost devices and sensors to measure Air quality in an area/room and predict advance disease.

Immediate alert system with nearest doctor consultation facility inside the module this can be done in Collection and reviewing of Datasheets for Air quality sensors and hardware. Using Fritzing simulator we will simulate the working of the sensors for initial testingCoding of Air quality sensors with C++ in IDE using supported libraries and logics. We will test the working of sensors in real time using Arduino Controller. With the help of a display application we will display the output of the sensors and Gather data for sensors to co relate with neurological disease data.

Code debugging and data(Obtained from sensors) is used to assign alerts and alarms. Final checking of Hardware components and Finalization of the Project and real time use case testing with the help of patients and result publication.

CHAPTER 2

LITERATURE REVIEW

Vrushali Ladekar and Rohin Daruwala – (2019 Published) Air pollution is becoming a major issue whose consequences, if not addressed, could result in serious health issues. Indoor air quality has gone unnoticed for a long time. Because we spend so much time indoors, it's critical to keep an eye on the air we breathe. The calculation and analysis of air parameters is needed to take suitable steps in a timely manner. A control module for air quality has been suggested to address this constraint. Indoor air quality is measured using various sensors in this Internet of Things project. Data is pushed and stored on the cloud, and graphs of real-time data are plotted for better visualisation. If the indoor air quality is poor, the message is displayed.

Noorfadzli Razak and Adizul Ahmad – (2018 Published) Air quality monitoring is necessary to ensure that the environments in which we work are safe for our health. Remote monitoring of environmental conditions such as air and water has been made possible by recent technological advances. This paper describes a remote air quality monitoring system based on myRIO-LabVIEW that can detect carbon monoxide levels and track air quality in indoor locations inside buildings. The MQ-7 CO sensor is attached to the Arduino Mega and MyRIO controllers and is used to detect CO. The LabVIEW programme on the user's computer processes the outputs remotely. To detect the rate of change of CO, the device was checked for over an hour at three different locations. MyRIO can save the CO rate data indefinitely while the substance is near these places. In this way, LabVIEW can process and show data to the user wirelessly or remotely. The average CO rate could then be estimated after that. Since both hardware and software can detect and log data wirelessly, the device is functioning properly. The data can then be further analysed by the users.

Fang Lide; Ding Zhenjun and all – (19 April 2010 Published) As air quality has worsened, air pollution representatives of automatic monitoring have earned greater attention. Sulfur dioxide is one of the most common pollutants in the air and a major aspect of pollution levels. The effect of sample gas temperature and flow rate on the

sulphur dioxide air quality control system is discussed in this article. The results show that the sample gas temperature and flow rate have a major impact on sulphur dioxide air quality monitoring system, and that if the relative error is not corrected, the relative error would surpass 8%. Finally, a correlation is presented, resulting in a relative error reduction of 8% to 2%.

S Kumar and A Jasuja – (2017 Published) This paper reviews and analyzes real-time air quality monitoring in order to create effective decisions in a timely manner. This paper describes a real-time standalone air quality monitoring system that includes PM 2.5, CO, CO₂, temperature, humidity, and air pressure, among other parameters. The Internet of Things is now commonly used in almost every industry, and it also plays a key role in our air quality control system. The integration of the Internet of Things and cloud computing brings a different strategy for increasing productivity data gathered and shared by the Raspberry Pi, a lowpower, low-cost ARM-based minicomputer. The framework is put to the test in Delhi, where the readings are compared to data given by the local environmental protection authority and tabulated. In IBM Bluemix Cloud, the values of the parameters computed are shown

Vasanth Pradeep V and Dr. V. Ilaiyaraja – (2018 Published) Using the MQ135 sensor, this paper created a device that can detect the amount of radioactive particles in the air. As the level of benzene in the air rises above a certain threshold, an alarm is activated. When the toxicity in the air reaches the threshold level, the cloud server displays the air quality in PPMs, and the air purifier automatically turns on and detoxifies the air. This system can be deployed anywhere, and when emissions levels exceed the safe zone, additional equipment, such as a dedicated server, can be activated. As harmful particles in the air, such as CO₂, methane, or alcohol, are observed, an alert is shown on the LCD monitor, or a notice is sent to all people present via a speaker or SMS via a GSM module.

Sumanth Reddy Enigella and Hamid Shahnasser (2019 Published) A real-time air quality management system dependent on crowd sensing is presented. The proposed method captures the air quality index from the atmosphere using low-cost, low-power devices with gas sensors. An android programme was used to send raw sensor data from the gas sensors to the cloud. The data is analyzed in the cloud. Using an open-source geospatial visual analytics platform like Leaflet and R programming, the processed data is illustrated in real time on a map.

Somphop Chanthakit and Choopan Rattanapoka (2018 Published) The air quality control unit is a hardware device that connects that sensors to monitor temperature, humidity, carbon monoxide (CO), ozone gas (O₃), and PM_{2.5} concentrations. The device's firmware turns it into a publisher, reading sensor data and sending it to a MQTT broker. Node-RED is used as a subscriber for receiving data from the MQTT Broker. We can conveniently design a flow to monitor and treat obtained data using Node-RED. The data will then be transmitted to the air quality monitoring dashboard, a flexible web application that displays data in gauges, text, and maps. Node-RED can also send a warning message to users via LINE Notify when the value of any data exceeds the configured range.

Jorge Silva and Pedro Lucas and all (2019 Published) With the environmental issues, interest in air quality monitoring is expected to rise in the near future. The majority of evidence, on the other hand, comes from a small number of government-owned sensors that can only capture a small portion of fact. As a result, improving data coverage necessitates decreasing sensor costs and making data more readily accessible. We'll do this by using a large range of low-cost sensors as the framework for an air quality monitoring network that can capture, aggregate, store, and view data. This platform would rely on streambased technology that can scale to accommodate a large number of sensors and users. The Nano Sen-AQM network that results would make large quantities of air quality data accessible to the public, with the goal of improving public health.

Manju K. Chattopadhyay (2017 Published) A concept for an environmental air quality monitoring system has been developed, which could track the concentrations of major air pollutant gases. The device makes use of low-cost air-quality control

nodes, which are made up of low-cost semiconductor gas sensors and Wi-Fi modules. Using semiconductor sensors, this device measures gas concentrations such as CO, CO₂, SO₂, and NO₂. The proposed work's main objective is to provide a low-cost infrastructure that allows for data collection and distribution to all stakeholders.

Kennedy O.Okokpujie, Etinosa Noma-Osaghae (2018 Published) Air pollution has an influence on our day-to-day lives as well as our quality of life. It threatens the planet's biodiversity and quality of life. Because of increasing industrial activity in recent years, there is a clear need to regulate air quality. People must be conscious of how their behaviours impact air quality. This project proposes a system for monitoring environmental impact. The Arduino microcontroller was used to build the device. The air pollution monitoring system was developed to constantly track and monitor air quality and log data to a central server, allowing data to be modified over the internet. The air quality was measured and analysed using the Parts per Million (PPM) metric. The developed system's air quality calculations were appropriate. The outcome was shown on the configured hardware's display interface and could be accessed from any smart mobile device via the cloud.

CHAPTER 3

EXISTING SYSTEM

3.1 INTRODUCTION:

The latest emission measurement technique employs either costly fixed-location equipment or specialised mobile equipment. Using dispersion models, the raw data collected in this manner is used to extrapolate the magnitude and concentration of emissions. This is a coarse-grained method of few and far between emission measurements. In addition, it is desirable to have access to real-time measurements to be able to quickly analyze and identify alarming levels of pollutants. Currently, access to such data is limited if not absent. It is available to and discernable by only a few who are well informed on the subject of pollution. Air Quality Monitoring using portable Multi-sensory Module for Neurological Disease Prevention. MQTT based Air quality Monitoring System using Node-MCU and Node-RED.

3.1.1 DRAWBACKS:

Sensor nodes drift in dynamic water, so localization methods based on non-fixed sensor nodes should be considered. Poison relocation is influenced by shear stream, violent stream, scattering, and corruption in convoluted water, for instance, the speed field. Compelling limitation calculations are still difficult to be given really. The contamination relocating measure is muddled; to improve on the examination issues, shift in weather conditions and compound responses just as transport and destiny models of poisons are overlooked in most contamination checking works utilizing sensor organizations. Restriction strategies considering the variables are difficult to be given.

CHAPTER 4

PROPOSED SYSTEM

To specifically Convert the Air detection Management System's Sensor data into transmitting over a fairly few distance using Arduino technology Data Acquisition from the sensor networks and display in a dashboard for sort of High level user experience Encryption of sensor data while transmission using perfect coding methods and achieving the output at a desired time without any hinderance, contrary to popular believe.

4.1 BLOCK DIAGRAM OF PROPOSED SYSTEM :

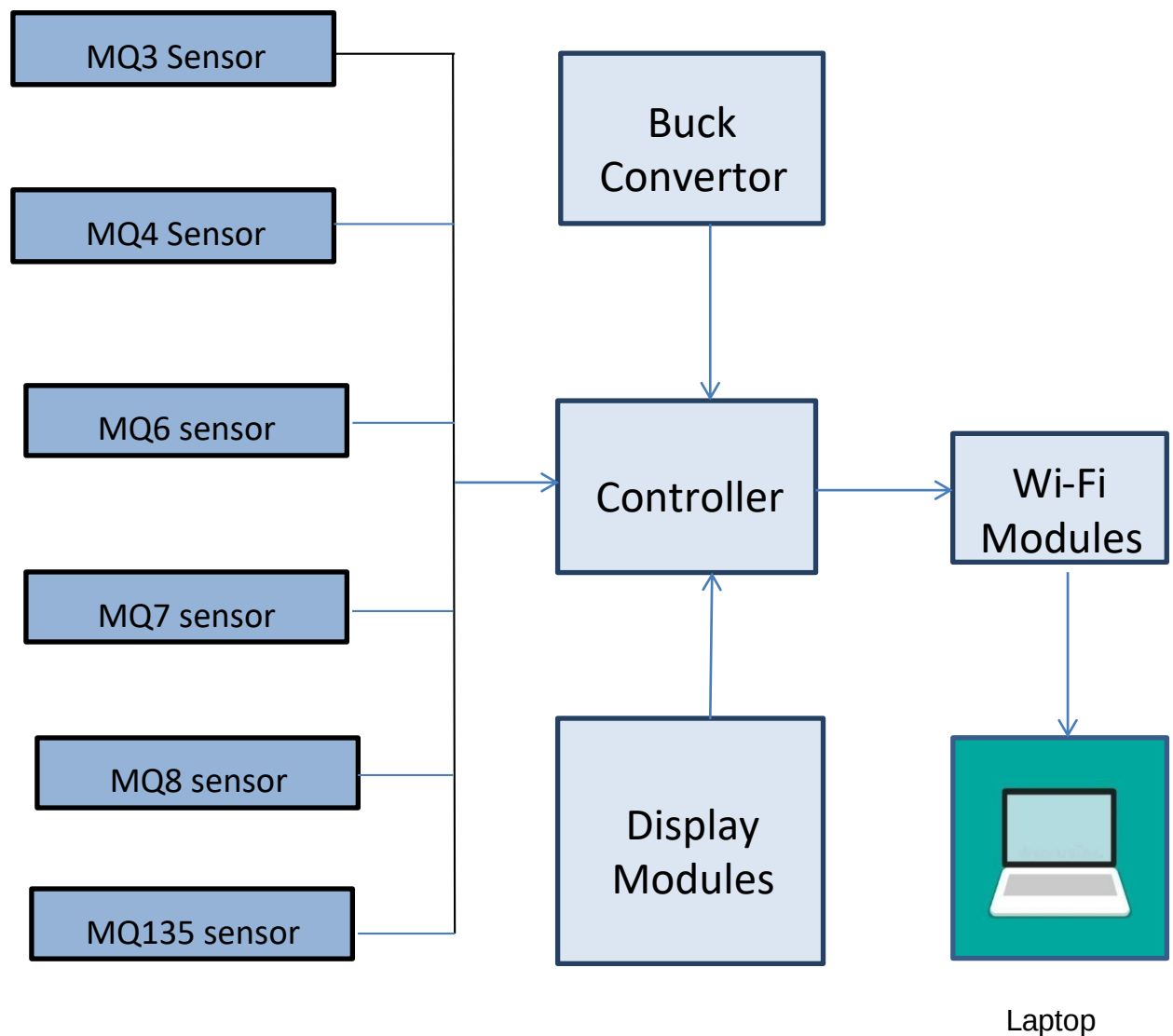


Fig:4.1 : Block diagram of proposed system

From the above figure, Sensors are connected to Arduino pins.ESP32 microcontroller,display module and buck converter is also connected to Arduino.ESP32 microcontroller has in-built wifi module and uses mqtt protocol to sent the data to Node-red platform.By using Node-red dashboard in laptop,we can get the alert message.

4.2 ARDUINO

It all depends on what you want to do with it. We'll go into all of the voltage divider's functions, which are described above.

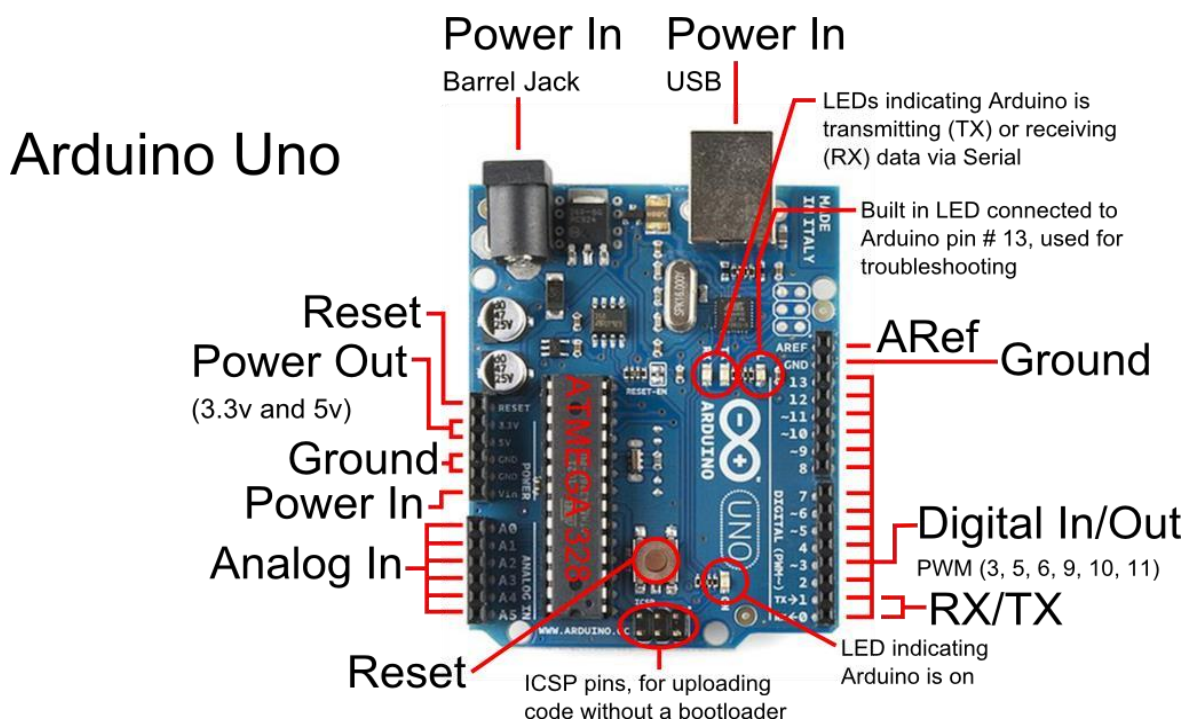


Fig:4.2 : Arduino Uno

From the above figure, to use the voltage divider as a sensor reading system, you must first determine the maximum voltage that the analogue inputs you are using to read the signal would enable. This is 5V on an Arduino. So, we always know what the optimal value for V_{out} should be. The V_{in} is clearly the amount of voltage on the circuit prior to meeting the first resistor. By looking at the Datasheet, we can able to determine the maximum voltage your sensor outputs; this is the maximum amount of voltage your sensor will allow through given the voltage in your circuit.

The only difference between powering an Analog Reference and reading a sensor is that you must compute the Vout value you choose to use as the analogue Reference.

The Arduino is capable of processing a wide range of electrical signals, both analogue and digital. It's crucial to understand the distinctions between these two types of signals, as well as how to use the data they provide.

4.2.1 DIGITAL

An electrical signal that may be the presence or absence of current, high and low voltages, or short pulses at a specific frequency and is transmitted as binary code.

Humans perceive the world in analogue form, while robots, computers, and circuits do so in digital form. The term "digital signal" refers to a signal that only has two states. These states can differ depending on the signal, but in general, they are either ON or OFF, with no in-between.

Everything in the Arduino world is done with digital signals, with the exception of Analog Input. The ON or HIGH signal of the Digital signal will match the system voltage, depending on the Arduino voltage, while the OFF or LOW signal will remain at 0V. This is a fancy way of saying that the HIGH signals on a 5V Arduino will be slightly less than 5V, while the HIGH signals on a 3.3V Arduino will be slightly less than 3.3V. Digital pins # 0 - # 13 are used by the Arduino to receive and transmit digital signals. Analog In pins may also be set to function as Digital pins. To convert Analog In pins to Digital pins, use the `pinMode(pinNumber, value)` command.

where `pinNumber` is an Analog pin (A0–A5) and `value` is Input or Output. To set up Digital pins, use the same command but specify a Digital pin instead of an Analog In pin for `pinNumber`. Since digital pins are set to input by default, you just need to change their `pinMode` to OUTPUT. Using the following command to read these pins: `pinNumber); digitalRead(pinNumber);`

pinNumber refers to the Digital pin to which the Digital part is attached. The digitalWrite command will return either a HIGH or a LOW signal. Using the order to transmit a digital signal to a pin: digitalWrite(pinNumber, value); digitalWrite(pinNumber, value);

where meaning is either HIGH or LOW and pinNumber is the number of the pin transmitting the signal.

A Pulse Width Modulation signal, which is a digital signal that acts like an analogue signal, may also be produced by the Arduino (PWM). PWM is available on Optical Pins #3, # 5, # 6, # 9, # 10, and #11. Use the following command to produce a PWM signal:

analogWrite(pinNumber, value); where pinNumber is a Digital Pin with PWM capacities and worth is a number between 0 (0%) and 255 (100%). On the Arduino UNO PWM pins are connoted by a ~ sign. For more data on PWM see the PWM worksheets or S.I.K. circuit 12. The Arduino manages electrical signs that are either info or yield.

Many of the electrical signals handled by the Arduino are either analogue or digital. Understanding the differences between these two kinds of signals, as well as how to exploit the information they contain, is critical.

4.2.2 ANALOG

Humans have an analogue perception of the globe. Everything we see and hear is a continuous stream of data reaching our senses. The temperatures we perceive are never completely hot or completely cold; rather, they fluctuate between our reasonable temperature ranges. Analog data is described by this continuous stream. Analog data is estimated using only ones and zeros of digital records, which is a complementary term to Analog.

In the Arduino world, an analogue signal is simply a signal that can be HIGH (on), LOW (off), or somewhere in between. This means that the voltage value of an Analog signal will be somewhere between 0 and 5 volts (unless you mess with the

Analog Reference pin). Analog allows you to send and receive data about devices that operate in percentages and on/off modes. The Arduino accomplishes this by sampling and matching the voltage signal sent to these pins to a voltage reference signal (5V). The Arduino then assigns a numerical value to the Analog signal based on its voltage when compared to the Analog Reference signal, which is somewhere between 0 and 1023. This number will then be used by the Arduino's digital system in calculations and drawings. Analog pins # 0 - # 5 are used by the Arduino to transmit analogue input. These pins are intended for use with Analog Output components, but they can also be used for Analog Input.

There is no need to instal anything, and you can read them with the command:

```
analogRead(pinNumber);
```

where pinNumber is the Analog In pin to which the the Analog component is connected. The analogRead command returns a number in the range of 0 to 1023.

Pulse Width Modulation is a digital signal that behaves like an analogue signal and can be output by the Arduino (PWM). PWM is available on Digital Pins #3, # 5, # 6, # 9, # 10, and #11. Use the following command to generate a PWM signal:

```
analogWrite(pinNumber, value);
```

where pinNumber is a Digital Pin with PWM capabilities and value is a number between 0 (0%) and 255 (100%). On the Arduino UNO PWM pins are signified by a ~ sign. For more information on PWM see the PWM worksheets or S.I.K. circuit 12.

The Arduino deals with electrical signals that are either input or output. Understanding the differences between these two types of signals, as well as how to exploit the information they contain, is critical.

4.2.3 OUTPUT SIGNALS

Output to the Arduino pins is always Digital, however there are two different types of Digital Output; regular Digital Output and Pulse Width Modulation Output (PWM). Output is only possible with Digital pins # 0 - # 13. The Digital pins are preset as Output pins, so unless the pin was used as an Input in the same sketch, there is no reason to use the pinMode command to set the pin as an Output. Should a situation

arise where it is necessary to reset a Digital pin to Output from Input use the command:

```
pinMode(pinNumber, OUTPUT);
```

where pinNumber is the Digital pin number set as Output. Using the following command to transmit a Digital Output signal: pinNumber, value);
digitalWrite(pinNumber, value);

where value is the signal and pinNumber is the digital pin that is outputting it. When sending a digital signal, the value may be HIGH (on) or LOW (off)

```
pinNumber, value);
```

```
analogWrite(pinNumber, value);
```

where pinNumber is a Digital Pin with PWM capacities and worth is a number between 0 (0%) and 255 (100%). On the Arduino UNO PWM pins are connoted by a ~ sign. For more data on PWM see the PWM worksheets or S.I.K. circuit 12. The Arduino manages electrical signs that are either info or yield.

Output may be transmitted to a variety of instruments, but it is up to the user to determine which type of Output signal is required, connect the hardware, and then type the appropriate code to use these signals correctly.

The Arduino deals with electrical signals that are either input or output. Understanding the differences between these two types of signals, as well as how to exploit the information they contain, is critical.

4.2.4 INPUT SIGNALS

Analog Input enters your Arduino through the Analog In pins # 0 - # 5. These signals originate from analog sensors and interface devices. These analog sensors and devices use voltage levels to communicate their information instead of a simple yes (HIGH) or no (LOW). For this reason you cannot use a digital pin as an input pin for these devices. Analog Input pins are used only for receiving Analog signals. It is

only possible to read the Analog Input pins so there is no command necessary in the setup() function to prepare these pins for input. Using the following instruction to read the Analog Input pins:

```
pinNumber = analogRead(pinNumber);
```

The Analog Input pin number is pinNumber. This function will return a value between 0 and 1023 for Analog Input. A reading of zero indicates 0 Volts, and a reading of 1023 indicates 5 Volts. The analogue sensors and interfaces emit these voltage values. If you have an Analog Input that could exceed $V_{cc} + .5V$ you may change the voltage that 1023 corresponds to by using the Aref pin. This pin sets the maximum voltage parameter your Analog Input pins can read. The Aref pin's preset value is 5V.

Either of the Digital Pins # 0 - # 13 can be used to connect to your Arduino for digital input. HIGH (on, 5V) or LOW (off, 0V) digital input signals are used (Off, 0V). Instead of this,

Input can come from many different devices, but each device's signal will be either Analog or Digital, it is up to the user to figure out which kind of input is needed, hook up the hardware and then type the correct code to properly use these signals.

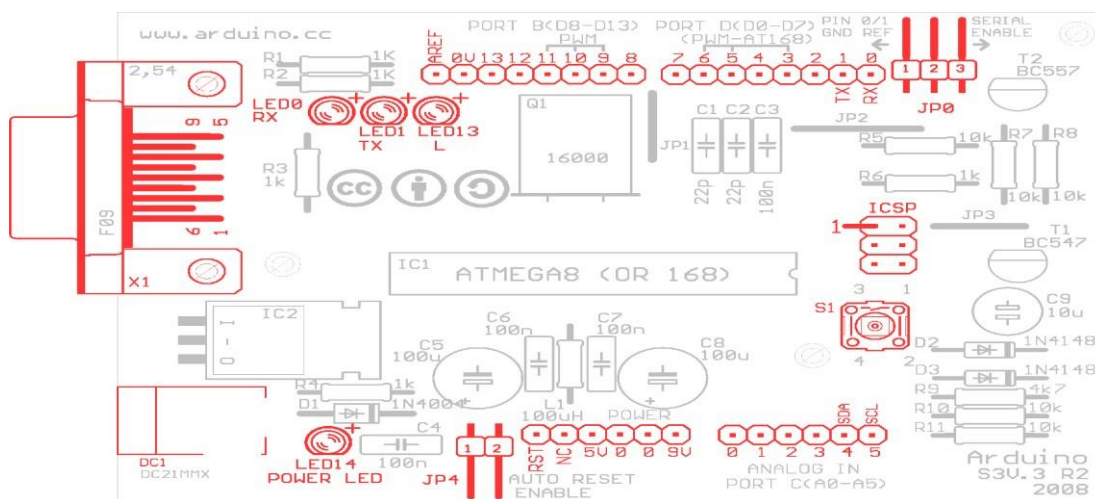


Fig:4.2.4 : Pin configuration of Arduino Uno

X1:

DE-9 serial connector

The RS-232 protocol is used to link a computer (or other device) to the internet. A serial cable with at least four pins connected (2, 3, 4, and 5) is needed. Only when JP0 is set to position 2-3 does it work.

DC1:

2.1 mm. power jack

Connects to an external power source. Positive in the middle. Regulator of voltage Works with controlled DC voltages of +7 to +20 volts (9v. to 12v. is recommended). External power may be connected using either the 9v. pin or the 5v. pin. (See POWER PINOUT for more information.) ICSP:

Header with 2x3 pins This is how you programme an Atmega with a bootloader. The location of cable pin 1 is indicated by the number 1 on both sides of the frame. It's used to upload sketches to Atmega ICs that don't have a bootloader (available only in Arduino IDE versions 0011 and 0012).

JP0:

Jumper with three pins This jumper, when set to position 2-3, allows serial connections to and from computers and devices (via the X1 connector). This is the default location. It disables serial communication and allows external pull-down resistors on pins 0 (RX) and 1 (TX) when in position 1-2. (TX). Use this just to avoid noise on RX (which appears to be incoming data to the Atmega), which can cause the sketch to fail to start.

JP4:

When in place 1-2, the 2 pins jumper activates the auto reset function, which is useful when uploading a sketch to Arduino and automatically resetting the Atmega. When uploading sketches, it eliminates the need to click the reset button (S1). If the COM Port speed on your device is not set to 19200bps, auto reset will not work. If you delete it, the auto-reset function will be disabled. When using sketches that require serial contact, this is a perfect way to avoid an unwanted Atmega reset. The DTR pulse on serial pin 4 is used for auto reset. When connecting X1 (serial connector), Arduino detects a DTR pulse, and some softwares send a DTR pulse when it begins or closes, causing Atmega to reset when it is not needed.

S1:

Button with a tactile feel This button resets the Atmega so that the uploaded sketch can be restarted or so that Arduino can receive a sketch through the serial connector (when auto reset is not active).

4.3 GAS SENSORS

4.3.1 MQ3 SENSOR



Fig:4.3.1 : MQ-3 Sensor

From the above figure, when the target alcohol gas is present, the sensor's conductivity increases, as does the gas concentration. MQ-3 gas sensor has a high sensitivity to alcohol and is resistant to gasoline, smoke, and vapour disturbance. Based on the amount of alcohol in the device, this sensor produces an analogue resistive output. As alcohol gas is present, the sensor's conductivity rises in tandem with the gas concentration.

There is a resistance within the sensor that depends based on whether or not alcohol is detected. The more alcohol there is, the lower the tolerance. This tolerance is used to calculate the amount of alcohol present. The sensor and the load resistor form a voltage divider, and the lower the sensor resistance, the higher the voltage reading.

4.3.1.1 FEATURES OF MQ-3 ALCOHOL SENSOR

The sensor has a resistance that changes depending on whether or not alcohol is detected. The smaller the tolerance, the more alcohol there is. The amount of alcohol present is calculated using this tolerance. A voltage divider is formed by the sensor and the load resistor, and the lower the sensor resistance, the higher the voltage reading.

4.3.2 MQ4 SENSOR:



Fig:4.3.2 : MQ4 Sensor

From the above figure, the MQ4 Methane Gas Sensor measures the amount of methane gas in the air and outputs an analogue voltage as a result. With a concentration sensing spectrum of 300 ppm to 10,000 ppm, leak detection is possible. The sensor might, for example, detect whether a gas stove was switched on but not lit. The sensor operates in temperatures ranging from -10 to 50°C and draws less than 150 mA at 5 V. By connecting five volts between the heating (H) pins, the sensor is held hot enough to operate properly. The sensor emits an analogue voltage on the other pins while the A or B pins are wired to five volts. A resistive load between the output pins and ground controls the detector's sensitivity.

4.3.2.1 MQ4 METHANE GAS SENSOR - GENERAL SPECIFICATIONS

- In a wide variety, good exposure to flammable gas
- Natural gas exposure is high.
- Longevity and low price
- a basic drive circuit

4.3.3 MQ6 SENSOR:



Fig:4.3.3 : MQ6 Sensor

From the above figure, this is an easy-to-use liquefied petroleum gas (LPG) sensor that detects LPG levels in the air (mostly propane and butane). Gas concentrations ranging from 200 to 10,000 parts per million can be detected by the MQ-6 (ppm). This is an easy-to-use liquefied petroleum gas (LPG) sensor that detects LPG levels in the air (mostly propane and butane).

4.3.3.1 LPG GAS SENSOR (MQ6) FEATURES

- LPG, isobutane, and propane have a high sensitivity.
- Small sensitivity to alcohol, smoke
- Detection Range of iso-butane propane 100-10,000 ppm
- Fast Response Time: <10s
- Simple drive circuit
- Heater Voltage: 5.0V

4.3.4 MQ7 SENSOR



Fig:4.3.4 : MQ7 Sensor

This CO gas sensor detects CO quantities in the air and produces an analogue voltage as a result. The sensor can measure concentrations as low as 10 parts per million to as high as 10,000 parts per million. The sensor works in temperatures ranging from -10 to 50°C and draws less than 150 milliamperes at 5 volts.

4.3.5 MQ8 SENSOR



Fig:4.3.5 : MQ8 Sensor

From the above figure, this is a simple hydrogen gas sensor that can be used to measure the presence of hydrogen in the air. The MQ-8 can detect hydrogen gas concentrations as low as 100 parts per million and as high as 10,000 parts per million (ppm).

The sensitivity of this sensor is high, and the response time is fast. The output of the sensor is an analogue resistance. The drive circuit is straightforward: simply supply 5V to the heater coil, add a load resistance, and link the output to an ADC.

The MQ-8 gas sensor's sensitive material is SnO_2 , which has a low conductivity in pure air. When hydrogen gas is present, the sensor's conductivity increases in accordance with the gas concentration. The change in conductivity can be converted to the corresponding output signal of gas concentration using a simple circuit. The MQ-8 gas sensor detects hydrogen gas with high sensitivity and does not interfere with other gases. This sensor is capable of detecting hydrogen, which is particularly useful in urban environments. It's a low-cost sensor that has a wide range of applications.

4.3.5.1 FEATURES

It has a large variety of exposure to hydrogen gas, as well as benefits such as long lifetime, low cost, and a simple drive circuit, among others.

4.3.5.2 MAIN APPLICATIONS

Domestic gas leakage alarms, commercial flammable gas alarms, and portable gas detectors all use it.

4.3.6 MQ135 SENSOR



Fig:4.3.6 : MQ135 Sensor

In air quality control devices, the MQ-135 Gas Sensors are suitable for detecting or calculating NH₃, NO_x, Alcohol, Benzene, Smoke, and CO₂. When you only want to detect one gas, the MQ-135 sensor module has a Digital Pin, which enables it to operate without a microcontroller. You'll need to use the analogue screw if you need to count gases in PPM. The analogue pin is TTL powered and operates at 5V, making it compatible with a huge variety of microcontrollers.

This sensor could be the right option for you if you need a sensor to detect or measure common air quality gases like CO₂, Smoke, NH₃, NO_x, Alcohol, or Benzene.

4.3.6.1 APPLICATIONS:

- Used to detect leakage/excess of gases like Ammonia, nitrogen oxide, alcohols, aromatic compounds, sulfide and smoke.
- Air quality monitors.

4.4 BUCK CONVERTER



Fig:4.4 : Buck Converter

Voltage regulators that use a buck converter or a step down converter. Since it's a DC-DC converter, it completes the job with the aid of a few transistor switches and an inductor. Buck converters operate in a manner that is identical to PWM 'dimming.' We've all heard of a PWM signal dimming lights.

4.4.1 WORKING OF A BUCK CONVERTER

The working of a buck converter can be broken down into a few steps.

STEP – 1:

The switch is turned on, allowing current to flow to and charge the output capacitor. Since the voltage across the capacitor cannot increase rapidly and the charging current is regulated by the inductor, the voltage across the capacitor during the switching cycle doesn't quite equal the highest voltage of the power supply.

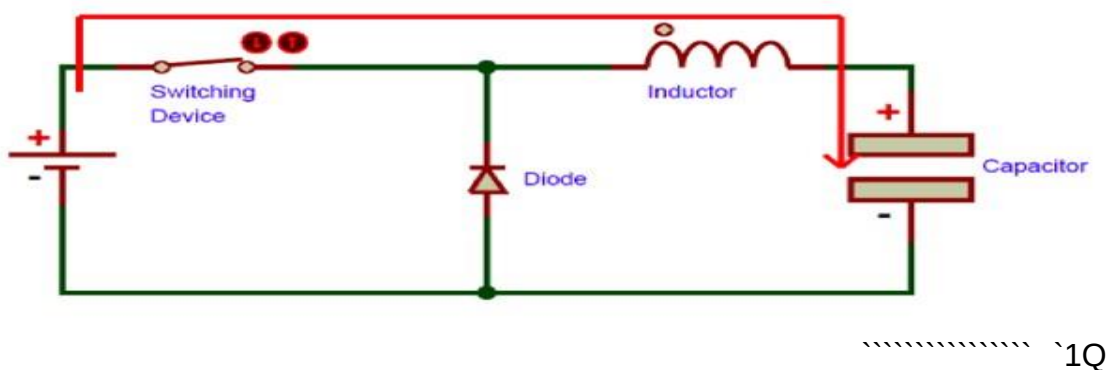


Fig:4.4.1.1 : Step 1 Working

STEP – 2:

The light has now been switched off. Since an inductor's current cannot shift unexpectedly, it induces a voltage through it. When the switch is switched off, this voltage is allowed to charge the capacitor and power the load via the diode, retaining current output current throughout the switching cycle.

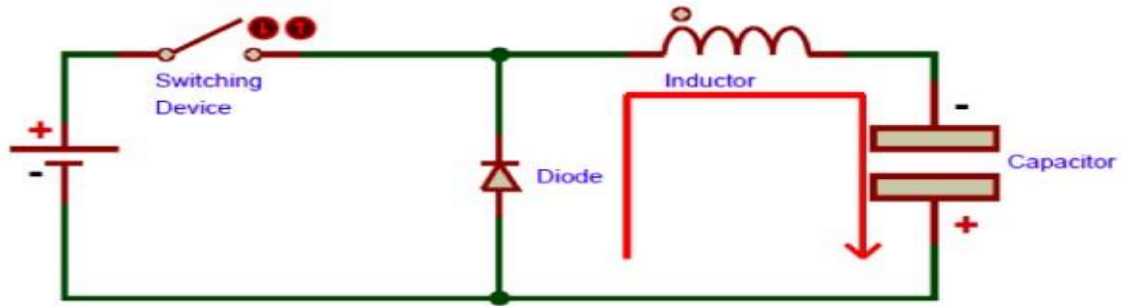


Fig:4.4.1.2 : Step 2 Sorking

These two steps are repeated thousands of times per second, resulting in a continuous stream of output.

Note that the design process outlined here is not meant to be an expert rant; rather, these are really specific equations for getting the first buck converter up and running.

Again, regulation is a topic that has not been discussed in this paper. If you want to make a controlled buck converter, you can use a buck controller IC like the LM2596, which is designed specifically for that purpose.

4.5 ESP32 MICROCONTROLLER:

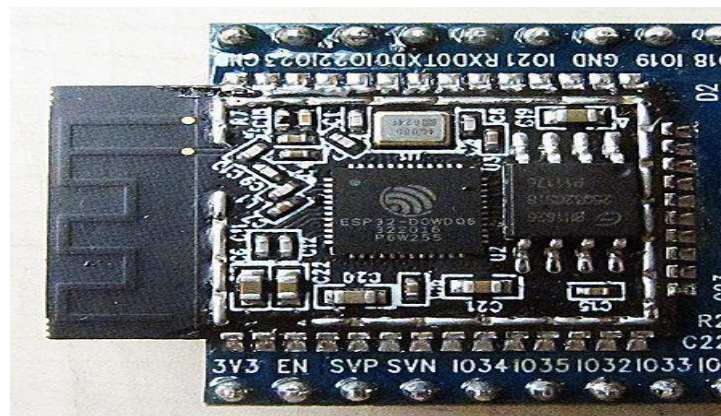


Fig:4.5 : ESP32 Microcontroller

The ESP32 is a low-cost, low-power system-on-a-chip (SoC) microcontroller with built-in Wi-Fi and dual-mode Bluetooth. Antenna switches, RF baluns, control amplifiers, low-noise receive amplifiers, filters, and power-management modules are all integrated into the ESP32 range, which also includes a Tensilica Xtensa LX6

microprocessor in dual-core and single-core models. A Shanghai-based Chinese company, Espressif Systems designed and produced the ESP32, which is manufactured by TSMC using their 40 nm process. [two] It is the ESP8266 microcontroller's successor.

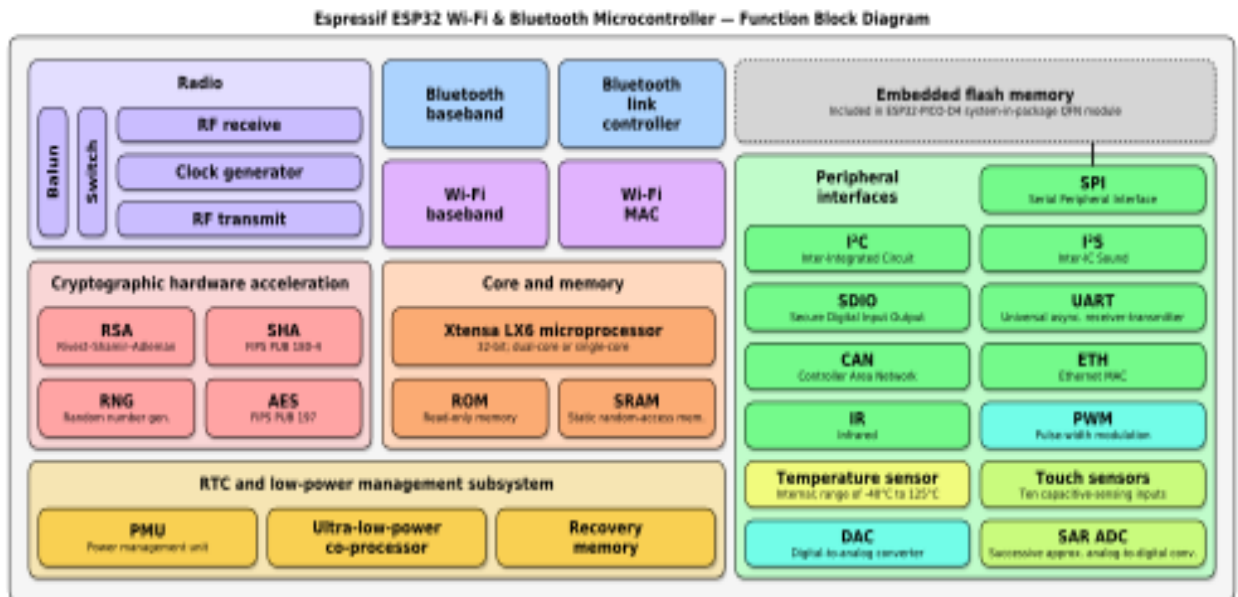


Fig:4.5.1 : ESP32 Function Block Diagram

From the above figure, Time Protocol support

- o CAN bus 2.0
- o Infrared remote controller (TX/RX, up to 8 channels)
- o Motor PWM
- o LED PWM (up to 16 channels)
- o Hall effect sensor
- o Ultra low power analog pre-amplifier

Security:

- o IEEE 802.11 standard security features all supported, including WPA, WPA2, WPA3

4.5.1 PROGRAMMING

ESP32 programming languages, systems, platforms, and environments include:

The ESP32 Arduino Core is supported by the Arduino IDE.

Espressif IoT Development Framework (IDF) – Espressif's official development framework for the ESP32 microcontroller.

MicroPython is a Python programming language. A microcontroller-friendly Python 3 implementation

Mesh Development Framework by Espressif

Espruino is a JavaScript SDK and firmware that highly resembles the Node.js framework.

4.5.2 OTHER CHIPS IN ESP32 FAMILY

In 2019, Espressif launched the ESP32-S2, a new chip based on a single Xtensa LX7 core CPU with up to 240 MHz clock speed. It has 320 kB SRAM, 128 kB ROM, and 16 kB RTC memory and can perform 64 FLOPS per loop. [52]

On December 1st, 2020, Espressif released the ESP32-S3, a new microcontroller with a dual-core Xtensa LX7 CPU core running at up to 240MHz, 384 KB of RAM, and another 384 KB of SRAM. The LX7 core on the Espressif platform provides new LX7 instruction set extensions aimed at enhancing the performance of Artificial Intelligence applications.

The S3 family incorporates the S2 family's improved hardware-level security, as well as other ESP32-S2 features such as USB On-The-Go support and an improved touchpad sensor implementation. There are up to 44 programmable GPIOs on the ESP32S3. It also includes some of the features found in the first generation of ESP32 microcontrollers, such as Bluetooth 5 and Bluetooth LE support, an SD/MMC host controller, and a "World Controller" that provides a completely isolated execution environment for security-critical applications.

4.6 DISPLAY MODULE



Fig:4.6 : ILI 9341 Display Module

From the above figure, the display controller for the ILI9341 TFT module is also called ILI9341. It's a low-cost, easy-to-use colour display that uses the SPI interface protocol and includes 4 or 5 control pins. This TFT monitor has a resolution of 240 x 320 pixels, or 76800 pixels. This module only accepts 3.3V and does not support 5V. (not 5V tolerant).

The ILI9341 controller chip is included in this LCD board, and it allows for different pin-saving SPI communication without losing data transmission speed.

Since the LCD module's interface is SPI, users would need 6 Dupont lines (SCK,MISO,MOSI,CS,D/C,RST) to control it. An SD card slot is also included in this module.

4.7 ARDUINO SOFTWARE

A text editor for writing code, a message area, a text console, a toolbar with buttons for basic functions, and a set of menus are all included in the Arduino Software (IDE). Through linking to them and importing programmes, it connects with the Arduino and Genuino hardware.

4.7.1 WRITING SKETCHES

Sketches are programmes created with the Arduino Software (IDE). These sketches were made in a text editor and saved with the.ino file extension. Cutting/pasting, as well as searching/replacing text, are all available in the editor. The message area shows errors and provides input when saving and exporting. The Arduino Program (IDE) outputs text to the console, which includes complete error messages and other information. The designed board and serial port are shown in the window's bottom righthand corner. You can verify and upload programmes, make, open, and save sketches, and open the serial monitor using the toolbar buttonsNB: Previous versions of the Arduino Software (IDE) used the extension.pde to save sketches. These files can be opened with version 1.0, but you'll be asked to save the sketch with the.ino extension when you save it.

The five menus Paper, Edit, Sketch, Resources, and Support include additional commands. The menus are context sensitive, which means they only display things that are important to the work being performed at the moment.

4.7.2 TOOLS

- *The coding is indented such that the opening and closing curly braces line up, and the sentences inside curly braces are indented even further.*
- Sketch from the Archives

Store a copy of the latest sketch in the.zip format. The archive is saved in the same directory as the drawing.

- Reload and fix the encoding

Fixes any inconsistencies between the editor's char map encoding and the char maps of other operating systems.

- *Board*

Choose the board you want to use. The outlines of the different boards can be found below.

- *Port*

This menu lists all of your machine's serial machines, both actual and virtual. When you open the top-level tools menu, it should immediately refresh.

☐ *Programmer*

When programming a board or chip using a hardware programmer rather than the onboard USB-serial link. You won't need this most of the time, but if you're burning a bootloader to a new microcontroller, you'll need it.

☐ *Burn Bootloader*

The items in this menu enable you to burn a bootloader onto an Arduino board's microcontroller. This is not needed for regular Arduino or Genuino board use, but it will come in handy if you buy a new ATmega microcontroller (which normally come without a bootloader). Before burning the bootloader on the target board, make sure you've picked the right board from the Boards menu.

4.7.3 HELP

A variety of documents related to the Arduino software can be found here (IDE). If you don't have internet connectivity, you can access Getting Started, Reference, this IDE guide, and other documents locally. The documents are a localised version of the online versions and may provide a link to our website.

This is the Help menu's only interactive feature: it selects the appropriate page for the function or order under the cursor from the local copy of the Reference.

4.7.4 SKETCHBOOK

The Arduino Software (IDE) is based on the idea of a sketchbook, which is a typical location for storing your programmes (or sketches). The Open button on the toolbar or the File > Sketchbook menu can be used to open the sketches in your sketchbook. The Arduino programme will build a directory for your sketchbook the first time you run it. With the Preferences dialogue, you can see or adjust the position of the sketchbook.

Files are saved with a .ino extension starting with version 1.0. The .pde extension was used in previous versions. In version 1.0 and later, you can still open .pde files; however, the programme will automatically rename the extension to .ino.

4.7.5 TABS, MULTIPLE FILES, AND COMPILATION

Allows you to handle drawings from several files (each of which appears in its own tab). There can be plain Arduino code (no extension), C files (.c extension), C++ files (.cpp extension), or header files (.h).

4.7.6 UPLOADING

Before uploading your drawing, make sure you choose the correct things from the Tools > Board and Tools > Port menus. The boards are listed in the following sections. The serial port on a Mac is possibly /dev/tty.usbmodem241 (for an Uno, Mega2560, or Leonardo), /dev/tty.usbserial-1B1 (for a Duemilanove or earlier USB board. On Windows, it's most likely COM1 or COM2 (for a serial board) or COM4, COM5, COM7, or higher (for a USB board); to find out, look in the ports section of the Windows System Manager for USB serial device. It should be /dev/ttyACMx, /dev/ttyUSBx, or something similar on Linux. If you've found the correct serial port and board, click the import button in the toolbar or choose the Upload object from the Sketch menu. Current Arduino boards will reboot and begin uploading automatically.. You'll need to click the reset button on the board just before beginning the upload on older boards (pre-Diecimila) that don't have auto-reset. When the sketch is posted, the RX and TX LEDs on most boards can blink. When the upload is complete, the Arduino Program (IDE) will show a message or an error.

When you upload a sketch, you're using the Arduino bootloader, which is a tiny programme loaded onto your board's microcontroller. It enables you to upload code without the need for any extra hardware. When the board resets, the

bootloader is active for a few seconds before starting whatever sketch was most recently uploaded to the microcontroller. When the bootloader begins, the onboard (pin 13) LED will blink (i.e. when the board resets).

4.7.7 LIBRARIES

Libraries add extra functionality to drawings, such as the ability to interact with hardware or manipulate data. Pick a library from the Sketch > Import Library menu to use it in a sketch. This will compile the library with your sketch and add one or more `#include` statements to the top of the sketch. Libraries take up more room on the board than your sketch since they are uploaded with it. Remove the `#include` statements from the top of your code if a sketch no longer needs a library.

In the reference section, there is a list of libraries. The Arduino app comes with a few libraries. Others are available for download from a number of places, like the Library Manager. You can import a library from a zip file and use it in an open sketch starting with version 1.0.5 of the IDE. To build a third-party library, follow these instructions.

4.7.8 THIRD-PARTY HARDWARE

The hardware directory in your sketchbook directory may be connected to help for third-party hardware. fabricated incorporate board definitions (which show up in the board menu), center libraries, bootloaders, and software engineer definitions. Make the equipment catalog first, at that point unfasten the outsider stage into its own subdirectory. (On the off chance that you use "arduino" as the sub-registry. Simply delete the program's directory to uninstall it.

The Arduino IDE 1.5 3rd party Hardware specification has more information on developing packages for third-party hardware.

4.7.9 SERIAL MONITOR

The Arduino or Genuino board's serial data is shown (USB or serial board). Type text and press enter or click the "send" button to send data to the device. From the drop-down menu, choose the baud rate that corresponds to the rate passed to Serial. Begin with a sketch. The Arduino or Genuino board will reset when you

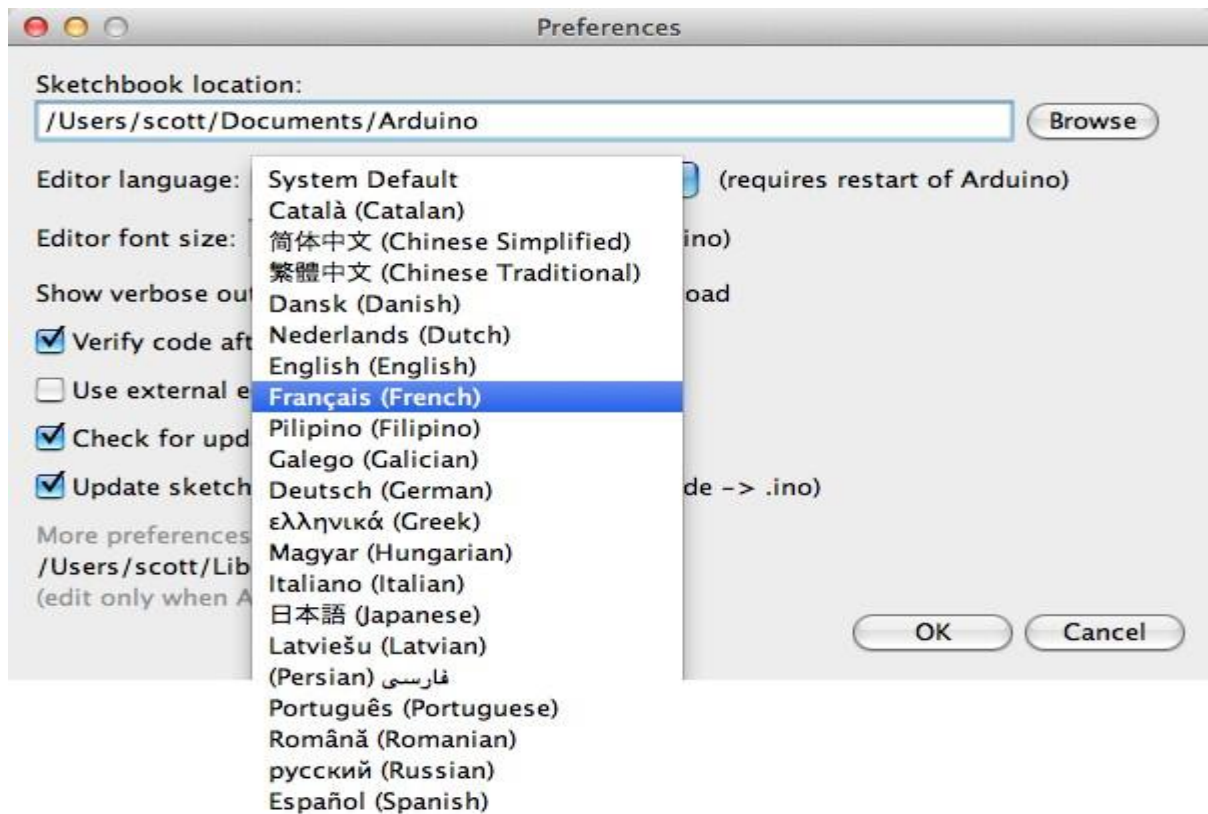
connect with the serial monitor on Windows, Mac, or Linux (restart your sketch execution from the beginning).

You can also communicate with the board using Processing, Flash, MaxMSP, and other programming languages (see the interfacing page for details).

4.7.10 PREFERENCES

In the preferences dialogue, you can set some preferences (found under the Arduino menu on the Mac, or File on Windows and Linux). The rest is in the preferences file, which is shown in the preference dialogue.

4.7.11 LANGUAGE SUPPORT



From the above figure, the Arduino Software (IDE) has been translated into 30+ languages since version 1.0.1. The IDE starts up in the language that your operating system has chosen by default. (Note that on Windows and most likely Linux, the area climate, which oversees money and date designs, decides this, not the language wherein the working framework is shown)

If you need to change the language physically, open the Preferences window in the Arduino Software (IDE). A dropdown menu of presently upheld dialects shows up close to the Editor Language. Pick your #1 language from the menu, and afterward restart the program to utilize it. The Arduino Program (IDE) will change to English if your working framework language isn't empowered.

Select System Default from the Editor Language drop-down to return the programme to its default setting of choosing the language depending on your operating system. When you restart the Arduino Program, this setting will take effect (IDE). Similarly, you must restart the Arduino Software (IDE) after updating your operating system's settings to upgrade it to the new default language.

4.7.12 BOARDS

The board selection has two effects: it controls the parameters used by the burning bootloader command when compiling and uploading sketches (such as CPU speed and baud rate), as well as the file and fuse settings used by the burn bootloader command. Only in the above do some of the board definitions differ, so double-check before burning the bootloader. even though you've been uploaded successfully with a specific range. A comparison table of the different boards can be found [here](#).

The Arduino Software (IDE) includes support for the following boards, which are all based on the AVR Core. The Boards Manager, which comes standard with the installation, lets you add support for a growing number of new boards based on various cores including the Arduino Due, Arduino Zero, Edison, Galileo, and others.

4.8 NODE RED

Node RED is a software application that allows us to create new and interesting connections between hardware, APIs, and online services.

It includes a browser-based editor that makes it simple to link flows to the palette's large number of nodes, which can all be deployed to the runtime with a single click.

The change node is another helpful node that allows you to change the message payload or add new properties. You may use this node to modify a message's properties, either by modifying existing ones, removing them, or adding new ones.

Continue with your MQTT theme in this example to see how you can add a new message property called `msg.payload`. It's important to remember that you've now successfully "switched" the message flow based on the incoming MQTT message.

You can use Node-RED to link nodes to generate flows that carry out your programming mission. Messages move between nodes, from input nodes to processing nodes and eventually to output nodes. Let's look at nodes, flows, and messages in more detail.

There are three main types of nodes:

1. Input Nodes (e.g. inject)
2. Output Nodes (e.g. debug)
3. Processing Nodes (e.g. function)

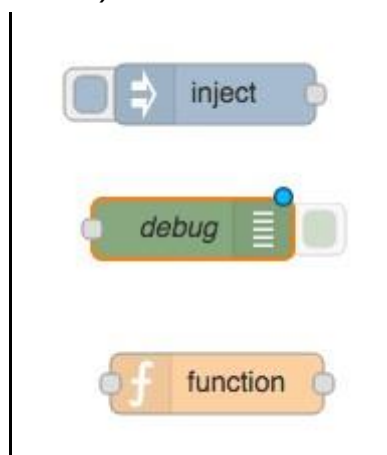


Fig 4.9 The main types of nodes: input, output and processing

From the above figure, you may use input nodes to enter data into a Node-RED application or "flow." They only have one output endpoint on their right hand, which is defined by the small grey rectangle. Input nodes are used to link data from other

services, such as Twitter, Google, serial, websockets, or tcp, or to manually enter data into a flow using the inject node.

You can send data outside of a Node-RED flow using output nodes. On their left hand, they have a single input endpoint. You can send data to other services using output nodes, such as Twitter, TCP, serial, or email nodes, or you can use the debug node to output to the debug pane.

You can process data using processing nodes. They have one or more output endpoints and an input endpoint. They allow you to convert data types (e.g. json, csv, xml) nodes, use data to trigger a message (e.g. trigger, delay) nodes, and write custom code that makes use of the data obtained (e.g. function node).

Some nodes, such as the inject and debug messages, have a button that enables you to actuate a node (in the case of the inject node) or activate and disable a node (in the case of the debug node) (in the case of the debug node).

Flows are made up of multiple nodes that are connected by output tabs that are linked to the input tabs of the next node in the flow. Messages travel across the nodes, bringing data from one node to the next.

Nodes in the Node-RED system consume input messages and generate output messages. Messages are JavaScript objects with at least one "payload" parameter, as seen here:

Listing 2.1 A basic Node-RED message structure

```
1.msg = {  
2.payload:"message payload"  
3.};
```

Nodes consume and generate messages, and the data they consume and produce is normally stored in `msg.payload`. Messages, on the other hand, may be expanded to provide additional parameters. To set a message's topic and add a new

parameter, position, for example, you could build a new msg object as shown in listing 2.2.

Listing 2.2 A more complex Node-RED message structure

```
1.msg = {  
2.payload:"message payload",  
3.topic:"error",  
4.location:"somewhere in space and time"  
5.};
```

4.8.1 BROWSER-BASED FLOW EDITING

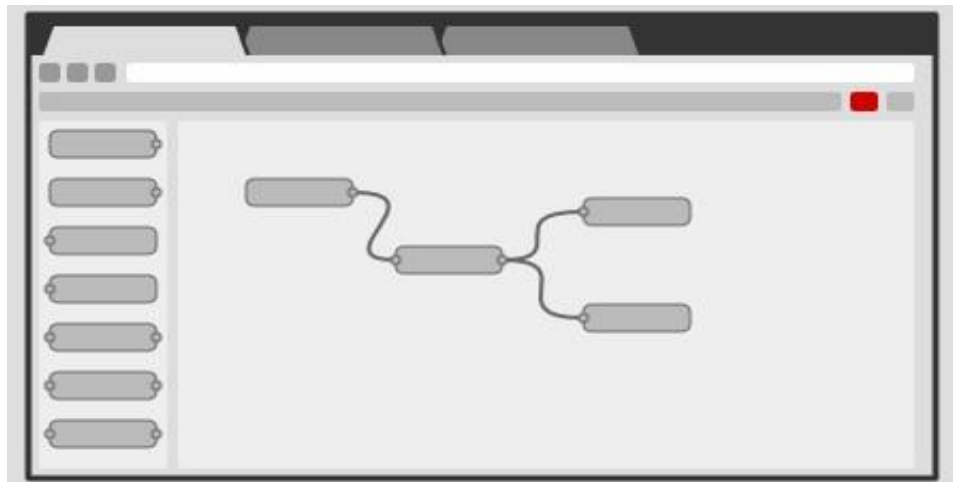


Fig:4.8.1.1 : Browser based flow editing

From the above figure, Node-RED comes with a browser-based flow editor that allows linking flows with the palette's many nodes a breeze. Flows can then be deployed to the runtime with only one press.

Inside the editor, a rich text editor can be used to create JavaScript functions.

You can save and reuse useful features, templates, and flows in a built-in library.

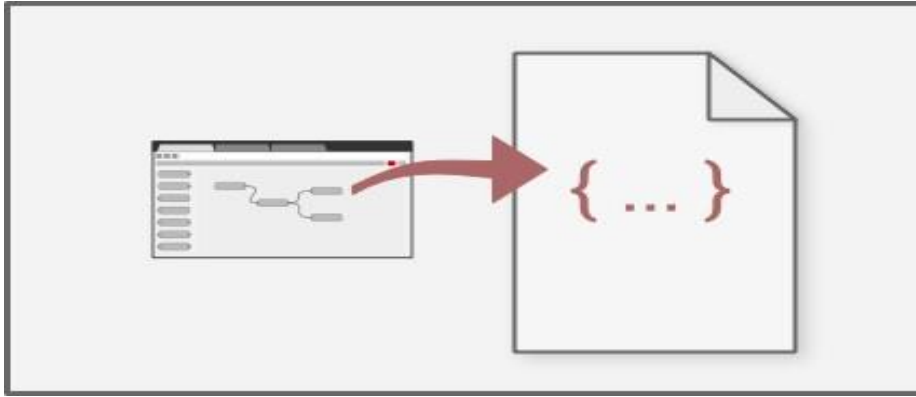


Fig:4.8.1.2 : Using json file save the document

From the above figure ,Node-RED flows are saved in JSON format, which can be conveniently imported and exported for sharing with others.

You can share your best flows with the rest of the world using an online flow library.

4.9 NEUROLOGICAL DISEASES

The brain, spine, and nerves that link them are all affected by neurological disorders. There are over 600 nervous system disorders, including brain tumours, epilepsy, Parkinson's disease, and stroke.

4.9.1 ACUTE SPINALCORD INJURY

The vertebrae are a collection of bones that made the spine. Your spinal cord moves through a canal in the middle of these bones on its way down. A bundle of nerves that transmits movement and feeling signals from the brain to the rest of the body is spinal cord.

Acute spinal cord injury (SCI) occurs when the spinal cord is bruised, partially torn, or completely torn as a result of a traumatic injury. In both children and adults, SCI is a common cause of permanent disability and death.

The following are the most common signs and symptoms of an acute spinal cord injury:

- Loss of voluntary muscle function in the chest, arms, or legs

- Muscle weakness
- Breathing difficulties
- Head, arm, or leg numbness or tingling
- Dysfunction of the bowels and bladder

4.9.2 ALZHEIMERS DISEASE

Alzheimer's disease is a neurologic disorder that causes the brain to shrink (atrophy) and the death of brain cells. Alzheimer's disease is the most common cause of dementia, which is described as a progressive loss of cognitive, behavioural, and social abilities that impairs a person's ability to function independently.

One of the first signs of Alzheimer's disease is forgetting recent events or conversations. As Alzheimer's disease progresses, a person can experience severe memory loss and lose the ability to carry out everyday activities.

Medications can improve or delay the progression of symptoms temporarily. These therapies may also help people with Alzheimer's disease preserve their independence and maximise their work. People with Alzheimer's disease and their families will benefit from a range of programmes and services..

4.9.3 ATAXIA

Ataxia is a nervous system degenerative disease. Slurred speech, stumbling, dropping, and incoordination are all signs of ataxia, as are slurred speech, stumbling, falling, and incoordination. Damage to the cerebellum, some part of the brain responsible for movement coordination, causes these symptoms. Ataxia is treated with a combination of medicine and therapy to symptoms will be relieved, and the quality of life will increase. Ataxia patients may have difficulty using their fingers and hands, arms, legs, walking, speaking, or moving their eyes. People of all ages are

affected by ataxia. Symptoms may appear at any age, ranging from childhood to late adulthood .The disease's complications are severe and sometimes debilitating.

Some forms of ataxia may result in death.

4.9.3.1 SYMPTOMS

- A lack of coordination.
- Unsteady gait and a proclivity to trip.
- Difficulty performing fine motor activities like chewing, writing, or buttoning a shirt.
- A change in speech pattern.
- Nystagmus (involuntary back-and-forth eye movements) • Swallowing difficulties

4.9.4 AMYOTROPHIC LATERAL SCLEROSIS

Amyotrophic lateral sclerosis (ALS), also known as Lou Gehrig's disease, is a progressive nervous system disease that affects nerve cells in the brain and spinal cord, causing muscle failure.

Muscle twitching and weakness in a leg, as well as slurred speech, are common symptoms of ALS. ALS eventually wreaks havoc on the muscles that allow you to move, talk, eat, and breathe. This deadly disease has no known cure.

4.9.4.1 SYMPTOMS

ALS symptoms and signs differ significantly from person to person, depending on which nerves are affected. The following are possible signs and symptoms:

- Difficulty walking or doing regular everyday tasks
- Tripping and falling
- Leg, foot, or ankle weakness

- Hand weakness or clumsiness
- Slurred speech or trouble swallowing
- Muscle cramps and twitching in your arms, shoulders, and tongue
- Inappropriate crying, smiling, or yawning

Environmental factors, such as the following, might trigger ALS.

- *Smoking:* The only known environmental risk factor for ALS is smoking. Women tend to be at the greatest risk, particularly after menopause.
- *Environmental toxin exposure:* Exposure to lead or other chemicals in the workplace or at home, according to some evidence, can be related to ALS. Despite extensive research, no single agent or chemical has been reliably linked to ALS.
- *Military service:* According to studies, people who have served in the military are more likely to develop ALS. It's unclear what aspect of military service can lead to the onset of ALS. Exposure to certain metals or chemicals, physical injuries, viral infections, and heavy exertion are all possible causes.

4.9.5 BELL'S PALSY:

Bell's palsy is a condition in which facial muscles become weak or paralysed for a short period of time. Facial paralysis may occur when the nerve that controls the facial muscles becomes inflamed, swollen, or cramped.

As a consequence, one side of the face droops or stiffens of the disease. On the affected hand, you can find it difficult to smile or close your eye. Bell's palsy is usually only temporary, with symptoms disappearing after a few weeks.

4.9.5.1 SYMPTOMS

- drooling

- trouble eating and drinking
- failure to make facial movements like smiling or frowning
- facial fatigue
- facial muscle twitches
- dry eye and mouth
- headache
- sensitivity to sound
- discomfort of the involved eye

4.9.5.2 RISK FACTORS:

Your risk of developing Bell's palsy increases if you:

- you're pregnant
- you've got asthma
- you've got a lung infection
- you've got a family history of the disease

4.9.6 BRAIN TUMORS

A brain tumour is a group of abnormal cells in your brain that forms a mass. Your brain is protected by a very rigid skull. Any expansion in such a small space can cause problems. Brain tumours can be malignant (cancerous) or benign (noncancerous) (benign). The pressure within your skull will rise as benign or malignant tumours grow larger.

4.9.6.1 SYMPTOMS

- A common symptom of a brain tumour is headaches. You may get headaches that:
 - are worst when you first wake up
 - happen when you're sleeping
 - are aggravated by coughing, sneezing, or exercise

You may also experience:

- vomiting
- double vision or blurred vision
- confusion
- seizures (especially in adults)
- a change in mental functioning
- weakening of a leg or part of the face

4.9.7 CEREBRAL ANEURYSM

CEREBRAL ANEURYSM A cerebral aneurysm (additionally called a brain aneurysm) is a feeble or slight spot on a cerebral conduit that lumps out and loads up with blood. An aneurysm that has blasted may put tension on nerves or brain tissue. It might likewise blast or burst, emptying blood into close by tissue (called a bleed). Hemorrhagic stroke, brain injury, unconsciousness, and even demise may all outcome from a burst aneurysm.

Some cerebral aneurysms, particularly the smallest ones, do not bleed or cause other complications. Aneurysms of this kind are normally discovered during imaging tests for other medical issues. Aneurysms can develop anywhere in the brain, but the majority form in the main arteries that run along the base of the skull.

4.9.7.1 SYMPTOMS

An aneurysm that is steadily increasing in size can push on tissues and nerves, resulting in:

- a dilated pupil in the eye
- discomfort above and below the eye
- numbness
- fatigue
- paralysis on one side of the face
- vision shifts or double vision

4.9.7.2 RISK FACTORS FOR DEVELOPING AN ANEURYSM

Aneurysms in the brain are often caused by hereditary risk factors, such as:

- artery-weakening hereditary connective tissue disorders
- polycystic kidney disorder (polycystic kidney disease) (in which numerous cysts form in the kidneys)
- arteriovenous malformations (blood flow is disrupted by tangled arteries and veins in the brain). Some AVMs appear at random or on their own.)
- a first-degree relative with a history of aneurysm (child, sibling, or parent).

4.9.8 EPILEPSY AND SEIZURES

Seizures are a symptom of epilepsy, which are abnormal movements or behaviour caused by abnormal electrical activity in the brain. However, not everybody who seems to have seizures has epilepsy, a category of conditions marked by a proclivity for frequent seizures.

4.9.8.1 CAUSES OF EPILEPSY

Epilepsy is a brain disorder characterised by abnormal electrical activity. Electrical signals are sent in a regular pattern by brain cells to communicate. These electrical signals become abnormal in epilepsy, resulting in a "electrical storm" that causes seizures. These storms can occur in a specific part of the brain or be generalised, depending on the type of epilepsy.

4.9.8.2 SYMPTOMS

There is no loss of consciousness in a clear partial seizure. Among the signs and symptoms are:

- dizziness
- limb tingling and twitching
- Taste, smell, sight, hearing, and touch changes

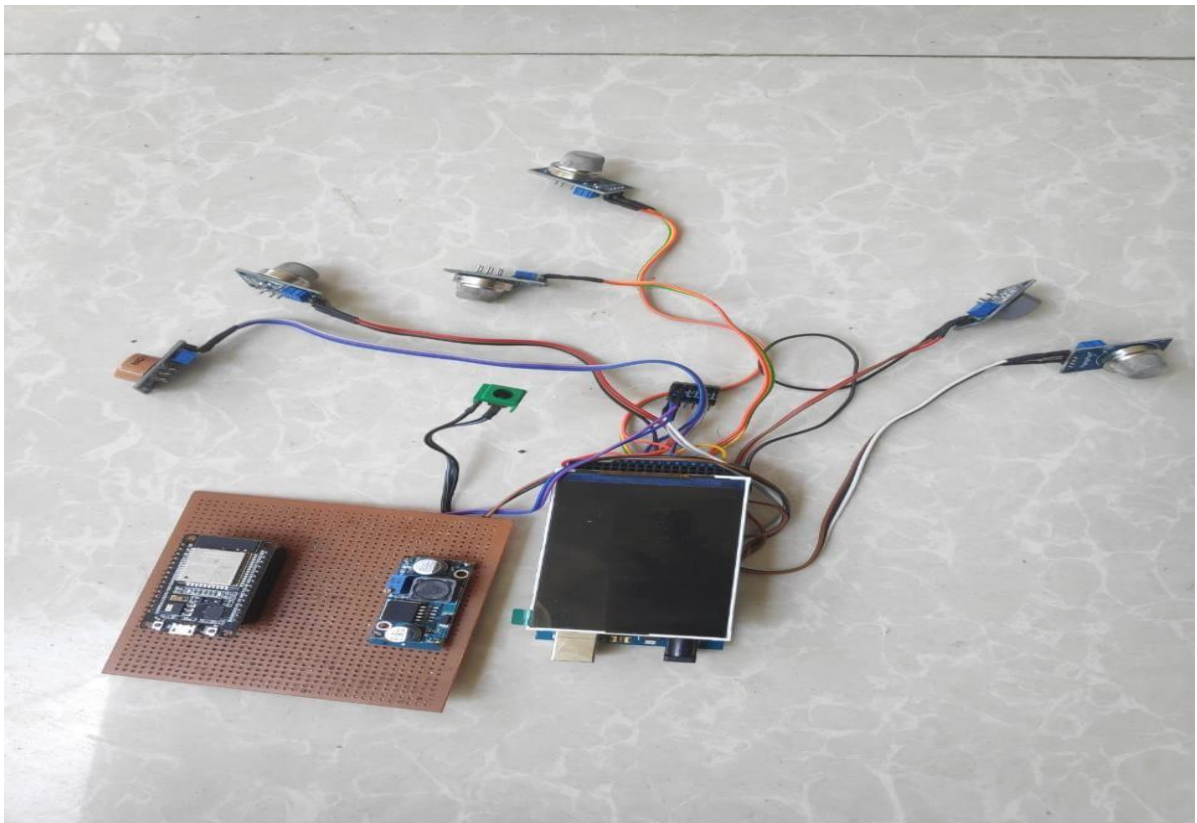
There is a lack of memory or consciousness in complex partial seizures. Some signs and symptoms include:

- staring blankly
- unresponsiveness

CHAPTER 5

RESULT

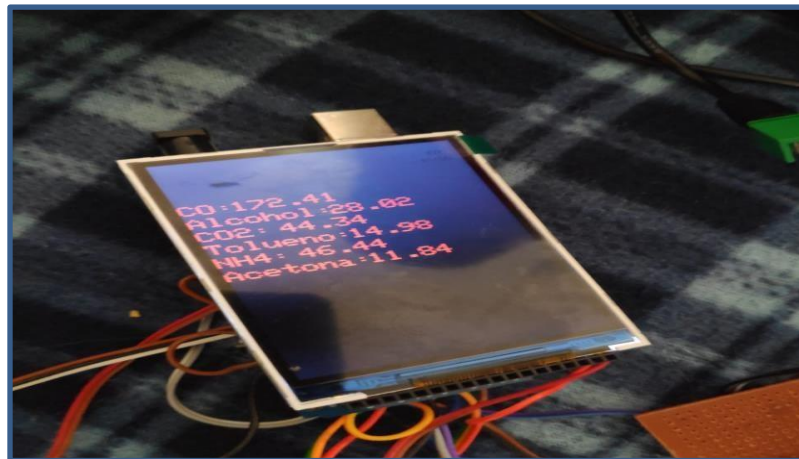
The sensors that are used in project are MQ-3,MQ-4,MQ-6,MQ-7,MQ8,MQ-135. The microcontroller is used to interface the gas sensors. Every sensor has one's own perspective. MQ3 is a gas detector that detects benzene whereas MQ4 is a gas detector that detects smoke and MQ6 is a gas detector that detects methane gas and MQ7 is a gas detector that detects carbon monoxide gas, MQ8 sensor is a gas detector that detects hydrogen gas and MQ135 sensor is a gas detector that detects carbon monoxide, alcohol, carbon dioxide, toluene, ammonia, acetone gases . These sensors were interfaced with microcontroller. Sensors detect the values and these values are given as input to Arduino and displayed on the display module. And this data will be sent to laptop through Wi fi module present in ESP32 microcontroller using mqtt protocol. These values will be sent to node red platform. There we can detect the diseases using java script. Hence, we can detect the diseases.



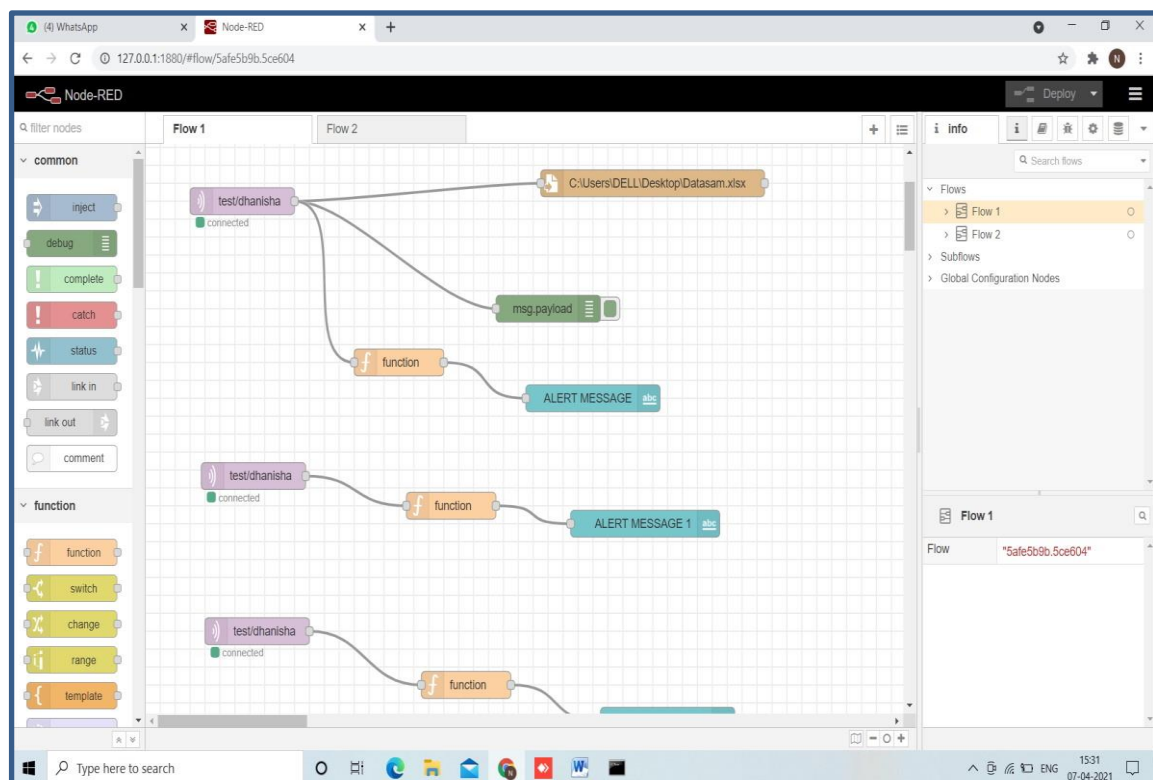
Hardware setup

From the above figure, this is hardware connections where all the sensors, display module, buck converter are connected to Arduino pins.

From the below figure indicates the values that are detected from each gas sensor and those values are displayed on the display module.

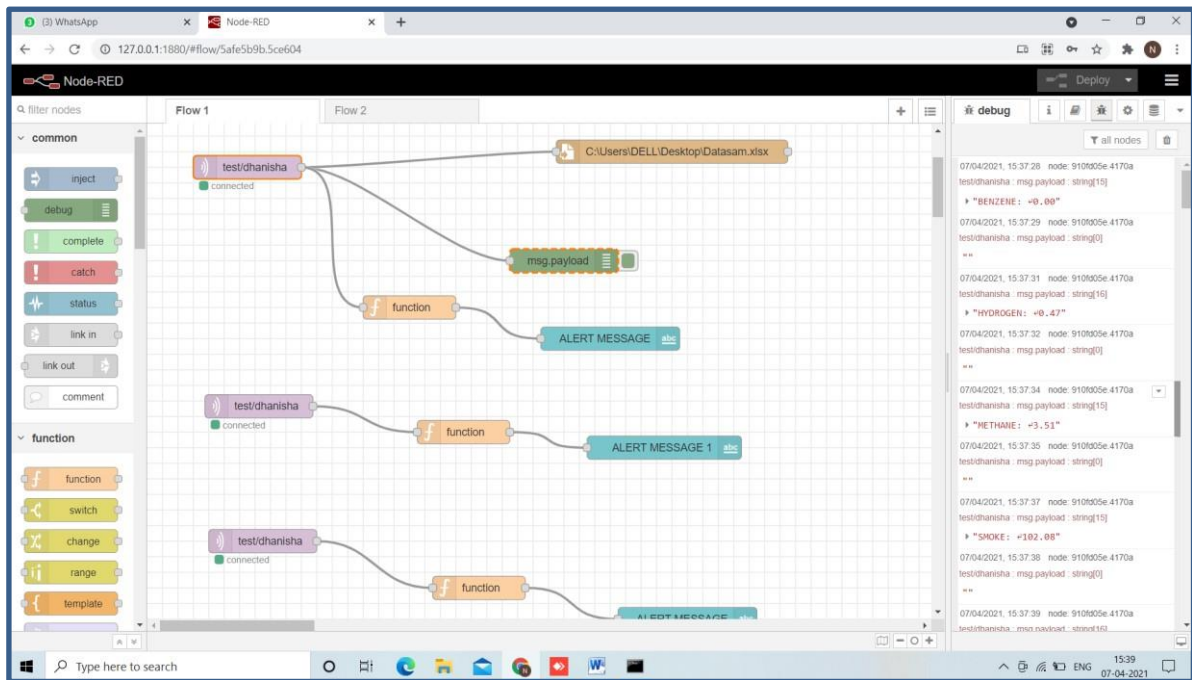


Values of gas sensors on display module



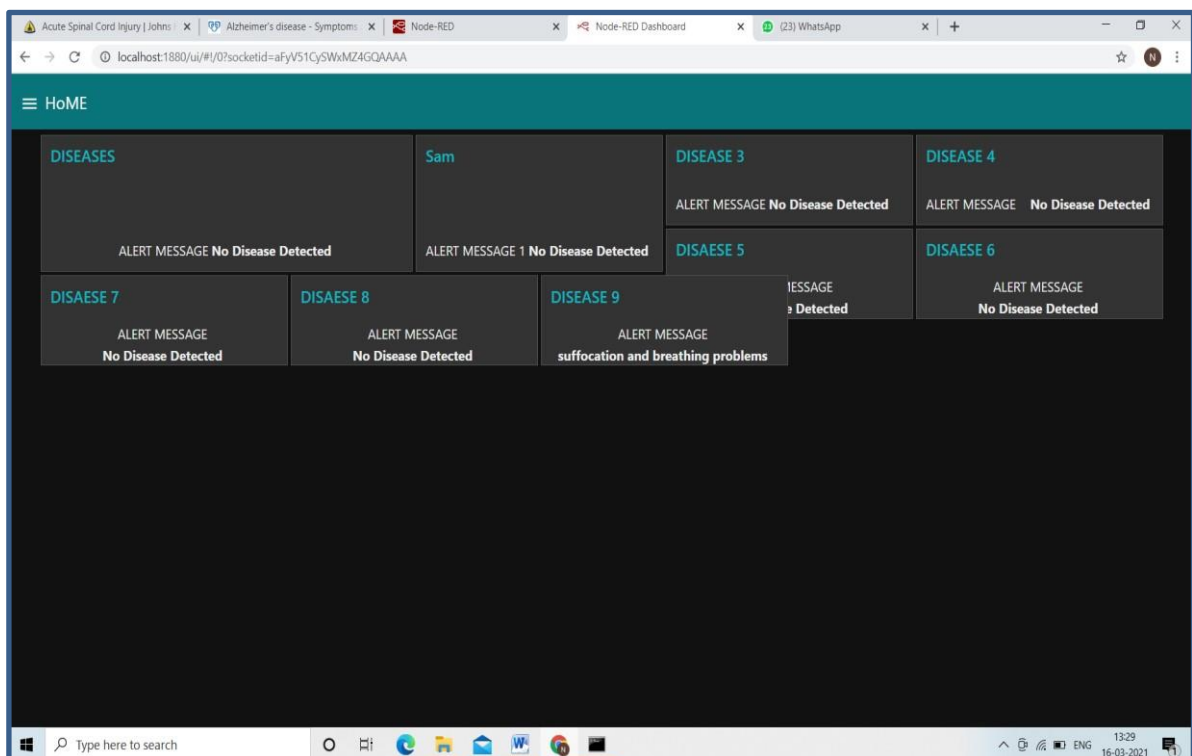
Node Red platform

The above figure is a node red platform. It's a flow editor that runs in a web browser. As part of the Internet of Things, it's used to connect hardware devices, APIs, and online services. It's used to collect data from the sensors and send us warning messages.



Debug window in node red platform

The above figure is the debug window in node red platform. Here we can see the values of each gas and these values will be changing every time depending on the present gas value in node red and as well as in display module which we got from the sensors.



Alert messages on node red dashboard

From the above figure,we can see the alert messages displaying on the node red dashboard. The message displayed on the dashboard changes everytime with the values given by the sensors.So,this is the final result and alert message.

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

Air pollutants are caused by a mixture of gaseous, volatile substances, semi volatile and particulate matter substances. All of these pollutants are released into the atmosphere as a result of human activity. Air pollution can cause a variety of health problems, including respiratory and neurological disorders, which pose a serious threat to human life. There is growing evidence that air pollutants can not only harm the central neural system, but may even increase the risk and mortality of Alzheimer's disease and other neurological diseases like Ataxia, Brain tumours, Acute spinal cord injury, Bells palsy and Amyotrophic lateral Sclerosis. By using various sensors such as MQ3, MQ4, MQ6, MQ7, MQ8 and MQ135. These sensors can detect gases such as Benzene, Smoke, Methane, Carbon dioxide, Hydrogen, Alcohol, Carbon monoxide, Ammonia, Acetone, Toluene. Using node red platform we can able to promote well being in humans by giving advance alerts and alarms over an area. By analyzing the air quality in a specific region, this project can be used to detect neurological diseases.

REFERENCES

- [1] Albert M.K Cheng ; Joe Air Jiang was presented this feature with the temperature- humidity sensor module at the parameter of NO₂_mean-0.168 by using the GRIMM's light- scattering fine- dust method at the accuracy of 76%.
- [2] Chiao-Lun Lee; Jin-Shyan Lee was presented this feature with the temperature-humidity sensor at the parameter of Temp- 0.412 using the LS estimation method at the accuracy of 26%
- [3] Giacomo Boracchi; Michalis Michaelides; was presented this feature with the VOC sensor module GSBT11-P110 at the parameter of Intercept-14.74 by using the method of GRIMM's light- scattering fine- dust at the accuracy of 72%
- [4] JeongGil Ko; Samuel Keene was presented this feature with the pollutant detection sensors at the parameter of NO₂_mean-0.168 using the proposed calibration method at the accuracy 58%.
- [5] Jungmo Ahn; Hyungi Kim; was presented this feature with the Temperature and Humidity sensor (DHT11) at the parameter of NO_DG-0.250 using a linear least square-based method for sensor calibration at the accuracy of 92%.
- [6] Manuel Roveri; Rita Paradiso was presented this feature with the Air Quality sensor (MQ 135) at the parameter of NO_mean-0.125 by using the L.S estimation Method at the accuracy of 67%.
- [7] Pooja Kumawat; Sabarimalai Manikandan . M was presented this feature by using the laser dust sensor, model PM2007 at the parameter of Temp- 0.412 by using the method of a linear least square-based method for sensor calibration at the accuracy of 47%.
- [8] Postolache.O ; Dias J. M Pereira; was presented this feature with the Temperature and Humidity sensor of module (DHT11) at the parameter of NO_mean-0.125 by using the modem uses LG U+ LTE B5/B7 FDD Cat. 4 as the communication method at the accuracy of 41%.

[9] Silva Girão .P; Postolache.G was presented this feature with the pollutant detection sensors of module NO_DG-0.250 at the parameter and the Chamber & gravimetric method is used at the accuracy of 83%.

[10] Sumanth Reddy Enigella; Hamid Shah Nasser was presented this feature with the Air Quality sensor of module (MQ 135) at the parameter of Intercept-14.74 Using air quality measurement values in most devices was dependent on a number of variables, including measurement methods with a 52 percent accuracy.

samdan report

by Plagiarism Checker

Submission date: 16-Apr-2021 03:11AM (UTC-0400)

Submission ID: 1507787517

File name: samdan1-merged_final.pdf (1.4M)

Word count: 11724

Character count: 57752

AN EMBEDDED APPLICATION FOR DETECTING ADVANCE NEUROLOGICAL AND BREATHING DISORDERS BY AIR QUALITY MEASUREMENT IN REAL TIME

Submitted in partial fulfilment of the requirements for the award of
Bachelor of Engineering degree in Electronics and Communication
Engineering
by

GADDAM SAI SAMHITHA (Reg. No. 37130126)
DANTU DHANISHA (Reg. No. 37130108)



DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

SCHOOL OF ELECTRICAL AND ELECTRONICS

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

APRIL - 2021



SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with "A" grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in



DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **GADDAM SAI SAMHITHA** (Reg. No.37130126) and **DANTU DHANISHA** (Reg. No.37130126) who carried out the project entitled "**AN EMBEDDED APPLICATION FOR DETECTING ADVANCE NEUROLOGICAL AND BREATHING DISORDERS BY AIR QUALITY MEASUREMENT IN REAL TIME**" under my supervision from.

Internal Guide

Dr. S.LAKSHMI, M.E., Associate Professor

Head of the Department

Dr. T. RAVI, M.E., Ph.D.

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

We GADDAM SAI SAMHITHA (Reg. No.37130126) and DANTU DHANISHA (Reg. No.37130108) hereby declare that the Project Report entitled "AN EMBEDDED APPLICATION FOR DETECTING ADVANCE NEUROLOGICAL AND BREATHING DISORDERS BY AIR QUALITY MEASUREMENT IN REAL TIME" done by us under the guidance of Dr. S.LAKSHMI,M.E.,Associate Professor is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering degree in ELECTRONICS AND COMMUNICATION ENGINEERING.

DATE:

PLACE:

- 1.
- 2.

SIGNATURE OF THE CANDIDATES

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to Board of Management of SATHYABAMA for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

We convey our thanks to **Dr. N.M. NANDITHA, M.E., Ph.D.**, Dean, School of Electrical and Electronics and **Dr. T. RAVI, M.E., Ph.D.**, Head of the Department, Dept. of Electronics and Communication Engineering for providing us necessary support and details at the right time during the progressive reviews.

We would like to express our sincere and deep sense of gratitude to our Project Guide **Dr. S. LAKSHMI, M.E., Associate Professor** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

We wish to express our thanks to all Teaching and Non-teaching staff members of the Department of Electronics and Communication Engineering who were helpful in many ways for the completion of the project.

ABSTRACT

Environmental and health issues have risen as living conditions and economic growth have improved. The haze weather phenomenon is becoming a serious problem as air quality continues to deteriorate. It has been discovered that there is a possible connection or causal relationship between exposure to air pollution and neurological diseases. The haze weather phenomenon is integrated into fog as an extreme weather warning in many geographical areas. The design and implementation of an embedded multisensory system for air pollution and air quality monitoring with disease prevention warnings and alarms is presented in this project. This device is made up of a microcontroller and various sensors for measuring gases in the atmosphere. This system also includes a model to predict diseases in a more efficient way other than the present conventional methods. Moreover an immediate alert system is also included with display and enables data collection of these sensors for future machine learning and artificial intelligence systems.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	i
	LIST OF FIGURES	v
	LIST OF ABBREVIATIONS	vi
1	INTRODUCTION	1
2	LITERATURE SURVEY	2
3	EXISTING SYSTEM	6
4	PROPOSED SYSTEM	7
	4.1 Block diagram of proposed System	7
	4.2 Arduino	8
	4.2.1 Digital	9
	4.2.2 Analog	10
	4.2.3 Output signals	11
	4.2.4 Input signals	12
	4.3 Gas sensors	15
	4.3.1 MQ3 sensor	15
	4.3.1.1 Features of MQ3 Alcohol sensor	15
	4.3.2 MQ4 sensor	16
	4.3.2.1 MQ4 Methane gas Sensor –	
	General Specifications	16
	4.3.3 MQ6 sensor	17
	4.3.3.1 LPG gas sensor features	17

4.3.4 MQ7 sensor	17
4.3.5 MQ8 sensor	18
4.3.5.1 Features	18
4.3.5.2 Main Applications	19
4.3.6 MQ135 sensor	19
4.3.6.1 Applications	19
4.4 Buck Converter	20
4.4.1 Working of a Buck Converter	20
4.5 ESP32 microcontroller	21
4.5.1 Programming	22
4.5.2 Other chips in eESP32 Family	23
4.6 Display module	23
4.7 Arduino Software	24
4.7.1 Writing sketches	24
4.7.2 Tools	24
4.7.3 Help	25
4.7.4 Sketchbook	26
4.7.5 Tabs, Multiple files, and compilation	26
4.7.6 Uploading	26
4.7.7 Libraries	27
4.7.8 Third-party Hardware	27
4.7.9 Serial monitor	28
4.7.10 Preferences	28
4.7.11 Language Support	28

	4.7.12 Boards	29
	4.8 Node red	29
	4.8.1 Browser-based Flow editing	32
	4.9 Neurological Diseases	33
	4.9.1 Acute spinalcord Injury	33
	4.9.2 Alzheimers Disease	34
	4.9.3 Ataxia	34
	4.9.3.1 Symptoms	35
	4.9.4 Amyotrophic Lateral Sclerosis	35
	4.9.4.1 Symptoms	35
	4.9.5 Bell's Palsy	36
	4.9.5.1 Symptoms	36
	4.9.5.2 Risk Factors	37
	4.9.6 Brain Tumors	37
	4.9.6.1 Symptoms	37
	4.9.7 Cerebral Aneurysm	38
	4.9.7.1 Symptoms	38
	4.9.7.2 Risk Factors	38
	4.9.8 Epilepsy and Seizures	39
	4.9.8.1 Causes of epilepsy	39
	4.9.8.2 Symptoms	39
5	RESULT	41
6	CONCLUSION	45
	REFERENCES	

LIST OF FIGURES

FIGURE NO	TITLE	PAGE NO
4.1	Block diagram of proposed system	7
4.2	Arduino UNO	8
4.2.4	PIN Configuration of Arduino UNO	13
4.3.1	MQ3 sensor	15
4.3.2	MQ4 sensor	16
4.3.3	MQ6 Sensor	17
4.3.4	MQ7sensor	17
4.3.5	MQ8 sensor	18
4.3.6	MQ135 Sensor	19
4.4	Buck Converter	20
4.4.1.1	Step-1 Working	20
4.4.1.2	Step-2 Working	21
4.5	ESP32 Microcontroller	21
4.5.1	ESP32 Function Block Diagram	22
4.6	Ili9341 Display Module	23
4.9	The main types of nodes-input output and processing	30
4.8.1.1	Browser-based flow editing	32
4.8.1.2	Using json file save the document	33

LIST OF ABBREVIATIONS

MQTT	Message Queuing Telemetry Transport
NodeMCU	Node Microcontroller Unit
S2	
TFT	Thin Film Transistor
SPI	Serial Peripheral Interface
IDE	Integrated Development Environment
JSON File	Java Script Object Notation
ALS	AMYOTROPHIC LATERAL SCLEROSIS

CHAPTER 1

INTRODUCTION

In this project We will develop a low-cost device capable of detecting neurological disorder in advance in this project. To analyse and give alerts based on the sensor data. To provide a healthy environment for breathing using data from sensors and analytics. As we already have a Portable Multi-Sensory Module for Neurological Disease Prevention MQTT Based Air Quality Measurement System using NodeMCU and Node-RED, we can use it to monitor air quality. We are proposing Using low cost devices and sensors to measure Air quality in an area/room and predict advance disease.

Immediate alert system with nearest doctor consultation facility inside the module this can be done in Collection and reviewing of Datasheets for Air quality sensors and hardware. Using Fritzing simulator we will simulate the working of the sensors for initial testing Coding of Air quality sensors with C++ in IDE using supported libraries and logics. We will test the working of sensors in real time using Arduino Controller. With the help of a display application we will display the output of the sensors and Gather data for sensors to co relate with neurological disease data.

Code debugging and data(Obtained from sensors) is used to assign alerts and alarms. Final checking of Hardware components and Finalization of the Project and real time use case testing with the help of patients and result publication.

CHAPTER 2

LITERATURE REVIEW

Vrushali Ladekar and Rohin Daruwala – (2019 Published) Air pollution is becoming a major issue whose consequences, if not addressed, could result in serious health issues. Indoor air quality has gone unnoticed for a long time. Because we spend so much time indoors, it's critical to keep an eye on the air we breathe. The calculation and analysis of air parameters is needed to take suitable steps in a timely manner. A control module for air quality has been suggested to address this constraint. Indoor air quality is measured using various sensors in this Internet of Things project. Data is pushed and stored on the cloud, and graphs of real-time data are plotted for better visualisation. If the indoor air quality is poor, the message is displayed.

Noorfadzli Razak and Adizul Ahmad – (2018 Published) Air quality monitoring is necessary to ensure that the environments in which we work are safe for our health. Remote monitoring of environmental conditions such as air and water has been made possible by recent technological advances. This paper describes a remote air quality monitoring system based on myRIO-LabVIEW that can detect carbon monoxide levels and track air quality in indoor locations inside buildings. The MQ-7 CO sensor is attached to the Arduino Mega and MyRIO controllers and is used to detect CO. The LabVIEW programme on the user's computer processes the outputs remotely. To detect the rate of change of CO, the device was checked for over an hour at three different locations. MyRIO can save the CO rate data indefinitely while the substance is near these places. In this way, LabVIEW can process and show data to the user wirelessly or remotely. The average CO rate could then be estimated after that. Since both hardware and software can detect and log data wirelessly, the device is functioning properly. The data can then be further analysed by the users.

Fang Lide; Ding Zhenjun and all – (19 April 2010 Published) As air quality has worsened, air pollution representatives of automatic monitoring have earned greater attention. Sulfur dioxide is one of the most common pollutants in the air and a major aspect of pollution levels. The effect of sample gas temperature and flow rate on the

sulphur dioxide air quality control system is discussed in this article. The results show that the sample gas temperature and flow rate have a major impact on sulphur dioxide air quality monitoring system, and that if the relative error is not corrected, the relative error would surpass 8%. Finally, a correlation is presented, resulting in a relative error reduction of 8% to 2%.

S Kumar and A Jasuja – (2017 Published) This paper reviews and analyzes real-time air quality monitoring in order to create effective decisions in a timely manner. This paper describes a real-time standalone air quality monitoring system that includes PM 2.5, CO, CO₂, temperature, humidity, and air pressure, among other parameters. The Internet of Things is now commonly used in almost every industry, and it also plays a key role in our air quality control system. The integration of the Internet of Things and cloud computing brings a different strategy for increasing productivity data gathered and shared by the Raspberry Pi, a lowpower, low-cost ARM-based minicomputer. The framework is put to the test in Delhi, where the readings are compared to data given by the local environmental protection authority and tabulated. In IBM Bluemix Cloud, the values of the parameters computed are shown

Vasantha Pradeep V and Dr. V. Ilaiyaraja – (2018 Published) Using the MQ135 sensor, this paper created a device that can detect the amount of radioactive particles in the air. As the level of benzene in the air rises above a certain threshold, an alarm is activated. When the toxicity in the air reaches the threshold level, the cloud server displays the air quality in PPMs, and the air purifier automatically turns on and detoxifies the air. This system can be deployed anywhere, and when emissions levels exceed the safe zone, additional equipment, such as a dedicated server, can be activated. As harmful particles in the air, such as CO₂, methane, or alcohol, are observed, an alert is shown on the LCD monitor, or a notice is sent to all people present via a speaker or SMS via a GSM module.

Sumanth Reddy Enigella and Hamid Shahnasser (2019 Published) A real-time air quality management system dependent on crowd sensing is presented. The proposed method captures the air quality index from the atmosphere using low-cost, low-power devices with gas sensors. An android programme was used to send raw sensor data from the gas sensors to the cloud. The data is analyzed in the cloud. Using an open-source geospatial visual analytics platform like Leaflet and R programming, the processed data is illustrated in real time on a map.

Somphop Chanthakit and Choopan Rattanapoka (2018 Published) The air quality control unit is a hardware device that connects that sensors to monitor temperature, humidity, carbon monoxide (CO), ozone gas (O3), and PM2.5 concentrations. The device's firmware turns it into a publisher, reading sensor data and sending it to a MQTT broker. Node-RED is used as a subscriber for receiving data from the MQTT Broker. We can conveniently design a flow to monitor and treat obtained data using Node-RED. The data will then be transmitted to the air quality monitoring dashboard, a flexible web application that displays data in gauges, text, and maps. Node-RED can also send a warning message to users via LINE Notify when the value of any data exceeds the configured range.

Jorge Silva and Pedro Lucas and all (2019 Published) With the environmental issues, interest in air quality monitoring is expected to rise in the near future. The majority of evidence, on the other hand, comes from a small number of government-owned sensors that can only capture a small portion of fact. As a result, improving data coverage necessitates decreasing sensor costs and making data more readily accessible. We'll do this by using a large range of low-cost sensors as the framework for an air quality monitoring network that can capture, aggregate, store, and view data. This platform would rely on streambased technology that can scale to accommodate a large number of sensors and users. The Nano Sen-AQM network that results would make large quantities of air quality data accessible to the public, with the goal of improving public health.

Manju K. Chattopadhyay (2017 Published) A concept for an environmental air quality monitoring system has been developed, which could track the concentrations of major air pollutant gases. The device makes use of low-cost air-quality control

nodes, which are made up of low-cost semiconductor gas sensors and Wi-Fi modules. Using semiconductor sensors, this device measures gas concentrations such as CO, CO₂, SO₂, and NO₂. The proposed work's main objective is to provide a low-cost infrastructure that allows for data collection and distribution to all stakeholders.

Kennedy O.Okokpuije, Etinosa Noma-Osaghae (2018 Published) Air pollution has an influence on our day-to-day lives as well as our quality of life. It threatens the planet's biodiversity and quality of life. Because of increasing industrial activity in recent years, there is a clear need to regulate air quality. People must be conscious of how their behaviours impact air quality. This project proposes a system for monitoring environmental impact. The Arduino microcontroller was used to build the device. The air pollution monitoring system was developed to constantly track and monitor air quality and log data to a central server, allowing data to be modified over the internet. The air quality was measured and analysed using the Parts per Million (PPM) metric. The developed system's air quality calculations were appropriate. The outcome was shown on the configured hardware's display interface and could be accessed from any smart mobile device via the cloud.

CHAPTER 3

EXISTING SYSTEM

3.1 INTRODUCTION:

The latest emission measurement technique employs either costly fixed-location equipment or specialised mobile equipment. Using dispersion models, the raw data collected in this manner is used to extrapolate the magnitude and concentration of emissions. This is a coarse-grained method of few and far between emission measurements. In addition, it is desirable to have access to real-time measurements to be able to quickly analyze and identify alarming levels of pollutants. Currently, access to such data is limited if not absent. It is available to and discernable by only a few who are well informed on the subject of pollution. Air Quality Monitoring using portable Multi-sensory Module for Neurological Disease Prevention. MQTT based Air quality Monitoring System using Node-MCU and Node-RED.

3.1.1 DRAWBACKS:

Sensor nodes drift in dynamic water, so localization methods based on non-fixed sensor nodes should be considered. Poison relocation is influenced by shear stream, violent stream, scattering, and corruption in convoluted water, for instance, the speed field. Compelling limitation calculations are still difficult to be given really. The contamination relocating measure is muddled; to improve on the examination issues, shift in weather conditions and compound responses just as transport and destiny models of poisons are overlooked in most contamination checking works utilizing sensor organizations. Restriction strategies considering the variables are difficult to be given.

CHAPTER 4

PROPOSED SYSTEM

To specifically Convert the Air detection Management System's Sensor data into transmitting over a fairly few distance using Arduino technology Data Acquisition from the sensor networks and display in a dashboard for sort of High level user experience Encryption of sensor data while transmission using perfect coding methods and achieving the output at a desired time without any hinderance, contrary to popular believe.

4.1 BLOCK DIAGRAM OF PROPOSED SYSTEM :

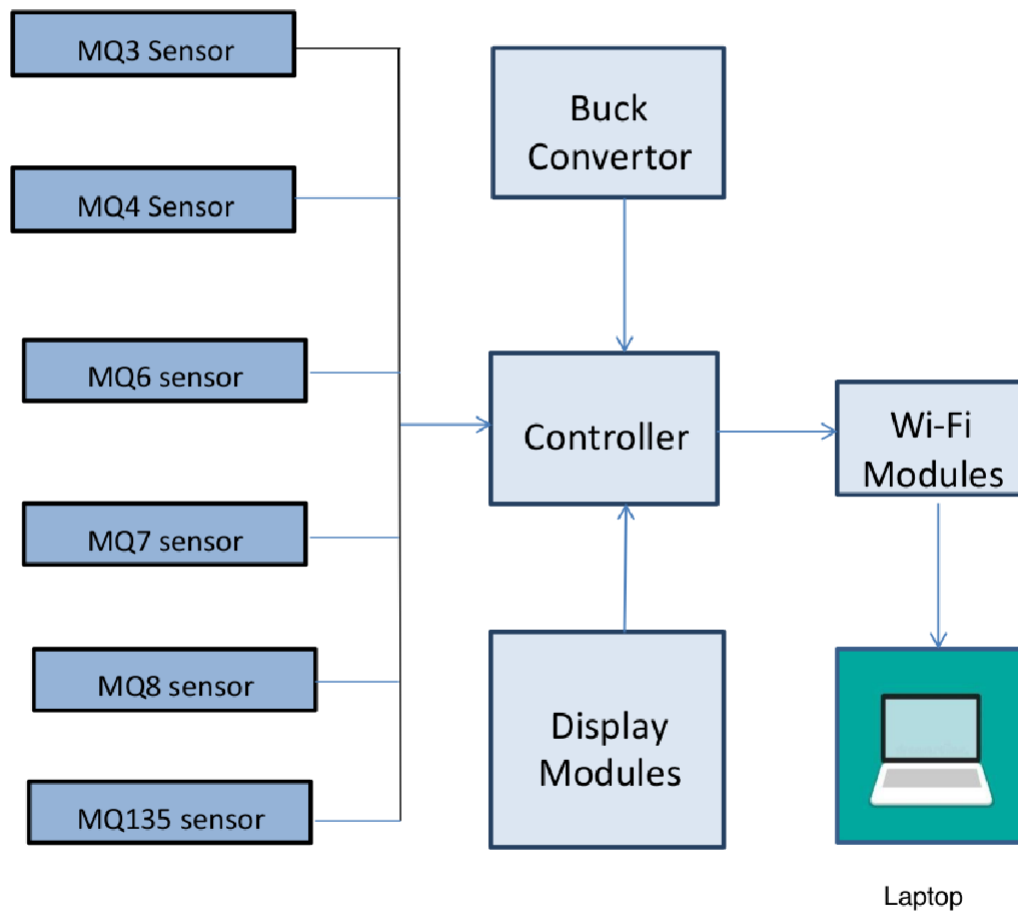


Fig:4.1 : Block diagram of proposed system

From the above figure, Sensors are connected to Arduino pins.ESP32 microcontroller,display module and buck converter is also connected to Arduino.ESP32 microcontroller has in-built wifi module and uses mqtt protocol to sent the data to Node-red platform.By using Node-red dashboard in laptop,we can get the alert message.

4.2 ARDUINO

It all depends on what you want to do with it. We'll go into all of the voltage divider's functions, which are described above.

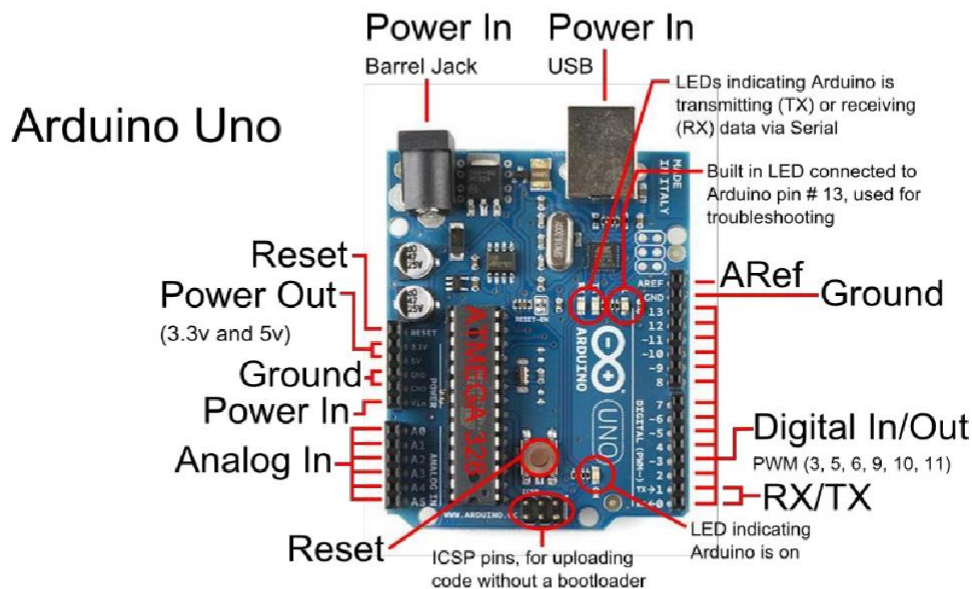


Fig:4.2 : Arduino Uno

From the above figure, to use the voltage divider as a sensor reading system, you must first determine the maximum voltage that the analogue inputs you are using to read the signal would enable. This is 5V on an Arduino. So, we always know what the optimal value for V_{out} should be. The V_{in} is clearly the amount of voltage on the circuit prior to meeting the first resistor. By looking at the Datasheet, we can able to determine the maximum voltage your sensor outputs; this is the maximum amount of voltage your sensor will allow through given the voltage in your circuit.

The only difference between powering an Analog Reference and reading a sensor is that you must compute the Vout value you choose to use as the analogue Reference.

The Arduino is capable of processing a wide range of electrical signals, both analogue and digital. It's crucial to understand the distinctions between these two types of signals, as well as how to use the data they provide.

4.2.1 DIGITAL

An electrical signal that may be the presence or absence of current, high and low voltages, or short pulses at a specific frequency and is transmitted as binary code.

Humans perceive the world in analogue form, while robots, computers, and circuits do so in digital form. The term "digital signal" refers to a signal that only has two states. These states can differ depending on the signal, but in general, they are either ON or OFF, with no in-between.

Everything in the Arduino world is done with digital signals, with the exception of Analog Input. The ON or HIGH signal of the Digital signal will match the system voltage, depending on the Arduino voltage, while the OFF or LOW signal will remain at 0V. This is a fancy way of saying that the HIGH signals on a 5V Arduino will be slightly less than 5V, while the HIGH signals on a 3.3V Arduino will be slightly less than 3.3V. Digital pins # 0 - # 13 are used by the Arduino to receive and transmit digital signals. Analog In pins may also be set to function as Digital pins. To convert Analog In pins to Digital pins, use the `pinMode(pinNumber, value)` command.

where `pinNumber` is an Analog pin (A0–A5) and `value` is Input or Output. To set up Digital pins, use the same command but specify a Digital pin instead of an Analog In pin for `pinNumber`. Since digital pins are set to input by default, you just need to change their `pinMode` to OUTPUT. Using the following command to read these pins: `pinNumber); digitalWrite(pinNumber);`

pinNumber refers to the Digital pin to which the Digital part is attached. The digitalWrite command will return either a HIGH or a LOW signal. Using the order to transmit a digital signal to a pin: pinNumber, value); digitalWrite(pinNumber, value);

where meaning is either HIGH or LOW and pinNumber is the number of the pin transmitting the signal.

A Pulse Width Modulation signal, which is a digital signal that acts like an analogue signal, may also be produced by the Arduino (PWM). PWM is available on Optical Pins #3, # 5, # 6, # 9, # 10, and #11. Use the following command to produce a PWM signal:

analogWrite(pinNumber, value); where pinNumber is a Digital Pin with PWM capacities and worth is a number between 0 (0%) and 255 (100%). On the Arduino UNO PWM pins are connoted by a ~ sign. For more data on PWM see the PWM worksheets or S.I.K. circuit 12. The Arduino manages electrical signs that are either info or yield.

Many of the electrical signals handled by the Arduino are either analogue or digital. Understanding the differences between these two kinds of signals, as well as how to exploit the information they contain, is critical.

4.2.2 ANALOG

Humans have an analogue perception of the globe. Everything we see and hear is a continuous stream of data reaching our senses. The temperatures we perceive are never completely hot or completely cold; rather, they fluctuate between our reasonable temperature ranges. Analog data is described by this continuous stream. Analog data is estimated using only ones and zeros of digital records, which is a complementary term to Analog.

In the Arduino world, an analogue signal is simply a signal that can be HIGH (on), LOW (off), or somewhere in between. This means that the voltage value of an Analog signal will be somewhere between 0 and 5 volts (unless you mess with the

Analog Reference pin). Analog allows you to send and receive data about devices that operate in percentages and on/off modes. The Arduino accomplishes this by sampling and matching the voltage signal sent to these pins to a voltage reference signal (5V). The Arduino then assigns a numerical value to the Analog signal based on its voltage when compared to the Analog Reference signal, which is somewhere between 0 and 1023. This number will then be used by the Arduino's digital system in calculations and drawings. Analog pins # 0 - # 5 are used by the Arduino to transmit analogue input. These pins are intended for use with Analog Output components, but they can also be used for Analog Input.

There is no need to instal anything, and you can read them with the command:

```
analogRead(pinNumber);
```

where pinNumber is the Analog In pin to which the the Analog component is connected. The analogRead command returns a number in the range of 0 to 1023.

Pulse Width Modulation is a digital signal that behaves like an analogue signal and can be output by the Arduino (PWM). PWM is available on Digital Pins #3, # 5, # 6, # 9, # 10, and #11. Use the following command to generate a PWM signal: `analogWrite(pinNumber, value);`

where pinNumber is a Digital Pin with PWM capabilities and value is a number between 0 (0%) and 255 (100%). On the Arduino UNO PWM pins are signified by a ~ sign. For more information on PWM see the PWM worksheets or S.I.K. circuit 12.

The Arduino deals with electrical signals that are either input or output. Understanding the differences between these two types of signals, as well as how to exploit the information they contain, is critical.

4.2.3 OUTPUT SIGNALS

Output to the Arduino pins is always Digital, however there are two different types of Digital Output; regular Digital Output and Pulse Width Modulation Output (PWM). Output is only possible with Digital pins # 0 - # 13. The Digital pins are preset as Output pins, so unless the pin was used as an Input in the same sketch, there is no reason to use the pinMode command to set the pin as an Output. Should a situation

arise where it is necessary to reset a Digital pin to Output from Input use the command:

```
pinMode(pinNumber, OUTPUT);
```

where pinNumber is the Digital pin number set as Output. Using the following command to transmit a Digital Output signal: pinNumber, value); digitalWrite(pinNumber, value);

where value is the signal and pinNumber is the digital pin that is outputting it. When sending a digital signal, the value may be HIGH (on) or LOW (off)

```
pinNumber, value);
```

```
analogWrite(pinNumber, value);
```

where pinNumber is a Digital Pin with PWM capacities and worth is a number between 0 (0%) and 255 (100%). On the Arduino UNO PWM pins are connoted by a ~ sign. For more data on PWM see the PWM worksheets or S.I.K. circuit 12. The Arduino manages electrical signs that are either info or yield.

Output may be transmitted to a variety of instruments, but it is up to the user to determine which type of Output signal is required, connect the hardware, and then type the appropriate code to use these signals correctly.

The Arduino deals with electrical signals that are either input or output. Understanding the differences between these two types of signals, as well as how to exploit the information they contain, is critical.

4.2.4 INPUT SIGNALS

Analog Input enters your Arduino through the Analog In pins # 0 - # 5. These signals originate from analog sensors and interface devices. These analog sensors and devices use voltage levels to communicate their information instead of a simple yes (HIGH) or no (LOW). For this reason you cannot use a digital pin as an input pin for these devices. Analog Input pins are used only for receiving Analog signals. It is

only possible to read the Analog Input pins so there is no command necessary in the `setup()` function to prepare these pins for input. Using the following instruction to read the Analog Input pins:

```
pinNumber = analogRead(pinNumber);
```

The Analog Input pin number is `pinNumber`. This function will return a value between 0 and 1023 for Analog Input. A reading of zero indicates 0 Volts, and a reading of 1023 indicates 5 Volts. The analogue sensors and interfaces emit these voltage values. If you have an Analog Input that could exceed $V_{cc} + .5V$ you may change the voltage that 1023 corresponds to by using the Aref pin. This pin sets the maximum voltage parameter your Analog Input pins can read. The Aref pin's preset value is 5V.

Either of the Digital Pins # 0 - # 13 can be used to connect to your Arduino for digital input. HIGH (on, 5V) or LOW (off, 0V) digital input signals are used (Off, 0V). Instead of this,

Input can come from many different devices, but each device's signal will be either Analog or Digital, it is up to the user to figure out which kind of input is needed, hook up the hardware and then type the correct code to properly use these signals.

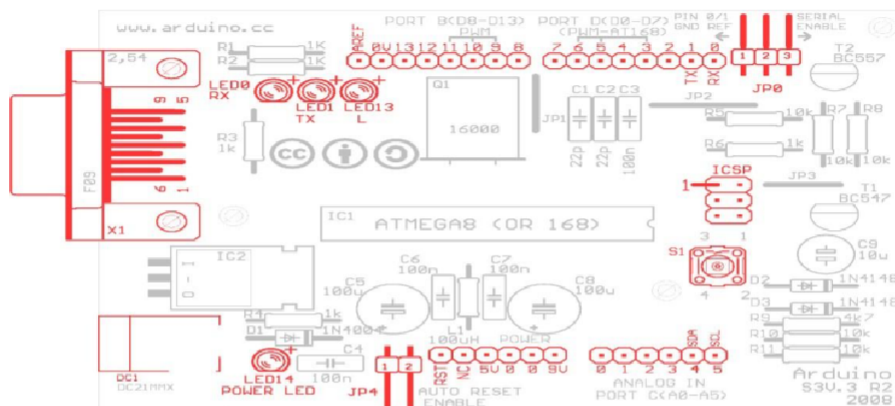


Fig:4.2.4 : Pin configuration of Arduino Uno

X1:

DE-9 serial connector

The RS-232 protocol is used to link a computer (or other device) to the internet. A serial cable with at least four pins connected (2, 3, 4, and 5) is needed.

Only when JP0 is set to position 2-3 does it work.

DC1:

2.1 mm. power jack

Connects to an external power source. Positive in the middle. Regulator of voltage Works with controlled DC voltages of +7 to +20 volts (9v. to 12v. is recommended). External power may be connected using either the 9v. pin or the 5v. pin. (See POWER PINOUT for more information.) ICSP:

Header with 2x3 pins This is how you programme an Atmega with a bootloader. The location of cable pin 1 is indicated by the number 1 on both sides of the frame. It's used to upload sketches to Atmega ICs that don't have a bootloader (available only in Arduino IDE versions 0011 and 0012).

JP0:

Jumper with three pins This jumper, when set to position 2-3, allows serial connections to and from computers and devices (via the X1 connector). This is the default location. It disables serial communication and allows external pull-down resistors on pins 0 (RX) and 1 (TX) when in position 1-2. (TX). Use this just to avoid noise on RX (which appears to be incoming data to the Atmega), which can cause the sketch to fail to start.

JP4:

When in place 1-2, the 2 pins jumper activates the auto reset function, which is useful when uploading a sketch to Arduino and automatically resetting the Atmega. When uploading sketches, it eliminates the need to click the reset button (S1). If the COM Port speed on your device is not set to 19200bps, auto reset will not work. If you delete it, the auto-reset function will be disabled. When using sketches that require serial contact, this is a perfect way to avoid an unwanted Atmega reset. The DTR pulse on serial pin 4 is used for auto reset. When connecting X1 (serial connector), Arduino detects a DTR pulse, and some softwares send a DTR pulse when it begins or closes, causing Atmega to reset when it is not needed.

S1:

Button with a tactile feel This button resets the Atmega so that the uploaded sketch can be restarted or so that Arduino can receive a sketch through the serial connector (when auto reset is not active).

4.3 GAS SENSORS

4.3.1 MQ3 SENSOR



Fig:4.3.1 : MQ-3 Sensor

From the above figure, when the target alcohol gas is present, the sensor's conductivity increases, as does the gas concentration. MQ-3 gas sensor has a high sensitivity to alcohol and is resistant to gasoline, smoke, and vapour disturbance. Based on the amount of alcohol in the device, this sensor produces an analogue resistive output. As alcohol gas is present, the sensor's conductivity rises in tandem with the gas concentration.

There is a resistance within the sensor that depends based on whether or not alcohol is detected. The more alcohol there is, the lower the tolerance. This tolerance is used to calculate the amount of alcohol present. The sensor and the load resistor form a voltage divider, and the lower the sensor resistance, the higher the voltage reading.

4.3.1.1 FEATURES OF MQ-3 ALCOHOL SENSOR

The sensor has a resistance that changes depending on whether or not alcohol is detected. The smaller the tolerance, the more alcohol there is. The amount of alcohol present is calculated using this tolerance. A voltage divider is formed by the sensor and the load resistor, and the lower the sensor resistance, the higher the voltage reading.

4.3.2 MQ4 SENSOR:



Fig:4.3.2 : MQ4 Sensor

From the above figure, the MQ4 Methane Gas Sensor measures the amount of methane gas in the air and outputs an analogue voltage as a result. With a concentration sensing spectrum of 300 ppm to 10,000 ppm, leak detection is possible. The sensor might, for example, detect whether a gas stove was switched on but not lit. The sensor operates in temperatures ranges from -10 to 50°C and draw lesser than 150 mA at 5 V. By connecting five volts between the heating (H) pins, the sensor is held hot enough to operate properly. The sensor emits an analogue voltage on the other pins while the A or B pins are wired to five volts. A resistive load between the output pins and ground controls the detector's sensitivity.

4.3.2.1 MQ4 METHANE GAS SENSOR - GENERAL SPECIFICATIONS

- In a wide variety, good exposure to flammable gas
- Natural gas exposure is high.
- Longevity and low price
- a basic drive circuit

4.3.3 MQ6 SENSOR:



Fig:4.3.3 : MQ6 Sensor

From the above figure, this is an easy-to-use liquefied petroleum gas (LPG) sensor that detects LPG levels in the air (mostly propane and butane). Gas concentrations ranging from 200 to 10,000 parts per million can be detected by the MQ-6 (ppm). This is an easy-to-use liquefied petroleum gas (LPG) sensor that detects LPG levels in the air (mostly propane and butane).

4.3.3.1 LPG GAS SENSOR (MQ6) FEATURES

- LPG, isobutane, and propane have a high sensitivity.
- Small sensitivity to alcohol, smoke
- Detection Range of iso-butane propane 100-10,000 ppm
- Fast Response Time: <10s
- Simple drive circuit
- Heater Voltage: 5.0V

4.3.4 MQ7 SENSOR



Fig:4.3.4 : MQ7 Sensor

This CO gas sensor detects CO quantities in the air and produces an analogue voltage as a result. The sensor can measure concentrations as low as 10 parts per million to as high as 10,000 parts per million. The sensor works in temperatures ranging from -10 to 50°C and draws less than 150 milliamperes at 5 volts.

4.3.5 MQ8 SENSOR



Fig:4.3.5 : MQ8 Sensor

From the above figure, this is a simple hydrogen gas sensor that can be used to measure the presence of hydrogen in the air. The MQ-8 can detect hydrogen gas concentrations as low as 100 parts per million and as high as 10,000 parts per million (ppm).

The sensitivity of this sensor is high, and the response time is fast. The output of the sensor is an analogue resistance. The drive circuit is straightforward: simply supply 5V to the heater coil, add a load resistance, and link the output to an ADC.

The MQ-8 gas sensor's sensitive material is SnO_2 , which has a low conductivity in pure air. When hydrogen gas is present, the sensor's conductivity increases in accordance with the gas concentration. The change in conductivity can be converted to the corresponding output signal of gas concentration using a simple circuit. The MQ-8 gas sensor detects hydrogen gas with high sensitivity and does not interfere with other gases. This sensor is capable of detecting hydrogen, which is particularly useful in urban environments. It's a low-cost sensor that has a wide range of applications.

4.3.5.1 FEATURES

It has a large variety of exposure to hydrogen gas, as well as benefits such as long lifetime, low cost, and a simple drive circuit, among others.

4.3.5.2 MAIN APPLICATIONS

Domestic gas leakage alarms, commercial flammable gas alarms, and portable gas detectors all use it.

4.3.6 MQ135 SENSOR



Fig:4.3.6 : MQ135 Sensor

In air quality control devices, the MQ-135 Gas Sensors are suitable for detecting or calculating NH_3 , NO_x , Alcohol, Benzene, Smoke, and CO_2 . When you only want to detect one gas, the MQ-135 sensor module has a Digital Pin, which enables it to operate without a microcontroller. You'll need to use the analogue screw if you need to count gases in PPM. The analogue pin is TTL powered and operates at 5V, making it compatible with a huge variety of microcontrollers.

This sensor could be the right option for you if you need a sensor to detect or measure common air quality gases like CO_2 , Smoke, NH_3 , NO_x , Alcohol, or Benzene.

4.3.6.1 APPLICATIONS:

- Used to detect leakage/excess of gases like Ammonia, nitrogen oxide, alcohols, aromatic compounds, sulfide and smoke.
- Air quality monitors.

4.4 BUCK CONVERTER



Fig:4.4 : Buck Converter

Voltage regulators that use a buck converter or a step down converter. Since it's a DC-DC converter, it completes the job with the aid of a few transistor switches and an inductor. Buck converters operate in a manner that is identical to PWM 'dimming.' We've all heard of a PWM signal dimming lights.

4.4.1 WORKING OF A BUCK CONVERTER

The working of a buck converter can be broken down into a few steps.

STEP – 1:

The switch is turned on, allowing current to flow to and charge the output capacitor. Since the voltage across the capacitor cannot increase rapidly and the charging current is regulated by the inductor, the voltage across the capacitor during the switching cycle doesn't quite equal the highest voltage of the power supply.

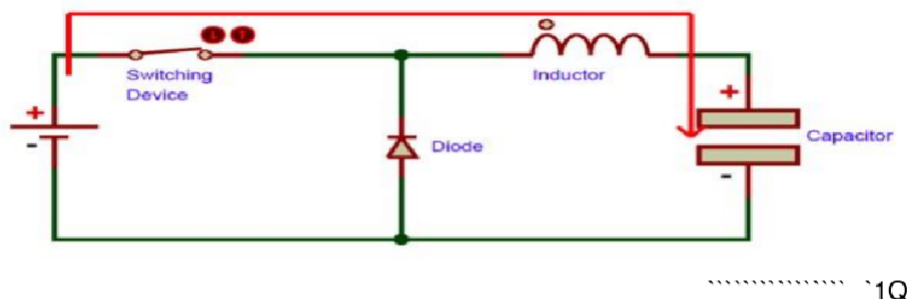


Fig:4.4.1.1 : Step 1 Working

STEP – 2:

The light has now been switched off. Since an inductor's current cannot shift unexpectedly, it induces a voltage through it. When the switch is switched off, this voltage is allowed to charge the capacitor and power the load via the diode, retaining current output current throughout the switching cycle.

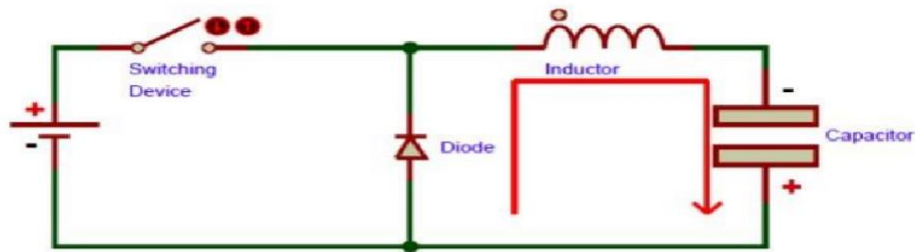


Fig:4.4.1.2 : Step 2 Sinking

These two steps are repeated thousands of times per second, resulting in a continuous stream of output.

Note that the design process outlined here is not meant to be an expert rant; rather, these are really specific equations for getting the first buck converter up and running.

Again, regulation is a topic that has not been discussed in this paper. If you want to make a controlled buck converter, you can use a buck controller IC like the LM2596, which is designed specifically for that purpose.

4.5 ESP32 MICROCONTROLLER:

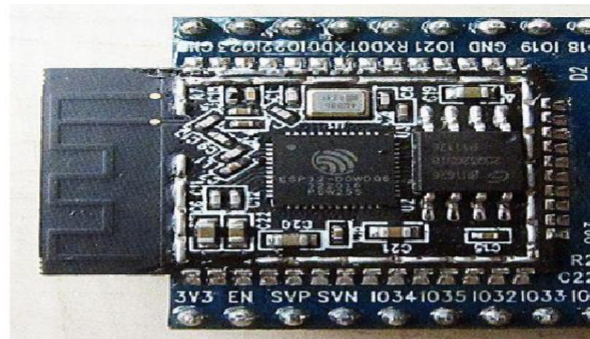


Fig:4.5 : ESP32 Microcontroller

The ESP32 is a low-cost, low-power system-on-a-chip (SoC) microcontroller with built-in Wi-Fi and dual-mode Bluetooth. Antenna switches, RF baluns, control amplifiers, low-noise receive amplifiers, filters, and power-management modules are all integrated into the ESP32 range, which also includes a Tensilica Xtensa LX6

microprocessor in dual-core and single-core models. A Shanghai-based Chinese company, Espressif Systems designed and produced the ESP32, which is manufactured by TSMC using their 40 nm process. [two] It is the ESP8266 microcontroller's successor.

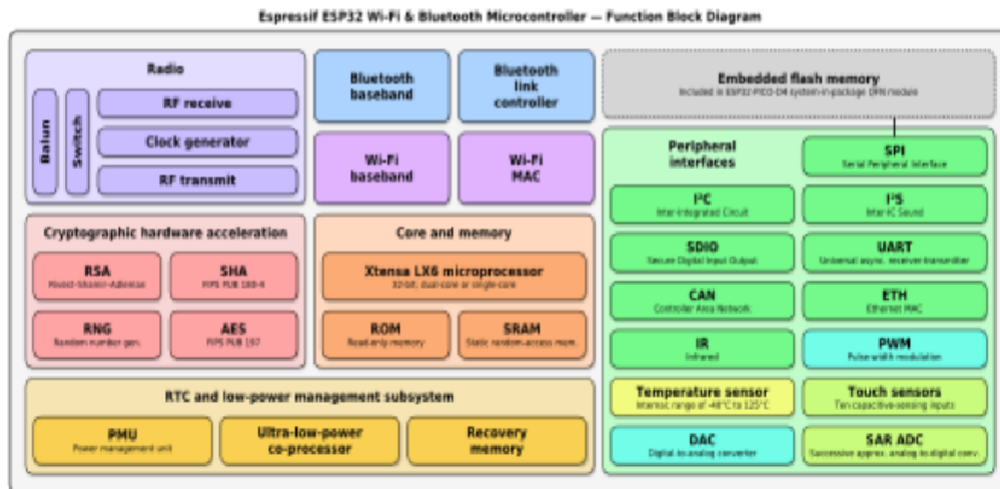


Fig:4.5.1 : ESP32 Function Block Diagram

From the above figure, Time Protocol support, CAN bus 2.0, Infrared remote controller (TX/RX, up to 8 channels), Motor PWM, LED PWM (up to 16 channels), Hall effect sensor, Ultra low power analog pre-amplifier, Security: IEEE 802.11 standard security features all supported, including WPA,

4.5.1 PROGRAMMING

ESP32 programming languages, systems, platforms, and environments include:

The ESP32 Arduino Core is supported by the Arduino IDE.

Espressif IoT Development Framework (IDF) – Espressif's official development framework for the ESP32 microcontroller.

MicroPython is a Python programming language. A microcontroller-friendly Python 3 implementation

Mesh Development Framework by Espressif

Espruino is a JavaScript SDK and firmware that highly resembles the Node.js framework.

4.5.2 OTHER CHIPS IN ESP32 FAMILY

In 2019, Espressif launched the ESP32-S2, a new chip based on a single Xtensa LX7 core CPU with up to 240 MHz clock speed. It has 320 kB SRAM, 128 kB ROM, and 16 kB RTC memory and can perform 64 FLOPS per loop. [52]

On December 1st, 2020, Espressif released the ESP32-S3, a new microcontroller with a dual-core Xtensa LX7 CPU core running at up to 240MHz, 384 KB of RAM, and another 384 KB of SRAM. The LX7 core on the Espress if platform provides new LX7 instruction set extensions aimed at enhancing the performance of Artificial Intelligence applications.

The S3 family incorporates the S2 family's improved hardware-level security, as well as other ESP32-S2 features such as USB On-The-Go support and an improved touchpad sensor implementation. There are up to 44 programmable GPIOs on the ESP32S3. It also includes some of the features found in the first generation of ESP32 microcontrollers, such as Bluetooth 5 and Bluetooth LE support, an SD/MMC host controller, and a "World Controller" that provides a completely isolated execution environment for security-critical applications.

4.6 DISPLAY MODULE



Fig:4.6 : ILI 9341 Display Module

From the above figure, the display controller for the ILI9341 TFT module is also called ILI9341. It's a low-cost, easy-to-use colour display that uses the SPI interface protocol and includes 4 or 5 control pins. This TFT monitor has a resolution of 240 x 320 pixels, or 76800 pixels. This module only accepts 3.3V and does not support 5V. (not 5V tolerant).

The ILI9341 controller chip is included in this LCD board, and it allows for different pin-saving SPI communication without losing data transmission speed.

Since the LCD module's interface is SPI, users would need 6 Dupont lines (SCK,MISO,MOSI,CS,D/C,RST) to control it. An SD card slot is also included in this module.

4.7 ARDUINO SOFTWARE

A text editor for writing code, a message area, a text console, a toolbar with buttons for basic functions, and a set of menus are all included in the Arduino Software (IDE). Through linking to them and importing programmes, it connects with the Arduino and Genuino hardware.

4.7.1 WRITING SKETCHES

Sketches are programmes created with the Arduino Software (IDE). These sketches were made in a text editor and saved with the.ino file extension. Cutting/pasting, as well as searching/replacing text, are all available in the editor. The message area shows errors and provides input when saving and exporting. The Arduino Program (IDE) outputs text to the console, which includes complete error messages and other information. The designed board and serial port are shown in the window's bottom righthand corner. You can verify and upload programmes, make, open, and save sketches, and open the serial monitor using the toolbar buttonsNB: Previous versions of the Arduino Software (IDE) used the extension.pde to save sketches. These files can be opened with version 1.0, but you'll be asked to save the sketch with the.ino extension when you save it.

The five menus Paper, Edit, Sketch, Resources, and Support include additional commands. The menus are context sensitive, which means they only display things that are important to the work being performed at the moment.

4.7.2 TOOLS

- The coding is indented such that the opening and closing curly braces line up, and the sentences inside curly braces are indented even further.
- Sketch from the Archives

Store a copy of the latest sketch in the.zip format. The archive is saved in the same directory as the drawing.

- Reload and fix the encoding

Fixes any inconsistencies between the editor's char map encoding and the char maps of other operating systems.

- *Board*

Choose the board you want to use. The outlines of the different boards can be found below.

- *Port*

This menu lists all of your machine's serial machines, both actual and virtual.

When you open the top-level tools menu, it should immediately refresh.

☐ *Programmer*

When programming a board or chip using a hardware programmer rather than the onboard USB-serial link. You won't need this most of the time, but if you're burning a bootloader to a new microcontroller, you'll need it.

☐ *Burn Bootloader*

The items in this menu enable you to burn a bootloader onto an Arduino board's microcontroller. This is not needed for regular Arduino or Genuino board use, but it will come in handy if you buy a new ATmega microcontroller (which normally come without a bootloader). Before burning the bootloader on the target board, make sure you've picked the right board from the Boards menu.

4.7.3 HELP

A variety of documents related to the Arduino software can be found here (IDE). If you don't have internet connectivity, you can access Getting Started, Reference, this IDE guide, and other documents locally. The documents are a localised version of the online versions and may provide a link to our website.

This is the Help menu's only interactive feature: it selects the appropriate page for the function or order under the cursor from the local copy of the Reference.

4.7.4 SKETCHBOOK

The Arduino Software (IDE) is based on the idea of a sketchbook, which is a typical location for storing your programmes (or sketches). The Open button on the toolbar or the File > Sketchbook menu can be used to open the sketches in your sketchbook. The Arduino programme will build a directory for your sketchbook the first time you run it. With the Preferences dialogue, you can see or adjust the position of the sketchbook.

Files are saved with a.ino extension starting with version 1.0. The.pde extension was used in previous versions. In version 1.0 and later, you can still open.pde files; however, the programme will automatically rename the extension to.ino.

4.7.5 TABS, MULTIPLE FILES, AND COMPILATION

Allows you to handle drawings from several files (each of which appears in its own tab). There can be plain Arduino code (no extension), C files (.c extension), C++ files (.cpp extension), or header files (.h).

4.7.6 UPLOADING

Before uploading your drawing, make sure you choose the correct things from the Tools > Board and Tools > Port menus. The boards are listed in the following sections. The serial port on a Mac is possibly /dev/tty.usbmodem241 (for an Uno, Mega2560, or Leonardo), /dev/tty.usbserial-1B1 (for a Duemilanove or earlier USB board. On Windows, it's most likely COM1 or COM2 (for a serial board) or COM4, COM5, COM7, or higher (for a USB board); to find out, look in the ports section of the Windows System Manager for USB serial device. It should be /dev/ttyACMx, /dev/ttyUSBx, or something similar on Linux. If you've found the correct serial port and board, click the import button in the toolbar or choose the Upload object from the Sketch menu. Current Arduino boards will reboot and begin uploading automatically.. You'll need to click the reset button on the board just before beginning the upload on older boards (pre-Diecimila) that don't have auto-reset. When the sketch is posted, the RX and TX LEDs on most boards can blink. When the upload is complete, the Arduino Program (IDE) will show a message or an error.

When you upload a sketch, you're using the Arduino bootloader, which is a tiny programme loaded onto your board's microcontroller. It enables you to upload code without the need for any extra hardware. When the board resets, the

bootloader is active for a few seconds before starting whatever sketch was most recently uploaded to the microcontroller. When the bootloader begins, the onboard (pin 13) LED will blink (i.e. when the board resets).

4.7.7 LIBRARIES

Libraries add extra functionality to drawings, such as the ability to interact with hardware or manipulate data. Pick a library from the Sketch > Import Library menu to use it in a sketch. This will compile the library with your sketch and add one or more `#include` statements to the top of the sketch. Libraries take up more room on the board than your sketch since they are uploaded with it. Remove the `#include` statements from the top of your code if a sketch no longer needs a library.

In the reference section, there is a list of libraries. The Arduino app comes with a few libraries. Others are available for download from a number of places, like the Library Manager. You can import a library from a zip file and use it in an open sketch starting with version 1.0.5 of the IDE. To build a third-party library, follow these instructions.

4.7.8 THIRD-PARTY HARDWARE

The hardware directory in your sketchbook directory may be connected to help for third-party hardware. fabricated incorporate board definitions (which show up in the board menu), center libraries, bootloaders, and software engineer definitions. Make the equipment catalog first, at that point unfasten the outsider stage into its own subdirectory. (On the off chance that you use "arduino" as the sub-registry. Simply delete the program's directory to uninstall it.

The Arduino IDE 1.5 3rd party Hardware specification has more information on developing packages for third-party hardware.

4.7.9 SERIAL MONITOR

The Arduino or Genuino board's serial data is shown (USB or serial board). Type text and press enter or click the "send" button to send data to the device. From the drop-down menu, choose the baud rate that corresponds to the rate passed to Serial. Begin with a sketch. The Arduino or Genuino board will reset when you

connect with the serial monitor on Windows, Mac, or Linux (restart your sketch execution from the beginning).

You can also communicate with the board using Processing, Flash, MaxMSP, and other programming languages (see the interfacing page for details).

4.7.10 PREFERENCES

In the preferences dialogue, you can set some preferences (found under the Arduino menu on the Mac, or File on Windows and Linux). The rest is in the preferences file, which is shown in the preference dialogue.

4.7.11 LANGUAGE SUPPORT



From the above figure, the Arduino Software (IDE) has been translated into 30+ languages since version 1.0.1. The IDE starts up in the language that your operating system has chosen by default. (Note that on Windows and most likely Linux, the area climate, which oversees money and date designs, decides this, not the language wherein the working framework is shown)

If you need to change the language physically, open the Preferences window in the Arduino Software (IDE). A dropdown menu of presently upheld dialects shows up close to the Editor Language. Pick your #1 language from the menu, and afterward restart the program to utilize it. The Arduino Program (IDE) will change to English if your working framework language isn't empowered.

Select System Default from the Editor Language drop-down to return the programme to its default setting of choosing the language depending on your operating system. When you restart the Arduino Program, this setting will take effect (IDE). Similarly, you must restart the Arduino Software (IDE) after updating your operating system's settings to upgrade it to the new default language.

4.7.12 BOARDS

The board selection has two effects: it controls the parameters used by the burning bootloader command when compiling and uploading sketches (such as CPU speed and baud rate), as well as the file and fuse settings used by the burn bootloader command. Only in the above do some of the board definitions differ, so double-check before burning the bootloader. even though you've been uploaded successfully with a specific range. A comparison table of the different boards can be found here.

The Arduino Software (IDE) includes support for the following boards, which are all based on the AVR Core. The Boards Manager, which comes standard with the installation, lets you add support for a growing number of new boards based on various cores including the Arduino Due, Arduino Zero, Edison, Galileo, and others.

4.8 NODE RED

Node RED is a software application that allows us to create new and interesting connections between hardware, APIs, and online services.

It includes a browser-based editor that makes it simple to link flows to the palette's large number of nodes, which can all be deployed to the runtime with a single click.

The change node is another helpful node that allows you to change the message payload or add new properties. You may use this node to modify a message's properties, either by modifying existing ones, removing them, or adding new ones.

Continue with your MQTT theme in this example to see how you can add a new message property called `msg.payload`. It's important to remember that you've now successfully "switched" the message flow based on the incoming MQTT message.

You can use Node-RED to link nodes to generate flows that carry out your programming mission. Messages move between nodes, from input nodes to processing nodes and eventually to output nodes. Let's look at nodes, flows, and messages in more detail.

There are three main types of nodes:

1. Input Nodes (e.g. inject)
2. Output Nodes (e.g. debug)
3. Processing Nodes (e.g. function)

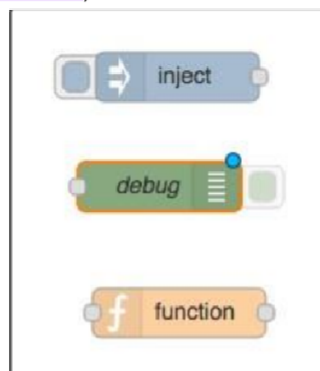


Fig 4.9 The main types of nodes: input, output and processing

From the above figure, you may use input nodes to enter data into a Node-RED application or "flow." They only have one output endpoint on their right hand, which is defined by the small grey rectangle. Input nodes are used to link data from other

services, such as Twitter, Google, serial, websockets, or tcp, or to manually enter data into a flow using the inject node.

You can send data outside of a Node-RED flow using output nodes. On their left hand, they have a single input endpoint. You can send data to other services using output nodes, such as Twitter, TCP, serial, or email nodes, or you can use the debug node to output to the debug pane.

You can process data using processing nodes. They have one or more output endpoints and an input endpoint. They allow you to convert data types (e.g. json, csv, xml) nodes, use data to trigger a message (e.g. trigger, delay) nodes, and write custom code that makes use of the data obtained (e.g. function node).

Some nodes, such as the inject and debug messages, have a button that enables you to actuate a node (in the case of the inject node) or activate and disable a node (in the case of the debug node) (in the case of the debug node).

Flows are made up of multiple nodes that are connected by output tabs that are linked to the input tabs of the next node in the flow. Messages travel across the nodes, bringing data from one node to the next.

Nodes in the Node-RED system consume input messages and generate output messages. Messages are JavaScript objects with at least one "payload" parameter, as seen here:

Listing 2.1 A basic Node-RED message structure

```
1.msg = {  
2.payload:"message payload"  
3.};
```

Nodes consume and generate messages, and the data they consume and produce is normally stored in msg.payload. Messages, on the other hand, may be expanded to provide additional parameters. To set a message's topic and add a new

parameter, position, for example, you could build a new msg object as shown in listing 2.2.

Listing 2.2 A more complex Node-RED message structure

```
1.msg = {  
2.payload:"message payload",  
3.topic:"error",  
4.location:"somewhere in space and time"  
5.};
```

4.8.1 BROWSER-BASED FLOW EDITING

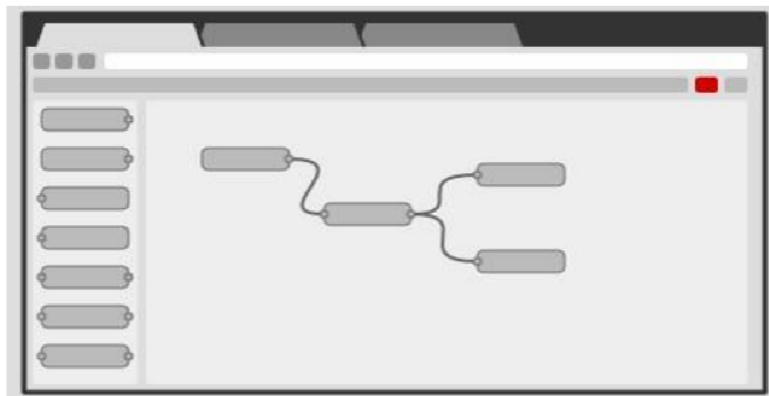


Fig:4.8.1.1 : Browser based flow editing

From the above figure, Node-RED comes with a browser-based flow editor that allows linking flows with the palette's many nodes a breeze. Flows can then be deployed to the runtime with only one press.

Inside the editor, a rich text editor can be used to create JavaScript functions.

You can save and reuse useful features, templates, and flows in a built-in library.

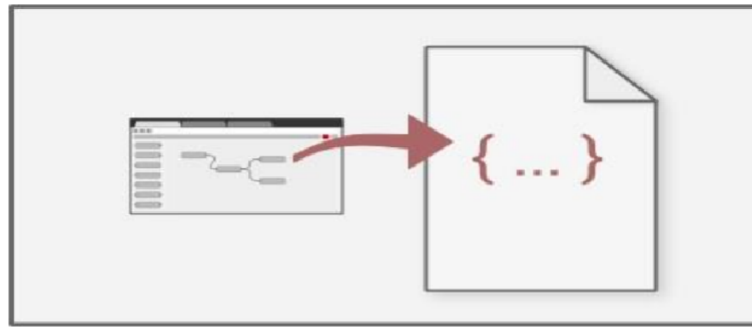


Fig:4.8.1.2 : Using json file save the document

From the above figure ,Node-RED flows are saved in JSON format, which can be conveniently imported and exported for sharing with others.

You can share your best flows with the rest of the world using an online flow library.

4.9 NEUROLOGICAL DISEASES

The brain, spine, and nerves that link them are all affected by neurological disorders. There are over 600 nervous system disorders, including brain tumours, epilepsy, Parkinson's disease, and stroke.

4.9.1 ACUTE SPINALCORD INJURY

The vertebrae are a collection of bones that made the spine. Your spinal cord moves through a canal in the middle of these bones on its way down. A bundle of nerves that transmits movement and feeling signals from the brain to the rest of the body is spinal cord.

Acute spinal cord injury (SCI) occurs when the spinal cord is bruised, partially torn, or completely torn as a result of a traumatic injury. In both children and adults, SCI is a common cause of permanent disability and death.

The following are the most common signs and symptoms of an acute spinal cord injury:

- Loss of voluntary muscle function in the chest, arms, or legs

- Muscle weakness
- Breathing difficulties
- Head, arm, or leg numbness or tingling
- Dysfunction of the bowels and bladder

4.9.2 ALZHEIMERS DISEASE

Alzheimer's disease is a neurologic disorder that causes the brain to shrink (atrophy) and the death of brain cells. Alzheimer's disease is the most common cause of dementia, which is described as a progressive loss of cognitive, behavioural, and social abilities that impairs a person's ability to function independently.

One of the first signs of Alzheimer's disease is forgetting recent events or conversations. As Alzheimer's disease progresses, a person can experience severe memory loss and lose the ability to carry out everyday activities.

Medications can improve or delay the progression of symptoms temporarily. These therapies may also help people with Alzheimer's disease preserve their independence and maximise their work. People with Alzheimer's disease and their families will benefit from a range of programmes and services..

4.9.3 ATAXIA

Ataxia is a nervous system degenerative disease. Slurred speech, stumbling, dropping, and incoordination are all signs of ataxia, as are slurred speech, stumbling, falling, and incoordination. Damage to the cerebellum, some part of the brain responsible for movement coordination, causes these symptoms. Ataxia is treated with a combination of medicine and therapy to symptoms will be relieved, and the quality of life will increase. Ataxia patients may have difficulty using their fingers and hands, arms, legs, walking, speaking, or moving their eyes. People of all ages are

affected by ataxia. Symptoms may appear at any age, ranging from childhood to late adulthood. The disease's complications are severe and sometimes debilitating.

Some forms of ataxia may result in death.

4.9.3.1 SYMPTOMS

- A lack of coordination.
- Unsteady gait and a proclivity to trip.
- Difficulty performing fine motor activities like chewing, writing, or buttoning a shirt.
- A change in speech pattern.
- Nystagmus (involuntary back-and-forth eye movements) • Swallowing difficulties

4.9.4 AMYOTROPHIC LATERAL SCLEROSIS

Amyotrophic lateral sclerosis (ALS), also known as Lou Gehrig's disease, is a progressive nervous system disease that affects nerve cells in the brain and spinal cord, causing muscle failure.

Muscle twitching and weakness in a leg, as well as slurred speech, are common symptoms of ALS. ALS eventually wreaks havoc on the muscles that allow you to move, talk, eat, and breathe. This deadly disease has no known cure.

4.9.4.1 SYMPTOMS

ALS symptoms and signs differ significantly from person to person, depending on which nerves are affected. The following are possible signs and symptoms:

- Difficulty walking or doing regular everyday tasks
- Tripping and falling
- Leg, foot, or ankle weakness

- Hand weakness or clumsiness
- Slurred speech or trouble swallowing
- Muscle cramps and twitching in your arms, shoulders, and tongue
- Inappropriate crying, smiling, or yawning

Environmental factors, such as the following, might trigger ALS.

- *Smoking:* The only known environmental risk factor for ALS is smoking. Women tend to be at the greatest risk, particularly after menopause.
- *Environmental toxin exposure:* Exposure to lead or other chemicals in the workplace or at home, according to some evidence, can be related to ALS. Despite extensive research, no single agent or chemical has been reliably linked to ALS.
- *Military service:* According to studies, people who have served in the military are more likely to develop ALS. It's unclear what aspect of military service can lead to the onset of ALS. Exposure to certain metals or chemicals, physical injuries, viral infections, and heavy exertion are all possible causes.

4.9.5 BELL'S PALSY:

Bell's palsy is a condition in which facial muscles become weak or paralysed for a short period of time. Facial paralysis may occur when the nerve that controls the facial muscles becomes inflamed, swollen, or cramped.

As a consequence, one side of the face droops or stiffens of the disease. On the affected hand, you can find it difficult to smile or close your eye. Bell's palsy is usually only temporary, with symptoms disappearing after a few weeks.

4.9.5.1 SYMPTOMS

- drooling

- trouble eating and drinking
- failure to make facial movements like smiling or frowning
- facial fatigue
- facial muscle twitches
- dry eye and mouth
- headache
- sensitivity to sound
- discomfort of the involved eye

4.9.5.2 RISK FACTORS:

Your risk of developing Bell's palsy increases if you:

- you're pregnant
- you've got asthma
- you've got a lung infection
- you've got a family history of the disease

4.9.6 BRAIN TUMORS

A brain tumour is a group of abnormal cells in your brain that forms a mass. Your brain is protected by a very rigid skull. Any expansion in such a small space can cause problems. Brain tumours can be malignant (cancerous) or benign (noncancerous) (benign). The pressure within your skull will rise as benign or malignant tumours grow larger.

4.9.6.1 SYMPTOMS

- A common symptom of a brain tumour is headaches. You may get headaches that:
 - are worst when you first wake up
 - happen when you're sleeping
 - are aggravated by coughing, sneezing, or exercise

You may also experience:

- vomiting
- double vision or blurred vision
- confusion
- seizures (especially in adults)
- a change in mental functioning
- weakening of a leg or part of the face

4.9.7 CEREBRAL ANEURYSM

CEREBRAL ANEURYSM A cerebral aneurysm (additionally called a brain aneurysm) is a feeble or slight spot on a cerebral conduit that lumps out and loads up with blood. An aneurysm that has blasted may put tension on nerves or brain tissue. It might likewise burst or burst, emptying blood into close by tissue (called a bleed). Hemorrhagic stroke, brain injury, unconsciousness, and even demise may all outcome from a burst aneurysm.

Some cerebral aneurysms, particularly the smallest ones, do not bleed or cause other complications. Aneurysms of this kind are normally discovered during imaging tests for other medical issues. Aneurysms can develop anywhere in the brain, but the majority form in the main arteries that run along the base of the skull.

4.9.7.1 SYMPTOMS

An aneurysm that is steadily increasing in size can push on tissues and nerves, resulting in:

- a dilated pupil in the eye
- discomfort above and below the eye
- numbness
- fatigue
- paralysis on one side of the face
- vision shifts or double vision

4.9.7.2 RISK FACTORS FOR DEVELOPING AN ANEURYSM

Aneurysms in the brain are often caused by hereditary risk factors, such as:

- artery-weakening hereditary connective tissue disorders
- polycystic kidney disorder (polycystic kidney disease) (in which numerous cysts form in the kidneys)
- arteriovenous malformations (blood flow is disrupted by tangled arteries and veins in the brain). Some AVMs appear at random or on their own.)
- a first-degree relative with a history of aneurysm (child, sibling, or parent).

4.9.8 EPILEPSY AND SEIZURES

Seizures are a symptom of epilepsy, which are abnormal movements or behaviour caused by abnormal electrical activity in the brain. However, not everybody who seems to have seizures has epilepsy, a category of conditions marked by a proclivity for frequent seizures.

4.9.8.1 CAUSES OF EPILEPSY

Epilepsy is a brain disorder characterised by abnormal electrical activity. Electrical signals are sent in a regular pattern by brain cells to communicate. These electrical signals become abnormal in epilepsy, resulting in a "electrical storm" that causes seizures. These storms can occur in a specific part of the brain or be generalised, depending on the type of epilepsy.

4.9.8.2 SYMPTOMS

There is no loss of consciousness in a clear partial seizure. Among the signs and symptoms are:

- dizziness
- limb tingling and twitching
- Taste, smell, sight, hearing, and touch changes

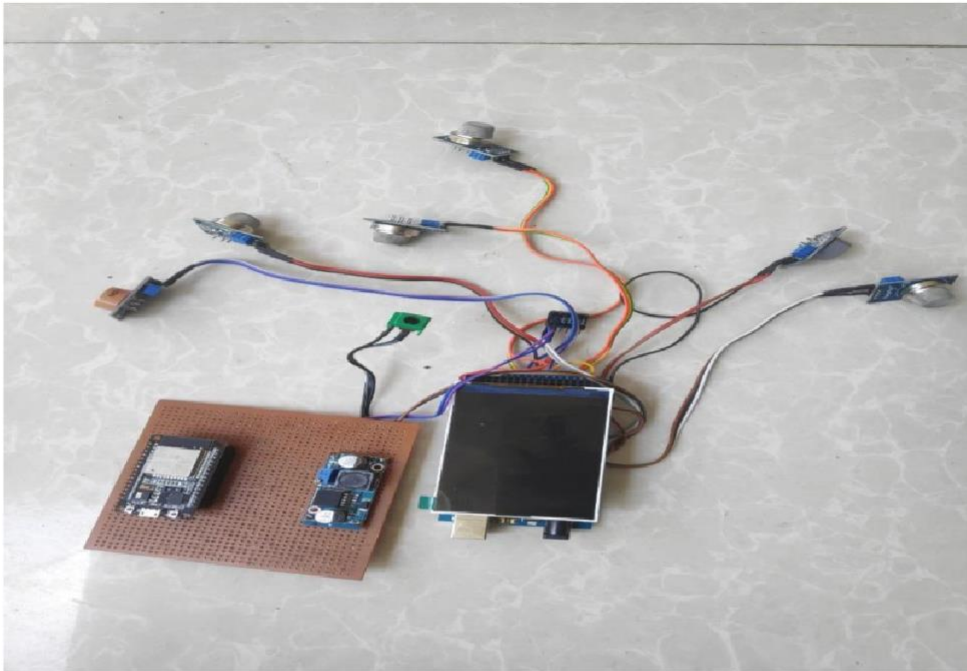
There is a lack of memory or consciousness in complex partial seizures. Some signs and symptoms include:

- staring blankly
- unresponsiveness

CHAPTER 5

RESULT

The sensors that are used in project are MQ-3, MQ-4, MQ-6, MQ-7, MQ8, MQ-135. The microcontroller is used to interface the gas sensors. Every sensor has one's own perspective. MQ3 is a gas detector that detects benzene whereas MQ4 is a gas detector that detects smoke and MQ6 is a gas detector that detects methane gas and MQ7 is a gas detector that detects carbon monoxide gas, MQ8 sensor is a gas detector that detects hydrogen gas and MQ135 sensor is a gas detector that detects carbon monoxide, alcohol, carbon dioxide, toluene, ammonia, acetone gases. These sensors were interfaced with microcontroller. Sensors detect the values and these values are given as input to Arduino and displayed on the display module. And this data will be sent to laptop through Wi fi module present in ESP32 microcontroller using mqtt protocol. These values will be sent to node red platform. There we can detect the diseases using java script. Hence, we can detect the diseases.



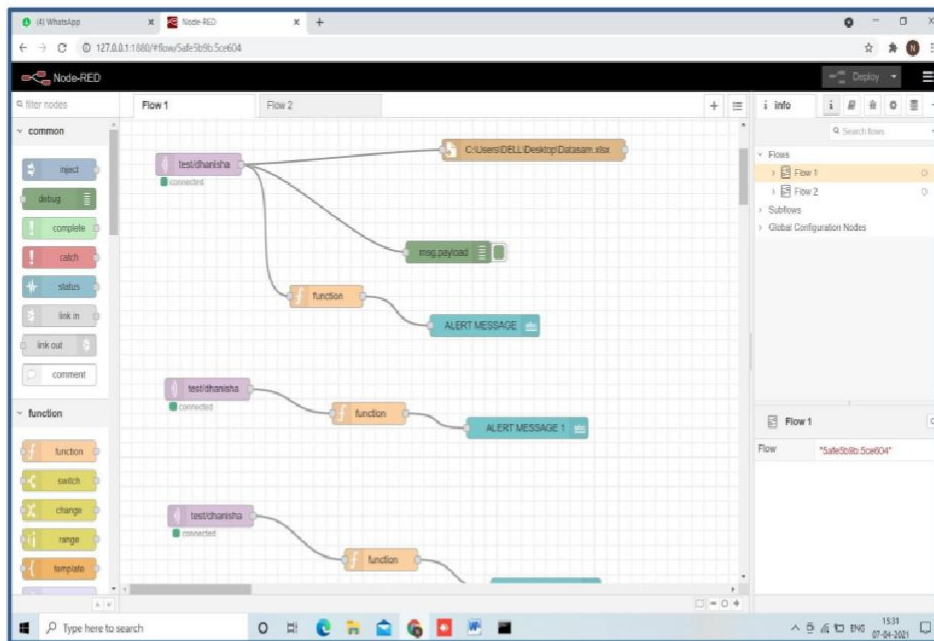
Hardware setup

From the above figure, this is hardware connections where all the sensors, display module, buck converter are connected to Arduino pins.

From the below figure indicates the values that are detected from each gas sensor and those values are displayed on the display module.

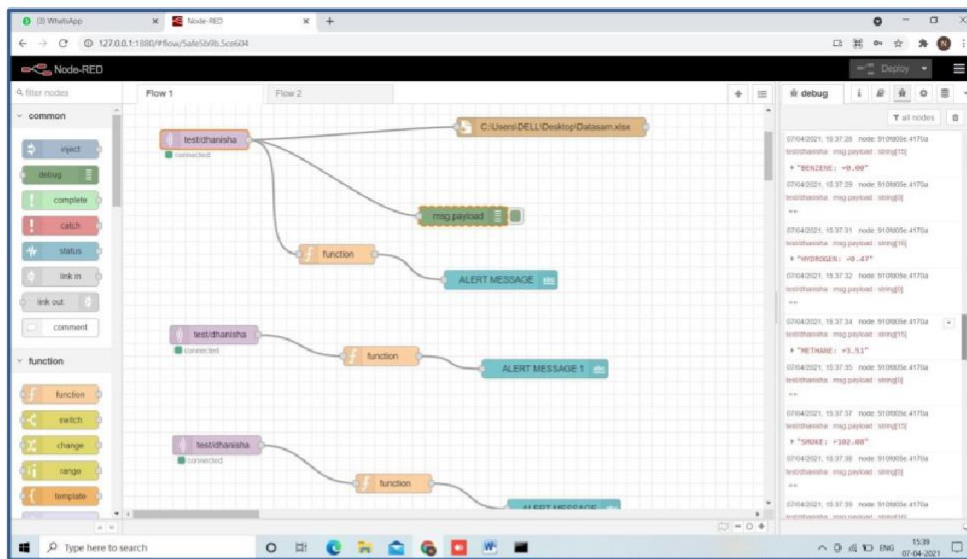


Values of gas sensors on display module



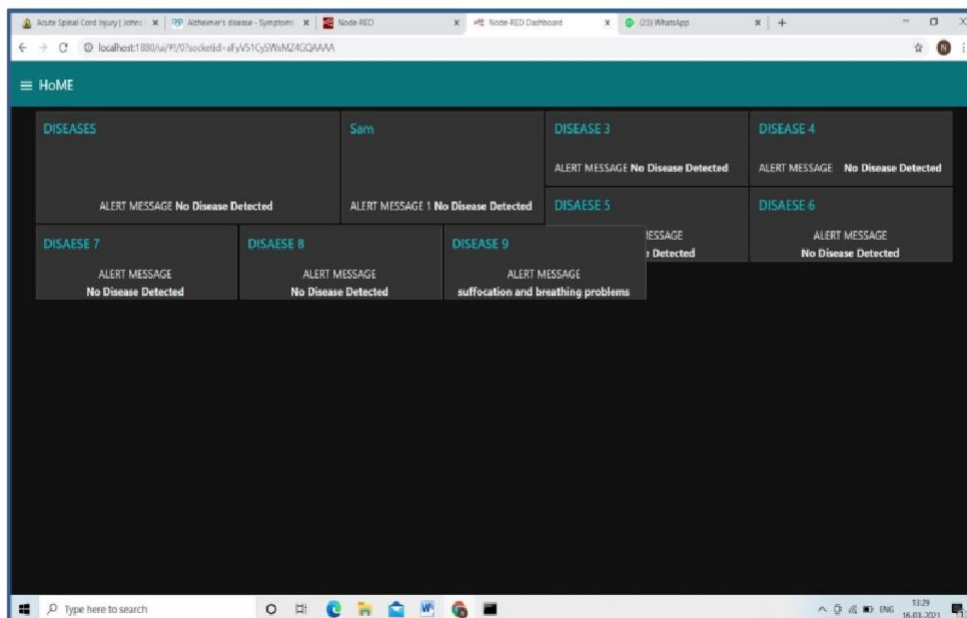
Node Red platform

The above figure is a node red platform. It's a flow editor that runs in a web browser. As part of the Internet of Things, it's used to connect hardware devices, APIs, and online services. It's used to collect data from the sensors and send us warning messages.



Debug window in node red platform

The above figure is the debug window in node red platform. Here we can see the values of each gas and these values will be changing every time depending on the present gas value in node red and as well as in display module which we got from the sensors.



Alert messages on node red dashboard

From the above figure,we can see the alert messages displaying on the node red dashboard. The message displayed on the dashboard changes everytime with the values given by the sensors.So,this is the final result and alert message.

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

Air pollutants are caused by a mixture of gaseous, volatile substances, semi volatile and particulate matter substances. All of these pollutants are released into the atmosphere as a result of human activity. Air pollution can cause a variety of health problems, including respiratory and neurological disorders, which pose a serious threat to human life. There is growing evidence that air pollutants can not only harm the central neural system, but may even increase the risk and mortality of Alzheimer's disease and other neurological diseases like Ataxia, Brain tumours, Acute spinal cord injury, Bells palsy and Amyotrophic lateral Sclerosis. By using various sensors such as MQ3, MQ4, MQ6, MQ7, MQ8 and MQ135. These sensors can detect gases such as Benzene, Smoke, Methane, Carbon dioxide, Hydrogen, Alcohol, Carbon monoxide, Ammonia, Acetone, Toluene. Using node red platform we can able to promote well being in humans by giving advance alerts and alarms over an area. By analyzing the air quality in a specific region, this project can be used to detect neurological diseases.

REFERENCES

- [1] Albert M.K Cheng ; Joe Air Jiang was presented this feature with the temperature- humidity sensor module at the parameter of NO₂_mean-0.168 by using the GRIMM's light- scattering fine- dust method at the accuracy of 76%.
- [2] Chiao-Lun Lee; Jin-Shyan Lee was presented this feature with the temperature-humidity sensor at the parameter of Temp-0.412 using the LS estimation method at the accuracy of 26%
- [3] Giacomo Boracchi; Michalis Michaelides; was presented this feature with the VOC sensor module GSBT11-P110 at the parameter of Intercept-14.74 by using the method of GRIMM's light- scattering fine- dust at the accuracy of 72%
- [4] JeongGil Ko; Samuel Keene was presented this feature with the pollutant detection sensors at the parameter of NO₂_mean-0.168 using the proposed calibration method at the accuracy 58%.
- [5] Jungmo Ahn; Hyungi Kim; was presented this feature with the Temperature and Humidity sensor (DHT11) at the parameter of NO_DG-0.250 using a linear least square-based method for sensor calibration at the accuracy of 92%.
- [6] Manuel Roveri; Rita Paradiso was presented this feature with the Air Quality sensor (MQ 135) at the parameter of NO_mean-0.125 by using the L.S estimation Method at the accuracy of 67%.
- [7] Pooja Kumawat; Sabarimalai Manikandan . M was presented this feature by using the laser dust sensor, model PM2007 at the parameter of Temp- 0.412 by using the method of a linear least square-based method for sensor calibration at the accuracy of 47%.
- [8] Postolache.O ; Dias J. M Pereira; was presented this feature with the Temperature and Humidity sensor of module (DHT11) at the parameter of NO_mean-0.125 by using the modem uses LG U+ LTE B5/B7 FDD Cat. 4 as the communication method at the accuracy of 41%.

[9] Silva Girão .P; Postolache.G was presented this feature with the pollutant detection sensors of module NO_DG-0.250 at the parameter and the Chamber & gravimetric method is used at the accuracy of 83%.

[10] Sumanth Reddy Enigella; Hamid Shah Nasser was presented this feature with the Air Quality sensor of module (MQ 135) at the parameter of Intercept-14.74 Using air quality measurement values in most devices was dependent on a number of variables, including measurement methods with a 52 percent accuracy.

samdan report

ORIGINALITY REPORT

15%

SIMILARITY INDEX

20%

INTERNET SOURCES

8%

PUBLICATIONS

12%

STUDENT PAPERS

PRIMARY SOURCES

1

static.sparkfun.com

Internet Source

4%

2

www.arduino.cc

Internet Source

3%

3

noderedguide.com

Internet Source

2%

4

Submitted to Woodcroft College

Student Paper

1%

5

components101.com

Internet Source

1%

6

www.coursehero.com

Internet Source

1%

7

www.mayoclinic.org

Internet Source

1%

8

eprints.gla.ac.uk

Internet Source

1%

9

en.wikipedia.org

Internet Source

1%

10	www.ninds.nih.gov Internet Source	1%
11	annualmeeting.conferenceseries.com Internet Source	1%
12	Kok Mun Ng, Muhamad Azri Haziq Mohd Suhaimi, Adizul Ahmad, N.A. Razak. "Remote Air Quality Monitoring System by Using MyRIO-LabVIEW", 2018 9th IEEE Control and System Graduate Research Colloquium (ICSGRC), 2018 Publication	1%
13	www.cs.uic.edu Internet Source	1%
14	Submitted to University of Moratuwa Student Paper	1%
15	Submitted to Universiti Teknikal Malaysia Melaka Student Paper	<1%
16	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	<1%
17	www.robotpark.com Internet Source	<1%
18	Vrushali Ladekar, Rohin Daruwala. "Indoor Air Quality Monitoring on AWS Using MQTT Protocol", 2019 10th International Conference	<1%

on Computing, Communication and Networking Technologies (ICCCNT), 2019

Publication

19

ieeexplore.ieee.org

Internet Source

<1%

20

www.ablkart.com

Internet Source

<1%

21

Somansh Kumar, Ashish Jasuja. "Air quality monitoring system based on IoT using Raspberry Pi", 2017 International Conference on Computing, Communication and Automation (ICCCA), 2017

Publication

<1%

22

Submitted to Roehampton University

Student Paper

<1%

23

Submitted to Melbourne Institute of Technology

Student Paper

<1%

24

Fang Lide, Ding Zhenjun, Li Jinhai, Li Honglian, Zhang Jike. "Temperature and flow rate compensation for air quality auto-monitoring system based on multi-sensor data fusion", 2010 The 2nd International Conference on Computer and Automation Engineering (ICCAE), 2010

Publication

<1%

Submitted to Universiti Teknologi Malaysia

25

Student Paper

<1%

26

Submitted to University of Florida

Student Paper

<1%

27

Sumanth Reddy Enigella, Hamid Shahnasser.
"Real Time Air Quality Monitoring", 2018 10th
International Conference on Knowledge and
Smart Technology (KST), 2018

Publication

<1%

28

linzeecraig.com

Internet Source

<1%

29

Submitted to Kookmin University

Student Paper

<1%

30

Somphop Chanthakit, Choopan Rattanapoka.
"MQTT Based Air Quality Monitoring System
using Node MCU and Node-RED", 2018
Seventh ICT International Student Project
Conference (ICT-ISPC), 2018

Publication

<1%

31

Submitted to Liverpool John Moores
University

Student Paper

<1%

32

Submitted to Royal College of Surgeons in
Ireland

Student Paper

<1%

33

Submitted to Engineers Australia

Student Paper

<1%

34

Submitted to Swinburne University of Technology

Student Paper

<1%

35

ataxia.org

Internet Source

<1%

36

www.hopkinsmedicine.org

Internet Source

<1%

37

Submitted to Indian Institute of Technology Guwahati

Student Paper

<1%

38

fromwww.jamaica-star.com

Internet Source

<1%

39

www.ijeart.com

Internet Source

<1%

40

Submitted to Walsall College

Student Paper

<1%

41

www.itead.cc

Internet Source

<1%

42

Submitted to National Institute of Technology, Rourkela

Student Paper

<1%

43

Submitted to Higher Education Commission Pakistan

<1%

44 Submitted to Visvesvaraya Technological University, Belagavi <1%
Student Paper

45 grietinfo.in <1%
Internet Source

46 www.wikizero.com <1%
Internet Source

47 Submitted to AAEC Estrella Mountain High School <1%
Student Paper

48 www.hindawi.com <1%
Internet Source

49 Submitted to Khulna University of Engineering & Technology <1%
Student Paper

50 Submitted to University of East London <1%
Student Paper

51 Submitted to University of Wisconsin, Madison <1%
Student Paper

52 simple-circuit.com <1%
Internet Source

53 www.ukessays.com <1%
Internet Source

54	www.scribd.com Internet Source	<1%
55	www.creoqode.com Internet Source	<1%
56	H. Chojer, P.T.B.S. Branco, F.G. Martins, M.C.M. Alvim-Ferraz, S.I.V. Sousa. "Development of low-cost indoor air quality monitoring devices: Recent advancements", Science of The Total Environment, 2020 Publication	<1%
57	Submitted to Institute of Research & Postgraduate Studies, Universiti Kuala Lumpur Student Paper	<1%
58	Jay H. Lehr. "Wiley's Remediation Technologies Handbook", Wiley, 2004 Publication	<1%
59	Submitted to Rochester Institute of Technology Student Paper	<1%
60	Submitted to University of Hertfordshire Student Paper	<1%
61	krishikosh.egranth.ac.in Internet Source	<1%
62	www.answers.com Internet Source	<1%

63

Submitted to Savitribai Phule Pune University

Student Paper

<1%

64

Submitted to Universiti Tenaga Nasional

Student Paper

<1%

65

github.com

Internet Source

<1%

66

Submitted to ABES Engineering College

Student Paper

<1%

67

Submitted to Nanyang Technological University

Student Paper

<1%

68

Submitted to National Institute of Business Management Sri Lanka

Student Paper

<1%

69

caca.essortment.com

Internet Source

<1%

70

Submitted to University of North Florida

Student Paper

<1%

71

digital.library.ryerson.ca

Internet Source

<1%

72

jamaicahospital.org

Internet Source

<1%

73

www.cooperhealth.org

Internet Source

<1%

74	Submitted to Ahsanullah University of Science and Technology Student Paper	<1%
75	Submitted to Sreenidhi International School Student Paper	<1%
76	Submitted to Universiti Malaysia Pahang Student Paper	<1%
77	Vasumathi Majety, G. Murali. "A remote epileptic patient supervising system", Advances in Modelling and Analysis B, 2018 Publication	<1%
78	fnfdoc.com Internet Source	<1%
79	dlnmh9ip6v2uc.cloudfront.net Internet Source	<1%
80	docplayer.net Internet Source	<1%
81	arduino.cc Internet Source	<1%
82	komponenelektronika.com Internet Source	<1%
83	www.riektron.co.za Internet Source	<1%
84	www.slideshare.net Internet Source	<1%

85	Ayesha Samreen, P. Sathish, N.Alivelu Manga. "Low Cost IoT Based Emission Monitoring System for Thermal Power Plants", 2019 Innovations in Power and Advanced Computing Technologies (i-PACT), 2019 Publication	<1%
86	Choopan Rattanapoka, Somphop Chanthakit, Apatsaraporn Chimchai, Amorntip Sookkeaw. "An MQTT-based IoT Cloud Platform with Flow Design by Node-RED", 2019 Research, Invention, and Innovation Congress (RI2C), 2019 Publication	<1%
87	Submitted to Madanapalle Institute of Technology and Science Student Paper	<1%
88	Submitted to Manchester Metropolitan University Student Paper	<1%
89	Submitted to Oberoi International School Student Paper	<1%
90	makingthing.com Internet Source	<1%
91	www.winsen-sensor.com Internet Source	<1%

92	"Progress in Intelligent Decision Science", Springer Science and Business Media LLC, 2021 Publication	<1%
----	--	-----

93	Anuj Kumar, I. P. Singh, S. K. Sud. "Energy efficient air quality monitoring system", 2011 IEEE SENSORS Proceedings, 2011 Publication	<1%
----	--	-----

94	YanJun Chen, Rami Ghannam, Hadi Heidari. "Air Quality Monitoring using Portable Multi-Sensory Module for Neurological Disease Prevention", 2019 UK/ China Emerging Technologies (UCET), 2019 Publication	<1%
----	---	-----

95	ukdiss.com Internet Source	<1%
----	-------------------------------	-----

Exclude quotes On

Exclude matches Off

Exclude bibliography On